

Program Transliterasi Antara Aksara Latin dan Aksara Jawa dengan Metode FSA

Vihl Atina
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Surakarta
v_atina@yahoo.com

YS. Palgunadi
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Surakarta
palgunadi@uns.ac.id

Wisnu Widiarto
Jurusan Informatika
Universitas Sebelas Maret
Jl. Ir. Sutami 36A Surakarta
bethoro_wisnu@yahoo.co.id

ABSTRAK

Aksara Jawa semakin ditinggalkan seiring dengan diperkenalkannya penulisan Latin atau Roman yang telah menjadi standar dunia internasional. Beberapa upaya telah dilakukan untuk melestarikan dan mensosialisasikan kembali aksara Jawa salah satunya adalah usaha mengintegrasikan aksara Jawa ke sistem informasi elektronik. Hal tersebut menyebabkan banyak muncul font aksara Jawa yang tidak didukung aplikasi untuk memudahkan dalam mempelajari penggunaan dan penulisan aksara Jawa sehingga perlu dibangun program transliterasi aksara Jawa.

Metode yang digunakan dalam membangun program transliterasi adalah dengan Metode Finite State Automata (FSA) jenis Deterministic Finite Automata (DFA) dinyatakan dengan (Q, Σ, q_0, F) dan diagram transisi. Diagram transisi DFA berpedoman pada aturan penulisan aksara Jawa dan pola pemenggalan suku kata untuk menghasilkan algoritma transliterasi.

Program dapat mentransliterasikan aksara Latin ke aksara Jawa maupun sebaliknya. Pengujian efisiensi dilakukan dengan menghitung waktu eksekusi program dengan data uji berupa teks, kalimat dan kata. Hasil uji efisiensi di peroleh grafik berupa model linear yang menunjukkan bahwa jumlah karakter dan tingkat kerumitan kata berbanding lurus dengan waktu eksekusi program.

Kata Kunci: aksara Jawa, aksara Latin, DFA, FSA, transliterasi

1. PENDAHULUAN

Aksara Jawa semakin ditinggalkan seiring dengan diperkenalkannya penulisan Latin atau Roman yang telah menjadi standar dunia internasional. Salah satu upaya yang dilakukan untuk melestarikan dan mensosialisasikan kembali aksara Jawa agar tidak hilang adalah usaha mengintegrasikan aksara Jawa ke sistem informasi elektronik agar setiap anggota aksara Jawa memiliki kode yang khas sehingga dapat dipakai pada komputer yang setara dengan huruf lain di dunia. Hal tersebut menyebabkan banyak muncul font aksara Jawa. Salah satunya adalah Font Hanacaraka yang merupakan hasil editing dari [1].

Banyaknya font aksara Jawa yang ditawarkan tidak didukung aplikasi yang membantu untuk memudahkan dalam mempelajari penggunaan dan penulisan aksara Jawa sehingga perlu dibangun program transliterasi aksara Jawa. Transliterasi merupakan suatu metode sistematis untuk mengkonversi (mengubah) karakter dari suatu jenis huruf atau bunyi ucapan (phonetic sound) ke dalam jenis huruf yang lain [2].

Salah satu penelitian mengenai program transliterasi aksara Jawa dilakukan oleh [3]. Penelitian ini mengkaji tentang pembuatan

aplikasi transliterasi kata/kalimat bahasa Jawa dalam huruf latin ke aksara Jawa. Implementasi pembuatan program ini yaitu menerapkan algoritma pengecekan terhadap rangkaian huruf pada kata/kalimat dalam huruf Latin, pemisahan suku kata dari rangkaian huruf tersebut, dan suku kata yang dihasilkan diubah menjadi aksara Jawa dalam format *font True Type*. Dari hasil penelitian ini disimpulkan bahwa aplikasi transliterasi masih memiliki beberapa kekurangan antara lain hanya mampu menangani transliterasi dari aksara Latin ke aksara Jawa tidak dapat sebaliknya, belum mampu menampilkan aksara Murda dan aksara Rekan, dan belum bisa melakukan pemenggalan kata tertentu atau penulisan kata dengan akhiran sehingga diperlukan aturan yang lebih lengkap dan kamus kata bahasa Jawa sehingga mampu menangani transliterasi terhadap berbagai macam jenis kata dalam bahasa Jawa.

