

Model Pembacaan Sort Message Service (SMS) Menggunakan Metode Intonasi Dalam Bahasa Indonesia Berbasis Android

Herwin
Program Studi Manajemen
Informatika
STMIK-AMIK Riau
herwin@stmik-amik-riau.ac.id

Torkis Nasution
Program Studi Teknik
Informatika
STMIK-AMIK Riau
torkisnasution@stmik-amik-riau.ac.id

Dwi Haryono
Program Studi Teknik
Informatika
STMIK-AMIK Riau
dwiaryono@stmik-amik-riau.ac.id

Abstrak

Keterbatasan penglihatan yang di alami oleh tunanetra seharusnya tidak menyebabkan berkurangnya kemampuan dalam menggunakan media komunikasi. Peran teknologi dalam mencukupi kekurangan yang di alami oleh individu tunanetra menjadi solusi. Desain teknologi yang tepat guna dan aplikatif akan mereduksi kekurangan individu sehingga mendekati atau sama dengan manusia normal. Dalam mencapai tujuan tersebut, dalam kajian ini akan dibuat penelitian untuk mengubah tulisan menjadi suara pada sort message service, sehingga setiap pesan yang masuk melalui telepon pintar dapat di ketahui maksud yang terkandung di dalamnya. Untuk mencapai target tersebut akan menggunakan metode yang digunakan ialah Diphone Concatenation. Pembangkit suara yang digunakan adalah pembangkit surat dibuat sendiri. Hasil akhir yang didapat ialah dapat membacakan pesan dengan format teks biasa maupun singkatan dengan kecepatan eksekusi sekitar 20 detik dan memiliki ekspresi kalimat tanya, kalimat berita dan kalimat perintah.

Key word: pembangkit, TTS, android.

1. Pendahuluan

Pada perkembangannya saat ini, *handphone* dapat melakukan berbagai hal diantaranya ialah telepon, mengirimkan *Short Message Service* (SMS), mengirimkan *Multimedia Message Service* (MMS), *games*, *audio player*, *video player*, *browsing* internet dan masih banyak lagi. Dilihat dari penggunaan semua layanan yang diberikan, SMS memiliki peningkatan jumlah pengguna yaitu sekitar 727 juta SMS per hari (Telkomsel, 2010). SMS semula hanyalah merupakan layanan yang bersifat komplementer terhadap dua layanan utama sistem *Global System for Mobile*

Communications (GSM) yaitu layanan *voice* dan *switched data*. Namun karena keberhasilan SMS yang tidak terduga, dengan ledakan pelanggan yang mempergunakannya, menjadikan SMS sebagai bagian integral dari layanan sistem.

Namun dalam perkembangannya, *handphone* tidak bisa masuk langsung ke semua kalangan masyarakat, terutama ke masyarakat yang memiliki kekurangan fisik. Sebagai contoh, para penderita tunanetra hampir tidak bisa menikmati semua layanan yang ada dikarenakan dalam interaksinya mengandalkan visualisasi kecuali untuk menerima telepon. Dari masalah ini, munculah gagasan bagaimana para penderita tunanetra ini bisa mendapatkan berita dan informasi yang ada pada *handphone*, walaupun tidak semua layanan bisa digunakan. Salah satunya ialah dengan penciptaan sebuah perangkat lunak yang mengandalkan indera peraba dan indera pendengaran penderita. Dengan indera peraba, penderita tuna netra bisa membaca dengan menggunakan huruf *Braille* yang timbul dengan syarat penderita mengerti dengan tulisan itu, karena huruf *Braille* berbeda dengan huruf abjad seperti biasanya. Kedua ialah dengan indera pendengaran, dengan cara ini bisa dibidang cara yang lebih efektif dibandingkan dengan membaca memakai huruf *Braille*, dengan hanya mendengarkan saja penderita tunanetra bisa mendapatkan informasi yang sama dengan apa yang tertulis. Permasalahan yang dapat di amati adalah:

- Tunanetra sebagai bagian dari masyarakat tidak mampu berinteraksi secara utuh, diperlukan peran teknologi, melalui aplikasi ini akan dicukupi kekurangan tersebut.
- Belum ada suatu aplikasi yang mudah digunakan untuk menbugah tulisan menjadi ucapan dengan memperhatikan tanda baca dalam standar pengucapan bahasa Indonesia.
- Pengalihan dari tulisan ke lisan akan memperjelas ke aslian penuturan, dan kemungkinan besar terjadi penambahan tanda baca, kata atau kalimat oleh penulis.

- d. Melakukan kegiatan ke ilmiah dalam rangka pencarian ke-ilmuan yang baru atau terbaru.

Dalam penelitian ini, akan dikembangkan sebuah perangkat lunak yang berdasarkan pada teknologi *text-to-speech* dalam Bahasa Indonesia dengan studi kasus sebagai pembaca SMS. Perangkat lunak yang akan dibuat ini akan mengkonversi *input* berupa teks dari sebuah atau beberapa SMS menjadi *output* berupa suara manusia yang dikemas dalam bentuk suatu *file audio* serta mempunyai intonasi dan ekspresi mengikuti tata bahasa baku Bahasa Indonesia.

2. Fonologi

Menurut Thomas Anung Basuki manusia mempunyai kemampuan untuk membedakan berbagai macam suara dengan range frekuensi 20 Hz sampai dengan 20.000 Hz. Dengan kemampuan ini manusia dapat mengeluarkan bunyi bahasa atau kata-kata. Suatu kata dapat terbentuk dengan penggabungan beberapa frekuensi tertentu menjadi satu. Fonologi merupakan ilmu yang mempelajari tentang seluk beluk bunyi bahasa. Fonologi dapat dibagi menjadi dua macam yaitu fonetik dan fonemik. Fonetik yaitu cabang studi fonologi yang mempelajari bunyi bahasa tanpa memperhatikan apakah bunyi-bunyi tersebut mempunyai fungsi sebagai pembeda makna atau tidak. Fonetika memiliki tiga cabang utama [4] :

1. fonetik artikulatoris yang mempelajari posisi dan gerakan bibir, lidah dan organ-organ manusia lainnya yang memproduksi suara atau bunyi bahasa
2. fonetik akustik yang mempelajari gelombang suara dan bagaimana mereka didengarkan oleh telinga manusia
3. fonetik auditori yang mempelajari persepsi bunyi dan terutama bagaimana otak mengolah data yang masuk sebagai suara Sedangkan fonemik yaitu cabang studi fonologi yang mempelajari bunyi bahasa dengan memperhatikan fungsi bunyi tersebut sebagai pembeda.

3. Stemming Bahasa Indonesia

Menurut Arman. A. Akhmad stemming [2] merupakan suatu proses untuk menemukan kata dasar dari sebuah kata. Dengan menghilangkan semua imbuhan (affiks) baik yang terdiri dari awalan (prefiks), sisipan (infiks), akhiran (sufiks) dan kombinasi dari awalan dan akhiran (konfiks) pada kata turunan. Stemming digunakan untuk mengganti bentuk dari suatu kata menjadi kata dasar sesuai dengan struktur morfologi Bahasa Indonesia yang baik dan benar.

