

# HASIL CEK\_Makalah1

*by Makalah 1 Bpk Suprihatin*

---

**Submission date:** 11-May-2022 09:52AM (UTC+0700)

**Submission ID:** 1833482723

**File name:** makal01.pdf (2.72M)

**Word count:** 113

**Character count:** 712

# **1** **APLIKASI FINITE STATE AUTOMATA UNTUK ALIHAKSARA TULISAN JAWA KE TULISAN LATIN**

**Suprihatin**

*Program studi Ilmu Komputer Universitas Ahmad Dahlan*

*Jl Prof. Dr. Soepomo SH., Janturan Yogyakarta 55164*

## **ABSTRAK**

Tulisan ini membahas pembuatan program alihaksara tulisan jawa ke tulisan latin mempergunakan Finite State Automata (FSA). FSA dipergunakan untuk mengenali token-token aksara jawa dan mengalihaksarakannya ke huruf latin. Token didefinisikan sebagai deretan aksara jawa. FSA yang terbentuk terdiri dari 100 status, dengan 98 status akhir. Ke 98 status akhir tersebut digunakan untuk mengalihaksarakan dan menampilkan token. Hasil program alihaksara dapat digunakan untuk mengalihaksarakan tulisan jawa ke tulisan latin juga dapat sebagai alat menulis tulisan jawa.

**Key Word :** *Aksarat, alihaksara, Finite State Automata*

## PENDAHULUAN

Paper ini mencoba program alihaksara dengan Delphi, yang berguna untuk mengalihaksarakan tulisan dalam aksara jawa ke tulisan dalam huruf latin., sehingga dengan alat dapat membantu dalam pembacaan tulisan dalam aksara jawa. Finite State Automata (FSA) sebagai mesin pengenalan bahasa tingkat 3 akan dipegunakan untuk mengenali deretan aksara jawa.

Permasalahannya yang timbul adalah: " Bagaimana mengimplementasikan Finite State Automata (FSA) ke program dengan kompiler Delphi untuk membuat alihaksara jawa ke latin".

## LANDASAN TEORI

Penelitian tentang alihaksara telah dilakukan oleh banyak peneliti antara lain: Kevin K. dan Jonathan G. memperkenalkan mesin alihaksara yang bisa mengalihaksara dari nama-nama Inggris ke aksara Jepang dan suara bacanya. Dalam penelitian ini mempergunakan WFSA (weight finite state acceptor) dan WFSTs (weight finite state transducers).

Sebuah WFSA adalah diagram status/transisi dengan bobot dan symbol dalam transisi-transisi, membuat beberapa barisan output yang serupa. Sebuah WFSTs adalah WFSA dengan sebuah pasangan symbol dalam transisinya, satu input satu output. ( Kevin, 1989). Aksara Arab mempunyai ligature yaitu dibuat dengan cara menggabung paling sedikit dua aksara yang berbeda (Srouji, 1993). Contoh lam-alif adalah ligatur yang terdiri dari huruf lam dan alif. Tulisan dalam aksara jawa juga mempunyai banyak ligatur sebagai misal T (k-na) adalah gabungan dari aksara T dan pasangan na.

### Pedoman Penulisan Aksara Jawa

Pemakaian aksara meliputi: Aksara carakan dan pasangannya, aksara murda dan pasangannya, aksara suara, dan aksara rekaan dan pasangannya. Aksara carakan dan pasangannya yaitu: **QWERTYUIOPASDFGHJKLZ** dan **qwertyuiopasdfghjklz** Aksara murda (**Na, Ka, Ta, Sa, Pa, Ga, Ba**) dan pasangannya: **X, C, V, B, N, M, <, x, c, v, b, n, m**, Aksara suara (**a, é, i, o, u**): **f, \_ , +, i , é**, Aksara rekaan (**kha, dza, fa, za, gha**) dan aksara pasangannya: **! ,@ ,# , \$ , % , ^ , & , \* , ( )**

