

Silabifikasi Kata Bahasa Korea Dalam Aksara Latin Berbasis Aturan dan Model Deterministic Finite Automata

Rouly Doharma Sihite¹, Aditya Wikan Mahastama²

¹Program Studi Manajemen Informatika, STMIK Widuri

Jl. Palmerah Barat 353, Jakarta

rouly.doharma@gmail.com

²Program Studi Teknik Informatika, Universitas Kristen Duta Wacana

Jl. Dr. Wahidin Sudirohusodo 5-25, Yogyakarta

mahas@staff.ukdw.ac.id

Abstract—Transliteration is still a challenge in helping people to read or write from one to another writing systems. Korean transliteration has been a topic of research to automate the conversion between Hangul (Korean writing system) and Latin characters. Previous works have been done in transliterating Hangul to Latin, using statistical approach (72.2% accuracy) and Extended Markov Models (54.9% accuracy).

This research focus on transliterating Latin (romanised) Korean words into Hangul, as many learners of Korean began using Latin first. Selected method is modeling the probable vowel and consonant forms and probable vowel and consonant sequences using Finite State Automata to avoid training. These models are then coded into rules which applied and tested to 100 random Korean words.

Initial test results only 40% success rate in transliterating due to the nature that consonants have to be labeled as initial or final of a syllable, and some consonants missed the modeled rules. Additional rules are then added to catch-up and merge these consonants into existing proper syllables, which increased the success rate to 92%. This result is analysed further and it is found that certain consonants sequence caused syllabification problem if exist in a certain position. Other additional rules was inserted and yields 99% final success rate which also is the accuracy of transliterating Korean words written in Latin into Hangul characters in compound syllables.

Intisari—Transliterasi atau alih aksara otomatis masih menjadi topik penelitian yang menarik. Transliterasi aksara Korea telah menjadi topik penelitian dalam otomatisasi konversi antara aksara Hangul menuju aksara Latin. Penelitian-penelitian yang telah dilakukan memberikan hasil akurasi 72,2% dengan pendekatan statistik dan 54,9% menggunakan pendekatan Extended Markov Models untuk transliterasi dari Hangul ke Latin.

Penelitian ini mengenai transliterasi kata bahasa Korea yang dituliskan menggunakan aksara Latin (romanisasi) untuk dialih aksarakan ke dalam aksara Hangul. Topik ini dipilih untuk membantu para pembelajar bahasa, karena sebagian besar pembelajar bahasa Korea memulai dengan aksara Latin terlebih

dahulu. Pendekatan yang dipilih adalah membangun model kemungkinan sekuens vokal dan konsonan sebagai penyusun kata, dan membangun model sekuens suku kata yang memungkinkan. Model ini akan dibangun menggunakan prinsip Finite State Automata (FSA) agar tidak memerlukan pelatihan. 100 kata acak dalam bahasa Korea digunakan sebagai data uji.

Hasil pengujian awal menunjukkan keberhasilan hanya sebesar 40% karena sejumlah kekurangan dalam metode pelabelan konsonan, sehingga terdapat konsonan yang tidak terlabeli sebagai initial atau final. Serangkaian aturan kemudian ditambahkan untuk mengatasi hal ini dan menghasilkan tingkat keberhasilan segmentasi dan transliterasi sebesar 92%. Hasil ini kemudian dianalisis kembali dan ditemukan bahwa rangkaian konsonan tertentu dapat menyebabkan kesalahan silabifikasi pada posisi tertentu dalam kata. Aturan tambahan berikutnya dapat memperbaiki kesalahan ini dan menghasilkan tingkat akurasi 99% dalam mentransliterasikan kata bahasa Korea dalam aksara Latin ke dalam bentuk suku kata aksara Hangul.

Kata kunci—transliterasi, Hangul, aksara Korea, Finite State Automata, berbasis aturan.

I. PENDAHULUAN

Transliterasi adalah salah satu bidang penelitian yang masih terbuka luas hingga saat ini. Transliterasi adalah konversi sebuah teks dari satu jenis aksara ke dalam bentuk aksara lainnya. Transliterasi diperlukan untuk (1) membantu mengeja bunyi sebuah aksara, misalnya dari aksara Kanji menjadi aksara Latin, (2) membantu konversi penulisan dari satu aksara ke aksara lainnya seperti dari aksara Latin ke aksara Jawa, serta (3) membantu pencarian dokumen (*document retrieval*) pada sekumpulan korpus yang menggunakan berbagai macam aksara, dengan masukan pencarian menggunakan satu aksara saja [1].

Bahasa Korea adalah salah satu bahasa yang memiliki aksara sendiri yaitu Hangul atau Hangeul. Aksara ini diciptakan oleh Raja Sejong dari dinasti Choson (1393-1910) dan diformulasikan pertama kali pada tahun 1446 dengan nama asli Hunmin chong-um, yang secara literal bermakna

"bunyi yang tepat untuk instruksi bagi rakyat", dengan tujuan mempermudah rakyatnya berkomunikasi secara tertulis [2].

Hangul kemudian ditetapkan menggantikan ikonografi aksara Cina (Hanja) yang sudah diimpor dan dipergunakan sebelumnya, karena dianggap kurang dapat merepresentasikan ide asli rakyat Korea. Korea sendiri saat ini menjadi salah satu kekuatan ekonomi, industri dan budaya dunia, sehingga minat untuk mempelajari bahasa dan script Korea (Hangul) semakin meningkat di kalangan bangsa-bangsa dunia.

Dari sisi ilmu bahasa, aksara Hangul merupakan sebuah aksara yang menarik untuk diteliti transliterasinya karena secara visual aksara Hangul tampak seperti sebuah abugida. Sebuah aksara disebut sebagai "abugida" jika satu karakter atau huruf melambangkan pengucapan satu suku kata [3]. Namun pada kenyataannya Hangul adalah sebuah pseudoabugida yang terdiri dari susunan alfabet murni yang diletakkan sedemikian rupa sehingga mewakili satu suku kata. Peletakan susunan alfabet ini memiliki aturan tertentu yang harus ditaati berdasarkan alfabet yang digunakan [4]. Masalah pada transliterasi tulisan Latin ke Hangul adalah bahwa dengan tulisan Latin, sebuah kata atau frasa bahasa Korea kadang dituliskan bersambung tanpa jeda suku kata, sedangkan para pembelajar bahasa Korea pada umumnya mengawali pembelajarannya dengan representasi Latin dari bahasa Korea, dan mengalami kesulitan dalam memecah-mecah suku kata ketika harus menuliskannya ke dalam aksara Hangul. Untuk itu diperlukan sebuah model yang dapat membantu merumuskan transliterasi yang tepat dari aksara Latin ke dalam Hangul, sehingga dapat dibangun aplikasi transliterasi otomatis untuk membantu pembelajaran penulisan aksara Hangul.