Imbuhan pada Bahasa Indonesia lebih kompleks bila dibandingkan dengan imbuhan pada Bahasa Inggris. Karena imbuhan pada Bahasa Indonesia terdiri

dari awalan, sisipan, akhiran, bentuk perulangan (repeated forms) dan kombinasi. Secara umum, pembentukan kata turunan dengan imbuhan mengikuti aturan penulisan kata yang ada di bagian sebelumnya. Berikut adalah beberapa informasi tambahan untuk melengkapi aturan tersebut.

4. Teknologi Text to Speech

Penelitian di bidang pensintesa ucapan mengalami perjalanan yang sangat panjang dan telah dimulai sejak lama. Salah satu catatan literatur awal yang berhubungan dengan sintesa ucapan adalah pernyataan seorang ahli matematika dan engineer terkenal yang bernama Leonhard Euler pada tahun 1761. Euler menyatakan "It would be a considerable invention indeed, that of a machine able to mimic speech, with its sounds and articulations. I think it is not impossible".

Berdasarkan hasil studi literatur dari berbagai sumber bacaan, perkembangan teknologi pensintesa ucapan dapat dibagi menjadi tiga kurun waktu. Kurun waktu pertama adalah sebelum 1930. Pada masa ini penelitian-penelitian lebih banyak ditujukan untuk memahami karakteristik sinyal ucapan serta pengembangan pensintesa ucapan berbasis mekanik atau elektromekanik. Kurun waktu kedua dimulai sejak tahun 1930-an sampai dengan ditemukannya komputer digital. Masa ini ditandai dengan pengembangan berbagai alat pensintesa ucapan menggunakan teknologi elektronik analog. Kurun waktu ketiga dimulai sejak ditemukannya komputer digital hingga sekarang. Pada masa ini, sintesa ucapan dilakukan menggunakan pendekatan-pendekatan pemrosesan sinyal digital.

4.1. Sistem Text to Speech

Pada prinsipnya, sistem Text to Speech (TTS) adalah suatu sistem yang dapat melakukan konversi dari teks menjadi ucapan. Dutoit dalam buku "An Introduction to Text-to-Speech Synthesis" mendefinisikan Text-to-Speech sebagai "production of speech by machines, by way of the automatic phonetization of the sentences to utter".

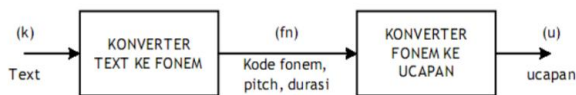
Pelton dalam buku "Voice Processing" menyatakan bahwa "The task of Text to Speech system is to convert plain text into speech ...". Dalam bagian lainnya, Pelton menyatakan pula "A very attractive advantage of text to speech is that any text can be read, vocabulary is not restricted to utterances that have been decided upon beforehand".

Berdasarkan sumber-sumber tersebut, suatu sistem TTS dapat didefinisikan sebagai "Sistem yang dapat mengubah suatu teks menjadi ucapan secara otomatis dengan cara fonetisasi (penyusunan fonem-fonem untuk membentuk ucapan)". Sebuah sistem TTS dapat

mengucapkan kata apapun, dan kosa katanya tidak terbatas. Pada dasarnya system TTS ini terdiri dari dua sub sistem, yaitu :

1. Bagian konverter teks ke fonem dan,
2. Bagian fonem ke ucapan

Bagian konverter teks ke fonem berfungsi mengolah masukan kalimat dalam bentuk teks yang diurutkan sesuai kode-kode tertentu yang direpresentasikan dengan kode fonem, durasi serta frekuensinya. Bagian ini disebut dengan NLP (Natural Language Processing) karena berfungsi mengolah teks yang berhubungan dengan linguistik. Sedangkan bagian konverter fonem ke ucapan akan menerima masukan kode-kode fonem, durasi dan frekuensinya sesuai dengan apa yang dihasilkan oleh proses sebelumnya, yang kemudian akan menghasilkan bunyi atau sinyal ucapan. Bagian ini kadang-kadang disebut modul DSP (Digital Signal Processing), karena sebagian besar proses yang dilakukannya adalah membangkitkan dan memanipulasi sinyal ucapan secara digital.



Gambar 1. Sistem Text to Speech

4.2 Digital Signal Processing

Ada beberapa metode untuk membangkitkan sinyal ucapan ini untuk mengimplementasikan bagian konverter fonem ke ucapan yaitu :

1. Formant Synthesizer
Metode ini bekerja berdasarkan suatu model matematis yang akan melakukan komputasi untuk membangkitkan sinyal ucapan yang diinginkan. Pembangkitan ucapan dilakukan berdasarkan parameter-parameter pembentuk ucapan. Beberapa parameter penting pembentuk ucapan tersebut adalah frekuensi-frekuensi formant, bandwidth serta amplitudanya. Pensintesa jenis ini telah lama digunakan pada berbagai aplikasi. Generasi terakhir pensintesa berbasis formant ini mampu menghasilkan ucapan dengan tingkat kejelasan ucapan yang baik, tetapi tidak mampu menghasilkan ucapan dengan tingkat kealamian yang tinggi.
2. Concatenative Synthesizer
Metode ini bekerja dengan cara menggabungkan potongan unit suara yang telah direkam sebelumnya. Teknik ini biasa disebut dengan diphone concatenation yang bekerja dengan menggabungkan diphone (dua buah fonem atau

lebih). TTS yang menggunakan teknik ini mampu mencapai tingkat kejelasan serta tingkat kealamian ucapan yang tinggi. Dua diphone processor engine yang tersedia dan banyak digunakan untuk kegiatan penelitian dan pengembangan adalah :

- a. Festival, dikembangkan di CSTR (Centre for Speech Technology Research) di University of Edinburgh;
- b. MBROLA, dikembangkan di Polytechnique de Mons, Belgia.

Kedua sistem tersebut bersifat terbuka dan dapat digunakan sebagai bagian dari TTS serta aplikasinya untuk tujuan penelitian dan pengembangan yang sifatnya non-komersial dan non-militer. MBROLA telah digunakan untuk implementasi TTS pada lebih dari 25 bahasa di dunia, termasuk untuk Bahasa Indonesia yang digunakan dalam pembangunan perangkat lunak ini.

