Thomas Anung Basuki

Pengenalan Suku Kata Bahasa Indonesia Menggunakan Finite-State Automata

Intisari

Tulisan ini membahas bagaimana mengenali suku-suku kata dalam kalimat Bahasa Indonesia menggunakan Finite State Automata, yaitu suatu model dari mesin pengenal yang mampu mengenali kelas bahasa yang disebut Bahasa Reguler

Abstract

This paper discusses how to recognize syllables in Indonesian sentences using Finite State Automata, a model of a recognizer which is able to recognize a language class called Regular Language

I. Latar Belakang

Bahasa Indonesia adalah bahasa nasional Bangsa Indonesia. Sebagai orang Indonesia, kita sering merasa tidak perlu lagi mempelajari Bahasa Indonesia. Ternyata banyak hal dalam Bahasa Indonesia yang belum kita kuasai.

Ketika kita masih duduk di bangku Sekolah Dasar, kita pernah belajar tentang suku kata dalam mata pelajaran Bahasa Indonesia. Ketika penulis mencoba memikirkan lagi tentang suku kata yang pernah diajarkan dulu, timbul satu pertanyaan yaitu apakah gunanya kita mempelajari tentang pemenggalan suku-suku kata ?

Setidaknya ada dua guna pemenggalan suku kata yang dapat penulis pikirkan, yaitu :

a) jika kita melihat peranan suku kata dalam bahasa tulisan, maka pemenggalan suku kata perlu dilakukan ketika kata yang kita tulis panjangnya melebihi batas kanan kertas,
b) jika kita melihat peranan suku kata dalam bahasa lisan, maka pemenggalan suku kata diperlukan untuk mengetahui bagaimana cara mengucapkan suatu kata.

Pada tulisan ini, pemenggalan suku kata ditujukan untuk mengetahui bagaimana pengucapan suatu kata Bahasa Indonesia. Dalam hal ini, suku kata dianggap sebagai satuan bahasa lisan. Hal ini perlu dipertegas karena kadangkala pemenggalan suku kata dalam bahasa lisan dan bahasa tulisan berbeda.

Dalam Matematika Diskret terdapat satu cabang ilmu yang khusus mempelajari tentang bahasa, yaitu Teori Bahasa Formal. Bahasa yang dibahas pada Teori Bahasa Formal adalah bahasa tulisan. Karena tulisan ini lebih ditujukan pada pengenalan suku

kata pada bahasa lisan, maka tentu ada beberapa hal yang tidak bisa didekati menggunakan Teori Bahasa Formal. Keuntungan dari penggunaan Teori Bahasa Formal adalah kita bisa memodelkan pengenalan suku kata menggunakan Finite State Automata, yaitu suatu mesin abstrak yang dapat mengenali bahasa.

II. Dasar Teori

II.1. Teori Bahasa Formal

Bahasa dalam bentuk tulisan sebenarnya terdiri atas simbol-simbol satuan yang jika dikombinasikan akan mempunyai arti yang berbeda-beda. Simbol-simbol yang bisa dipergunakan dalam sebuah bahasa tentunya terbatas jumlahnya, yang membentuk sebuah himpunan dan disebut sebagai **abjad** (*alphabet*).

Kadangkala digunakan istilah karakter yang maknanya sama dengan simbol. Deretan karakter membentuk *string*. **Bahasa** (*language*) didefinisikan sebagai himpunan semua string yang dapat dibentuk dari suatu abjad. Kaidah/aturan pembentukan kata/kalimat disebut **tata bahasa** (*grammar*).

Jika x adalah suatu string maka $|x|$ adalah panjang x yaitu jumlah simbol yang terdapat dalam x. String kosong, dinotasikan dengan ε, adalah string dengan panjang 0 (string yang tidak mempunyai simbol di dalamnya).

