PARSER TULISAN *HIJAIYYAH* TANPA *HAROKAT* DENGAN FINITE STATE AUTOMATA

Suprihatin
Program Studi Sistem Informasi FMIPA
Universitas Ahmad Dahlan
pakprih@yahoo.co.id

ABSTRAK

Huruf Hijaiyyah dalam penulisannya dibedakan menjadi huruf depan, tengah, belakang ataupun yang berdiri sendiri. Pengetikan kalimat Hijaiyyah dengan komputer tanpa program bantu akan sangat lambat, karena harus memilih huruf depan, tengah, belakang ataupun huruf yang berdiri sendiri. Tujuan penelitian ini adalah membuat program yang dapat mempercepat penulisan kalimat Hijaiyyah, dan dibatasi untuk kalimat yang tanpa harokat. Program dibuat dengan teknik parser mempergunakan Finite State Automata, bahasa yang digunakan yaitu Delphi. Hasilnya berupa program yang dapat menyusun huruf-huruf dengan otomatis sesuai aturan yang berlaku tanpa memikirkan apakah huruf dapat disambung ataupun tidak dapat disambung, sehingga akan mempercepat pengetikan.

Kata kunci: Parser, Finite State Automata, Hijaiyyah, Delphi.

PENDAHULUAN

Komputer sebagai alat bantu menjadi bertambah penting, seiring dengan perkembangan kehidupan manusia. Komputer dapat dipergunakan untuk membantu dalam penulisan beberapa bentuk huruf. Penulisan huruf arab (*hijaiyyah*) akan mengalami kesulitan jika hanya mempergunakan pengolahan kata biasa, karena hurufnya akan berubah jika tidak berdiri sendiri, dan penyusunannya juga terbalik dari kiri ke kanan. Sebagai misal:

- huruf م ا (mim, alif) akan menjadi ما
- huruf ه ل ل ا (alif, lam, lam, ha) akan menjadi ﷲ

Tulisan dalam huruf latin tidak mengenal huruf depan, tengah, belakang ataupun yang berdiri sendiri. Penulisan huruf latin tinggal disambung saja, atau dipisahkan dengan blank (spasi). Penulisan huruf *hijaiyyah* harus memperhatikan apakah huruf itu di depan menyambung, disambung (di tengah, di belakang) ataupun tidak menyambung (disambung, berdiri sendiri).

Aturan ini dapat digambarkan dengan Finite State Automata (FSA), sehingga penelitian ini akan membahas peran FSA dalam pengaturan penulisan huruf *hijaiyyah*. Sebagai tindak lanjutnya akan dikodekan dengan program Delphi, sehingga dapat langsung sebagai produk program komputer.

PERMASALAHAN

Masalah yang dibahas dalam tulisan ini adalah:

1. Bagaimana mengambarkan struktur penyusunan huruf-huruf *hijaiyyah* ke dalam FSA.
2. Bagaimana implementasi FSA ke dalam bahasa pemrograman Delphi sehingga penulisan tulisan *hijaiyyah* dapat lebih mudah.

KAJIAN PUSTAKA

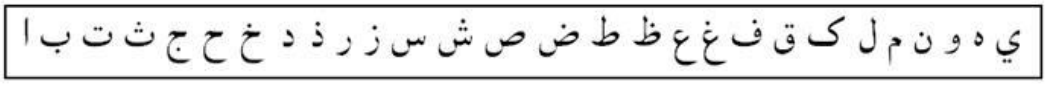
Finite State Automata (FSA) dapat dipergunakan untuk transliterasi aksara jawa ke latin (Suprihatin, 2004). Penelitian ini membahas bagaimana peran FSA membantu dalam transilterasi aksara jawa, yaitu: input berupa aksara-aksara jawa sedangkan output berupa suku kata-suku kata yang dapat dibaca oleh masyarakat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

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 kalimat tersebut.