Penelitian [4] mengkaji tentang metode *Weighted Finite State Automata* (WFSA) untuk menerjemahkan tulisan bahasa Inggris ke bahasa Jepang pada mesin transliterasi. Metode WFSA diterapkan dalam 5 model yaitu : model pertama digunakan untuk menghasilkan urutan kata dalam bahasa Inggris; model kedua digunakan untuk mengubah urutan kata bahasa Inggris menjadi urutan suara bahasa Inggris; model ketiga digunakan untuk melakukan pemetaan urutan dalam suara bahasa Jepang; model keempat digunakan menggabungkan suara vokal Jepang ke dalam simbol baru yaitu : aa, ii, uu, ee, dan oo; dan model kelima digunakan untuk memetakan suara Jepang ke simbol katakana.

Penelitian [5] mengkaji tentang metode *Finite State Automata* (FSA) untuk membagi kalimat menjadi fonem atau suku kata yang diterapkan pada perangkat lunak *Text To Speech* (TTS) yaitu : pengubah teks sebagai input menjadi suara sebagai output. Dalam penelitian ini metode FSA untuk mencari pola suku kata dibagi menjadi tiga tingkatan yaitu : tingkatan pertama mengenali pola VK dan KV; tingkatan kedua mengenali pola V, VK, KV, KVK, KKV, KVKV, KKKVK; dan tingkatan ketiga mengenali pola VKK, KVKK dan KVKKK. Pada perangkat lunak yang dibangun ini proses dimulai dengan *open file* berupa teks bahasa Indonesia, melakukan pengenalan huruf vokal dan konsonan, pengklasifikasian suku kata dengan metode FSA, mengubah suku kata menjadi rangkaian suara sebagai output.

Program transliterasi yang akan dibangun menggunakan metode *Finite State Automata* (FSA) jenis *Deterministic Finite Automata* (DFA) dinyatakan dengan (Q, Σ, q_0, F) dan diagram transisi. Diagram transisi DFA berpedoman pada aturan penulisan aksara Jawa dan pola pemenggalan suku kata sehingga dapat menghasilkan algoritma transliterasi. Algoritma transliterasi digunakan untuk membangun program transliterasi aksara Jawa sehingga dapat memudahkan dalam mempelajari penggunaan dan penulisan aksara Jawa.

2. DASAR TEORI

2.1 Finite State Automata (FSA)

FSA merupakan automata berhingga yang memiliki sekumpulan status dan kontrol yang yang bergerak dari status ke status lain. Kontrol terhadap FSA dapat bersifat deterministik (automata tidak dapat berada di lebih dari satu status pada saat bersamaan) atau nondeterministik (automata dapat berada di beberapa status pada saat yang bersamaan) sehingga FSA terdiri dari 2 jenis yaitu : *Deterministic Finite Automata (DFA)* dan *Non Deterministik Finite Automata (NFA)*.

Setiap FSA bisa diasosiasikan dengan sebuah diagram transisi, yaitu suatu graf 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 [6].

2.2 Deterministic Finite Automata (DFA)

Setiap bahasa reguler bisa dikenali oleh DFA. Secara formal suatu DFA dinyatakan dengan (Q, Σ, q_0, F) di mana [7] :

- 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 :

1. Mula-mula DFA akan berada pada status q_0 , kepala pita pada simbol pertama pada pita.
2. Selanjutnya kepala pita akan membaca simbol-simbol dari pita dan bergeser maju.
3. Untuk setiap simbol, DFA akan berpindah status sesuai dengan fungsi δ .
4. 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 reguler), dan bila tidak maka string masukan ditolak (tidak dikenali).

2.3 Pedoman Pemenggalan Suku Kata

Pedoman pemenggalan suku kata yang secara umum digunakan adalah sebagai berikut berikut ini [8] :

1. Apabila kata memiliki dua huruf vokal berturut-turut, maka pemenggalan dilakukan di antara kedua huruf vokal tersebut.
2. Apabila kata memiliki huruf konsonan di antara dua huruf vokal, maka pemenggalan dilakukan sebelum huruf konsonan tersebut.
3. Apabila kata memiliki dua huruf konsonan berurutan, maka pemenggalan dilakukan setelah huruf konsonan pertama.
4. Apabila kata memiliki tiga huruf konsonan berurutan atau lebih, maka pemenggalan dapat dilakukan diantara huruf konsonan pertama dan kedua.
5. Apabila kata mendapatkan imbuhan yang mengalami perubahan bentuk, maka pemenggalan dilakukan sebagai satu kesatuan.

2.4 Aksara Jawa

Aksara Jawa terdiri dari 7 macam yaitu [1] :

1. Aksara Carakan

Aksara Carakan adalah huruf utama dalam aksara Jawa terdiri dari 20 huruf dasar. Aksara ini tetap hidup (berbunyi vokal) meskipun tidak diberi sandhangan.