II. KAJIAN LITERATUR

Penelitian mengenai pengembangan model otomatisasi transliterasi telah dilakukan terhadap sejumlah aksara dari beberapa bahasa-bahasa Asia, di antaranya adalah dari aksara Jepang (bahasa Jepang) terhadap aksara Latin (bahasa Prancis) dalam rangka penggalian informasi dari jaringan WWW [5]. Penelitian oleh Omae, dkk. ini mengurai struktur penulisan sebuah kata dalam aksara Jepang dengan sebuah kata beraksara Latin berbahasa Prancis ke dalam struktur morfologi masing-masing, kemudian membandingkan keduanya untuk memperoleh persepsi apakah kedua kata tersebut merujuk terhadap sebuah kata yang sama. Meskipun bukan merupakan transliterasi langsung, tetapi penelitian ini menggunakan teknik penguraian yang sama menuju transliterasi, dan tujuannya adalah mencari ketepatan transliterasi kata bahasa Prancis ke dalam aksara Jepang.

Penelitian mengenai otomatisasi transliterasi berikutnya adalah transliterasi istilah asing dari aksara Latin ke aksara Ibrani oleh Goldberg dan Elhadad [6] yang melakukan transliterasi kata asing dari aksara Latin ke aksara Ibrani dengan tujuan mendeteksi secara otomatis ada atau tidaknya kata asing serapan dalam sebuah kalimat berbahasa Ibrani. Pendekatan yang digunakan adalah klasifikasi Naïve Bayes dan *Over-Generation Process* yang menggunakan representasi fonemik suatu kata. Sementara untuk aksara

Asia Selatan, terdapat penelitian studi pustaka mengenai metode-metode transliterasi kata aksara Latin (bahasa Inggris) ke dalam aksara Punjabi oleh Brar dan Kaur [7]. Penelitian ini membandingkan pendekatan-pendekatan berbasis *direct mapping*, *rule-based*, dan *Statistical machine translation*, dengan kesimpulan bahwa belum terdapat metode yang memuaskan, dengan hasil terburuk dicapai oleh pendekatan *direct mapping*.

Untuk transliterasi dari istilah Bahasa Inggris dengan aksara Latin ke dalam Bahasa Korea dengan aksara Hangul telah diteliti oleh Lee dan Choi [8], menggunakan pendekatan statistik mengenai istilah Bahasa Korea yang paling sering digunakan sebagai ekuivalensi istilah Bahasa Inggris tertentu. Penelitian ini menggunakan pendekatan STM (*statistical Transliteration Model*) yang bebas-bahasa dengan dua cara: menggunakan pivot dan tidak (langsung). Hasil penelitian ini menyimpulkan bahwa kedua cara diatas harus digabungkan (*hybrid*) untuk memberikan hasil yang lebih baik dari pada terpisah, yaitu 72,2%.

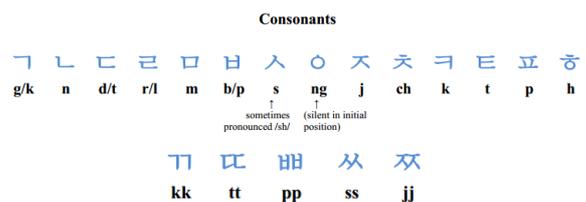
Sementara itu, penelitian Jung, Hong dan Paek [9] menciptakan model berbasis tabel pemetaan fonetik, kemudian dilanjutkan dengan pendekatan statistika yang menggunakan ekstensi Jendela Markov untuk memprediksi transliterasi dari bunyi suku kata Bahasa Inggris yang dituliskan dengan aksara Latin agar mendapatkan ekuivalensi bunyi yang sama saat dituliskan dengan aksara Hangul. Model Jendela Markov yang diusulkan mampu mengungguli penggunaan otomata dengan 54,9% dibanding 36,4%.

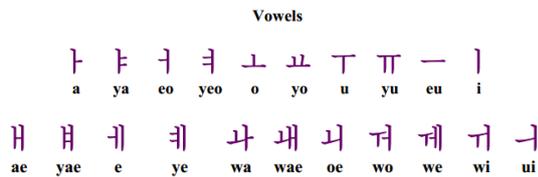
Berdasarkan kajian literatur yang telah dilakukan, maka penelitian ini bertujuan mengembangkan otomatisasi transliterasi penulisan kata bahasa Korea menggunakan aksara Latin ke dalam aksara Korea (Hangul) menggunakan pendekatan otomata melalui pembangunan model FSA yang spesifik dan sesuai untuk proses silabifikasi, dengan harapan bahwa pembuatan model yang spesifik dapat merumuskan aturan yang lebih sesuai dan dapat memberikan hasil lebih baik. Pendekatan otomata dipilih karena tidak memerlukan pelatihan terlebih dulu.

A. Aksara Hangul

Aksara Hangul diciptakan oleh Raja Sejong dari dinasti Choson (1393-1910) dan mulai disusun pada tahun 1443. Fungsi utama diciptakannya Hangul adalah untuk mempermudah literasi rakyat dan mengizinkan komunikasi tertulis untuk menggantikan ikonografi aksara Cina (Hanja) yang sebelumnya diimpor dari Cina untuk mengekspresikan kata-kata Bahasa Korea [2].

Aksara Hangul merupakan alfabet yang terdiri dari 19 karakter konsonan dan 22 karakter vokal yang merepresentasikan bunyi pada Bahasa Korea [4].

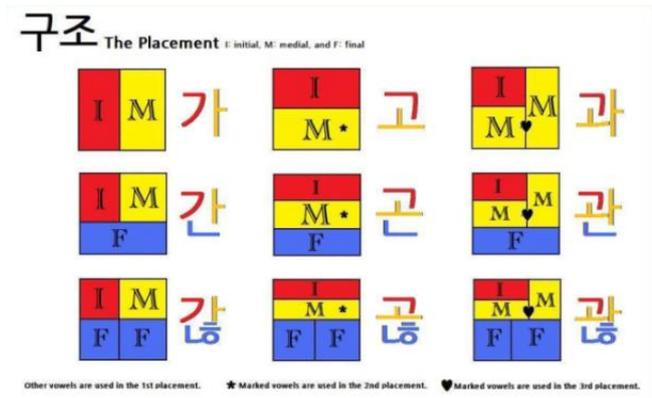




Gambar. 1. Alfabet Hangul

Aksara Hangul dituliskan dalam bentuk satuan suku kata, di mana masing-masing karakter diletakkan dengan aturan posisi tertentu pada blok suku kata berdasarkan golongannya: initial (karakter awal), medial (karakter tengah), dan final (karakter akhir). Bentuk asli dari alfabet Hangul sesungguhnya merepresentasikan pula anatomi mulut saat pengucapannya, namun untuk saat ini falsafah asli pembuatan bentuk alfabet ini sudah tidak begitu populer. Koleksi bentuk dan nama alfabet Hangul ditunjukkan oleh Gambar 1.