4.3. Tahapan Text-to-Speech

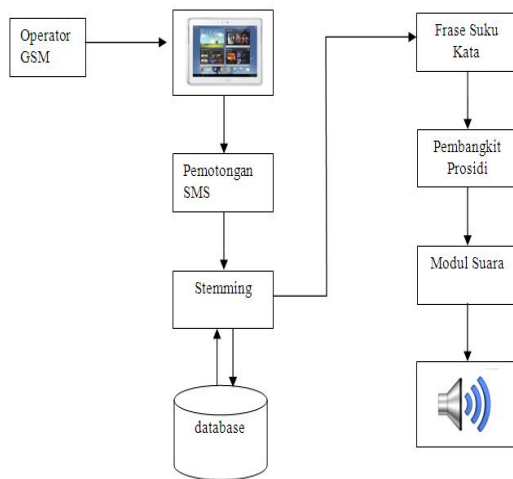
Algoritma pembangkit suara menggabungkan frame-frame pada unit suara yang saling overlap langsung ke dalam pemrosesan suara berdomain waktu. teknik yang digunakan adalah spectral smoothing dalam memanipulasi spectral terpatah-patah yang terjadi pada perbatasan antara unit suara-unit suara yang akan digabungkan. Selain itu pembangkit sura melakukan kompresi dan dekomposisi database suaranya sehingga ukurannya lebih kecil.

Untuk mendapatkan ucapan yang lebih alami, ucapan yang dihasilkan harus memiliki intonasi (prosody). Prosodi adalah perubahan nilai pitch (frekuensi dasar) selama pengucapan kalimat dilakukan. Pada prakteknya, informasi pembentuk prosodi berupa data-data pitch serta durasi pengucapannya untuk setiap fonem yang dibangkitkan. Nilai-nilai yang dihasilkan diperoleh dari suatu model prosodi. Prosodi bersifat sangat spesifik untuk setiap bahasa, sehingga model yang diperlukan untuk membangkitkan data-data prosodi menjadi sangat spesifik juga untuk suatu bahasa. Beberapa model umum prosodi pernah dikembangkan, tetapi untuk digunakan pada suatu bahasa masih perlu banyak penyesuaian yang harus dilakukan. Pada sistem yang menggunakan teknik diphone concatenation, sistem harus didukung oleh suatu diphone database yang berisi rekaman segmen-segmen ucapan yang berupa diphone. Ucapan dalam suatu bahasa dibentuk dari satu set bunyi yang mungkin berbeda untuk setiap bahasa, oleh karena itu setiap bahasa harus dilengkapi dengan diphone database yang berbeda.

5. Model Sistem

Sistem yang dibuat memiliki beberapa tahapan utama yang dilakukan, diantaranya adalah :

1. Pengambilan teks dari SMS *inbox*
 2. Proses segmentasi SMS
 3. Proses *stemming* SMS dan pengecekan kata di *database*
 4. Pemenggalan suku kata Bahasa Indonesia
 5. Pembangkitan prosodi atau intonasi
- Untuk lebih detailnya bisa perhatikan di gambar dibawah ini.



Gambar 2. Arsitektur Sistem

5.1. Pengambilan Teks dari SMS *Inbox*

Pengambilan teks SMS ini akan diambil dari *folder Inbox* yang ada di perangkat *handphone*. Pengambilan SMS dilakukan oleh bahasa pemrograman java. Yang dihasilkan dari pengambilan SMS ini berupa teks asli sebelum diproses.

5.2. Segmentasi SMS

Proses pemenggalan kalimat menjadi penting untuk mengetahui isi dari pesan, diawali dengan pemenggalan kalimat ke dalam kata. Selanjutnya akan di periksa apakah berupa singkatan, bila ternyata singkatan maka perlu di sesuaikan dengan kamus yang telah dibuat.

1. **Segmentasi Kalimat**, dilakukan apabila didalam teks SMS terdapat karakter “.,!?:“-” yang tujuannya membagi teks SMS menjadi beberapa bagian kalimat. Contoh sederhananya ialah : Lg dkmps nih.. bntn lg plgnya! Bsk jg mau kkmps lg. mhn maaf lahir batin ya, Smg kt smua dprtemukan kmbli. Karena menemukan tanda koma (,), titik (.) dan tanda seru (!) maka kalimat dipecah menjadi empat segment kalimat yaitu
 - a. Lg dkmps nih
 - b. bntn lg plgnya
 - c. Bsk jg mau kkmps lg.
 - d. mhn maaf lahir batin ya
 - e. Smg kt smua dprtemukan kmbli

2. **Segmentasi Kata**, dilakukan bila dalam kalimat terdapat karakter spasi, dan selanjutnya dipecah menjadi beberapa kata.

Tabel 1. Hasil Segmentasi Kalimat dan Kata

No.	Segmentasi	Kata
1.	Lg dkmps ne	- Lg - Dkmps - Ne
2.	bntn lg plgnya	- Btr - Lg - Plgnya
3.	Bsk jg mau kkmps lg	- Bsk - Jg - Mau - Kkmps - Lg
4.	Smg kt smua dprtemukan kmbli	- Smg - Kt - Smua - Dprtemukan - Kmbli
5.	Q dtg skr	- Q - Dtg - Skr

5.3. Proses *Stemming* SMS

Proses ini berfungsi untuk memotong atau memangkas sebuah kata yang memiliki imbuhan baik di awal, akhir dan campuran. Dalam penelitian ini tidak disertakan imbuhan yang berupa sisipan dan perulangan karena struktur kata Bahasa Indonesia yang terlalu spesifik sehingga kurang memungkinkan untuk dibuat. dalam pemrosesan yang sesuai. Pada proses *stemming* ini seluruh kata dicek satu persatu apakah memiliki imbuhan atau tidak. Proses ini sebenarnya hanya mencari kata dasar yang memiliki imbuhan dan apabila didapatkan kata dasar maka akan melakukan pengecekan ke *database* dan jika kata dasar tadi berupa singkatan dan ada di *database* maka kata dasar awal akan digantikan dengan kata dasar yang ada pada *database*. Dan jika tidak, maka kata langsung dicek ke *database*, jika tidak ditemukan maka kata awal akan diproses langsung walaupun kata tersebut tidak dalam kata baku yang ada pada Bahasa Indonesia atau EYD (Ejaan Yang Disempurnakan). Dengan kata lain sistem ini maContoh proses *stemming* dan pengecekan *database* dapat dilihat pada SMS ke-1, dengan pola **Prefiks 1 + Prefiks 2 + Kata dasar + Sufiks 3 + Sufiks 2 + Sufiks 1** dengan urutan awalan 1, awalan 2,

akhiran 1, akhiran 2 dan akhiran 3 untuk pengecekan dan berlaku aturan sebagai berikut:

imbuh1 = ['menge','penge']
 imbuh2 = ['meng','peng','meny','peny']
 imbuh3 = ['mem','pem','men','pen','per','ter','ber']
 imbuh4 = ['me','pe','ke','se','di','be']
 akhir1 = ['kan','lah','nya']
 akhir2 = ['an']
 akhir3 = ['i']

a. Kata “Lg”

- Awalan 1 : gagal memenuhi semua aturan
- Awalan 2 : gagal memenuhi semua aturan
- Akhiran 1 : gagal memenuhi semua aturan
- Akhiran 2 : gagal memenuhi semua aturan
- Akhiran 3 : gagal memenuhi semua aturan

Karena dari semua aturan diatas tidak semuanya terpenuhi, maka kata tersebut langsung dicek ke *database*, dan jika ada dalam *database* maka kata “lg” diubah menjadi kata “lagi” yang sesuai.