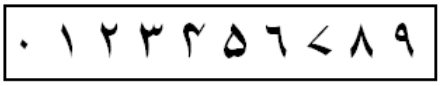
Secara formal Finite State Automata (FSA) didefinisikan sebagai sebuah 5 tupel (Q, Σ, δ, q0, F), dimana Q: himpunan berhingga status, Σ: himpunan berhingga simbol input (Alpabet), q0 dalam Q adalah status awal, F $\subseteq$ Q adalah himpunan status akhir (finish) dan δ: fungsi transisi yang memetakan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

Huruf *hijaiyyah* dan angka *hijaiyyah* yang akan dibahas dalam penelitian ini dapat dilihat pada Gambar 1 dan Gambar 2.

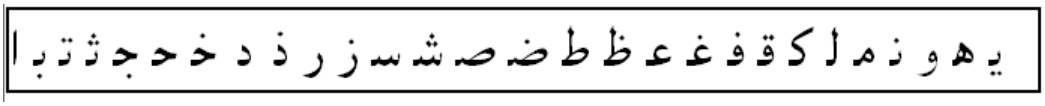


Gambar 1 Huruf Hijaiyyah berdiri sendiri

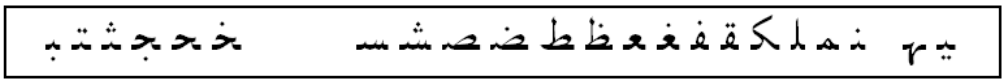


Gambar 2 Angka Hijaiyyah

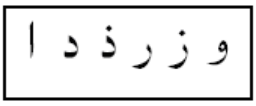
Berikut adalah tabel huruf *hijaiyyah* berdasarkan letaknya:



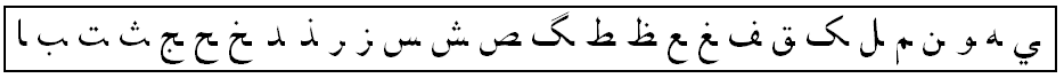
Gambar 3 Huruf Hijaiiyah jika letaknya di depan



Gambar 4 Huruf Hijaiiyah jika letaknya di tengah



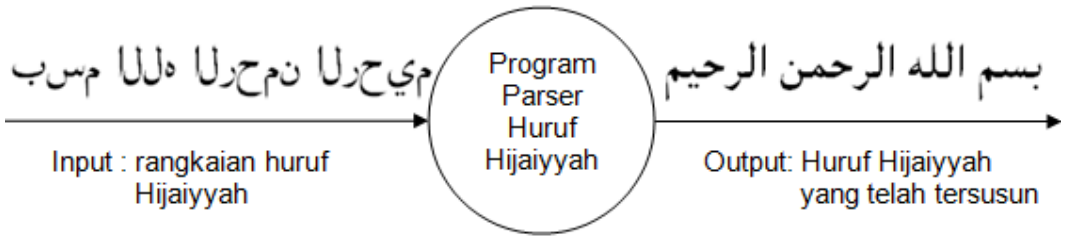
Gambar 5 Huruf Hijaiyyah yang tidak dapat diletakkan di tengah



Gambar 6 Huruf Hijaiyyah jika letaknya dibelakang

METODE PENELITIAN

Penelitian ini diawali dengan membuat FSA kata dari susunan huruf *hijaiyyah*. Langkah selanjutnya aturan-aturan parser FSA tersebut. Diagram alir programnya dapat digambarkan seperti Gambar 7.



