ISBN : 9-793-688-02-5

Implementasi Program Bahasa Jawa Walikan Dengan Viginere Chiper dan Finite State Automata

Suprihatin
Program Studi Ilmu Komputer Universitas Ahmad Dahlan Yogyakarta
Jalan Prof. Dr. Soepomo, Janturan Yogykarta, Telp (0274) 381523

Abstrak

Tujuan penelitan ini membuat program bahasa jawa walikan. Vigenere Chipher sebagai dasar transformasi konsonan huruf jawa, FSA sebagai pengenal konsonan-konsonan huruf jawa. Hasilnya suatu program dalam Delphi yang dapat mentrasformasikan kalimat dalam bahasa jawa ke bahasa jawa walikan.

Kata kunci: *jawa walikan, Vegenere Chiper , Finite State Automata, Delphi*

I. Pendahuluan

Suatu kalimat adakalanya tidak mau diketahui oleh orang selain yang diajak bicara, maka suatu kalimat akan disandikan atau disamarkan. Criptografi adalah suatu cabang ilmu yang mempelajari sandi. Salah satu metode criptografi yaitu vigenere chiper, yang akan dipakai sebagai metode criptografi konsonan huruf jawa.

Bahasa jawa adalah salah satu bahasa daerah yang merupakan bagian kebudayaan nasional Indonesia, yang hidup dan tetap dipergunakan dalam masyarakat Jawa. Masyarakat Yogyakarta sangat tahu apa itu bahasa jawa walikan (kebalikan), kebanyakan kaum tua sangat mahir dalam mempergunakan bahasa ini. Masyarakat muda yang jauh dari tempat lahirnya (Gunung Ketur) bahasa jawa walikan ini mungkin tahu caranya tetapi tidak dapat secara cepat dan tepat dalam mempergunakan bahasa ini. Untuk itu akan dibuat program yang dapat menstransformasikan tulisan latin jawa ke bahasa walikan jawa.

Tulisan jawa latin dapat dikenali menjadi konsonan-konsonan. Konsonan-konsonan dalam suatu kata ataupun kalimat dapat dikenali dengan suatu mesin pengenal bahasa yaitu: Finite State Automata (FSA). Finite State Automata adalah bagian dari cabang ilmu dari Teori Bahasa dan Automata.

II. Dasar Teori

Criptografi adalah cabang ilmu yang mempejari sandi, vigenere chiper adalah salah satu metode yang dipakai. Vigenere chiper dengan t periode, s karakter alpabet, dan dengan t karakter kunci: $k_1k_2k_3...k_t$ akan memetakan plaintex m $=m_1m_2m_3...$ ke ciphertext $c = c_1c_2c_3$ . . . adalah didefinisikan atas karakter-karakter individualnya oleh $c_i = (m_i + k_i)$ mod s dimana subcript i dalam k dimoduluskan terhadap t (Menezes, 1997).

Secara formal Finite State Automata (FSA) didefinisikan sebagai sebuah 5 tupel (Q, Σ, δ, q0, F), dimana Q : himpunan berhingga status, Σ: himpunan berhingga simbol input (Alpabet), q0 dalam Q adalah status awal, $F \subseteq Q$ adalah himpunan status akhir (finish) dan δ: fungsi transisi yang memetakan Q x Σ ke Q(Hopcroft, 1979).

Sebuah FSA dapat digambarkan sebagai graff berarah yang titik-titikya merupakan status-statusya. Jika sebuah transisi dari status q ke status p dalam input a, maka sebuah garis berlabel a akan menghubungkan status q ke status p dalam graff tersebut.

III. Rancangan Program

Konsonan dalam bahasa jawa terdiri dari 20 yaitu: h, n, c, r, k, d, t, s, w, l, p, dh, j, y, ny, m, g, b, th, ng. Walikan jawa mempergunakan vigenere chiper dengan t = 1, dan k = 10 dan untuk vokal ('a', 'e', 'i', 'o', 'u', 'é') tidak akan sandikan. sehingga jika ditabelkan akan menjadi:

Tabel 1 Pemetaan konsonan jawa

no	KP	Rumus (no+10) mod 20	KC
0	h	10	p
1	n	11	dh
2	c	12	j
3	r	13	y
4	k	14	ny
5	d	15	m
6	t	16	g
7	s	17	b
8	w	18	th
9	l	19	ng
10	p	0	h
11	dh	1	n
12	j	2	c
13	y	3	r
14	ny	4	k
15	m	5	d
16	g	6	t
17	b	7	s
18	th	8	w
19	ng	9	l

Contoh suatu plaintext: **mari mas**

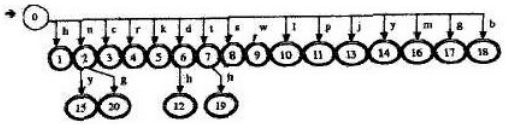
Akan disandikan

m → d
a → a
r → y
i → i

m → d
a → a
s → b

Sehingga chipertextnya menjadi: **dayi dab**

Finite state automata dapat berguna untuk mengenali konsonan-konsonan jawa, jika digambarkan adalah sebagai berikut:



Gambar1. FSA pengenalan konsonan jawa

Parsing digunakan untuk memecah kalimat atau kata menjadi konsonan-konsonan yang dikenali karakter-karakter yang tidak dikenali. FSA digunakan untuk memparsing kalimat, sebagai contoh:

Plaintext: **mari mas,**

Parsernya:

0 =m= >16, 0 =a=> 0, 0 =r=> 4, 0=i= 0, 0 = =>0, 0=m=>16, 0=a=>0, 0=s=>8

Chipertext: **dayi dab**

3.1. Implementasi program

Finite State Automata di atas dapat diimplementasikan sebagai berikut:

```
function FSA(status:integer;k:char):
integer;
var q: integer;
begin
 case status of
 0: if k = 'h' then q := 1  else
    if k = 'n' then q := 2  else
    if k = 'c' then q := 3  else
    if k = 'r' then q := 4  else
    if k = 'k' then q := 5  else
    if k = 'd' then q := 6  else
    if k = 't' then q := 7  else
    if k = 's' then q := 8  else
    if k = 'w' then q := 9  else
    if k = 'l' then q := 10  else
    if k = 'p' then q := 11  else
    if k = 'j' then q := 13  else
    if k = 'y' then q := 14  else
    if k = 'm' then q := 16  else
    if k = 'g' then q := 17  else
    if k = 'b' then q := 18  else
     q := err1;
 2: if k = 'y' then q := 15 else
    if k = 'g' then q := 20 else
    q := err1;
 6: if k = 'h' then q := 12 else
    q := err1;
 7: if k = 'h' then q := 19 else
    q := err1;
 else q := err1;
 end;
 result := q;
end;
```

Prosedur 1 FSA

Konstanta err1 (error 1) pada prosedur FSA dapat diisi dengan berapa saja selain 0 sd 19.

Konsonan dapat terbentuk oleh dua karakter sebagai misalnya dalam kata **nganggur**, konsonan **ng** terbentuk oleh dua karakater (n dan g), jika diparser akan menjadi:

0 =n= >2=g=>20, 0=a=>0, 0=n= 2=g =>20, 0=g=>16, 0=u=>0, 0=r=>4

Prosedur lookhead berguna mencari status akhir

```
function lookhead(p,i:integer):integer;
var look : integer ;
begin
 if i > panjang then look := err1
 else look := FSA(p,kal[i]);
lookhead := look;
end;
```

Prosedur 2. lookhead

variabel panjang adalah panjang kalimat yang dimasukan.

Kata-kata yang diawali vokal ('a','e','i','o','u','é') dalam penulisan jawa harus ditambah konsonan **h,** sebagai misal: ana maka

dalam tulisan jawa ditulis dengan **hana** sehingga diperlukan prosedur untuk proses ini. Berikut adalah prosedurnya:

```
procedure Normal(w:string ; var s :
string);
var i,p : integer; blank : boolean;
begin
 blank := true; p := length(w);
 s := '';
 for i := 1 to p do
 begin
  if (w[i]in['a','e','i','o','u','é'])
    and blank
      then s := s + 'h' + w[i]
      else s := s + w[i] ;
 if w[i] = ' ' then blank := true
    else blank := false;
end;
end;
```

Prosedur 3. Normal

Status-status yang diperoleh dari FSA dapat langsung dikonversikan ke konsonan-konsonan hasil pemetaan, prosedurenya adalah sebagai berikut:

```
function stattochar(stat:integer):string;
const
kons:array[0..19] of string =
 ('p','dh','j','y','ny','m','g','b',
  'th','ng','h','n','c','r','k','d',
  't','s','w','l');
begin
 result:= kons[stat] ;
end;
```

Prosedur 4. stattochar

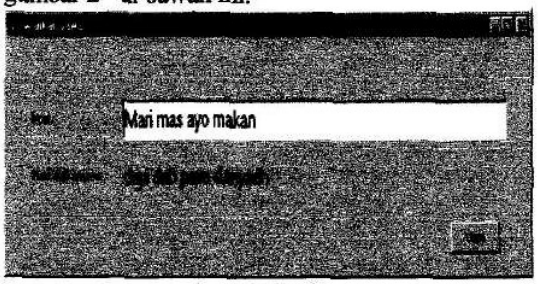
Procedure yang terakhir adalah prosedur parser yang berguna memparser kalimat input dan outputnya berupa chipertext.

```
procedure Parser;
var
  kalW,kal,kalN : string;
  q,i, lihat,panjang : integer;
begin
 kal := edit1.Text; Kecilsemua(kal,kalN);
 Normal(kalN,kal); panjang := length(kal);
 q := 0; i := 1; kalW := '';
 while ( i <= panjang ) do
 begin
  q := FSA(q,kal[i]);
  if q = err1 then
  begin
   kalW := KalW  + kal[i]; q := 0;
  end  else
  begin
   lihat := lookhead(q,i+1);
    if lihat= err1 then
     begin
      kalW := KalW + stattochar(q-1);
      q := 0;
     end;
  end; i := i + 1;
 end;
 label2.Caption := kalw;
end;
```

Prosedur 5. Parser

3.2. Hasil program

Rancangan formnya terdiri dari 3 label, 1 edit text, dan satu button seperti di bawah ini. Prosedur parser akan dipanggil jika edit text diisikan. Hasil program terlihat seperti pada gambar 2 di bawah ini.



gambar 2. hasil program

IV. Penutup

Telah dibuat program walikan jawa dengan vegenere chiper dan FSA mempergunakan Delphi. Program berguna mempercepat dalam bahasa walikan.

V. Referensi

1. Hopcropft JE, and Ullman J.D, 1979, *Introduction to Automata Theory, language and Computation* , Addison Wesley, Massachusets
2. Menezes Alfred J, van Oorschot Paul C, and Vanstone Scott A, 1997, *Handbook of Applied cryptographi*, CRC Press, New York