

ISSN : 1979-7656

Vol. 2 No.1 Juli 2009

TEKNOMATIKA

JURNAL INFORMATIKA DAN KOMPUTER

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER
JENDERAL ACHMAD YANI
YOGYAKARTA**

TEKNOMATIKA

JURNAL INFORMATIKA DAN KOMPUTER

Penanggung Jawab:

Ketua STMIK Jenderal Achmad Yani

Edhy Tri Cahyono, S.Si., MM.

Pemimpin Redaksi:

Damar Widodo S.Si

Dewan Redaksi:

Ari Cahyono, ST., M.T

Choerun Asnawi, S.Kom

Dayat Subekti, S.Si.

Alamat Redaksi:

STMIK Jenderal Achmad Yani

Jln. Beo 35 Demangan Baru Yogyakarta

Telpon (0274) 552489, 552851 faks. (0274) 557228

e-mail: teknomatika@stmikayani.ac.id

Jurnal TEKNOMATIKA diterbitkan oleh STMIK Jenderal Achmad Yani Yogyakarta, berisi artikel hasil penelitian atau kajian ilmiah di bidang informatika dan komputer dalam arti yang seluas-luasnya. Redaksi menerima tulisan ilmiah yang berkaitan dengan bidang tersebut.

DAFTAR ISI

Bahasa Bebas Konteks Untuk Proses Transliterasi Latin Ke Jawa (Suprihatin)	1 - 8
Implementasi <i>Search Engine Optimization (SEO)</i> Pada Pemasaran Usaha Kecil Menengah (UKM) Studi Kasus pada Gealgeol.com (Arif Himawan)	9-28
Pengambilan Data Buku di Internet Melalui ISBN Menggunakan Ajax dan PHP (Abdul Kadir)	29-36
Perancangan dan Implementasi Sistem Informasi Kerja Praktek dan Tugas Akhir Berbasis Teknologi <i>Wireless Application Protocol (WAP)</i> Studi Kasus <i>Fakultas Teknik Univeristas Janabadra</i> (Fatsyahrina Fitriastuti)	37-50
Konsep Dan Penerapan ERP Pada Perancangan Sistem Informasi Akademik (Fajrillah)	51-58
Analisis Nilai Link Popularity Untuk Mengukur Tingkat Popularitas Dengan Studi Kasus www.stmikayani.ac.id (Damar Widodo)	59-64
Sistem Pencarian Data Buku di Perpustakaan Menggunakan Perintah Suara (<i>Voice</i>) (Dayat Subekti)	65-86
Kesalahan Dalam Desain Web (Choerun Asnawi)	87-96

BAHASA BEBAS KONTEKS UNTUK PROSES TRANSLITERASI LATIN KE JAWA

Suprihatin
Program Studi Sistem Informasi
Fakultas Matematika dan Ilmu Pengetahuan
Universitas Ahmad Dahlan
pakprih@yahoo.co.id

Abstrak

Pelajaran menulis aksara jawa sangat dirasakan susah oleh siswa-siswa, ataupun orang-orang yang pernah mempelajari. Penelitian ini bertujuan untuk membuat aturan produksi atau tatabahasa bebas konteks untuk proses transliterasi latin ke jawa. Tatabahasa bebas konteks dipergunakan memparser kalimat menjadi sukukata-sukukata, selanjutnya ditransliterasikan ke aksara jawa.. Hasil penelitian ini berupa tatabahas/aturan produksi yang dapat mentransliterasikan latin ke jawa. Agoritma aturan produksi akan dibuat, dan algoritma ini dapat dikodingkan sehingga dapat dicobakan dikomputer. Input program nantinya berupa tulisan latin dan oput berupa tulisan jawa.

Kata kunci: Parser, Tatabahasa, Bebas Konteks, Transliterasi, Jawa

PENDAHULUAN

Sudah bukan rahasia umum lagi bahwa anak-anak sekolah khususnya SD, dan SMP di DIY, Jawa Tengah, dan Jawa Timur kesulitan dalam mempelajari tulisan jawa. Adanya komputer sebagai alat bantu sangat penting bagi penggunaanya, baik untuk pengetikan, olah data, sistem informasi, pendidikan, ataupun untuk penghitungan-penghitungan matematika. Mempergunakan komputer dimungkinkan dibuat alat untuk mempermudah penulisan tulisan jawa. Oleh karena itu penelitian ini akan membuat algoritma program komputer untuk membantu penulisan aksara jawa.