Karena bahasa adalah sebuah himpunan dari string, maka untuk mendefinisikan suatu bahasa bisa dilakukan dengan menuliskan semua string yang menjadi anggotanya. Bagaimana kita bisa melakukannya jika jumlah string yang menjadi anggota bahasa tersebut banyak sekali atau bahkan tidak berhingga ? Pada Teori Bahasa Formal, hal ini dilakukan dengan mendefinisikan tata bahasanya.

Tata Bahasa $G = (T,N,S,P)$, di mana

- T adalah himpunan berhingga simbol-simbol terminal
- N adalah himpunan berhingga simbol-simbol non terminal
- S adalah simbol awal, $S \in N$
- P adalah himpunan berhingga aturan produksi yang setiap elemennya berbentuk $\alpha \rightarrow \beta$, $\alpha,\beta \in (T \cup N)^+$, α harus berisi minimal 1 simbol non terminal

Sentential form adalah semua string yang dapat diturunkan dari simbol awal S dengan menggunakan aturan produksi P. *Kalimat* (**sentence**) adalah sentential form yang tidak mengandung simbol non terminal. Bahasa yang dihasilkan dari G dinotasikan dengan $L(G)$, yaitu himpunan kalimat yang dapat diturunkan dari S dengan menggunakan P.

II.2. Teori Automata

Berasal dari bahasa Yunani *automatos*, yang berarti sesuatu yang bekerja secara otomatis (mesin). Dalam tulisan ini akan dipergunakan istilah *automaton* sebagai bentuk tunggal dan *automata* sebagai bentuk jamak.

Teori Automata adalah teori tentang mesin abstrak yang :

- bekerja sekuensial
- menerima input
- mengeluarkan output

Pengertian mesin di tulisan ini, bukan hanya mesin elektronis/mekanis saja melainkan segala sesuatu (termasuk perangkat lunak) yang memenuhi ketiga ciri di atas. Penggunaan automata pada perangkat lunak terutama pada pembuatan kompiler bahasa pemrograman.

Setelah kita mengetahui definisi bahasa dan automata, pertanyaan selanjutnya adalah apakah hubungan antara teori automata dan bahasa formal ? Secara garis besar ada dua fungsi automata dalam hubungannya dengan bahasa, yaitu :

- fungsi automata sebagai pengenal (*RECOGNIZER*) string-string dari suatu bahasa, dalam hal ini bahasa sebagai masukan dari automata
- fungsi automata sebagai pembangkit (*GENERATOR*) string-string dari suatu bahasa, dalam hal ini bahasa sebagai keluaran dari automata

Dalam tulisan ini, pembahasan akan ditekankan pada fungsi pertama dari automata.

Untuk mengenali string-string dari suatu bahasa, akan dimodelkan sebuah automaton yang memiliki komponen sebagai berikut :

- pita masukan, yang menyimpan string masukan yang akan dikenali;
- kepala pita (*tape head*), untuk membaca/menulis ke pita masukan;
- Finite State Controller (FSC), yang berisi status-status dan aturan-aturan yang mengatur langkah yang dilakukan oleh automaton berdasarkan status setiap saat dan simbol masukan yang sedang dibaca oleh kepala pita;
- pengingat (*memory*), untuk tempat penyimpanan dan pemrosesan sementara

Automaton pengenal, setelah membaca string masukan dan melakukan langkah-langkah pemrosesan yang diperlukan, akan mengeluarkan keputusan apakah string tersebut dikenali atau tidak.

Konfigurasi adalah suatu mekanisme untuk menggambarkan keadaan suatu mesin pengenal , yang terdiri atas :

- status FSC
- isi pita masukan dan posisi kepala pita
- isi pengingat

Mesin pengenal bersifat *deterministik* bila dalam setiap konfigurasi, hanya ada satu kemungkinan yang dapat dilakukan mesin, jika tidak mesin pengenal bersifat *non deterministik*.