2. Aksara Pasangan
Aksara Pasangan adalah aksara carakan yang berubah bentuknya apabila aksara di depannya mati (diambil huruf konsonannya saja).
3. Aksara Wilangan/Angka
Wilangan adalah angka aksara Jawa yang penggunaannya seperti angka pada huruf latin.
4. Aksara Swara
Aksara Swara adalah huruf hidup atau vokal utama: A, I, U, E, O dalam kalimat. Biasanya digunakan pada awal kalimat atau untuk nama dengan awalan vokal yang mengharuskan penggunaan huruf besar.
5. Aksara Rekan
Aksara Rekan digunakan untuk penulisan kata-kata serapan yang berasal dari bahasa asing terutama berbahasa Arab.
6. Aksara Murda
Aksara Murda yang digunakan untuk menuliskan awal kalimat dan kata yang menunjukkan nama diri, gelar, kota, lembaga, dan nama-nama lain yang kalau dalam Bahasa Indonesia kita gunakan huruf besar.
7. Aksara Sandhangan
Sandhangan adalah tanda baca atau pelengkap huruf dalam aksara Jawa. Sebagian besar sandhangan digunakan untuk membuat bunyi tertentu aksara carakan atau pasangannya, sedangkan yang lainnya berfungsi sebagai tanda baca.

2.5 Aturan Penulisan Aksara Jawa

Pedoman tata tulis aksara Jawa yang disesuaikan dengan kebutuhan penulisan aksara Jawa modern, yaitu sebagai berikut [9] :

1. Susunan aksara Jawa dihadirkan dalam bentuk suku kata terbuka (a) sehingga disebut aksara Carakan atau telanjang yang berarti belum mendapatkan pasangan, sandhangan, maupun tanda baca.
2. Untuk mendapatkan bunyi sesuai suara vokal maka diperlukan sandhangan. Sandhangan terdiri dari :
 - a. Sandhangan swara : wulu (i), suku (u), pepet (e), taling (è), taling tarung (o), dan taling cerek (é).
 - b. Sandhangan sigeg : layar (r), wignyan (h) dan cecek (ng).
 - c. Sandhangan wyanjana : cakra (ra), cakra keret (re) dan cakra ru (ru).
3. Untuk mendapatkan bunyi kosonan (sigeg) maka diperlukan pasangan atau sandhangan pangkon.
4. Penempatan sandhangan disesuaikan dengan sifat pasangan aksara tersebut berada, apakah pasangan aksara tersebut sejajar dengan aksara yang dipasangi atau tidak.
5. Penulisan aksara ra dan la yang mendapat sandhangan pepet diganti dengan pa cerek dan nga leled.
6. Penulisan kata yang bersifat susun tiga
 - a. Penulisan pasangan susun tiga dimungkinkan pada beberapa pasangan yang letaknya sejajar dengan aksara yang dipasangnya.
 - b. Penulisan pasangan susun tiga tidak dimungkinkan pada beberapa pasangan yang letaknya dibawah aksara yang dipasangnya, oleh karenanya aksara di depannya harus disandhangi pangkon.

3. METODOLOGI

Langkah-langkah yang dilakukan dalam penelitian tugas akhir ini adalah sebagai berikut:

1. Tahap Persiapan
Mendefinisikan masalah dan melakukan studi literatur.

2. Tahap Analisis Kebutuhan
Mendeskripsikan program yang akan dibangun secara umum, menentukan fungsi-fungsi yang harus ada dan memilih data yang akan digunakan.
3. Tahap Penyusunan Algoritma
Menerapkan metode DFA dinyatakan dengan (Q, , q0,F) dan diagram transisi, serta menyajikan algoritma dalam bentuk kata-kata.
4. Tahap Implementasi
Membuat database dengan MySQL dan menulis kode program dalam bahasa pemrograman PHP dan *Javascript*.
5. Tahap Pengujian
Menguji kebenaran algoritma dengan *flowchart* dan tabel keputusan, serta menguji efisiensi algoritma dengan kata uji dan kalimat uji.
6. Tahap Panulisan Laporan
Menulis hasil penelitian ke dalam laporan.