Pemakaian sandangan terdiri dari: sandangan bunyi vokal, sandangan penanda konsonan penutup suku kata. Sandangan bunyi vokal (wulu, pepet, *suku*, taling, *taling tarung*): / ~ . > > ? Sandangan Penanda Konsonan Penutup Suku Kata ( *wigyan*, layar, cecak, pangkon): 1 3 ` 2 Penanda Gugus Konsonan meliputi (cakra, *keret*, pengkal, panjangan wa, panjangan la): - = [ o p serta angka (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) ditulis: 0 J , € 4 5 \_ P N F

Tanda baca dalam aksara jawa terdiri dari: pada adeg-adeg, pada guru, **pada** pancak, pada lingsa, pada lungsi, pada pangkat, pada ageng, purwopada, madyapada, dan wasanapada. Pada tulisan ini akan dibatasi hanya untuk: pada adeg-adeg, **pada** lingsa, pada lungsi, pada pangkat. Pada adeg-adeg ( { ) dan pada lungsi ( } ) dipakai untuk awal dan akhir kalimat. Pada lingsa (;): dipakai sebagai tanda koma (,) Pada pangkat (:): dipakai sebagai tanda titik dua (:), mengapit angka contoh :0J45PNF: = 0145789, mengapit petikan langsung contoh : QW: = "ana"

### Teori Bahasa Otomata

Bahasa yang diterima oleh finite state automata secara gampang disebut dengan Ekspresi singkat dan dinamakan dengan ekspresi reguler. Yaitu yang diturunkan dengan operasi konkatinasi(penyambungan) dan closure (penutup) dalam himpunan-himpunan string ( Hopcroft, 1979).

Ekspresi reguler atas himpunan simbol  $\Sigma$  didefinisikan secara rekursif (Kelley,1999):

1.  $\phi$  dan  $\epsilon$  adalah ekspresi-ekspresi reguler
2. a adalah ekspresi reguler untuk setiap  $a \in \Sigma$
3. Jika r dan s ekspresi reguler, maka  $r U s$  ,  $rs$  , dan  $r^*$  juga ekspresi-ekspresi reguler
4. Tidak ada deretan symbol-simbol yang lain yang merupakan ekspresi-ekspresi reguler.

Tulisan dalam aksara jawa merupakan string (deretan simbol) aksara jawa dengan aturan penulisan tertentu. Aturan tersebut disebut dengan tatabahasa penulisan, jika dilanggar akan menyebabkan kesalahan dalam pembacaan atau tidak bisa dibaca sama sekali. Mempergunakan tatabahasa penulisan tersebut akan terbentuk kalimat-kalimat dalam aksara jawa. Kumpulan kalimat-kalimat tersebut disebut bahasa.

Bahasa didefinisikan sebagai himpunan string simbol. Secara formal, sebuah tatabahasa terdiri dari 4 komponen sebagai berikut: ( Dulimarta, 1996) dengan: T : Himpunan berhingga tidak kosong dari simbol-simbol terminal. N : Himpunan berhingga simbol-simbol non terminal. S : Simbol awal  $S \in N$ , yang merupakan salah satu anggota dari himpunan simbol non terminal. P: Himpunan berhingga aturan produksi yang setiap elemennya ditulis dalam bentuk  $\alpha \rightarrow \beta$  yaitu  $\alpha$  dan  $\beta$  adalah string yang dibentuk dari himpunan  $(T \cup N)$  dan  $\alpha$  harus berisi paling sedikit satu simbol Nonterminal,

Secara formal Finite State Automata (FSA) didefinisikan sebagai sebuah 5 tupel  $(Q, \Sigma, \delta, q_0, F)$ , dimana Q : himpunan berhingga status,  $\Sigma$ : himpunan berhingga simbol input (Alphabet),  $q_0$  dalam Q adalah status awal,  $F \subseteq Q$  adalah himpunan status akhir (finish) dan  $\delta$ : fungsi transisi yang memetakan  $Q \times \Sigma$  ke Q (Hopcroft, 1979). Sebuah FSA dapat digambarkan sebagai graff berarah, dengan titik-titiknya merupakan status-statusnya.