b. Kata “dkmps”

Karena kegunaan sistem menangani kata yang disingkat, maka imbuhan juga perlu memiliki pola yang disingkat. Dengan begitu aturan menjadi bertambah

imbuh1 = ['menge','penge']
 imbuh2=['meng','peng','meny','peny','**mnge**’,’**png e**’]
 imbuh3 = ['mem', 'pem', 'men', 'pen', 'per', 'ter', 'ber',
 ’**mng**’,’**png**’,’**mny**’,’**pnny**’]
 imbuh4 = ['me', 'pe', 'ke', 'se', 'di', 'be', ’**mm**’, ’**pm**’,
 ’**mn**’, ’**pn**’, ’**pr**’, ’**tr**’, ’**br**’]
 imbuh5= [’,m’,’p’,’k’,’s’,’d’,’b’]
 akhir1 = ['kan','lah','nya']
 akhir2 = ['an']
 akhir3 = ['i']

- Awalan 1 : memenuhi aturan imbuh5
- Kata menjadi “d” dan “kmps”
- Kata “kmps” cek di *database*

Kata “kmps” terdapat di *database*, dengan begitu digantikan menjadi kata “kampus”

- Kata “d” bermaksud awalan “di”, dan digantikan
- Setelah itu digabungkan kembali dan menjadi kata “dikampus”

c. Kata “nih”

- Awalan 1 : gagal memenuhi semua aturan
 - Awalan 2 : gagal memenuhi semua aturan
 - Akhiran 1 : gagal memenuhi semua aturan
 - Akhiran 2 : gagal memenuhi semua aturan
 - Akhiran 3 : gagal memenuhi semua aturan
- Karena dari semua aturan diatas tidak semuanya terpenuhi, maka kata tersebut langsung dicek ke *database*, dan jika tidak ada dalam *database* maka kata “nih” tidak diubah sama sekali.

d. Kata “plgnya”

- Awalan 1 : gagal memenuhi semua aturan
- Awalan 2 : gagal memenuhi semua aturan
- Akhiran 1 : memenuhi aturan akhir1
- Kata menjadi “plg” dan “nya”
- Kata “plg” cek di *database*
- Kata “plg” terdapat di *database*, dengan begitu digantikan menjadi kata “pulanganya”
- Kata “nya” bermaksud akhiran “nya”, dan tidak digantikan
- Setelah itu digabungkan kembali dan menjadi kata “pulanganya”

Tabel 2. Hasil Stemming SMS

No.	Segmentasi	Kata	Hasil Stemming dan pengecekan database
1.	Lg dkmps	Lg	Lagi
		dkmps	Di kampus
2.	bntr lg plg	Bntr	Bentar
		Lg	Lagi
		Plg	Pulang
3.	Bsk jg mau kkmpps lg	Bsk	Besok
		Jg	Juga
		Kkmpps	Ke kampus
		Lg	Lagi
4.	mhn maaf lahir batin ya	Mhn	Mohon
		Maaf	Maaf
		Lahir	Lahir
		Batin	Batin
		Ya	Ya
5.	Smg kt smua dprtemukan kmbli	Smg	Semoga
		Kt	Kita
		Smua	Semua
		Dprtemukan	Dipertumakan
		Kmbli	Kembali

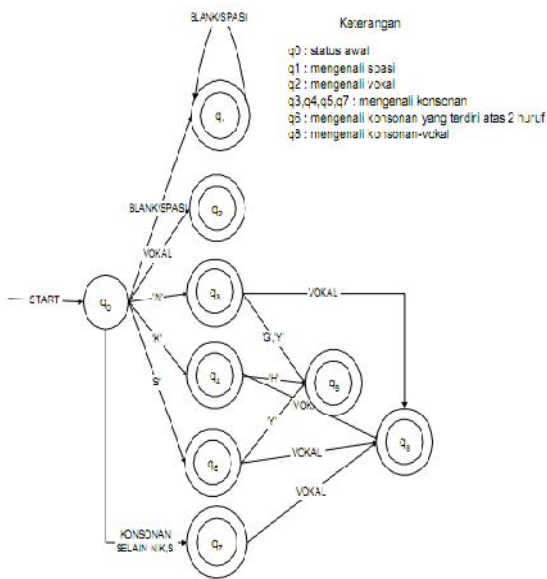
5.4. Pemenggalan Suku Kata

Pada proses ini semua kata akan dipenggal menjadi beberapa suku kata yang sesuai dengan kaidah Bahasa Indonesia. Adapun proses pemenggalan suku kata ini dengan cara.

$$kt_{ijkl} = U_{l=1}^{jsk} sk_{ijkl}$$

Menurut Thomas Anung Basuki Pada Teori Bahasa Formal, setiap bahasa memiliki suatu aturan tata bahasa yang baku dan konsisten. Dalam kenyataannya bahasa yang dipakai oleh manusia merupakan kesepakatan dari para pemakai bahasa tersebut. Akibatnya tata bahasa yang digunakan seringkali tidak konsisten. Oleh karena itu dapat dimaklumi jika Teori Bahasa Formal tidak dapat memodelkan bahasa manusia secara sempurna. Dalam perancangan perangkat lunak ini digunakan FSA sebagai mesin

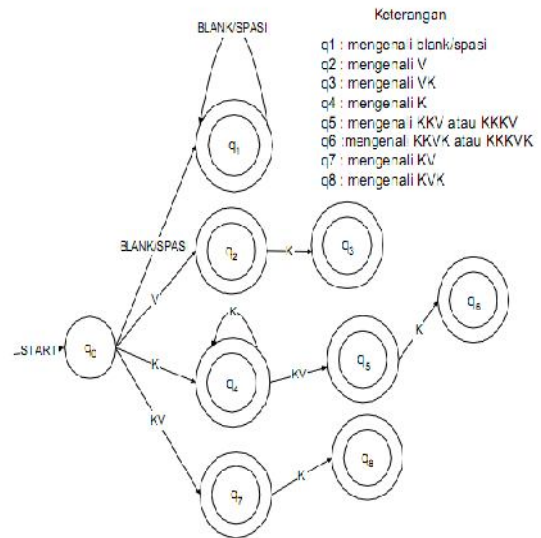
abstrak yang akan mengenali dan memisahkan suku kata dalam kalimat. FSA yang akan digunakan dirancang dalam tiga tingkatan. Pada tingkatan pertama yang akan dikenali adalah pola-pola : V,K atau KV. Hasil pengenalan FSA pada suatu tingkatan menjadi masukan bagi FSA tingkatan berikutnya. Pada tingkatan kedua FSA akan mengenali suku kata dengan pola V, VK, VKK, KV, KVK, KKV, KKVK, KKKV, KKKVK. Dengan menggunakan pemodelan bertingkat, akan mempermudah pemisahan suku kata. Untuk memperjelasnya, dapat dilihat dari contoh berikut ini. Pada saat kita membaca dua huruf pertama kata **anak** dan kata **anda** kita belum dapat memutuskan apakah pemisahan suku kata akan dilakukan di antara kedua huruf tersebut atau tidak. Setelah membaca huruf ketiga, barulah bisa diputuskan di mana harus dilakukan pemisahan suku kata. Jika huruf ketiga berupa sebuah konsonan maka pemisahan dilakukan setelah huruf kedua (kata **anda** akan menjadi **an-da**). Sedangkan jika huruf ketiga adalah sebuah vokal, maka harus ditelusuri mundur dan memisahkan suku kata setelah huruf pertama (kata **anak** akan menjadi **a-nak**).