4. PENYUSUNAN ALGORITMA

4.1 Transliterasi Aksara Latin ke Jawa

Aksara Latin terdiri dari huruf, angka, spasi dan tanda baca. Angka, spasi dan tanda baca dapat ditransliterasikan ke aksara Jawa secara langsung, sedangkan untuk huruf harus melihat pola suku katanya terlebih dahulu karena ditransliterasikan ke aksara Jawa berdasarkan pola suku kata. Tanda baca hanya tanda baca titik dan tanda koma yang dapat ditransliterasikan ke aksara Jawa. Setiap kata dibatasi dengan 4 susunan konsonan berurutan karena tidak ada kata yang terdiri dari lebih 4 konsonan berturut di tengah kata. Pola suku kata terdiri dari 12 ya itu : V, VK, VKK, KV, KVK, KVKK, KKV, KKVK, KKVKK, KKKV, KKKVK

dan KKKVKK. Berpedoman pada batasan dan pola suku kata maka DFA dapat dinyatakan sebagai berikut :

- Q : {q0, q1, q2, q3, q4, q5, q6, q7, q8}
 : {aksara Latin (huruf, angka, spasi, tanda baca)}
 S : q0
 F : {q1, q3, q4, q5, q6, q8}

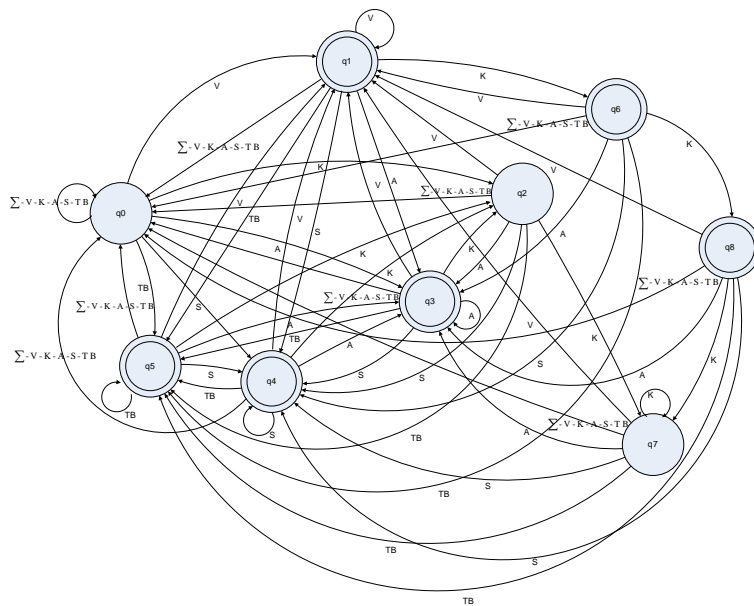
Transisi DFA transliterasi aksara Latin ke aksara Jawa dapat dilihat pada Gambar 1.

4.2 Transliterasi Aksara Jawa ke Latin

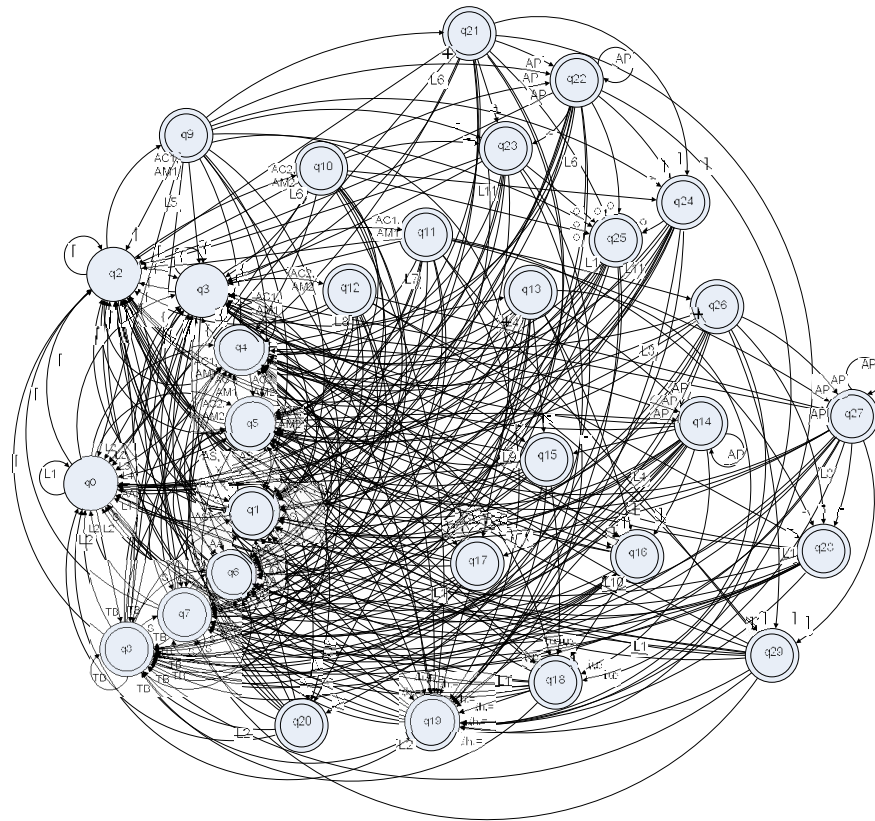
Aksara Jawa terdiri dari aksara Carakan, aksara Pasangan, aksara Murda, aksara Rekan, aksara Swara, aksara Angka dan Sandhangan. Aksara Jawa ditransliterasikan ke aksara Latin dengan berpedoman pada aturan penulisan aksara Jawa sehingga DFA dapat dinyatakan sebagai berikut :