## RANCANGAN PROGRAM

Langkah pertama mengenali bentuk-bentuk dasar akasar jawa, selanjutnya membuat fontnya, selanjutnya membuat ligature-ligatur font-font gabungan. Pembuatan font digunakan perangkat lunak *Soffty*. Langkah kedua mengenali bentuk struktur token ( suku kata ) jawa, strukturnya dapat dirumuskan dengan ekspresi reguler. Adapun ekspresi regulernya adah sebagai berikut:

$$\begin{aligned} & t + ta_1 + ta_1b_1 + ta_1c_1 + ta_1b_1c_1 + ta_1c_2 + ta_1b_1c_2 + ta_2 + ta_2b_1 + ta_2c_2 + ta_2b_1c_2 + \\ & ta_3 + ta_3b_1 + ta_4 + ta_4b_1 + ta_4c_2 + ta_4b_1c_2 + ta_41 + ta_4b_11 + ta_4c_21 + ta_4b_1c_21 + \\ & ta_5 + ta_5b_1 + ta_51 + ta_5b_11 + tb_1 + tb_1c_1 + tb_1c_2 + tb_11 + tb_1c_11 + tbc_21 + \\ & tc_1 + tc_11 + tc_2 + tc_21 + t1 + t2 + >t + >ta_1 + >ta_1b_1 + >ta_1c_2 + >ta_1b_1c_2 + \\ & >ta_1? + >ta_1b_1? + >ta_1c_2? + >ta_1b_1c_2? + >tb_1 + >tb_1c_2 + >tb_1? + >tb_1c_2? + \\ & >tb_11 + >tb_1c_21 + >tb_1?1 + >tb_1c_2?1 + >tc_2 + >tc_2? + >tc_21 + >t c_2?1 + \\ & >t? + >?1 + >t1 + >tb_2 + >tb_2a_1 + >tb_2a_1c_2 + >tb_2a_1c_2? + >tb_2a_1? \\ & >tb_2c_2 + >tb_2c_2? + >tb_2c_2?1 + >tb_2c_21 + >tb_2? + >tb_2?1 + >tb_21 \\ & e + ea_1 + e1 + f + tb_2 + tb_2a_1 + tb_2a_1c_1 + tb_2a_1c_2 + tb_2a_2 + tb_2a_2c_2 + tb_2a_3 + \\ & tb_2a_4 + tb_2a_4c_2 + tb_2a_4c_21 + tb_2a_41 + tb_2a_5 + tb_2a_51 + tb_2c_1 + tb_2c_11 \\ & tb_2c_2 + tb_2c_21 + tb_21 + t \setminus + t \setminus a_1 + t \setminus 1 \end{aligned}$$

dengan t, b<sub>1</sub>, b<sub>2</sub>, a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>, a<sub>4</sub>, a<sub>5</sub>, c<sub>1</sub>, c<sub>2</sub>, e, f adalah ekspresi reguler dari himpunan-himpunan:

$T = \{ Q, W, E, R, T, Y, U, I, O, P, A, S, D, F, G, H, J, K, L, Z, X, C, V, B, N, M, <, !, @, \#, \$, \% \}$

$B1 = \{ w, e, r, t, y, u, o, p, s, d, f, g, h, j, k, l, , x, c, v, b, n, m, , , ^, \&, (, ), [ \}$