II.3. Klasifikasi Bahasa Menurut Chomsky

Untuk menyelesaikan suatu masalah, mula-mula harus dikenali dulu dengan baik masalah yang sebenarnya dihadapi. Salah satu caranya adalah dengan mengklasifikasikan masalah tersebut. Dengan demikian dapat dikenali ciri-ciri masalah tersebut, dan dapat lebih difokuskan penyelesaian masalah tersebut pada ciri-ciri yang berhasil dikenali. Pada masalah bahasa, klasifikasi tersebut sudah pernah dilakukan. Chomsky membagi bahasa menjadi 4 kelas berdasarkan tata bahasanya. Keempat kelas tersebut adalah :

1. **Regular Grammar**/**Regular Language** (**RG**/**RL**)
   Mesin Pengenal : **Finite State Automata (FSA)**
   Dapat dibagi menjadi dua subkelas :
   a) *left linear grammar/left linear language* (LLG/LLL), jika P berbentuk $A \rightarrow Bx|x$
   b) *right linear grammar/right linear language* (RLG/RLL), jika P berbentuk $A \rightarrow xB|x$, di mana $A,B \in N$ dan $x \in T^*$
2. **Context Free Grammar**/**Context Free Language** (**CFG**/**CFL**)
   Mesin Pengenal : **Push Down Automata (PDA)**
   Ciri-ciri : bentuk produksi $P \quad A \rightarrow \beta$, di mana A $\in N$ dan $\beta \in (T \cup N)^*$
3. **Context Sensitive Grammar**/**Context Sensitive Language** (**CSG**/**CSL**)
   Mesin Pengenal : **Linear Bounded Automata (LBA)**
   Ciri-ciri : bentuk produksi $P \quad \alpha \rightarrow \beta$, $\alpha,\beta \in (T \cup N)^+$, $|\alpha| \leq |\beta|$
4. **Unrestricted Grammar**/**Unrestricted Language** (**UG**/**UL**)
   Mesin Pengenal **: Turing Machine ™**
   Ciri-ciri : bentuk produksi $P \quad \alpha \rightarrow \beta$, $\alpha,\beta \in (T \cup N)^+$

Pada tulisan ini hanya akan dibahas Bahasa Reguler (kelas bahasa yang pertama), yang merupakan kelas bahasa yang paling sederhana, beserta mesin pengenalnya (FSA). Dengan menggunakan FSA sudah cukup mampu mengenali suku-suku kata Bahasa Indonesia.

II.4. F S A

Setiap jenis automata mempunyai keunikan yang membuatnya berbeda fungsinya dengan automata yang lain. Berikut ini akan dibahas sifat-sifat FSA :

- pita masukan hanya bisa dibaca, berisi string yang berasal dari suatu abjad,

- setelah membaca satu simbol pada pita, kepala pita akan maju ke posisi simbol berikutnya,
- kepala pita tidak bisa mundur,
- mempunyai sejumlah berhingga status, setiap saat FSA berada pada status tertentu.

Setiap FSA bisa diasosiasikan dengan sebuah diagram transisi, yaitu suatu geraf berarah sebagai berikut :

- setiap simpulnya mewakili setiap status pada FSA
- jika ada transisi dari status *p* ke status *q* pada input *a*, maka ada busur dari *p* ke *q* berlabel *a*
- status awal ditandai dengan kata START, status akhir ditandai dengan 2 lingkaran

Jadi fungsi dari diagram transisi adalah untuk menggambarkan cara kerja suatu FSA.

II.5. Deterministic FSA (DFSA)

Pada bab 2.2 sudah disebutkan bahwa sebuah automaton bisa bekerja secara deterministik ataupun non deterministik. Setiap bahasa reguler bisa dikenali oleh DFSA. Secara formal suatu DFSA dinyatakan dengan $(Q, \Sigma, \delta, q_0, F)$ di mana

Q = himpunan berhingga status

Σ = himpunan berhingga simbol masukan (alfabet)

δ = fungsi transisi yang memetakan $Q \times \Sigma$ ke Q

q_0 = status awal, $q_0 \in Q$

F = himpunan status akhir, $F \subseteq Q$

Cara kerja :

- mula-mula DFSA akan berada pada status q_0, kepala pita pada simbol pertama pada pita,
- selanjutnya kepala pita akan membaca simbol-simbol dari pita dan bergeser maju,
- untuk setiap simbol, DFSA akan berpindah status sesuai dengan fungsi δ,
- proses akan berakhir bila simbol masukan pada pita sudah habis,
- bila pada akhir proses dicapai status akhir maka string masukan diterima (dikenali sebagai string dari bahasa regular), dan bila tidak maka string masukan ditolak (tidak dikenali).