- Q : {q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15, q16, q17, q18, q19, q20, q21, q22, q23, q24, q25, q26, q27, q28, q29}
 : {aksara Jawa (carakan, pasangan, murda, rekan, angka, swara, sandhangan)}
 S : q0
 F : {q1, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15, q16, q17, q18, q19, q20, q21, q22, q23, q24, q25, q26, q27, q28, q29}

Transisi DFA transliterasi aksara Jawa ke aksara Latin dapat dilihat pada Gambar 2.



Gambar 1. Diagram Transisi DFA Transliterasi Aksara Latin ke Jawa



Gambar 2. Diagram Transisi DFA Transliterasi Aksara Jawa ke Latin

Tabel 1. Hasil Transliterasi Teks Aksara Latin ke Aksara Jawa

5. PENGUJIAN

Pengujian efisiensi algoritma dilakukan dengan menghitung waktu eksekusi program. Data yang diujikan berupa teks, kalimat dan kata.

5.1 Transliterasi Aksara Latin ke Jawa

5.1.1 Teks Latin

Pengujian teks Latin diambil dari Serat Rangsang Tuban sepanjang 1 paragraf dengan jumlah kata sebanyak 59. Hasil transliterasi teks aksara Latin ke aksara Jawa dapat dilihat pada Tabel 1.

Input Aksara Latin	Output Aksara Jawa
Serat Rangsang Tuban	ꦱꦺꦫꦠ꧀ ꦫꦁꦱꦁ꧀ ꦠꦸꦧꦤ
nyanyosaken	ꦤꦚꦚꦺꦴꦱꦺꦏꦺꦤ
lalampahanipun pangéran	ꦭꦭꦩꦭꦥꦲꦤꦶꦥꦸꦤ ꦥꦁꦒꦺꦫꦤ
kakalih ing Nagari Tuban	ꦏꦏꦭꦶꦁ ꦤꦁꦫꦶ ꦠꦸꦧꦤ
ajujuluk Sang Pangéran	ꦲꦗꦸꦗꦸꦭꦸꦏ ꦱꦁ ꦥꦁꦒꦺꦫꦤ
Wanih Kusuma sarta Sang	ꦮꦤꦶꦁ ꦏꦸꦱꦸꦩꦤ ꦱꦠꦫ ꦱꦁ
Pangéran Adipati Anom	ꦥꦁꦒꦺꦫꦤ ꦲꦢꦶꦥꦠꦶ ꦲꦤꦺꦩ
Warsa Kusuma, ngantos	ꦮꦫꦱꦤ ꦏꦸꦱꦸꦩꦤ, ꦤꦁꦠꦺꦴꦱꦺꦴꦠ
satamating cariyos,	ꦱꦠꦩꦠꦶꦁ ꦕꦫꦶꦲꦺꦴꦱ
petikan saking Serat	ꦥꦺꦠꦶꦏꦤ ꦱꦁꦶꦁ ꦱꦺꦫꦠ
Wéddaparaya,	ꦮꦺꦢꦢꦥꦫꦺ
anggitanipun Empu	ꦲꦁꦒꦶꦠꦤꦶꦥꦸꦤ ꦺꦩꦥꦸ
Manehguna ing Nagari	ꦩꦤꦺꦁꦒꦸꦤꦤ ꦲꦶꦁ ꦤꦁꦫꦶ
Lamongan kaanggit déning	ꦭꦩꦺꦁꦤ ꦏꦲꦁꦒꦶꦠ ꦢꦺꦤꦶꦁ
Ki Padmasastra, Tiyang	ꦏꦶ ꦥꦢꦩꦱꦱꦠꦫ ꦠꦶꦲꦁ
mardika ingkang marsudi	ꦩꦫꦢꦶꦏꦤ ꦲꦶꦁꦏꦁ ꦩꦫꦱꦸꦢꦶ
kasusastran Jawi Ing	ꦏꦱꦱꦸꦱꦠꦫꦤ ꦗꦮꦶ ꦲꦶꦁ
Surakarta, Nalika taun	ꦱꦸꦫꦏꦫꦠꦤ, ꦤꦭꦶꦏꦤ ꦠꦤ
1900 Kaecap wonten	ꦭꦲꦏꦺꦩꦠꦺꦴꦤ ꦮꦺꦤꦠꦺꦤ
ing pangecapan Budi	ꦲꦶꦁ ꦥꦁꦺꦕꦩꦥꦤ ꦧꦸꦢꦶ
Utama Ing Surakarta 1912.	ꦠꦸꦧꦤ ꦲꦶꦁ ꦱꦸꦫꦏꦫꦠꦤ 1912.