Gambar 3. Diagram Transisi FSA Tingkatan Pertama

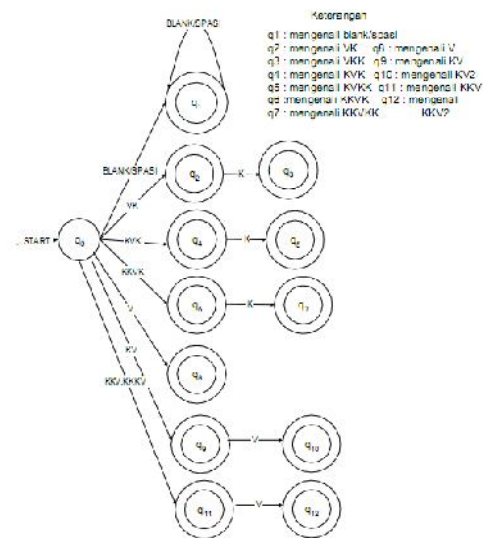
Penelusuran mundur ini bertentangan dengan prinsip kerja FSA yang hanya bisa bergerak maju. Untuk mengatasinya perlu dimodelkan FSA tingkatan pertama yang mengenali pola V, K dan KV. Pada tingkatan pertama kata anak akan dipisahkan menjadi **a-na-k** (V-KV-K) dan pada tingkatan kedua akan menjadi **a-nak**. Pemisahan sesudah huruf pertama ini terjadi karena tidak dikenal suku kata berpola VKV (sesuai dengan pedoman pemisahan suku kata).

Sedangkan kata anda pada tingkatan pertama akan dipisahkan menjadi **a-n-da** (V-K-KV), yang lalu dipisahkan menjadi **an-da**. Hal ini sesuai dengan aturan pemisahan suku kata dasar Bahasa Indonesia (sesuai dengan pedoman pemisahan suku kata).



Gambar 4. Diagram Transisi FSA Tingkatan Kedua

Dari kelakuan FSA tingkatan kedua, tampak bahwa pola suku kata VKK, KVKK dan KKVKK belum bisa dikenali. Untuk itu diperlukan FSA tingkatan ketiga yang mampu mengenalnya. Jika FSA ini menemukan pola VK-K, KVK-K dan KKVK-K, dia akan mengenalnya sebagai pola suku kata VKK, KVKK dan KKVKK.



Gambar 6. Diagram Transisi FSA Tingkatan Ketiga

Selain itu FSA pada tingkatan 3 dapat mengenali diftong. Dalam Bahasa Indonesia dikenal 3 macam diftong yaitu : „au“, „ai“ dan „oi“. Namun kemunculan dua vokal tersebut secara berturutan belum tentu berupa diftong. Kata-kata seperti „kacau“, „pantai“, „sepoi“ adalah contoh kata yang mengandung diftong. Kata „kaidah“, „yaitu“ dan „bau“ adalah contoh kata yang tidak mengandung diftong. Karena ketidakkonsistenan ini maka tidak mungkin mengenali diftong secara sempurna tanpa melakukan analisis semantik. Analisis semantik tidak dapat dimodelkan dengan FSA.

Pada penelitian ini tidak disertakan pemecahan kata dengan pola kata vokal yang saling berhimpitan, dikarenakan adanya ketidakkonsistenan dalam pengenalan diftong atau bukan diftong. Berikut ini adalah hasil pemenggalan kata dari dua SMS sebelumnya yang dijelaskan.

Tabel 3. Hasil Pemenggalan Suku Kata

No.	Kata	Suku Kata
1.	lagi - dikampus - nih	La-gi di-kam-pus nih
2.	bentar - lagi - pulangnya	Ben-tar La-gi Pu-lang-nya
3.	- besok - juga - mau - kekampus - lagi	Be-sok Ju-ga Mau Ke-kam-pus La-gi
4	- mohon - maaf - lahir - batin - ya	Mo-hon Maaf la-hir Ba-tin ya
5.	- semoga - kita - semua - dipertemukan - kembali	Se-mo-ga Ki-ta Se-mua di-per-te-mu-kan kem-ba-li

6. Pembangkitan Prosodi

Proses terakhir sebelum pembangkitan prosodi dimulai ialah mengubah semua karakter yang ada menjadi kode sampa yang sebelumnya telah dituliskan pada bab 2. Untuk penelitian ini penulis menggunakan kode sampa dan durasi dasar setiap fonem telah ditentukan sebelumnya oleh penulis.

Tabel 4. Kode Sampa dan Frekuensi Dasar

Fonem	SAMPA	Durasi (ms)
a	V	65
B	B	65
C	Ts	124
D	D	48
E	e	48
d	@	84
F	F	100
G	G	90
H	H	60
I	I	60
J	dZ	84
K	K	87
L	L	54
M	M	90
N	N	65
O	Q	83
P	P	97
Q	K	87
R	R	90
S	S	100
T	T	100
U	U	51
V	F	100
W	W	61
X	X	
Y	J	90
Z	Z	72
Ny	nY	41
Ng	N	61
	-	10

Setelah semua huruf dirubah menjadi fonem, berbekal dengan proses sebelumnya yaitu pemenggalan kalimat, pemenggalan kata dan pemenggalan suku kata dan perubahan huruf menjadi fonem, proses pembangkitan prosodi bisa dilakukan. Hasil akhir dari pembangkitan prosodi [3] untuk setiap fonem memiliki pola seperti pada gambar 7.

m 90 0 115 57 109

Gambar 7. Atribut Fonem yang telah Diberikan Durasi dan Frekuensi

Studi kasus pembangkitan Prosodi untuk kalimat :
 besok juga mau ke kampus lagi

Tahap 1 :

Menentukan angka pembagi untuk setiap kata.