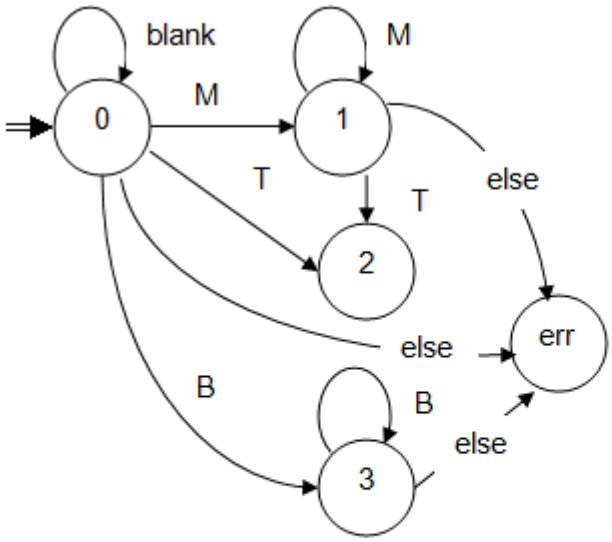
Gambar 7 Diagram alir program

Input berupa huruf *hijaiyyah* yang berdiri sendiri, angka *hijaiyyah* dan *blank*, output berupa susunan huruf sesuai dengan aturan penulisan.

RANCANGAN PROGRAM

Input digolongkan menjadi 3, yaitu huruf-huruf yang dapat menyambung, huruf-huruf yang tidak dapat menyambung, serta bilangan-bilangan, FSA dapat digambarkan seperti Gambar 8.

Dimulai dari status 0, jika ditemui huruf yang dapat menyambung (M) maka ke status 1, jika bertemu huruf yang tidak dapat menyambung (T) maka ke status 2, jika bertemu bilangan ke status 3, dan seterusnya. Algoritma untuk FSA di atas dapat dilihat pada Algoritma 1.



M= {ب,ت,ث,ج,ح,خ,س ,ش,ص,ض,ط,ظ,ع,غ,ف,ق,ك ,ل,م,ن,ه,ي}

T= { ا,د,ذ,ر,ز, و }

B= {٠,١,٢,٣,٤,٥,٦,٧,٨,٩}

Gambar 8 FSA susunan huruf *hijaiyyah*

Algoritma 1 FSA susunan huruf hijaiyyah

```
input: k: char; q: integer
output: integer
variabel lokal st: integer;
begin
  st := 0;
  case q of
    0: if k in T then st := 2 else
      if k in M then st := 1 else
      if k in B then st := 3 else
      if k = ' ' then st := 0 else st := err;
    1: if k in T then st := 2 else
      if k in M then st := 1 else st := err;
    3: if k in B then st := 3 else st := err;
  end;
  result := st;
end;
```

Algoritma parser dipergunakan untuk penyusunan huruf *hijaiyyah* dengan mempergunakan FSA. Algoritma parsernya dapat dilihat pada **Error! Reference source not found.**.

Algoritma 2 Parser

```
begin
  hitung panjang: kal;
  inisialisasi: q := 0; i := 1; depan := true;
  kalW := '';
  selagi (i <= panjang) lakukan begin
    inisialisasi: c := ' ';
    hitung: q := delta(kal[i], q);
    cari kal[i] di tabel masukan pencarian ke dk;
    case q of
      0: jika  kal[i] = ' ' maka kalw := ' ' + kalw;
      1: begin
        jika posisi huruf di depan maka begin
          lihat := status selajutnya;
          jika lihat = err maka begin
            c := huruf berdiri sendiri;
            q := 0; depan := true;
          end jika tidak begin
            c := huruf depan;
            depan := false;
          end;
        end jika tidak begin
          lihat := status selajutnya;
          jika lihat = err maka begin
            c := huruf belakang;
            q := 0; depan := true;
          end jika tidak c := huruf tengah;
        end;
        kalw := c + kalw;
      end;
      2: begin
        if depan then c := huruf sendiri
        jika tidak c := huruf belakang;
        q := 0; depan := true;
        kalw := c + kalw;
      end;
      3: begin
        simpan huruf ke s sampai ditemukan bukan bilangan;
        catat indeknya (i);
        kalw := s +kalw; i := i - 1; depan := true;
        q := 0;
      end;
    end;
    i := i + 1;
  end;
  tampilkan kalw;
end;
```

PENGKODEAN

Pengkodean dimulai dari struktur datanya sebagai berikut:

Kode Program 1 Tipe data dan variabel privat

```
type
  rec = record
    c: char;
    d, t, b, s: byte;
  end;
  larik = array[1..28] of rec;

  private
    err: integer;
    dkar: larik;
    B, M, T: set of char;
```

Tipe data **rec** berupa rekaman terdiri dari c: karakter yang diketikan, d: kode ascii huruf didepan, t: kode ascii huruf tengah, b: kode ascii huruf belakang, dan s: kode ascii huruf berdiri sendiri. Tipe data larik berupa *array* (larik) **rec** sejumlah 28. Variabel privat terdiri dari **err** status error, **dkar** daftar rekaman huruf, B, M, T adalah himpunan huruf sesuai Gambar 1.