Hasil transliterasi teks aksara Latin adalah aksara Jawa dengan waktu eksekusi program selama 50,97 detik.

5.1.2 Kalimat Latin

Kalimat yang diujikan dikelompokkan menjadi kalimat latin sederhana dan kalimat latin rumit.

Tabel 2. Tabel Uji Waktu Eksekusi Kalimat Latin Sederhana

No	Kalimat	Pjng	Uji1 (s)	Uji2 (s)	Uji3 (s)	Waktu Rata2 (s)
1	Iki budaya jawa.	16	1,875	1,849	1,857	1,86
2	Janaka ratu Madukara.	21	2,566	2,553	2,492	2,54
3	Damel dhaharan kagem tamu.	26	3,214	3,206	3,227	3,22
4	Pelajaran agama lan matematika	31	3,699	3,709	3,723	3,71
5	Dina Rebo ana upacara ing sekolahan.	36	4,157	4,177	4,147	4,16

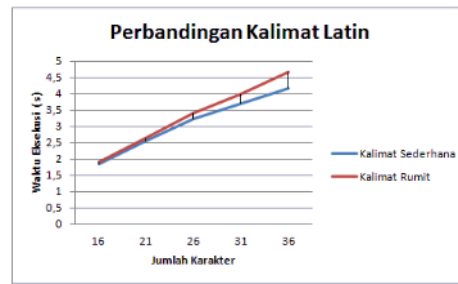
Tabel 3. Tabel Uji Waktu Eksekusi Kalimat Latin Rumit

No	Kalimat	Pn jng	Uji1 (s)	Uji2 (s)	Uji3 (s)	Waktu Rata2 (s)
1	Tio gadhah arto.	16	1,910	1,897	1,878	1,90
2	Pardi mblandong kayu.	21	2,681	2,634	2,631	2,65
3	Dhimas sikile gringgingen	26	3,409	3,406	3,417	3,41
4	Mangsa rendheng banyune bludag.	31	4,011	4,043	3,957	4,00
5	Godhonge wit bayem padha nglinthing.	36	4,696	4,684	4,628	4,67

Dari hasil perhitungan waktu eksekusi program pada Tabel 2 dan Tabel 3 terdapat perbedaan waktu eksekusi program antara kalimat sederhana dan kalimat rumit. Perbandingan waktu eksekusi program pada kalimat sederhana dan kalimat rumit berdasarkan jumlah karakter dapat dilihat pada Tabel 4 dan disajikan dalam bentuk grafik pada Gambar 3.

Tabel 4. Tabel Perbandingan Kalimat Latin Sederhana dan Rumit

No	Panjang Karakter	Rata2 Kalimat Sederhana (s)	Rata2 Kalimat Rumit (s)
1	16	1,86	1,90
2	21	2,54	2,65
3	26	3,22	3,41
4	31	3,71	4,00
5	36	4,16	4,67



Gambar 3. Grafik Perbandingan Kalimat Sederhana dan Rumit

Dari grafik pada Gambar 3. menunjukkan bahwa semakin rumit kalimat yang diinputkan maka semakin lama waktu eksekusi program.

5.1.3 Kata Latin

Kata yang diujikan dikelompokkan menjadi kata latin sederhana dan kata latin rumit.

Tabel 5. Tabel Uji Waktu Eksekusi Kata Latin Sederhana

No	Kata	Panjang Karakter	Uji1 (s)	Uji2 (s)	Uji3 (s)	Waktu Rata2 (s)
1	amanat	6	0,829	0,818	0,819	0,822
2	gegana	6	0,867	0,872	0,864	0,868
3	dhawah	6	0,876	0,852	0,857	0,862
4	adiguna	7	0,931	0,947	0,927	0,935
5	carakan	7	0,978	0,976	0,966	0,973
6	nyamari	7	0,965	0,979	0,979	0,974
7	maratuwa	8	1,183	1,123	1,170	1,159
8	dèwadaru	8	1,083	1,094	1,105	1,094
9	dhaharan	8	1,175	1,174	1,152	1,167
10	selapanan	9	1,229	1,211	1,255	1,232
11	ngayawara	9	1,277	1,269	1,269	1,272
12	nyilakani	9	1,265	1,249	1,253	1,256
13	disalahake	10	1,351	1,393	1,395	1,380
14	nyalawados	10	1,380	1,398	1,406	1,395
15	ngrabèkakè	10	1,393	1,393	1,409	1,398