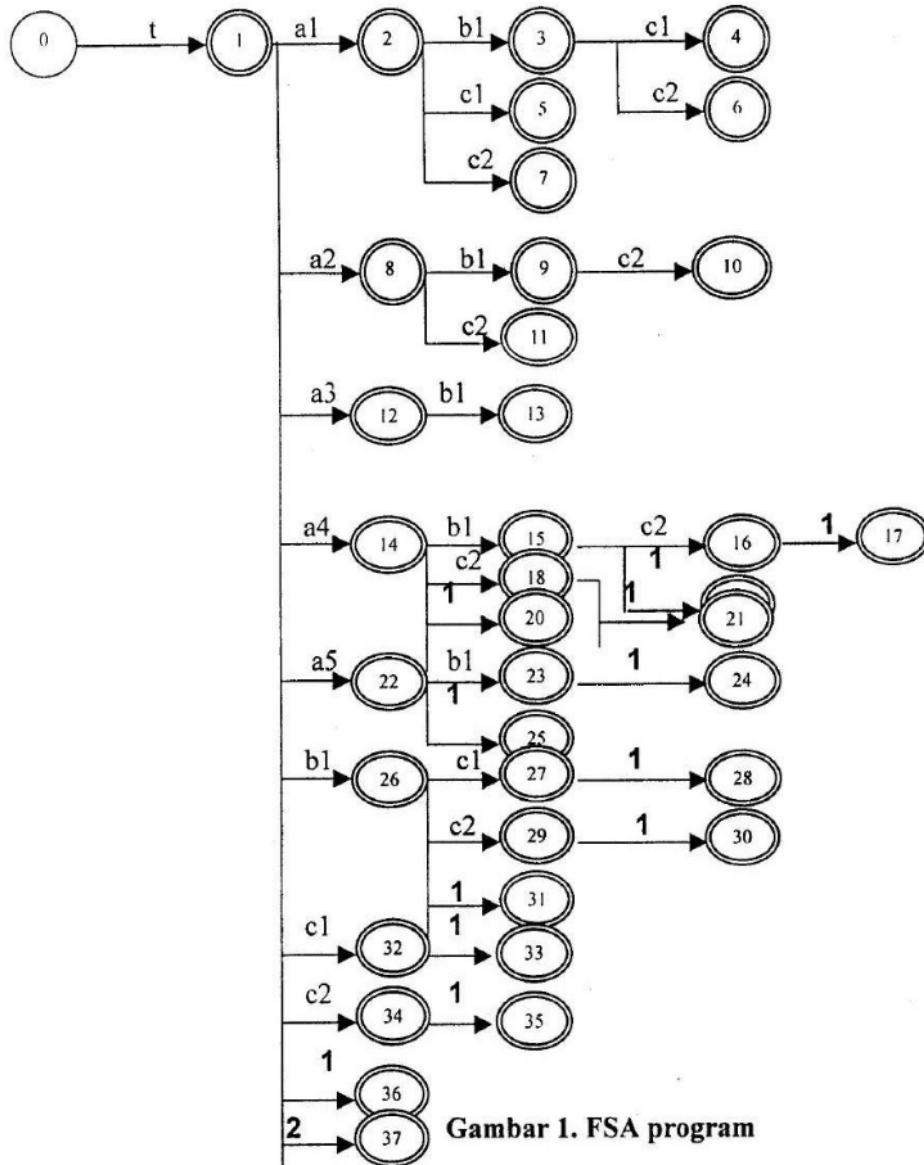
$B2 = \{ q, i, a, *, \}$

$A_1 = \{ , 3 \}$   $A_2 = \{ 6, 8 \}$   $A_3 = \{ 7, 9 \}$   $A_4 = \{ / \}$   $A_5 = \{ \sim \}$