Penelitian ini huruf hijaiyyah-nya dapat dirancang sendiri dengan editing font, ataupun download di internet. Setelah font ada dicatat kode ASCII-nya sebagai acuan tabel/daftar huruf seperti pada varabel privat. Kode selanjutnya adalah koding dari Algoritma 1 yaitu sebagai berikut:

Kode Program 2 Delta (FSA)

```
function TForm1.delta(k: char; q: integer): integer;
var st: integer;
begin
  st := 0;
  case q of
    0: if k in T then st := 2 else
      if k in M then st := 1 else
      if k in B then st := 3 else
      if k = ' ' then st := 0 else st := err;
    1: if k in T then st := 2 else
      if k in M then st := 1 else st := err;
    3: if k in B then st := 3 else st := err;
  end;
  result := st;
end;
```

Daftar huruf tidak dicantumkan, karena tergantung dari rancangan font-nya. Selanjutnya kode program untuk memparser huruf adalah sebagai berikut:

Kode Program 3 Parser

```
procedure TForm1.Parser;
var
  s,kalW,kal : string;
  c : char;
  q,i, lihat,panjang : integer;
  dk : rec;
  depan : boolean;

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

procedure caribilangan(var i:integer; var s : string);
begin
  s := '';
  while kal[i] in B do begin s := s + kal[i]; i := i + 1;
  end;
end;

begin
  kal := edit1.Text;
  panjang := length(kal);
  q := 0; i := 1; depan := true;
  kalW := '';
  while (i <= panjang) do begin
    c := ' ';
    q := delta(kal[i],q);
    cari(kal[i],dk);
    case q of
      0: if kal[i] = ' ' then kalw := ' ' + kalw;
      1: begin
        if depan then begin
          lihat := lookhead(q,i+1);
          if lihat = err then begin
            c := chr(dk.s);
            q := 0; depan := true;
          end else begin
            c := chr(dk.d);
            depan := false;
          end;
        end else begin
          lihat := lookhead(q,i+1);
          if lihat = err then begin
            c := chr(dk.b);
            q := 0; depan := true;
          end else c := chr(dk.t);
        end;
        kalw := c + kalw;
      end;
      2: begin
        if depan then c := chr(dk.s) else c := chr(dk.b);
        q := 0; depan := TRUE;
```

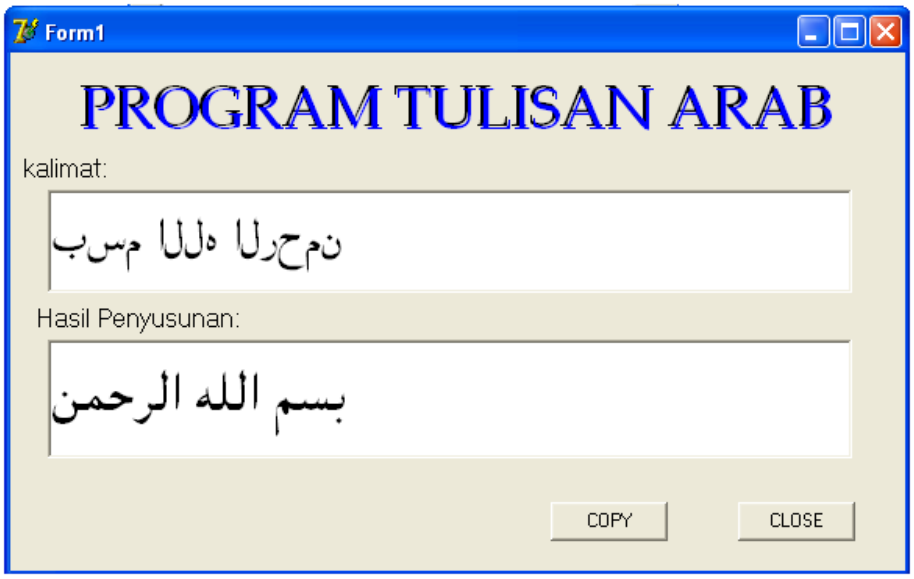
Kode Program 3 Parser (lanjutan)

```
          kalw := c + kalw;
        end;
        3: begin
          caribilangan(i,s);
          kalw := s +kalw; i := i - 1; depan := true;
          q := 0;
        end;
      end;
      i := i + 1;
    end;
    edit2.Text := kalw;
end;
```