Tabel 6. Tabel Uji Waktu Eksekusi Kata Latin Rumit

No	Kata	Panjang Karakter	Uji1 (s)	Uji2 (s)	Uji3 (s)	Waktu Rata2 (s)
1	ambruk	6	0,915	0,925	0,925	0,922
2	kendho	6	0,912	0,905	0,917	0,911
3	jlogro	6	0,969	0,982	0,926	0,959
4	ongklok	7	1,029	1,031	1,036	1,032
5	gondhok	7	1,070	1,111	1,133	1,105
6	klungsu	7	1,087	1,091	1,081	1,086
7	mingklik	8	1,213	1,304	1,217	1,245
8	jlènrèh	8	1,317	1,245	1,254	1,272
9	domblong	8	1,243	1,285	1,244	1,257
10	kesempyok	9	1,338	1,325	1,352	1,338
11	nyanglek	9	1,359	1,379	1,349	1,362
12	blandhong	9	1,370	1,389	1,385	1,381
13	gondhongen	10	1,489	1,511	1,491	1,497
14	slinthutan	10	1,586	1,574	1,579	1,580
15	nggondheli	10	1,522	1,508	1,516	1,515

Dari hasil perhitungan waktu eksekusi program Tabel 5 dan Tabel 6 terdapat hubungan antara jumlah karakter dan waktu eksekusi program. Perbandingan waktu eksekusi program pada kata latin sederhana dan kata latin rumit berdasarkan jumlah karakter dapat dilihat pada Tabel 7 dan disajikan dalam bentuk grafik pada Gambar 4.

Tabel 11. Tabel Perbandingan Kalimat Jawa Sederhana dan Rumit

No	Panjang Karakter	Rata2 Kalimat Sederhana (s)	Rata2 Kalimat Rumit (s)
1	11	1,76	2,54
2	14	2,39	3,20
3	18	2,82	3,88
4	22	3,75	4,35
5	24	4,21	4,99



Gambar 5. Grafik Perbandingan Kalimat Sederhana dan Rumit

Dari grafik pada Gambar 5. menunjukkan bahwa semakin rumit kalimat yang diinputkan maka waktu eksekusi program lebih cepat.

5.2.3 Kata Jawa

Kata yang diujikan dikelompokkan menjadi kata jawa sederhana dan kata jawa rumit.

Tabel 12. Tabel Uji Waktu Eksekusi Kata Jawa Sederhana

No	Kata	Pjng	Uji1 (s)	Uji2 (s)	Uji3 (s)	Waktu Rata2 (s)
1	ꦩꦏꦏꦶ	2	0,517	0,489	0,475	0,494
2	ꦩꦏꦏꦶ	2	0,483	0,502	0,517	0,501
3	ꦩꦏꦏꦶꦭꦶ	3	0,762	0,789	0,776	0,776
4	ꦩꦏꦏꦶꦭꦶ	3	0,649	0,612	0,614	0,625
5	ꦩꦏꦏꦶꦭꦶꦭꦶ	4	0,905	0,907	0,908	0,907
6	ꦩꦏꦏꦶꦭꦶꦭꦶ	4	0,914	0,925	0,925	0,921
7	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶ	5	1,083	1,084	1,082	1,083
8	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶꦭꦶ	5	1,107	1,107	1,094	1,103

Tabel 13. Tabel Uji Waktu Eksekusi Kata Jawa Rumit

No	Kata	Pnjng	Uji1 (s)	Uji2 (s)	Uji3 (s)	Waktu Rata2 (s)
1	ꦩꦏꦏꦶꦭꦶꦭꦶ	2	0,749	0,765	0,749	0,754
2	ꦩꦏꦏꦶꦭꦶꦭꦶ	2	0,716	0,695	0,693	0,701
3	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶ	3	1,038	1,011	1,031	1,027
4	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶ	3	0,971	1,006	0,987	0,988
5	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶꦭꦶ	4	1,270	1,290	1,281	1,280
6	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶꦭꦶꦭꦶ	4	1,217	1,260	1,228	1,235
7	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶꦭꦶꦭꦶꦭꦶ	5	1,365	1,339	1,368	1,357
8	ꦩꦏꦏꦶꦭꦶꦭꦶꦭꦶꦭꦶꦭꦶꦭꦶꦭꦶ	5	1,339	1,323	1,298	1,320

Dari hasil perhitungan waktu eksekusi program pada Tabel 12 dan Tabel 13 terdapat hubungan antara jumlah karakter dan waktu eksekusi program. Perbandingan waktu eksekusi program pada kata sederhana dan kata rumit berdasarkan jumlah karakter dapat dilihat pada Tabel 14 dan disajikan dalam bentuk grafik pada Gambar 6.