Alfabet Hangul dituliskan dengan penempatan khusus untuk membentuk sebuah suku kata *compound* Hangul, dengan aturan penempatan ditunjukkan seperti pada Gambar 2 [10].



Gambar. 2. Aturan Komposisi Suku Kata Alfabet Hangul [10]

B. Romanisasi Aksara Hangul

Romanisasi aksara Hangul adalah cara merepresentasikan pengucapan alphabet Hangul pada Bahasa Korea menggunakan aksara Latin. Istilah romanisasi merujuk pada transliterasi aksara sebuah bahasa menuju aksara Latin. Romanisasi ini distandarisasi, dahulu menggunakan standar McCune-Reischauer yang diformulasikan pada 1939, dengan contoh transliterasi seperti pada Tabel 1[11].

TABEL I
CONTOH ROMANISASI AKSARA HANGUL MENURUT MCCUNE-REISCHAUER

Hangul	Romanisasi
라 면	ramyŏn
한 자	hancha
영 창	yŏngchang
서울대	sŏuldae
킬 자비	kiljabi
말 갈 족	malgaljok

카 버 치	kabŏch'i
실 증	silchŭng

Standar McCune Reischauer ini lebih mudah untuk dipelajari bagi pembelajar asing karena memiliki representasi bunyi asli alfabet tersebut saat digunakan dalam percakapan. Namun standar ini kurang dapat memberikan ekuivalensi alfabet Latin yang konsisten untuk setiap alfabet Hangul. Oleh karena itu pada tahun 2000, Kementerian Budaya, Olahraga dan Pariwisata Korea Selatan mengeluarkan standar baru yang mengatur transliterasi alfabet dengan beberapa alfabet Hangul dapat memiliki lebih dari satu representasi penulisan dalam aksara Latin untuk menyesuaikan pengucapan aslinya, dengan memperhatikan letak alfabet pada suatu suku kata. Contoh standar baru yang disebut juga “Revised Romanization” (RR) ini ditunjukkan oleh Tabel 2 [12].

TABEL II
CONTOH ROMANISASI AKSARA HANGUL MENURUT KEMENTERIAN BUDAYA, OLARHAGA DAN PARIWISATA KOREA SELATAN

Hangul	Romanisasi
라 면	ramyeon
한 자	hanja
영 창	yeongchang
서울대	seoulde
킬 자비	kiljabi
말 갈 족	malgaljok
카 버 치	kabeochi
실 증	siljeung

Romanisasi yang digunakan pada penelitian ini adalah mengikuti Revised Romanization (RR), dengan pengecualian kk menjadi gg, tt menjadi dd, pp menjadi bb sesuai penggunaan populer.

C. Finite State Automata

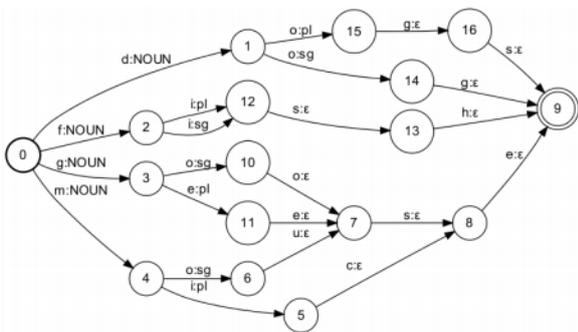
Finite State Automata (FSA) atau dapat disebut juga dengan *Finite State Machine* (FSM) adalah model komputasi matematika yang digunakan dalam perancangan program komputer dan sirkuit logika sekuensial. Ia adalah sebuah mesin abstrak yang hanya dapat berada dalam satu *state* dari sejumlah *state* yang terbatas. Mesin ini dapat berpindah dari satu *state* ke *state* lainnya hanya jika terdapat pemacu atau syarat yang terpenuhi, dan perpindahan ini disebut dengan transisi. FSA pada umumnya terdiri dari serangkaian *state* dan syarat untuk perpindahan dari satu *state* ke *state* lainnya.

FSA dimodelkan dalam bentuk graf berarah (digraf) dengan hanya satu node awal dan node akhir, di mana node awal digambarkan dengan lingkaran bergaris tebal dan node akhir digambarkan dengan lingkaran bergaris tepi ganda. Node merupakan simbol *state*, dan biasanya diberi nomor dengan *state* awal memiliki nomor terkecil, sebagai ilustrasi kemajuan langkah. *Transisi* digambarkan sebagai *edge* yang menghubungkan antara dua buah node *state*, dan syarat transisi dituliskan di dekat *edge* tersebut. Graf FSA dapat memiliki *loop*, dan syarat opsional yang dituliskan sebagai epsilon (ε).

Finite State Transducer (FST) adalah sebuah FSA yang memetakan antara dua himpunan: himpunan input dan himpunan output [13]. Dengan demikian syarat transisi juga mencantumkan output yang dihasilkan jika melalui transisi tersebut. Menurut Gribkoff, sebuah Deterministic Finite State Transducer adalah sebuah 7-tupel $(Q, \Sigma, \Gamma, \delta, \omega, q_0, F)$ [14], di mana:

1. Q adalah sebuah himpunan terbatas yang disebut **states**
2. Σ adalah sebuah himpunan terbatas yang disebut **alphabet**
3. Γ adalah sebuah himpunan terbatas yang disebut **output alphabet**
4. $\delta : Q \times \Sigma \rightarrow Q$ adalah **fungsi transisi**
5. $\omega : Q \times \Sigma \rightarrow \Gamma$ adalah **fungsi output**
6. $q_0 \in Q$ adalah **state awal**
7. $F \subseteq Q$ adalah himpunan **state yang diterima**

Contoh dari sebuah FST deterministik ditunjukkan oleh Gambar 3, yaitu mesin untuk menentukan apakah kata benda hewan dalam Bahasa Inggris ini termasuk jamak atau bukan berdasarkan rangkaian alfabet penyusunnya.