Aksara jawa pada dasarnya mempunyai 20 aksara biasa disebut dengan carakan. Sandangan ada 4 macam yaitu: Sandangan Swara (a, i, u, e, é, o), Panyigeging Wanda(r, ng) , Wiyanjana (r, y) , dan Panjingan (l, w). Dua puluh aksara dan sandangan akan membentuk banyak sekali ligatur (gabungan satu atau lebih aksara yang berbeda). Ligatur tidak dapat ditulis dengan satu-satu aksara (karakter) tetapi harus dibuat satu karakter untuk mewakili ligature tersebut, sebagai misal: yꦧꦸ pyar) yang merupakan gabungan dari aksara pa, pengkal, dan panyigeging wanda r (layar)

Tulisan jawa banyak mengandung ligatur-ligaturnya, cara penulisannya juga banyak yang tidak konsisten, sebagai misal huruf mati (*sigegan*) kadang memakai *layar* untuk r, memakai cecak untuk ng, ataupun memakai pangkon, ataupun juga dapat memakai pasangan. Ketidakkonsistenan yang lain masih banyak, sebagai misal ra disandangi e akan berubah ligaturnya bukan gabungan aksara ra dan sandangan e tetapi menjadi pa ceret, begitu juga aksara la jika disandangi e akan menjadi nga lelet.

Finite State Automata (FSA) dapat dipergunakan untuk transliterasi aksara jawa ke latin (Suprihatin, 2004). Penelitian ini membahas bagaimana peran FSA membantu dalam transliterasi aksara jawa, yaitu: input berupa aksara-aksara jawa sedangkan output berupa suku kata-suku kata yang dapat dibaca oleh masyarakat umum.

FSA juga dapat digunakan memenggal kalimat menjadi suku kata-suku kata (Suprihatin, 2005). Penelitian ini jika dilanjutkan dapat sebagai alat mengubah teks ke ucapan (*Text to Speech*). Sehingga jika dimasukan kalimat ke program hasilnya adalah ucapan atau bacaan dari kalimat tersebut.

Penelitian kali ini mempergunakan aturan produksi untuk mengenali dan memparser kalimat menjadi sukukata-sukukata, selanjutnya dengan mengacu tabel transliterasi sukukata-sukukata ini ditransliterasikan ke aksara jawa.

RUANG LINGKUP

Ketidakkonsistenan ini membuat penelitian ini terbatas, sehingga ruang lingkungnya meliputi: aksara carakan, sandangan swara, Wiyanjana (r, y) , dan Panjangan (l, w). Pasangan dan bilangan di luar penelitian ini. Tulisan dengan huruf kapital akan dibuat dengan huruf kecil.

PERMASALAHAN

Masalah yang dibahas dalam tulisan ini adalah: Apakah Aturan Produksi yang dibuat nantinya dapat membantu dalam mentransliterasikan ke tulisan jawa. Aturan Produksi nantinya akan dites mempergunakan input yang sesuai ruang lingkungnya sehingga akan menghasilkan output berupa hasil transliterasi sesuai dengan inputnya.

KAJIAN PUSTAKA

Secara formal tata bahasa/*grammar* didefinisikan sebagai sebuah 4 tupel $G(N, \Sigma, S, P)$, dimana G : tata bahasa, N himpunan berhingga Nonterminal, Σ : himpunan berhingga simbol input, S dalam N adalah simbol awal, P : adalah aturan produksi dalam bentuk $\alpha \rightarrow \beta$ (Hopcroft, 1979).

Tatabahasa bebas konteks mempunyai sifat $P \subseteq N^*(\Sigma \cup N)$ (Kelly, 1999).

Ini berarti sisi kiri aturan produksi P hanya mengandung 1 Nonterminal, sedangkan sisi kanan terdiri dari gabungan simbol input dan Nonterminal. Sebuah tata bahasa bebas konteks akan menghasilkan bahasa bebas konteks.

METODE PENELITIAN

Penelitian ini diawali dengan melihat struktur suku kata jawa. Kata-kata yang diawali dengan vokal akan ditambahkan huruf (konsonan) *h*, karena kata-kata yang diawali dengan vokal hasil transliterasinya sama jika diawali dengan konsonan *h*. Pola struktur suku kata terdiri dari 4 yaitu: KV (Konsonan Vokal, contoh: **patri, tari**), KVK (contoh: **krakal, marut**), KKV (contoh: **krakal, mlayu**), KKVK (contoh: **krambil, tyas**). Sesuai ruang lingkupnya maka struktur KVK, dan KKVK tidak diikutkan dalam penelitian ini. Dua struktur yang akan dibahas dalam penelitian ini yaitu: KV, KKV, untuk mempermudah akan diganti dengan istilah jawa yaitu: LS (*Legeno, Swara*), LPS (*Legeno, Panjangan, Swara*). Struktur suku kata jawa LS dan LPS dapat dijadikan satu saja menjadi LPS, dengan P dapat berupa kosong.