- Kata pertama

Batas atas kata pertama, $ProsodiAwalAtas = 40$
 Batas atas kata kedua, $ProsodiAwalBawah = 6$

m 90 0 115 57 109

Pembagi, $ProsodiAtas = 40 - 6$
 - Kata kedua sampai ke-n

Batas atas kata pertama, $ProsodiTengahAtas = 60$
 Batas atas kata kedua, $ProsodiTengahBawah = 10$
 Pembagi, $ProsodiTengah = 60 - 10$
 - Kata terakhir

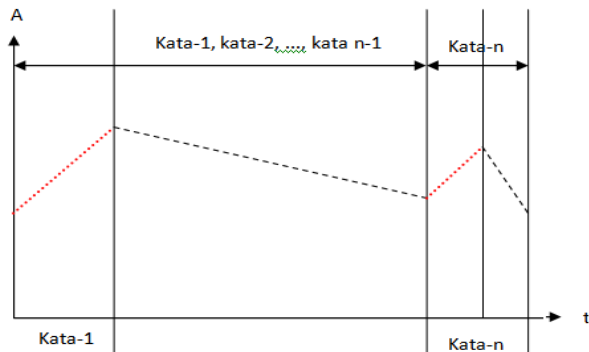
Batas atas kata pertama, $ProsodiAkhirAtas = 30$
 Batas atas kata kedua, $ProsodiAkhirBawah = 1$
 Pembagi, $ProsodiAkhir = 30 - 1$

Tahap 2 :

Dalam proses pemenggalan kata dan suku kata didapatkan kata pertama, kata-kata tengah dan kata akhir dengan hasil sebagai berikut

Kata pertama : besok (be-sok)
 Kata tengah : juga (ju-ga), mau (mau), ke (ke),
 kampus (kam-pus)
 Kata akhir : lagi (la-gi)

Tahap 3 :



Sumber Arry Akhmad Arman, 2008

Gambar 7. Kurva Prosodi Per Segmen

Pemberian atribut pada kata pertama, yaitu kata be-sok. Kata pertama mempunyai pola frekuensi yang menaik. Untuk pemberian frekuensi pada kata pertama (suku kata pertama dan suku kata kedua)

$bagi = ProsodiTengah / s$
 $frek = bagi + 100$
 $posisi = frek/2$
 $frek2 = (ProsodiTengah / (s+1)) + 100$
 Untuk suku kata "ju" :
 $bagi = 50/1$
 $frek = 50 + 100$

$posisi = 150/2$
 $frek2 = (50/(1+1)) + 100$

hasil yang didapatkan dari suku kata ini adalah :

dZ 84 0 150 75 125
 u 83 0 150 75 125
 Untuk suku kata "ga" :
 $bagi = 50/2$
 $frek = 25 + 100$
 $posisi = 125/2$
 $frek2 = (50/(2+1)) + 100$

hasil yang didapatkan dari suku kata ini adalah :

g 90 0 125 62 116
 V 65 0 125 62 116
 Untuk suku kata "mau" :
 $bagi = 50/3$
 $frek = 16 + 100$
 $posisi = 116/2$
 $frek2 = (50/(3+1)) + 100$

hasil yang didapatkan dari suku kata ini adalah :

m 90 0 116 58 112
 V 65 0 116 58 112
 U 51 0 116 58 112
 Untuk suku kata "ke" :
 $bagi = 50/4$
 $frek = 12 + 100$
 $posisi = 112/2$
 $frek2 = (50/(4+1)) + 100$

hasil yang didapatkan dari suku kata ini adalah :

k 87 0 112 56 110
 @ 84 0 112 56 110
 Untuk suku kata "kam" :
 $bagi = 50/5$
 $frek = 10 + 100$
 $posisi = 110/2$
 $frek2 = (50/(5+1)) + 100 75$

hasil yang didapatkan dari suku kata ini adalah :

k 87 0 110 55 108
 V 65 0 110 55 108
 M 90 0 110 55 108
 Untuk suku kata "pus" :
 $bagi = 50/6$
 $frek = 8 + 100$
 $posisi = 108/2$
 $frek2 = (50/(6+1)) + 100$

hasil yang didapatkan dari suku kata ini adalah :

p 97 0 108 54 107
 U 51 0 108 54 107
 s 100 0 108 54 107

Kata terakhir memiliki rumus yang berbeda pula dengan kata pertama dan kata tengah, kata ini akhir dari kalimat dan memiliki intonasi yang menaik.

$$\begin{aligned} \text{bagi} &= \text{ProsodiAkhir} / 2 + s - 1 \\ \text{frek} &= \text{bagi} + 100 \\ \text{posisi} &= \text{frek} / 2 \\ \text{frek2} &= (\text{ProsodiAkhir} / (s+1)) + 100 \end{aligned}$$

Untuk suku kata "la" :

$$\begin{aligned} \text{bagi} &= 29/2 + 1 - 1 \\ \text{frek} &= 14 + 100 \\ \text{posisi} &= 114/2 \end{aligned}$$

$$\begin{aligned} \text{frek2} &= (29/(1+1)) + 100 \\ \text{hasil yang didapatkan dari suku kata ini adalah :} \\ 1\ 5\ 4\ 0\ 1\ 1\ 4\ 5\ 7\ 1\ 1\ 4 \\ V\ 6\ 4\ 0\ 1\ 1\ 4\ 5\ 7\ 1\ 1\ 4 \end{aligned}$$

Untuk suku kata "gi" :

$$\begin{aligned} \text{bagi} &= 29/2 + 2 - 1 \\ \text{frek} &= 15 + 100 \\ \text{posisi} &= 115/2 \\ \text{frek2} &= (29/(2+1)) + 100 \end{aligned}$$

$$\begin{aligned} \text{hasil yang didapatkan dari suku kata ini adalah :} \\ g\ 9\ 0\ 0\ 1\ 1\ 5\ 5\ 7\ 1\ 0\ 9 \\ I\ 6\ 0\ 0\ 1\ 1\ 5\ 5\ 7\ 1\ 0\ 9 \end{aligned}$$

Dari hasil pembangkitan prosodi ini diperoleh hasil yang siap oleh modul pembangkit suara menjadi *file* suara. Untuk beberapa kalimat yang diakhiri dengan tanda tanya (?) di akhir kalimat diberikan pemberian durasi yang lebih panjang daripada durasi yang digunakan pada kalimat biasa. Sedangkan untuk kalimat perintah atau yang diakhiri dengan tanda seru (!) diberikan durasi yang lebih pendek daripada durasi yang digunakan pada kalimat biasa.

6.2. Perangkat Lunak

Terdapat berbagai macam dan jenis *engine* untuk membangun sebuah sistem *text to speech*, dalam penelitian ini digunakan *engine* pembangkit suara sendiri, sedianya menggunakan MBROLA, namun tidak tersedia untuk sistem operasi android. Sehingga dibuat sendiri.

6.3. Deskripsi Sistem

Pengguna perangkat *handphone* yang semakin hari semakin bertambah jumlah penggunaannya. Namun dalam kemajuannya ini, banyak keterbatasan dalam penggunaan perangkat. Diantaranya para penderita tunanetra yang tidak bisa membaca pesan singkat atau SMS sebagaimana orang pada biasanya. Untuk mengatasi hal tersebut, diperlukan suatu aplikasi yang dapat membacakan isi pesan SMS sehingga akan memudahkan penderita tunanetra untuk mengikuti

perkembangan teknologi. Aplikasi ini dibuat dengan menggunakan *speech engine* sendiri, dengan suara yang di rekam sendiri.