T_{lex} maps words to morphological features
 dog → NOUN sg (singular)
 dogs → NOUN pl (plural)

Gambar. 3. Contoh Finite State Transducer (FST) Deterministik

III. METODE PENELITIAN

Penelitian ini diawali dengan kegiatan persiapan, yaitu pembelian alat dan bahan serta pengumpulan literatur pendukung, baik dalam bentuk artikel di jurnal, prosiding maupun dalam bentuk buku dan kamus. Pada tahapan persiapan ini didapatkan informasi awal mengenai cara penyusunan dan penulisan aksara Hangul dalam bentuk suku kata, yang ternyata memiliki aturan-aturan tertentu yaitu bahwa:

- a. Sebuah suku kata Hangul selalu terdiri dari satu nukleus (pusat) berupa vokal, yang dapat didampingi di sisi kiri dan kanannya oleh masing-masing satu buah konsonan.
- b. Konsonan awal dinamakan “initial”, vokal tengah “medial” dan konsonan akhir “final”.
- c. Terdapat aturan khusus untuk menyusun suku kata Hangul berdasarkan komposisi initial, medial dan

final yang terlibat. Aturan ini dituangkan dalam aturan Hangul Jamo

- d. Terdapat bagian aksara Hangul pada Unicode yang dapat membentuk Jamo melalui kombinasi rumus tertentu yang mengolah nilai initial, medial dan final.
- e. Initial dan final memiliki tabel keanggotaan masing-masing, seperti ditunjukkan oleh Tabel 3. Terdapat juga sejumlah konsonan yang dapat berperan sebagai initial dan final. Konsonan juga dapat berupa konsonan panjang yang diromanisasikan sebagai aksara Latin ganda.
- f. Vokal medial tidak selalu berupa vokal biasa namun juga bisa berupa diftong dan nasal, oleh karena itu representasinya dalam romanisasi tidak selalu dapat diwakili oleh satu aksara Latin, seperti ditunjukkan oleh Tabel 4.

TABEL III
 KONSONAN INITIAL DAN FINAL AKSARA HANGUL

Romanisasi Konsonan	Hangul Jamo		Unicode
Initial saja			
dd	ㄸ	ㄹ	ᄄ
r	ㄷ	ㄹ	ᄅ
bb	ㅃ	ㅅ	ᄈ
jj	ㅈ	ㅊ	ᄍ
Final saja			
gs	ㄱ	ㄴ	ᆪ
nj	ㄴ	ㄷ	ᆬ
nh	ㄴ	ㄹ	ᆭ
lg	ㄷ	ㄹ	ᆰ
lm	ㅃ	ㅅ	ᆱ
lb	ㅃ	ㅅ	ᆲ
ls	ㅃ	ㅅ	ᆳ
lt	ㅃ	ㅅ	ᆴ
lp	ㅃ	ㅅ	ᆵ
lh	ㅃ	ㅅ	ᆶ
bs	ㅃ	ㅅ	ᆹ
ng	ㅇ	ㅇ	ᆼ
Initial dan final			
g	ㄱ	ㄴ	ᄀ
	ㄱ	ㄴ	ᆨ
gg	ㄱ	ㄴ	ᄁ
	ㄱ	ㄴ	ᆩ
n	ㄴ	ㄷ	ᄂ
	ㄴ	ㄷ	ᆫ
d	ㄷ	ㄹ	ᄃ
	ㄷ	ㄹ	ᆮ
l	ㄷ	ㄹ	ᆯ
m	ㅃ	ㅅ	ᄆ
	ㅃ	ㅅ	ᆷ
b	ㅃ	ㅅ	ᄇ
	ㅃ	ㅅ	ᆸ
s	ㅅ	ㅈ	ᄉ
	ㅅ	ㅈ	ᆺ
ss	ㅅ	ㅈ	ᄊ

	ㅃ	ㅅ	ᆻ
j	ㅈ	ㅊ	ᄌ
	ㅊ	ㅊ	ᆽ
c	ㅌ	ㅎ	ᄎ
	ㅎ	ㅎ	ᆾ
k	ㅋ	ㅋ	ᄏ
	ㅋ	ㅋ	ᆿ
t	ㅌ	ㅌ	ᄐ
	ㅌ	ㅌ	ᇀ
p	ㅍ	ㅍ	ᄑ
	ㅍ	ㅍ	ᇁ
h	ㅎ	ㅎ	ᄒ
	ㅎ	ㅎ	ᇂ

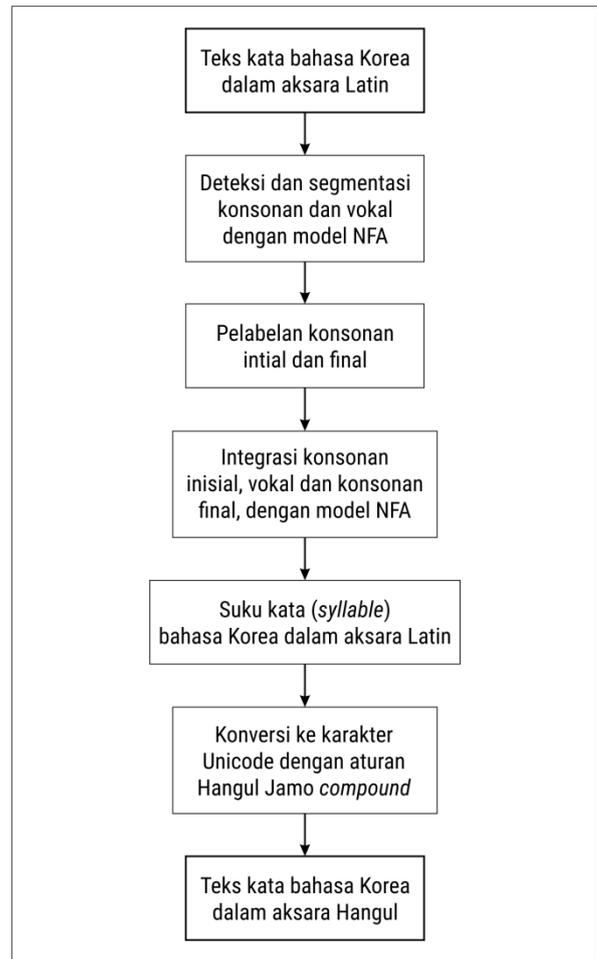
Informasi ini kemudian digunakan sebagai basis pengetahuan untuk menyusun model FSA yang akan digunakan untuk mengidentifikasi vokal dan konsonan, serta melakukan proses segmentasi suku kata (silabifikasi).

TABEL IV
ROMANISASI VOKAL HANGUL

Vokal	
Karakter pertama	Probabilitas sekuens dalam Romanisasi
a	a, ae
y	ya, yae, ye, yeo, yo, yu, yi
e	e, eo, eu
w	wa, wae, we, weo, wi
o	o, oe
i	i
u	u

A. Alur Penelitian

Setelah semua persiapan selesai dilakukan, maka penelitian dimulai dengan mengikuti alur penelitian pada Gambar 4. Teks bahasa Korea dalam aksara Latin akan menjadi masukan bagi sistem untuk selanjutnya diproses menggunakan aturan-aturan yang dibuat menurut rancangan model FSA.