Aksara *Legeno* terdiri dari 20 aksara yaitu: *h, n, c, r, d, t, s, w, l, p, dh, j, y, ny, m, g, b, th, ng*. Aksara *Panjangan* terdiri dari: *r, y, w, l*, dan aksara swara terdiri dari: *a, i, u, e, é, o*. Kumpulan huruf *legeno, panjangan, dan swara* yaitu: *h, n, c, r, d, t, s, w, l, p, j, y, m, g, b, a, i, u, e, é, o*. Struktur suku kata dan kumpulan huruf ini akan dibentuk suatu tata bahasa $G(N, \Sigma, S, P)$ dengan:

$N = \{K, L, L_1, L_2, L_3, P, T, S\}$

$\Sigma = \{h, n, c, r, d, t, s, w, l, p, j, y, m, g, b, a, i, u, e, \acute{e}, o, \text{blank}, \text{titik}\}$

$S = K$

$P = \{ K \rightarrow \text{blank}K$

$K \rightarrow \text{titik}$

```

K → LPSTK
L → h|nL1| c | r| d L2| t L3| s| w| l| p| j| y| m| g| b
L1 → y|g|ε
L2 → h|ε
L3 → h|ε
P → r| y| w| l| | ε
S → a| i| u| e| é| o
T → ε // untuk aksi menuliskan transliterasinya
}

```

ALGORITMA

Algoritma terdiri dari 4 algoritma yaitu: Algoritma pengenalan struktur sukukata, Algoritma pengenalan *legeno*, Algoritma Pengenalan Panjang, Algoritma Pengenalan Swara, Algoritma Tulis Hasil Transliterasi.

Algoritma pengenalan struktur sukukata

```

procedure K;
begin
  bacahead;
  if head=' ' then K else
  if head='.' then begin end else
  begin ke := ke-1;L;P;S;tulis; K; end;
end;

```

Perintah *bacahead* adalah membaca input pada posisi head (*ke*). Jika head=blank maka ulangi perintah K. Jika head = titik maka berhenti, jika yang lain posisi head mundur satu dan laksanakan perintah L, P, dan S, kemudian perintah tulis untuk menuliskan *legeno*, *panjang*, dan *swara* sesuai aturan penulisan, selanjutnya perintah K lagi. Jadi procedure K adalah perintah rekursi.

Algoritma pengenalan *legeno*

```

procedure L;
begin
  bacahead;
  case head of
    'h':Legeno:= transl[1];
    'n':begin Legeno:= transl[2]; L1 end;
    'c':Legeno:= transl[3];
    'r':Legeno:= transl[4];
    'k':Legeno:= transl[5];

```

```
'd':begin Legeno:= transl[6]; L2;end;
't':begin Legeno:= transl[7]; L3; end;
's':Legeno:= transl[8];
'w':Legeno:= transl[9];
'l':Legeno:= transl[10];
'p':Legeno:= transl[11];
'j':Legeno:= transl[13];
'y':Legeno:= transl[14];
'm':Legeno:= transl[16];
'g':Legeno:= transl[17];
'b':Legeno:= transl[18];
end;
end;
procedure L1;
begin
  bacahead;
  case head of
    'y':legeno:= transl[15] ;
    'g':legeno:= transl[20];
  else ke := ke-1;
  end;
end;
procedure L2;
begin
  bacahead;
  if head='h'then legeno:= transl[12]
  else ke := ke-1;
end;
procedure L3;
begin
  bacahead;
  if head='h'then legeno:= transl[19]
  else ke := ke-1;
end;
```

Variabel *trans1* adalah string untuk mentrasliterasikan *legeno*. Jika *head* = huruf h maka *legeno* berisi aksara *ha* atau *trans1[1]*. Jika *head* = huruf n maka *legeno* berisi *trans1[2]* = *na* dan akan dilihat karakter/input selanjutnya dengan perintah L1. Jika *head* = huruf c maka *ca* dan seterusnya. Procedure L1

mengenali input ng, dan ny. Procedure L2 mengenali input th. Procedure L3 mengenali input dh.

Algoritma Pengenalan Panjang

```

procedure P;
begin
  bacahead;
  case head of
    'r', 'w', 'l', 'y': Panjang := head;
  else begin Panjang := ' ' ; ke := ke-1; end;
  end;
end;

```

Procedure P adalah mengenali panjang jika input = r, w, l, atau y, jika tidak berarti panjangnya tidak ada atau = blank.