Fungsi `lookhead` adalah fungsi untuk melihat status selajutnya, prosedur `caribilangan` adalah untuk mencari huruf-huruf bilangan. Jika status berada di 0 dan input berupa blank maka tulis saja. Jika status berada di 1 maka tulis huruf yang dapat menyambung. Jika status berada di 2 maka tulis huruf yang tidak dapat menyambung. Jika status berada di 3 maka tulis bilangan-bilangan tersebut.

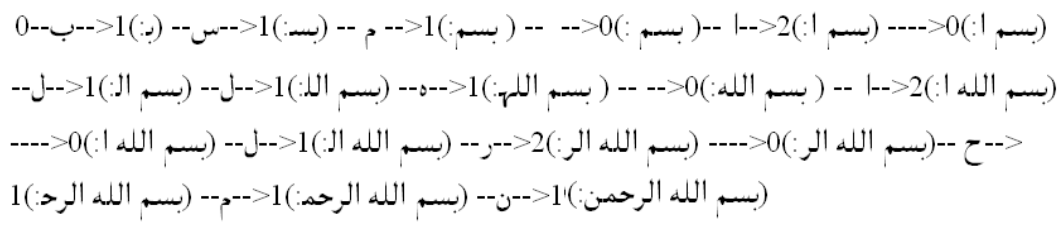
HASIL PROGRAM

Rancangan formnya terdiri dari 1 label, 2 edit text, dan dua button seperti di bawah ini. Edit 1 untuk memasukan huruf-huruf yang diketikan, dan edit 2 akan mengeluarkan susunan huruf sesuai dengan aturan penulisan. Tombol copy untuk mengkopi ke *clipboard* hasil susunan huruf. Tombol *close* untuk keluar program. Hasil program terlihat seperti pada Gambar 9 di bawah ini.



Gambar 9 Hasil *running* program

Proses parser hasil tampilkan di atas adalah sebagai berikut:



Gambar 10 Proses parser

KESIMPULAN DAN SARAN

Kesimpulannya yaitu telah dibuat program yang dapat menyusun huruf-huruf dengan otomatis sesuai aturan yang berlaku tanpa memikirkan apakah huruf dapat disambung ataupun tidak dapat disambung, sehingga akan mempercepat pengetikan. Program ini dapat dilanjutkan dengan menambahkan aturan harokat, program juga dapat dikembangkan menjadi program kaligrafi, dengan membuat *font hijaiyyah* yang indah.

DAFTAR PUSTAKA

Firar Utdirartatma, 2001, *Teknik Kompilasi*, J&J Learnig, Jakarta.

Hopcropft JE, and Ullman J.D, 1979, *Introduction to Automata Theory, language and Computation* , Addison Wesley, Massachusets.

Suprihatin, 2004, *Aplikasi Finite State Automata untuk Alihaksara tulisan Jawa ke Tulisan Latin*, Proseding Seminar Nasional Universitas Muhammadiyah, Semarang.

Suprihatin, 2005, *Finite State Automata untuk Parsing (Pemenggalan) Suku Kata dalam Bahasa Indonesia,* Proseding Seminar Nasional Universitas Negeri, Yogyakarta.