Gambar. 4. Blok Diagram Alur Penelitian

Model FSA pertama digunakan untuk melakukan identifikasi vokal dan konsonan dari kata yang dimasukkan, karena romanisasi Hangul belum tentu menggunakan satu karakter untuk merepresentasikan satu vokal atau konsonan dalam aksara Hangul.

Setelah itu, sesuai aturan hangul, dilakukan pelabelan terhadap konsonan untuk menandai apakah konsonan tersebut termasuk initial (konsonan awal) atau final (konsonan akhir) dalam sebuah suku kata.

Model FSA ke dua digunakan untuk melakukan segmentasi terhadap suku kata yang ada di dalam kata tersebut melalui deteksi sekuens konsonan dan vokal, serta integrasi komponen yang mungkin tercerai dan belum berhasil direkatkan dalam deteksi *pass* pertama. Setelah setiap suku kata terdeteksi, maka suku kata tersebut kemudian dikonversi ke Hangul Unicode menggunakan kode Hangul Jamo yang telah disediakan. Luaran terakhir dari sistem adalah aksara *compound* Hangul yang sesuai dengan kata yang diinputkan.

B. Perancangan Model Finite State Automata

Rancangan Finite State Automata (FSA) yang dibuat dituangkan dalam bentuk *state diagram* yang mengikuti prinsip Deterministic Finite Automata (DFA) di mana seluruh probabilitas transisi dari satu *state* ke *state* yang lain sudah dituliskan dengan tegas dan tidak memberi ruang untuk

transisi epsilon [15]. Transisi epsilon adalah transisi di mana sistem tidak menerima input apapun saat terjadi perubahan *state*. Hasil rancangan FSA untuk identifikasi konsonan ditunjukkan oleh Tabel 5, sementara rancangan FSA untuk identifikasi vokal ditunjukkan oleh Tabel 6.

TABEL V
RANCANGAN FSA UNTUK IDENTIFIKASI KONSONAN

Konsonan		
Karakter pertama	Probabilitas sekuens	Model FSA
g	g, gg (initial), gs (final)	
d	d, dd (initial)	
b	b, bb (initial), bs (final)	
s	s, ss	
j	j, jj	
n	n, nj (final), nh (final), ng (final)	
l	l, lg, lm, lb, ls, lt, lp, lh (semua final)	

TABEL VI
RANCANGAN FSA UNTUK IDENTIFIKASI VOKAL

Vokal		
Karakter pertama	Probabilitas sekuens	Model FSA
a	a, ae	
y	ya, yae, ye, yeo, yo, yu, yi	
e	e, eo, eu	
w	wa, wae, we, weo, wi	
o	o, oe	

Langkah berikutnya adalah menyusun model FSA untuk segmentasi suku kata (silabifikasi) berdasarkan probabilitas susunan antara vokal dan konsonan pada komposisi suku kata Hangul. Model FSA yang dihasilkan ditunjukkan oleh Tabel 7.

TABEL VII
RANCANGAN FSA UNTUK SEGMENTASI SUKU KATA

Probabilitas Sekuens	Rancangan FSA
k[initial] – v – k[final]	
k[initial] – v	
v	
v – k[final]	

Tanda lingkaran ganda (*final state*) menunjukkan bahwa *input* dapat berhenti lengkap pada *state* tersebut. Jika pada *final state* masih terdapat garis berarah menuju ke *state* lainnya, maka masih terdapat pilihan atau kemungkinan untuk tidak berhenti dan melanjutkan ke *state* selanjutnya.

C. Pembuatan Sistem Silabifikasi dan Transliterasi

Rancangan Finite State Automata (FSA) yang telah dibuat kemudian dituangkan menjadi serangkaian aturan (*rules*) yang dipakai untuk menelusuri *string* masukan berupa kata bahasa Korea yang ditulis dalam aksara Latin (romanisasi). Aturan ini kemudian dikodekan terlebih dahulu dalam bahasa pemrograman Borland Delphi kemudian dilakukan *porting* menuju bahasa *scripting* PHP dan dimuat dalam *web hosting* agar hasilnya dapat diakses dan digunakan oleh khalayak ramai. Selain pengkodean, dibuat sebuah tabel konversi konsonan vokal ke angka Unicode yang sekaligus berperan sebagai tabel identifikasi initial, medial dan final. Tabel ini disimpan dalam senarai karena isinya tidak berubah dan tetap, sehingga tidak membutuhkan basis data. Tabel konversi ini selengkapnya ditunjukkan oleh Tabel 8. Keterangan: AU: Angka Unicode, Ltn: Latin, Hgl: Hangul.

TABEL VIII
KONVERSI INITIAL, MEDIAL DAN FINAL KE ANGKA UNICODE

Initial			Medial			Final		
AU	Latin	Hangul	AU	Latin	Hangul	AU	Latin	Hangul
1	g	ㄱ	1	a	ㅏ	1	g	ㄱ
2	gg	ㄲ	2	ae	ㅑ	2	gg	ㄲ
3	n	ㄴ	3	ya	ㅓ	3	gs	ㄴ
4	d	ㄷ	4	yae	ㅕ	4	n	ㄴ
5	dd	ㄸ	5	eo	ㅗ	5	nj	ㄴ
6	r	ㄹ	6	e	ㅛ	6	nh	ㄴ
7	m	ㅁ	7	yeo	ㅜ	7	d	ㄷ
8	b	ㅂ	8	ye	ㅠ	8	l	ㄷ
9	bb	ㅃ	9	o	ㅠ	9	lg	ㄷ
10	s	ㅅ	10	wa	ㅝ	10	lm	ㄷ
11	ss	ㅆ	11	wae	ㅞ	11	lb	ㄷ
12		ㅇ	12	oe	ㅟ	12	ls	ㄷ
13	j	ㅈ	13	yo	ㅠ	13	lt	ㄷ
14	jj	ㅉ	14	u	ㅠ	14	lp	ㄷ
15	c	ㅊ	15	weo	ㅠ	15	lh	ㄷ
16	k	ㅋ	16	we	ㅠ	16	m	ㅁ

17	t	ㅌ	17	wi	ㅍ	17	b	ㅂ
18	p	ㅍ	18	yu	ㅠ	18	bs	ㅃ
19	h	ㅎ	19	eu	ㅡ	19	s	ㅅ
			20	yi	ㅣ	20	ss	ㅆ
			21	i	ㅣ	21	ng	ㅇ
						22	j	ㅈ
						23	c	ㅊ
						24	k	ㅋ
						25	t	ㅌ
						26	p	ㅍ
						27	h	ㅎ

Kegunaan dari angka Unicode ini adalah, jika sebuah kombinasi konsonan dan vokal sudah diketahui angka Unicodenya sesuai dengan posisinya (initial, medial atau final), maka suku kata Hangul untuknya dapat dibuat dengan mendapatkan Hangul Code Point menggunakan rumus berikut:

$$HCP = f + (m-1)*28 + (i-1)*588 + 44032 \quad (1)$$

- HCP : Hangul Code Point
- f : final
- m : medial
- i : initial

Hangul Code Point yang didapatkan akan menunjuk ke aksara *compound* Hangul yang sesuai yang telah terdapat pada bagian tabel Unicode untuk Hangul, sehingga tidak perlu dilakukan komposisi penempatan secara manual, karena *compound* ini sudah dibuat mengikuti aturan baku penulisan *compound* Hangul. Contoh konversi karakter Hangul menjadi suku kata ditunjukkan oleh Tabel 9.