Algoritma Pengenalan Swara

```

procedure S;
begin
  bacahead;
  if head in ['a', 'e', 'i', 'o', 'u', 'é'] then swara := head
  else
  begin
    swara := '=';
  end;
end;

```

Karakter swara yaitu: 'a', 'e', 'i', 'o', 'u', 'é'

Algoritma Tulis Hasil Transliterasi

```

procedure tulis;
var s : string;
begin
  s := '';
  case panjang of
    'x': // panjang x
      case swara of
        'a':begin s := s + 'a';end;
        'i':begin s := s + 'i';end;
        'o':begin s := s + 'o' + 'i';end;
        . . . //dst
      end;
    . . . //dst
  end;
  iowr := iowr + s;
// iowr: variabel global bertipe string untuk menyimpan hasil transliterasi
end;

```

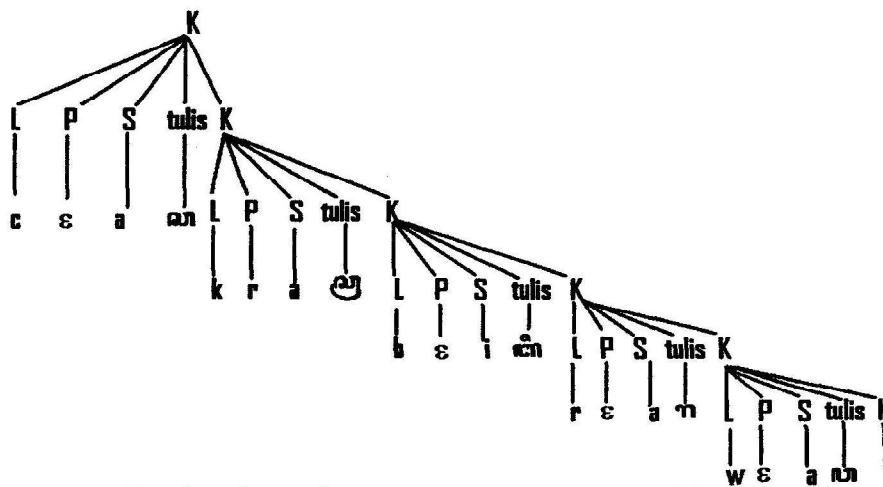
Jika panjang = ra maka jika swara = a maka hasil transliterasi = swara + cakra,
 jika swara = I maka hasil transliterasi = swara + cakra + wulu, jika swara = o
 maka hasil transliterasi = taling + swara + cakra + tarung, dan seterusnya.

HASIL UJI ALGORITMA

Tahab ini akan disimulasikan hasil proses transliterasi dengan beberapa input, Berikut contoh transliterasinya:

Contoh1: Input: cacrabirawa.

Hasil pohon parsernya:

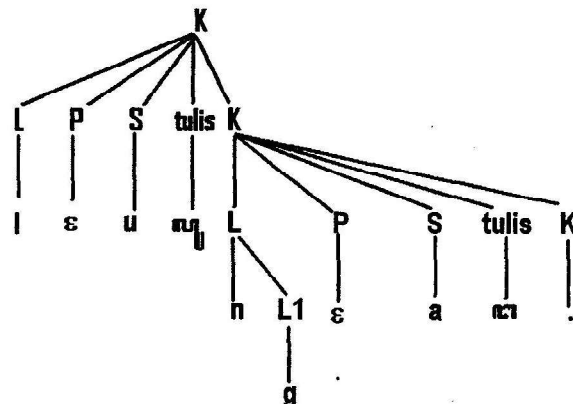


Gambar 1. gambar pohon parser input cacrabirawa.

Hasil transliterasinya: **caꦕꦫꦧꦶꦫꦮ**

Contoh2: Input: lunga.

Hasil pohon parsernya:



Gambar 2. gambar pohon parser input lunga.

Hasil transliterasinya: **ꦭꦸꦁ**

KESIMPULAN DAN SARAN

Telah dibentuk aturan produksi beserta algoritma untuk mentrasliterasikan latin ke jawa. Saran selanjutnya dapat dikodekan dan ditambahkan aturan tentang *segegan* (huruf mati) baik berupa *layar, cecak, wigyan, pasangan, ataupun pangkon*.

DAFTAR PUSTAKA

1. Firar Utdirartatma, 2001, *Teknik Kompilasi*, J&J Learnig, Jakarta
2. Hopcroft JE, and Ullman J.D, 1979, *Introduction to Automata Theory Language and Computation* , Addison Wesley, Massachusets
3. Kelly D., 1999, *Otomata dan Bahasa-Bahasa Formal*, Prenhallindo, Jakarta
4. Suprihatin, 2004, *Aplikasi Finite State Automata untuk Alihaksara tulisan Jawa ke Tulisan Latin*, Proseding Seminar Nasional Universitas Muhammadiyah, Semarang
5. Suprihatin, 2005, *Finite State Automata untuk Parsing (Pemenggalan) Suku Kata dalam Bahasa Indonesia*, Proseding Seminar Nasional Universitas Negeri, Yogyakarta