II.6. Bahasa Indonesia

Bahasa Indonesia mengenal bahasa tulisan maupun bahasa lisan. Kadangkala terdapat beberapa perbedaan dalam kedua jenis bahasa ini.

Dalam bahasa lisan, dikenal istilah *fonem*, yang merupakan kesatuan bahasa terkecil yang dapat membedakan arti. Dalam bahasa tulisan, *fonem* dilambangkan dengan huruf. Dengan kata lain, huruf adalah tulisan dari *fonem*. Seringkali istilah fonem disamakan dengan huruf, padahal tidak selamanya berlaku demikian.

Fonem dibagi menjadi vokal dan konsonan. Bahasa Indonesia mengenal 5 vokal yaitu : a, e, i ,o, u, dan 25 konsonan yaitu : b, c, d ,f , g , h, j , k, kh, l, m, n, ng, ny, p, q, r, s, sy, t, v, w, x, y, z. Konsonan kh, ng, ny dan sy adalah contoh fonem yang terdiri atas dua huruf. Selain itu dikenal pula istilah diftong, yaitu gabungan 2 vokal yang membentuk kesatuan bunyi, yaitu : au, ai, oi. Pada beberapa buku referensi, diftong digolongkan sebagai vokal pula.

Abjad

Abjad yang digunakan dalam Bahasa Indonesia terdiri atas 52 huruf, yaitu 26 huruf besar (A sampai dengan Z) dan 26 huruf kecil (a sampai dengan z). Selain itu dikenal 10 simbol untuk angka yaitu 0 sampai dengan 9.

Persukuan

Menurut Kamus Besar Bahasa Indonesia suku kata adalah struktur yang terjadi dari satu atau urutan fonem yang merupakan bagian kata. Setiap suku kata ditandai dengan sebuah vokal (termasuk diftong). Bahasa Indonesia mengenal beberapa pola umum suku kata, ialah :

a) V[1] *a*-nak, ba-*u*
b) VK *an*-da, da-*un*
c) KV *se*-bab, man-*di*
d) KVK *lan*-tai, ma-*kan*
e) KKV *pra*-ha-rai, sas-*tra*
f) KKVK *frik*-si, kon-*trak*
g) VKK *eks*, *ons*

[1] V berarti vokal dan K berarti konsonan

h) KVKK *pers*, kon-*teks*
i) KKVKK kom-*pleks*
j) KKKV in-*stru*-men
k) KKKVK *struk*-tur