1. **Analisis Input**, isi pesan atau SMS merupakan satu-satunya masukan agar sistem ini dapat berjalan. Adapun karakteristik SMS yang diperhatikan sebagai berikut:
 - a. Jika isi SMS sesuai dengan EYD dalam Bahasa Indonesia, isi SMS langsung dapat dibacakan tanpa ada masalah atau pengucapan yang gagal.
 - b. Jika isi SMS tidak sesuai dengan EYD, maka dilakukan langkah-langkah *stemming* dan pengecekan ke *database*. Jika tidak ditemukan kemungkinan pembacaan pesan akan dieja sesuai dengan huruf yang ada, gagal atau ada beberapa kata singkatan yang tidak bisa terbacakan.
2. **Analisis Output**, keluaran (*output*) dari aplikasi ini merupakan suara yang dapat didengarkan sesuai dengan isi SMS dengan tingkat kealamian suara yang cukup jelas, sehingga pengguna dapat mengetahui isi dari pesan tanpa harus membacanya, terutama untuk penderita tunanetra.

6.4. Batasan Aplikasi

Batasan-batasan dalam pengembangan perangkat lunak untuk menyelesaikan masalah pembacaan SMS yang mengandung kata singkatan dalam bahasa sehari-hari adalah sebagai berikut.

1. Kamus kata yang digunakan pada sistem ini disimpan dalam *file database* dengan nama kata.
2. Isi dari kamus kata dimasukkan oleh penulis dan bisa ditambah oleh *users* bila kata yang dimaksud belum terdapat dalam *database*, dan tidak bisa di edit jika terjadi kesalahan.
3. Hasil dari penyempurnaan kata yang disingkat hanya bisa dilakukan jika terdapat dalam *database* dan memenuhi aturan *stemming* (jika terdapat imbuhan yang menempel pada kata dasar) disimpan dalam *file input.txt*. Dengan kata lain sistem ini sangat bergantung pada kamus jika kata mengandung kata singkatan.
4. Hasil dari pengubahan huruf ke fonem disimpan di dalam *file input.pho*
5. Hasil suara yang dihasilkan dari teks untuk selanjutnya diucapkan disimpan dalam *file speech.wav*.

6.5. Perancangan Sistem

Dalam perancangan dalam penelitian ini terdiri atas perancangan *database* dan perancangan arsitektur. Perancangan *database* untuk pembangunan aplikasi pembaca SMS ini terdiri dari satu *table* yang bisa dilihat di tabel 5.

Tabel 5. Struktur Tabel Kata

Field	Type	Width	Deskripsi
Id	Integer	11	Primary key
Singkat	Varchar	255	Singkatan
Eyd	Varcyar	255	sebenarnya

Field singkat berfungsi sebagai penyimpanan kata yang disingkat dan *field* eyd berfungsi sebagai penyimpanan kata yang sesuai dengan EYD pada Bahasa Indonesia.

Dalam penelitian ini sistem aplikasi pembaca SMS dibangun dengan susunan komponen yang terdiri dari komponen utama, yakni proses segmentasi, proses *stemming*, proses pemenggalan suku kata dan proses pembangkitan prosodi.

6.6. Implementasi

Program ini terdiri atas beberapa class, yang memuat beberapa modul, adapun modul yang digunakan adalah :

Tabel 5. Implementasi Modul

No.	Fungsi	Keterangan
1.	tambahdata()	Fungsi yang digunakan untuk menambahkan data singkatan dan data kata EYD ke dalam <i>database</i>
2.	pecahkata()	Fungsi yang digunakan untuk melakukan proses segmentasi kalimat dan kata.
3.	cek_prefiks()	Fungsi yang digunakan untuk menangani kasus kata yang memiliki imbuhan baik diawal, akhir ataupun campuran dan melakukan proses pengecekan terhadap <i>database</i>
4.	cek_kombinasi()	Fungsi yang digunakan untuk memeriksa kembali kata yang telah terlanjur dilakukan proses <i>stemming</i> dan kembali melakukan proses pengecekan terhadap <i>database</i>
5.	num2string()	Fungsi yang digunakan untuk menggantikan angka yang ada dalam teks menjadi karakter yang bisa dibacakan oleh sistem
6.	pecah()	Fungsi yang digunakan untuk menangani pemenggalan suku kata setelah sebelumnya dilakukan normalisasi teks oleh fungsi <i>stemming</i> dan num2string

7.	get_prosodi()	Fungsi yang digunakan agar suara yang akan diucapkan menjadi lebih alamiah.
8.	gspeech()	Fungsi yang digunakan untuk menjalankan men-generate speech yang akan mengeksekusi file input.pho yang telah diisikan dengan prosodi
9.	speech()	Fungsi yang digunakan untuk memutar suara yang dihasilkan proses sebelumnya.

Antar muka berguna untuk memudahkan *users* untuk menggunakan aplikasi ini. Adapun fungsi *keypad* atau tombol di perangkat *handphone* berguna untuk membacakan urutan SMS. Berikut daftar untuk berinteraksi dengan perangkat :

Tabel 6. Tombol Interaksi Pengguna

Tombol	Aksi
Tombol Options	Menambahkan isi data singkatan dan kata EYD
Tombol Exit	Keluar dari Aplikasi
Tombol „#“	Petunjuk Penggunaan
Tombol „*“	About Aplikasi
Tombol „1“	Membaca urutan SMS terbaru urutan ke-1
Tombol „2“	Membaca urutan SMS terbaru urutan ke-2
Tombol „3“	Membaca urutan SMS terbaru urutan ke-3
Tombol „4“	Membaca urutan SMS terbaru urutan ke-4
Tombol „5“	Membaca urutan SMS terbaru urutan ke-5
Tombol „6“	Membaca urutan SMS terbaru urutan ke-6
Tombol „7“	Membaca urutan SMS terbaru urutan ke-7
Tombol „8“	Membaca urutan SMS terbaru urutan ke-8
Tombol „9“	Membaca urutan SMS terbaru urutan ke-9
Tombol „0“	Memainkan suara SMS yang telah dipilih sebelumnya

6.7. Uji Coba

Tabel 7. Pengujian Sistem

Bentuk Pengujian	Metode Pengujian	Data Uji yang digunakan	Keterangan (Tujuan)
Pengujian Modul Program	<i>Black Box Testing</i>	Data <i>dummy</i> yang menginterpretasikan data yang sebenarnya	Menguji kebenaran dari setiap modul fungsi yang dikerjakan

7. Pengujian

Studi kasus yang diujicobakan terhadap sistem ini diantaranya dengan memberikan SMS masukan dengan

variasi kata yang berbeda-beda, skenario pengujian dengan memperdengarkan hasil pembacaan SMS ke *user* dan menuliskan apa yang mereka dengar.