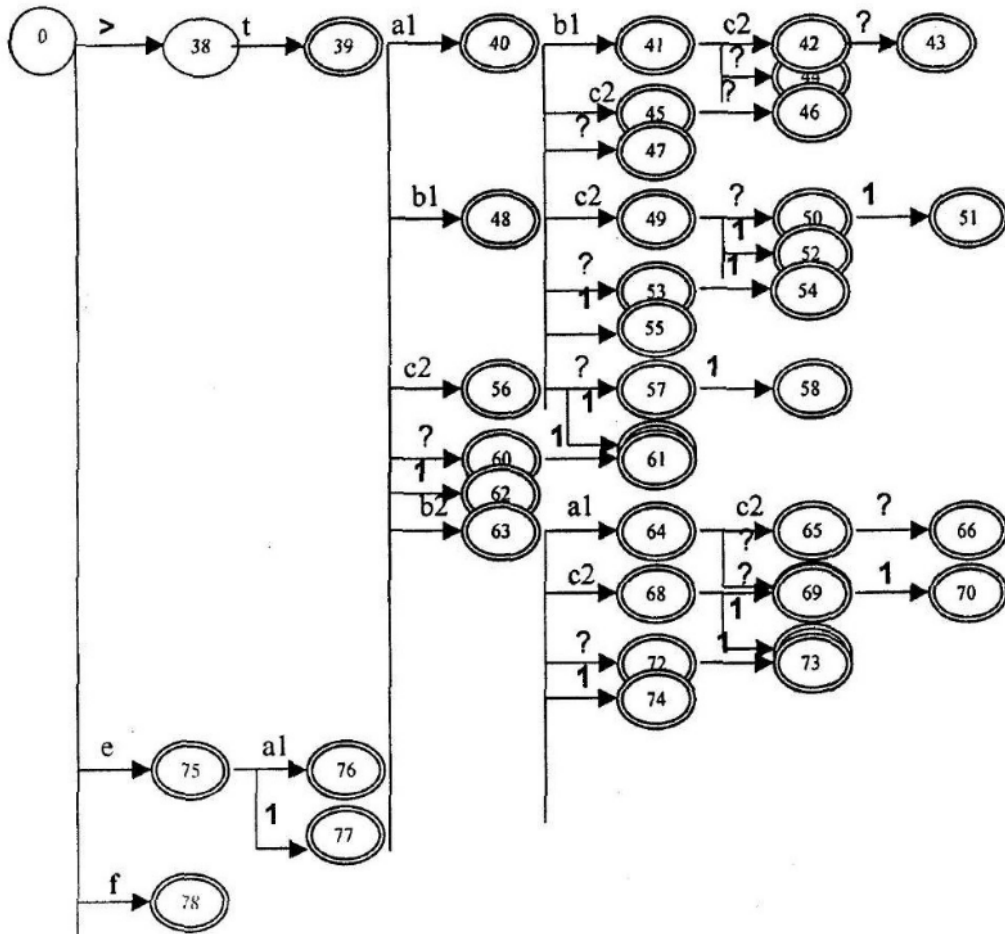
$C_1 = \{ =, ], , . \}$   $C_2 = \{ - \}$

$E = \{ [, \_, +, f, ", ', , \}$  dan  $F = \{ 0, 4, 5, \epsilon, : , ; , \{ , \} \}$

FSA yang terbentuk adalah

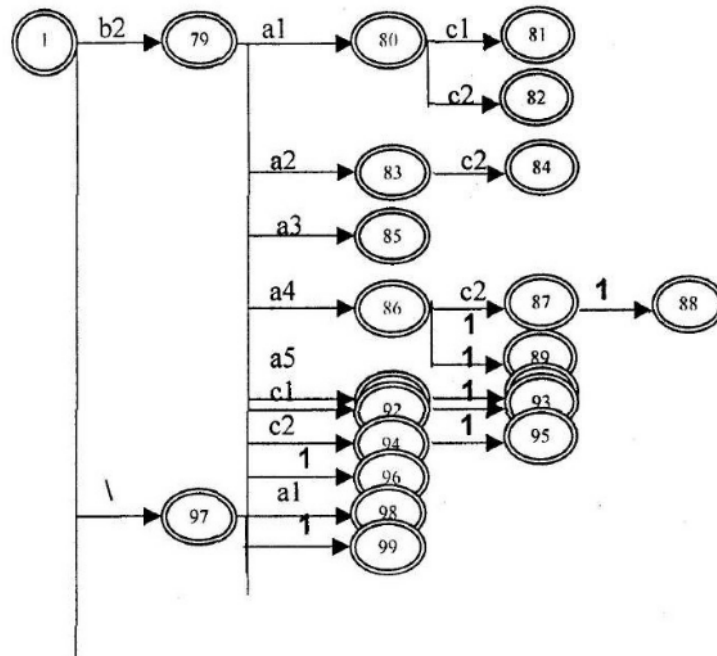


Gambar 1. FSA program



Gambar 1. FSA program (lanjutan)