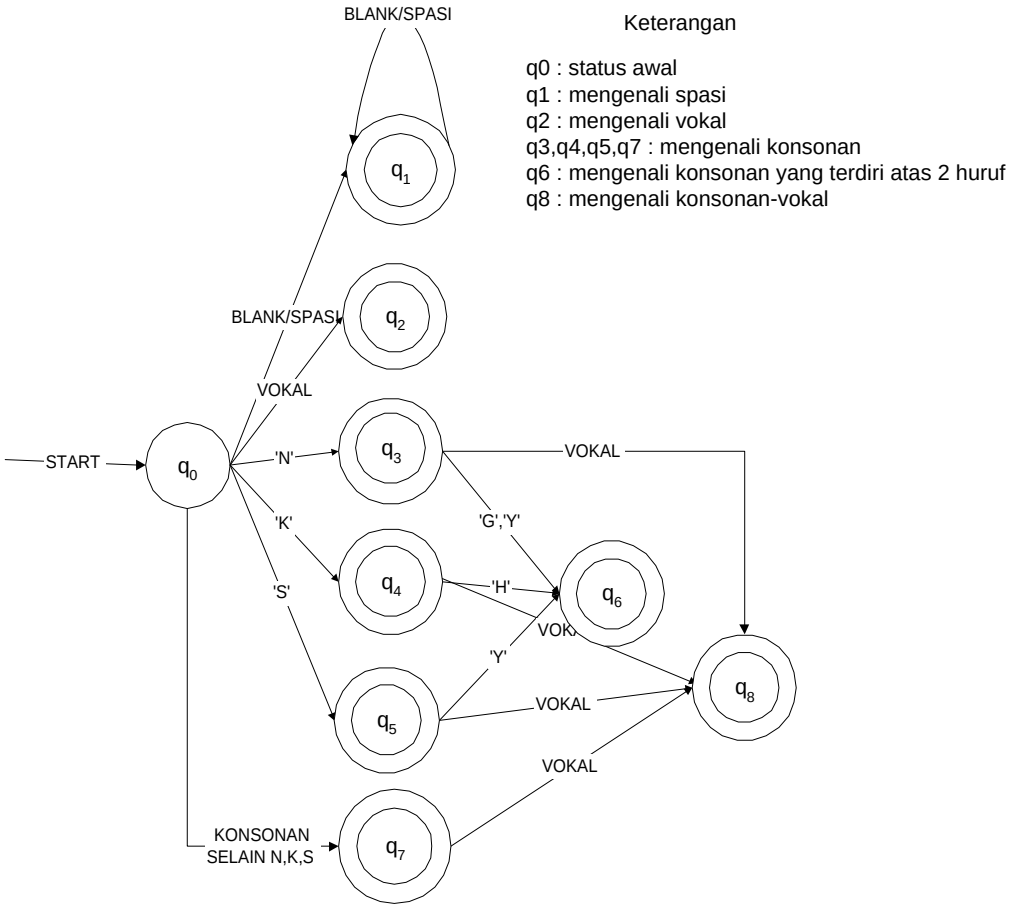
Untuk memenggal suku kata, dapat digunakan pedoman berikut ini :

a) Kalau di tengah kata terdapat dua vokal berturutan (selain diftong), pemisahan dilakukan di antara kedua vokal tersebut.
b) Kalau di tengah kata terdapat konsonan di antara dua vokal, pemisahan dilakukan sebelum konsonan tersebut.
c) Kalau di tengah kata terdapat dua konsonan atau lebih, pemisahan dilakukan setelah konsonan pertama.
d) Imbuhan dan partikel yang biasanya ditulis serangkai dengan kata dasar, pada penyukuan dipisahkan.

Pedoman a sampai dengan c, bisa diterapkan pada perancangan perangkat lunak ini. Pedoman d tidak bisa diterapkan, karena pedoman tersebut lebih tepat diterapkan pada bahasa tulisan. Pada bahasa lisan tidak dikenal imbuhan maupun partikel, yang ada hanyalah bagaimana suku kata tersebut diucapkan.