TABEL IX
CONTOH PERHITUNGAN HANGUL CODE POINT

http://gernot-katzers-spice-pages.com/var/korean_hangul_unicode.html

Suku kata:	JA	MO	HAN	GEUL
Initial	J ㅈ (13)	M ㅁ (7)	H ㅎ (19)	G ㄱ (1)
Medial	A ㅏ (1)	O ㅑ (9)	A ㅏ (1)	EU ㅡ (19)
Final	- (0)	- (0)	N ㄴ (4)	L ㄹ (8)
Hangul code point (dec)	51088	47784	54620	44544
Hangul code point (hex)	C790	BAA8	D55C	AE00
Unicode (HTML Unicode)	U+C790 자	U+BAA8 모	U+D55C 한	U+AE00 글
Hangul	자	모	한	글

D. Pengujian

Perangkat lunak yang telah dibuat kemudian diuji menggunakan 100 buah kata bahasa Korea dalam aksara Latin yang diambil dari kamus Bahasa Inggris-Bahasa Korea agar memperoleh pemisahan suku kata dan bentuk akhir *compound* Hangul yang benar sebagai pembanding

hasil keluaran perangkat lunak. 100 buah kata tersebut diambil secara acak dan contohnya ditunjukkan oleh Tabel 10.

TABEL X
CONTOH KATA UJI

Kata	Arti dalam Bahasa Inggris	Suku Kata
gyejwa	account	gye jwa
haengdong	action	haeng dong
daeryag	about, around	dae ryag
bihaengjang	airport	bi haeng jang
ddaraseo	along	dda ra seo
geum-aeg	amount	geum aeg
peurogeuraem	program	peu ro geu raem
chaeneol	channel	chae neol
kaerigteo	character	kae rig teo
bulpyeong	charge	bul pyeong
babbeuda	busy	ba bbeu da
sa-eob	business	sa eob
dalryeog	calendar	dal ryeog
oechim	call	oe chim
sajingi	camera	sa jin gi
hwajangsil	can	hwa jang sil
chwiso	cancel	chwi so
gibbeuda	content	gi bbeu da
gyeyag	contract	gye yag
iyagi	conversation	i ya gi

IV. HASIL DAN PEMBAHASAN

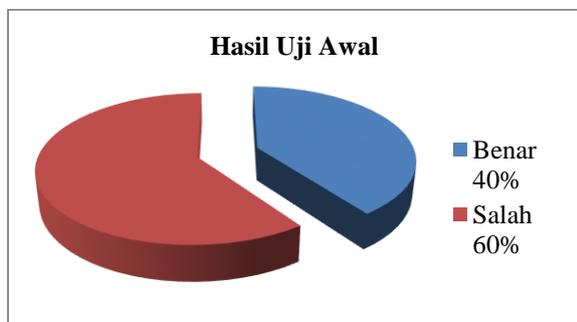
Pengujian awal menggunakan murni rancangan FSA yang telah dibuat menunjukkan hasil yang tidak memuaskan untuk proses silabifikasi (pemisahan suku kata). Jika proses silabifikasi tidak benar maka konsekuensinya adalah proses konversi *compound* Hangul Jamo juga akan mengalami kesalahan. Contoh hasil pengujian awal ini ditunjukkan oleh Tabel 11.

TABEL XI
CONTOH HASIL PENGUJIAN AWAL

Kata	Hasil Pengujian dengan Sistem		Hasil
	Silabifikasi	Angka Unicode	
gyejwa	gye j wa	[1-8-0] [13-0-0] [12-10-0]	Salah
haengdong	haeng dong	[19-2-21] [4-9-21] [12-0-0]	Benar
daeryag	dae r yag	[4-2-0] [6-0-0] [12-3-1]	Salah
bihaengjang	bi h aeng jang	[8-21-0] [19-0-0] [12-2-21] [13-1-21]	Salah
ddaraseo	dda r a s eo	[5-1-0] [6-0-0] [12-1-0] [10-0-0] [12-5-0]	Salah
geum-aeg	geum aeg	[1-19-16] [12-0-0] [12-2-1]	Benar
peurogeuraem	peu r o g eu r aem	[18-19-0] [6-0-0] [12-9-0]	Salah

		[1-0-0] [12-19-0] [6-0-0] [12-2-16]	
gong-gyeoghada	gong gyeog ha da	[1-9-21] [12-0-0] [1-7-1] [19-1-0] [4-0-0] [12-1-0]	Salah
dab-byeon	dab byeon	[4-1-17] [12-0-0] [8-7-4]	Benar
hae gyeol	hae g yeol	[19-2-0] [1-0-0] [12-7-8]	Salah

Kesalahan yang paling banyak muncul pada pengujian awal ini adalah terdapat konsonan yang tidak berperan sebagai salah satu dari initial atau final, dengan kata lain ia tidak terikat dengan nukleus manapun. Konsonan ini biasanya terimpit oleh dua vokal sehingga sulit untuk menentukan ia adalah initial atau final, dan akhirnya hanya diberi label konsonan saja. Hasil pengujian untuk 100 kata uji ditunjukkan oleh grafik pada Gambar 5.



Gambar. 5. Hasil Uji Awal

Untuk mengatasi konsonan yang gagal diidentifikasi sebagai initial atau final, ditambahkan aturan-aturan berikut seperti pada Gambar 6, yang dituangkan dalam bentuk logika pemrograman menurut bahasa Borland Delphi. Aturan tambahan ini meniadakan konsonan yang sendirian dengan menggabungkannya ke segmen apapun yang ada di belakangnya. Langkah ini diambil berdasarkan pengamatan yang dilakukan terhadap hasil-hasil pengujian awal yang salah.