Gambar 1 FSA program (lanjutan)

### IMPLENTASI PROGRAM

Hasil rancangan program diimplimentasikan dan contoh hasil running programnya adalah sebagai berikut:



Gambar 2 menunjukkan tampilan program utama dengan masukan suatu string aksara jawa, kelurannya berupa tampilan tulisan jawa dan hasil alihaksaranya. Gambar 3 tampilan memilih aksara jawa dengan cara mengklik aksaranya dapat juga dengan bantuan papan ketik.

Agar program berjalan sesuai dengan kaidah penulisan aksara jawa, maka diperlukan testing program. Testing dilakukan untuk masing-masing status, jika tidak ada kesalahan berarti program telah benar.

## KESIMPULAN DAN SARAN

### Kesimpulan

1. Banyak menemui kesulitan dalam mengalihaksarakan dikarenakan aturan penulisan aksara jawa banyak yang tidak konsisten salah satu yang paling berat adalah masalah berarti ganda (ambiguous).
2. Struktur token aksara jawa ada 98 bentuk yang diwakili oleh status akhir dari FSA program
3. Program telah berjalan sesuai dengan kaidah penulisan aksara jawa, dengan tidak ada kesalahan pada 98 status FSA-nya dan masalah-masalah ketidak konsistenan telah teratasi.

### Saran

1. Program dapat dikembangkan untuk alihaksara latin ke jawa dengan cara membalik algoritma, yaitu memparser kalimat menjadi suku kata-suku kata, selanjutnya mengalihaksarakan suku kata-suku kata tersebut.
2. Program dapat dikembangkan untuk alihaksara yang lain dengan cara memodifikasi FSA dan algoritmanya.
3. Program dikembangkan untuk penulisan jawa yang digabung dengan program pengolahan kata seperti **MS Word**.

## DAFTAR PUSTAKA

- Darusuprpta, 1996, *Pedoman, Penulisan Aksara Jawa*, Yayasan Pustaka Nusantara, Yogyakarta.
- Dulimarta, H. S., 1996 *Teori Automata dan bahasa Formal*, Teknik Informatika Universitas Atma Jaya, Yogyakarta.
- Hopcroft J.E. and Ullman J.D., 1979, *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, Massachusetts.
- Kelley D, 1995, *Otomata dan Bahasa-Bahasa Formal*. Prenhallindo, Jakarta
- Kevin K. and Jonathan G., 1998 *Machine Tansliteration*
- <http://acl.ldc.upenn.edu/J/J98/J98-4003.pdf>
- Slamet S. dan Suhartanto H, 1992, *Teknik Kompilasi* , PT Elex Media Komputindo, Jakarta.
- Srouji. J. and Berry. D, 1993 *Arabic Formating with ditroffffortid* , [http://se.uwaterloo.ca/~dberry/FTP\\_SITE/reprints.journals.conferences/arabic.journal.paper.pdf](http://se.uwaterloo.ca/~dberry/FTP_SITE/reprints.journals.conferences/arabic.journal.paper.pdf).

# HASIL CEK\_Makalah1

---

## ORIGINALITY REPORT

---

11%

SIMILARITY INDEX

11%

INTERNET SOURCES

0%

PUBLICATIONS

0%

STUDENT PAPERS

---

## PRIMARY SOURCES

---

1

[eprints.uad.ac.id](http://eprints.uad.ac.id)

Internet Source

11%

---

Exclude quotes On

Exclude matches Off

Exclude bibliography On