III. Perancangan Perangkat Lunak

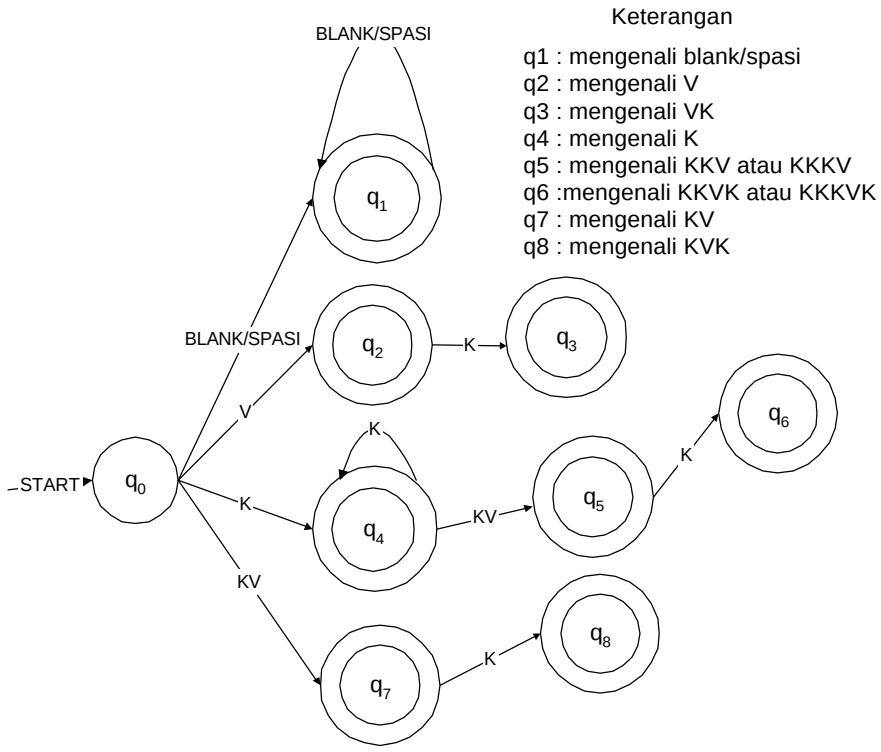
Pada Teori Bahasa Formal, setiap bahasa memiliki suatu aturan tata bahasa yang baku dan konsisten. Dalam kenyataannya bahasa yang dipakai oleh manusia merupakan konvensi/kesepakatan dari para pemakai bahasa tersebut. Akibatnya tata bahasa yang digunakan seringkali tidak konsisten. Oleh karena itu dapat dimaklumi jika Teori Bahasa Formal tidak dapat memodelkan bahasa manusia secara sempurna.



Gambar 1. Diagram Transisi FSA tingkatan pertama

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 tesebut 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**).



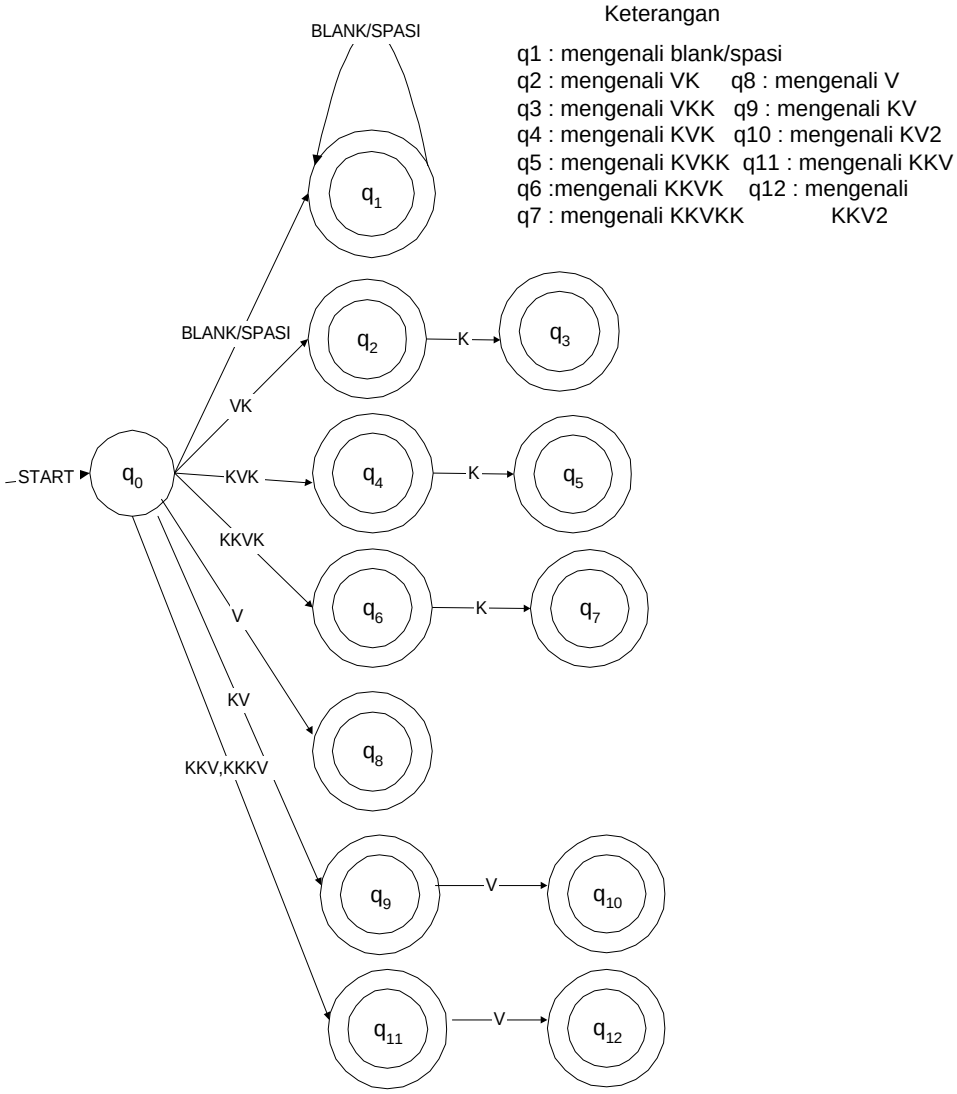
Gambar 2. Diagram Transisi FSA tingkatan kedua