Gambar. 6. Kode Program Aturan Tambahan Pertama

```

for j:=1 to sylcount do begin
  initial[j]:=trim(initial[j]);
  final[j]:=trim(final[j]);
  if syllabel[j]='k' then begin
    syllabel[j]:='e';
    syllable[j+1]:=syllable[j]+
      syllable[j+1];
    initial[j+1]:=initial[j]+
      initial[j+1];
    initial[j]:='';
    syllable[j]:='';
  end;
end;

```

Aturan yang ditambahkan tersebut memberi efek positif terhadap hasil pengujian ulang, seperti ditunjukkan

oleh Tabel 12. Meskipun demikian masih terdapat beberapa kesalahan yang perlu diperbaiki seperti ditunjukkan oleh Tabel 13.

TABEL XII
CONTOH HASIL PENGUJIAN SETELAH PENAMBAHAN ATURAN

Kata	Hasil Pengujian dengan Sistem		Hasil	Hangul
	Silabifikasi	Angka Unicode		
gyejwa	gye jwa	[1-8-0] [13-10-0]	Benar	계좌
haengdong	haeng dong	[19-2-21] [4-9-21]	Benar	행동
daeryag	dae ryag	[4-2-0] [6-3-1]	Benar	대략
bihaengjang	bi haeng jang	[8-21-0] [19-2-21] [13-1-21]	Benar	비행장
ddaraseo	dda ra seo	[5-1-0] [6-1-0] [10-5-0]	Benar	따라서
geum-aeg	geum aeg	[1-19-16] [12-2-1]	Benar	금액
peurogeuraem	peu ro geu raem	[18-19-0] [6-9-0] [1-19-0] [6-2-16]	Benar	프로그램
gong-gyeoghada	gong gyeog ha da	[1-9-21] [1-7-1] [19-1-0] [4-1-0]	Benar	공격하다
dab-byeon	dab byeon	[4-1-17] [8-7-4]	Benar	답변
hae gyeol	hae gyeol	[19-2-0] [1-7-8]	Benar	해결

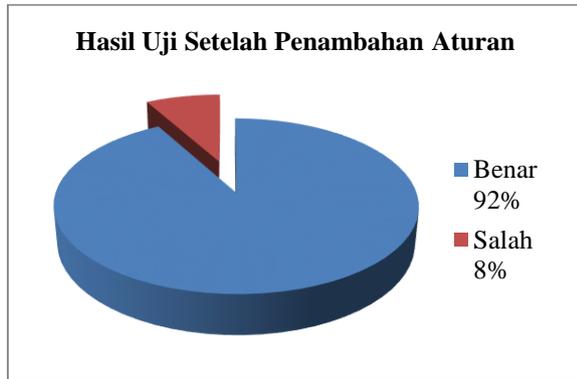
TABEL XIII
KESALAHAN SILABIFIKASI

Kata	Segmentasi seharusnya	Segmentasi hasil sistem
hyeolaeg	hyeol aeg	hyeo laeg
sajingi	sa jin gi	sa ji ngi
bulpyeong	bul pyeong	bulp yeong
silpae	sil pae	silp ae
jasingam	ja sin gam	ja sing am
munhwa	mun hwa	munh wa
hyeonjae	hyeon jae	hyeonj ae
hwagsilhi	hwag sil hi	hwags ilh i

Kesalahan silabifikasi akibat kekeliruan segmentasi suku kata yang masih tersisa adalah kesalahan identifikasi konsonan sebagai initial atau vokal. Tidak terdapat pola konsisten untuk kesalahan-kesalahan ini karena konsonan menempel secara acak pada nukleus vokal di sebelah depan atau belakangnya. Namun demikian, hasil pengujian setelah penambahan aturan ini mengalami kenaikan yang signifikan untuk jumlah kata yang dapat ditransliterasi dengan benar, seperti ditunjukkan oleh grafik pada Gambar 7.

Untuk lebih meningkatkan akurasi, maka kembali ditambahkan aturan-aturan yang terdapat pada kode pada Gambar 8. Aturan-aturan ini memang bersifat sangat

kasuistik yaitu khusus terhadap rangkaian karakter konsonan lp, lh, nh, nj, ng, gs, dengan mempertimbangkan pola rangkaian karakter yang sering muncul pada suku kata bahasa Korea dari kamus.



Gambar. 7. Hasil Uji Setelah Penambahan Aturan

```

if a='g' then begin //G
  if b='g' then cutstatus:='2ki'
  else if b='s' then begin
    if e='sil' then
      cutstatus:='1kf' else cutstatus:='2kf'
    end else cutstatus:='1k';
end
if a='l' then begin //L
  if b='g' then begin
    if (c='ga') or (c='ge') or (c='go')
      then cutstatus:='1kf' else
        cutstatus:='2kf';
    end
    else if b='m' then cutstatus:='2kf'
    else if b='b' then begin
      if (c='ba') or (c='bi') or
        (c='bo') then cutstatus:='1kf' else
        cutstatus:='2kf';
    end
    else if b='s' then cutstatus:='2kf'
    else if b='t' then cutstatus:='2kf'
    else if b='p' then begin
      if (c='py') or (e='pae') then
        cutstatus:='1kf' else cutstatus:='2kf'
      end
    else if b='h' then begin
      if (c='ha') or (c='hi') then
        cutstatus:='1kf' else cutstatus:='2kf'
      end
    else if f='eo' then cutstatus:='1kf'
    else cutstatus:='1k';
end
if a='n' then begin //N
  if b='j' then begin
    if (c='ja') or (c='je') or (c='jj')
      then cutstatus:='1kf' else
        cutstatus:='2kf';
    end
    else if b='h' then begin
      if (c='ha') or (c='he') or (c='hw')
        then cutstatus:='1kf' else
        cutstatus:='2kf'
      end
    else if b='g' then begin
      if d='geul' then cutstatus:='1kf'
      else cutstatus:='2kf';
      if e='gam' then cutstatus:='1kf'
      else cutstatus:='2kf';
    end
    else cutstatus:='1k';
end
    
```

Gambar. 8. Kode program aturan tambahan ke dua

Penambahan aturan khusus ini mampu meningkatkan tingkat akurasi hingga mencapai 99% dengan tinggal satu kata yang tidak dapat disilabifikasi dengan benar yaitu kata “sajingi” yang seharusnya disilabifikasi sebagai “sa jin gi” tetapi hasil dari sistem memberikan “sa jing i”. Pertimbangan tidak memberikan aturan khusus bagi digram “ng” adalah karena banyak sekali penggunaan konsonan “ng” pada suku kata bahasa Korea, sehingga dikhawatirkan penambahan aturan akan menyebabkan kesalahan pada silabifikasi yang sebelumnya sudah benar. Persentase akurasi final dari penelitian ini disajikan oleh grafik pada Gambar 9.