Tabel 14. Tabel Perbandingan Kata Jawa Sederhana dan Rumit

No	Panjang Karakter	Rata2 Kata Sederhana (s)	Rata2 Kata Rumit (s)
1	2	0,50	0,73
2	3	0,70	1,01
3	4	0,91	1,26
4	5	1,10	1,34



Gambar 6. Grafik Perbandingan Kata Jawa Sederhana dan Rumit

Hasil pengujian dari grafik Gambar 6. berupa model linear yang menunjukkan bahwa jumlah karakter dan tingkat kerumitan kata berbanding lurus dengan waktu eksekusi.

6. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Program Transliterasi antara Aksara Latin dan Aksara Jawa dibangun dengan Metode *Finite State Automata* (FSA) jenis *Deterministic Finite Automata* (DFA) dinyatakan dengan (Q, , q0,F) dan diagram transisi. Diagram transisi DFA yang dibuat berpedoman pada aturan penulisan aksara Jawa dan pola pemenggalan suku kata. Algoritma yang dihasilkan berdasarkan diagram DFA digunakan untuk membangun program transliterasi guna membantu memudahkan dalam mempelajari penggunaan dan penulisan aksara Jawa. Program dapat mentransliterasikan aksara Latin ke aksara Jawa maupun sebaliknya. Pengujian efisiensi dilakukan dengan menghitung waktu eksekusi program

dengan data uji berupa teks, kalimat dan kata. Hasil uji efisiensi diperoleh grafik berupa model linear yang menunjukkan bahwa jumlah karakter dan tingkat kerumitan kata berbanding lurus dengan waktu eksekusi program.

6.2 Saran

Berdasarkan kesimpulan di atas dapat diberikan saran sebagai berikut:

1. Program transliterasi yang dibangun hanya mentransliterasikan karakter yang diinputkan pada form program, sehingga dapat dikembangkan untuk dapat mentransliterasikan file dalam bentuk pdf, txt, atau format dokumen lainnya.
2. Program transliterasi ini jika dibutuhkan dapat dikembangkan dalam aplikasi berbasis mobile untuk lebih membantu memudahkan dalam mempelajari penggunaan dan penulisan aksara Jawa.

7. DAFTAR PUSTAKA

- [1] Sayoga, B., 2004, *Hanacaraka Truetype Font Untuk Perangkat Lunak Komputer Berbasis Sistem Operasi*

- Windows*, http://pallawa.com/sites/default/files/tutorial_jw.pdf, diakses pada tanggal 14 April 2011 (12:47)
- [2] Arms, W., 1999, *Glossary of Digital Libraries-Transliteration*, M.I.T.Press, <http://www.cs.cornell.edu/wya/diglib/ms1999/glossary.html>, diakses tanggal 7 April 2011 (11:51)
- [3] Cahyono, D. D., 2006. *Transliterasi Huruf Latin ke Aksara Jawa dengan Truetype Font*, Yogyakarta : UGM
- [4] Knight, K., Jonathan, G., 1998, *Machine Transliteration*. Computational Linguistics Vol. 24, No. 4.
- [5] Wasista, S., Elly, P., dan Novita, A., 2008, *Sistem Pembaca Teks Bahasa Indonesia Menggunakan Metode Finite State Automata (FSA)*, Spektrum Teknologi Vol. 15 No. 2.
- [6] Hopcroft, J. E., dan Ullman, J. D., 1979, *Introduction to Automata Theory, Languages and Computation*, Addison Wesley Publishing Company, Reading, Massachusetts
- [7] Hopcroft, J. E., Rajeev, M., dan Jeffrey, D. U., 2007, *Teori Bahasa dan Otomata*, Yogyakarta : Andi
- [8] Badudu, J. S., 1978. *Pelik-Pelik Bahasa Indonesia*, Pustaka Prima : Bandung
- [9] Prasaja, S. A., 2008, *Aksara Jawa Dulu, Sekarang Dan Yang Akan Datang*, Yogyakarta : UGM