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 b 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 c pemisahan suku kata).

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 mengenalinya. Jika FSA ini menemukan pola VK-K, KVK-K dan KKVK-K, dia akan mengenalinya sebagai pola suku kata VKK, KVKK dan KKVKK.



Gambar 3. Diagram Transisi FSA tingkatan 3

Selain itu FSA pada tingkatan 3 dapat mengenali diftong. Dalam Bahasa Indonesia dikenal 3 macam diftong yaitu : ‘au’,’ai’ da ‘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 ketakkonsistenan ini maka tidak mungkin mengenali diftong secara sempurna tanpa melakukan analisis semantik. Analisis semantik tidak dapat dimodelkan dengan FSA.

Pada tulisan ini pengenalan diftong dibatasi pada hal-hal yang dapat dimodelkan oleh FSA. Semua pasangan vokal ‘au’, ‘ai’ dan ‘oi’ akan dianggap sebagai diftong.

IV. Penutup

Tulisan ini bertujuan untuk mengenali persukuan kata dalam kalimat Bahasa Indonesia. Dengan mengenali suku kata dalam bahasa lisan, maka dapat dikembangkan perangkat lunak untuk mengubah teks tertulis menjadi suara. Pengenalan suku kata yang dilakukan pada tulisan ini masih mengandung kelemahan, yaitu ketidakmampuannya untuk membedakan diftong dengan gabungan dua vokal. Kritik dan saran tentang tulisan ini akan penulis terima dengan senang hati.

Daftar Pustaka

PUSAT PEMBINAAN DAN PENGEMBANGAN BAHASA DEPARTEMEN PENDIDIKAN DAN KEBUDAYAAN [1979]. *Pedoman Umum Ejaan Bahasa Indonesia Yang Disempurnakan*. PN Balai Pustaka, Jakarta.

BADUDU, J.S [1978]. *Pelik-Pelik Bahasa Indonesia*. Pustaka Prima, Bandung.

BADUDU, YUS [1979]. *Membina Bahasa Indonesia Baku, Seri 1*. Pustaka Prima, Bandung.

AHO, A.V., dan J.D. ULLMAN [1972]. *The Theory of Parsing, Translation, and Compiling Volume I : Parsing.* Prentice-Hall, Englewood Cliffs, New Jersey.

HOPCROFT, J.E. dan J.D. ULLMAN [1979]. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishing Company, Reading, Massachusetts.

TREMBLAY, J.P. dan P.G. SORENSON [1985]. *The theory and Practice of Compiler Writing*. McGraw-Hill, New York.

Penulis

Thomas Anung Basuki adalah dosen Jurusan Ilmu Komputer, Universitas Katolik Parahyangan.

e-mail: anung@home.unpar.ac.id