Gambar. 9. Hasil Uji Akhir

V. KESIMPULAN DAN SARAN

Penelitian telah berhasil dilaksanakan pada beberapa tahap awal dan didapatkan kesimpulan sebagai berikut:

1. Penerapan rancangan FSA yang murni mengacu pada aturan identifikasi konsonan dan vokal serta aturan penyusunan *compound* Hangul dapat mengidentifikasi konsonan dan vokal dengan benar.
2. Keberhasilan proses silabifikasi sangat mempengaruhi keberhasilan transliterasi, karena dasar penyusunan Hangul Jamo adalah suku kata hasil silabifikasi.
3. Aturan pelabelan yang diterapkan untuk memberi label initial dan final pada konsonan hasil identifikasi, tidak dapat memastikan seluruh konsonan selalu diberi label final atau initial sehingga mempengaruhi proses silabifikasi dan menghasilkan tingkat akurasi silabifikasi dan transliterasi sebesar 40%.
4. Penambahan aturan agar konsonan yang tidak teridentifikasi sebagai initial atau final disatukan dengan suku kata di belakangnya mampu meningkatkan tingkat akurasi silabifikasi dan transliterasi menjadi sebesar 92%.
5. Penambahan aturan khusus untuk digram lp, lh, nh, nj, ng, gs serta aturan perekatan konsonan “l” terhadap suku kata depan yang diakhiri “eo” mampu meningkatkan tingkat akurasi silabifikasi dan transliterasi menjadi sebesar 99%.

Berikut adalah beberapa saran yang dapat diimplementasikan untuk mendapatkan hasil yang lebih optimal:

1. Menambahkan aturan yang memperhatikan aspek kombinasi konsonan-vokal yang sering digunakan dalam penulisan Hangul atau tata bahasa Korea dalam menentukan apakah sebuah konsonan dapat menjadi initial atau vokal, dan terikat pada nukleus vokal yang mana. Aturan ini ditambahkan pada langkah segmentasi suku kata.
2. Menambahkan data uji untuk menjajagi kemungkinan aturan-aturan lain yang perlu ditambahkan.

VI. RENCANA KELANJUTAN PENELITIAN

Penelitian ini telah menghasilkan evaluasi singkat mengenai unjuk kerja dari sistem silabifikasi kata bahasa Korea yang dituliskan dengan aksara Latin dan transliterasinya ke dalam aksara Hangul. Hasil yang didapatkan sudah cukup baik yaitu 99%, tetapi masih memerlukan perbaikan lanjutan untuk mencapai hasil yang lebih baik dan merata. Beberapa hal yang diusulkan untuk dilakukan pada penelitian berikutnya adalah:

1. Melakukan evaluasi aturan-aturan yang diterapkan pada proses pengujian hingga saat ini, untuk mendapatkan hasil yang optimal.
2. Menambahkan aturan-aturan yang diperlukan untuk mencapai hasil optimal yang diinginkan dan dapat berlaku umum untuk semua konsonan dan vokal.
3. Melanjutkan pengembangan sistem untuk mengintegrasikan aturan-aturan pada nomor 2 (dua) dan melakukan pengujian kembali terhadap penerapan aturan-aturan tersebut.
4. Menambah data uji agar validitas model yang telah dibuat dapat lebih teruji untuk berbagai kasus.
5. Melibatkan penutur dan ahli bahasa Korea untuk konversi transliterasi romanisasi sehari-hari di mana kombinasi antar konsonan tertentu dapat menghasilkan pengucapan yang berbeda dengan aksara aslinya.

UCAPAN TERIMA KASIH

Penelitian telah berhasil dilaksanakan dengan dukungan dana dari Direktorat Riset dan Pengabdian Masyarakat Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian Riset, Teknologi dan Pendidikan Tinggi (Penelitian Dosen Pemula). Penulis juga ingin menghaturkan terima kasih kepada Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) Sekolah Tinggi Manajemen Informatika dan Komputer Widuri.

DAFTAR PUSTAKA

- [1] Kharusi, N. S., Salman, A. 2011. The English Transliteration of Place Names in Oman. *Journal of Academic and Applied Studies* Vol. 1(3) September 2011, hal. 1–27.
- [2] Anon. *History of Hangul – Part 1*. https://zkorean.com/hangul/history_of_hangul. Diakses tanggal 25 Mei 2016.
- [3] Anon. *Abugida*. http://www.cs.mcgill.ca/~rwest/link-suggestion/wpcd_2008-09_augmented/wp/a/Abugida.htm. Diakses tanggal 25 Mei 2016.
- [4] Trombley, G., Bullen, R., Bong, S., Ham, M. 2015. *Korean From Zero!* From Zero, LLC., Henderson.
- [5] Omae, H., Tsuji, K., Kageura, K., Imai, M., Anzai, Y. *Extraction of Compound Word Translations from Non-parallel Japanese-French Text in World Wide Web*. <http://slis.sakura.ne.jp/>. Diakses tanggal 25 Mei 2016.
- [6] Goldberg, Y., Elhadad, M. 2008. Identification of Transliterated Foreign Words in Hebrew Script.

- Computational Linguistics and Intelligent Text Processing* Volume 4919 of the series Lecture Notes in Computer Science, hal. 466-477.
- [7] Brar, D., dan Kaur, R. 2014. A Review of Transliteration system from English to Punjabi. *International Journal of Advanced Research in Computer Science and Software Engineering* Vol. 4, ed. 7, Juli 2014
- [8] Lee, J.S. dan Choi, K-S. 2014. *English to Korean Statistical Transliteration for Information Retrieval*. Department of Computer Science, Korea Advanced Institute of Science and Technology, Taejon
- [9] Jung, S.Y, Hong, S.L, Paek, E. 2000. *An English to Korean Transliteration Model of Extended Markov Window*. Information Technology Lab., LG Electronics Institute of Technology, Seoul
- [10] Lee, C. Y. 2004. *Essential Grammar For Korean as a Second Language*
- [11] McCune, G.M., Reischauer, E.O. 1939. Romanization of The Korean Language. *Transactions of the Korea Branch of the Royal Asiatic Society* Vol. XXIX, Seoul
- [12] South Korean Ministry of Culture & Tourism. *Romanization of Korean* (2000). https://web.archive.org/web/20070916025652/http://www.korea.net/korea/kor_loca.asp?code=A020303. Diakses pada 25 Mei 2016 dari tautan arsip.
- [13] Mohri, M. 1997. Finite-State Transducers in Language and Speech Processing. *Computational Linguistics* Vol. 23 issue 2, Juni 1997, hal. 269-311.
- [14] Gribkoff, E. 2013. *Finite State Transducers*. Catatan Pengajaran: University of California, Davis.
- [15] Mogensen, T. Æ. 2010. *Basics of Compiler Design*. University of Copenhagen, Copenhagen, Denmark.