Tabel 8. Studi Kasus

No	Isi SMS	Kecepatan Eksekusi (detik)	Penulisan Benar (orang)	Penulisan Salah (orang)	Keterangan
1	Lg dkmps nih.. bntn lg plgnya! Bsk jg mau kkmps lg. mhn maaf lahir batin ya, Smg kt smua dprtemukan kmbli	20	8	2	Ekspresi untuk tanda seru kurang terdengar
2	Khusus Plgn Indosat, discount menarik Merchant Indosat Square di Istana Plaza. Info : hubungi 100	15	7	3	Penggunaan kata asing membuat kesalahan dalam pengucapan, karena belum dimasukkan ke dalam <i>database</i>
3	Saya mau makan	10	10	0	-
4	Selamat malam, lg apa?	10	10	0	Eskpresi kalimat tanya terdengar
5	Btmn dgn siapa?	10	7	3	Kata "Btmn" dieja karena tidak memenuhi aturan <i>stemming</i> meski kata "tmn" terdapat dalam <i>database</i>

7.1. Studi Kasus

Studi kasus yang diujicobakan terhadap sistem ini diantaranya dengan memberikan SMS masukan dengan

variasi kata yang berbeda-beda, skenario pengujian dengan memperdengarkan hasil pembacaan SMS ke *user* dan menuliskan apa yang mereka dengar.

Tabel 9. Studi Kasus

No	Isi SMS	Kecepatan Eksekusi (detik)	Penulisan Benar (orang)	Penulisan Salah (orang)	Keterangan
1	Lg dkmps nih.. bntn lg plgnya! Bsk jg mau kkmps lg. mhn maaf lahir batin ya, Smg kt smua dprtemukan kmbli	20	8	2	Ekspresi untuk tanda seru kurang terdengar
2	Khusus Plgn Indosat, discount menarik Merchant Indosat Square di Istana Plaza. Info : hubungi 100	15	7	3	Penggunaan kata asing membuat kesalahan dalam pengucapan, karena belum dimasukkan ke dalam <i>database</i>
3	Saya mau makan	10	10	0	-
4	Selamat malam, lg apa?	10	10	0	Eskpresi kalimat tanya terdengar
5	Btmn dgn siapa?	10	7	3	Kata "Btmn" dieja karena tidak memenuhi aturan <i>stemming</i> meski kata "tmn" terdapat dalam <i>database</i>

7.2. Analisa Hasil

Hasil eksperimen yang menentukan keberhasilan sistem dapat dilihat dari beberapa hal, diantaranya:

1. *Intelligibility*, faktor ini yang diharapkan *users* agar keluaran suara yang dihasilkan sistem dapat dimengerti dengan mudah. Faktor ini bisa dikatakan berhasil karena suara yang dikeluarkan sangat jelas.
2. *Variability*, faktor yang dapat membuat perubahan suara seperti kecepatan, nada, tekanan kata dan sebagainya. Faktor ini berhasil dilakukan karena adanya tahapan pembangkitan prosodi untuk mengatur nada dan tekanan suara.
3. *Naturalness*, tingkat kealamian sangat diharapkan oleh *users*, dengan hasil suara yang memiliki tingkat kealamian yang tinggi sehingga *users* akan mudah mencerna maksud dari tulisan yang ada sehingga suara yang dihasilkan dapat disamakan dengan orang yang sedang berbicara. Dari hasil implementasi pembangkitan prosodi yang dikemukakan oleh bapak Arry Akhmad Arman sistem ini cukup berhasil dikarenakan sistem dapat mengenali segmen kalimat, walaupun hasilnya masih terbata-bata untuk beberapa kata yang diucapkan.
4. Kecepatan eksekusi, seberapa cepat sistem dapat bekerja. Sehingga kecepatan juga sangat dibutuhkan dalam mengeksekusi ketika *users* menekan tombol dengan fungsi tertentu dan berapa lama sistem akan menjalankan perintah itu. Dalam hal kecepatan, sistem ini dinilai kurang *responsive* dalam menangani pembacaan SMS, dalam penelitian sistem memerlukan waktu rata-rata sekitar 15 detik untuk mendapatkan suara yang dihasilkan.

8. Simpulan

Berdasarkan hasil penelitian dapat disimpulkan beberapa hal sebagai berikut :

1. Untuk membuat program yang dapat membacakan isi SMS dengan kasus terdapat beberapa kata singkatan, algoritma *stemming* yang digunakan sekarang cukup mampu menangani masalah pencarian kata dasar dengan syarat semua aturan dipenuhi dengan baik. *Finite State Automata* dapat digunakan dengan baik untuk pemenggalan suku kata, walaupun ada beberapa yang tidak bisa diaplikasikan karena ketidakkonsistennya Bahasa Indonesia.
2. Model Intonasi yang dibuat belum dapat menghasilkan tingkat kealamian untuk dapat dikategorikan *naturalness*.
3. Hasil implementasi sistem pada emulator dapat berjalan lancar, baik yang tanpa melewati pemeriksaan *database* maupun yang melewati

pemeriksaan *database* dengan catatan kata yang dimaksud terdapat dalam *database*.

9. Saran

Untuk mengembangkan penelitian ini, dapat dipertimbangkan berapa hal sebagai berikut:

1. Perlu membuat sendiri pembangkit intonasi dalam bahasa Indonesia pada sistem operasi Android
2. Penanganan kata yang mengandung singkatan tidak perlu lagi mengandalkan *database* sehingga sistem dapat lebih cepat mengerjakan perintah.
3. Pembuatan *speech synthesizer* sendiri, dan tidak menggunakan pembangkit suara yang sudah ada, namun membuat sendiri.
4. Tambahan fitur pengiriman SMS dengan memanfaatkan aplikasi *Screen Reader* Bahasa Indonesia.

Referensi

- [1] Arman. A. Akhmad, 2003. *Konversi dari Teks ke Ucapan*, Departemen Teknik Elektro ITB, Bandung.
- [2] Arman. A. Akhmad, 2003, *Pengembangan Aplikasi Teknologi Bahasa Studi Kasus Pengembangan Text To Speech Bahasa Indonesia*, Departemen Teknik Elektro ITB, Bandung.
- [3] Arman. A. Akhmad, 2003, *Proses Pembentukan dan Karakteristik Sinyal Ucapan*, Departemen Teknik Elektro ITB, Bandung.
- [4] Dutoit. Thierry. 1997. *An Introduction to Text-to-Speech Synthesis*, Kluwer Academic Publisher, Dordrecht.
- [5] Dutoit. Thierry. 1993. *High Quality Text-to-Speech Synthesis of the French Language*. Faculte Polytechnique de Mons, Belgia.
- [6] Pelton, Gordon E. 1993. *Voice Processing*. Newyork: McGraw-Hill.
- [7] Pressman, R.S. 2001. *Software Engineering: A Practitioners Approach 5th Edition*. Newyork: McGraw-Hill.
- [8] Subali, Muhammad. 2004. *Model Linier Dinamik Sebagai Dasar Penyelesaian Diphone Pada Sistem Pensitesaan Suara Concatenative Dalam Bahasa Indonesia*. Universitas Gunadarma, Jakarta.