WEB SERVICE APPLICATION PROGRAMMING INTERFACE (API) TRANSLITERASI AKSARA JAWA KE AKSARA LATIN DENGAN *BACKTRACKING* SEBAGAI METODE PENYUSUNAN KATA

SKRIPSI

Oleh :
SILVA AHMAD ZAKY
NIM. 15650133

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2019

WEB SERVICE APPLICATION PROGRAMMING INTERFACE (API) TRANSLITERSI AKSARA JAWA KE AKSARA LATIN DENGAN *BACKTRACKING* SEBAGAI METODE PENYUSUNAN KATA

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh:
SILVA AHMAD ZAKY
NIM. 15650133

JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2019

LEMBAR PERSETUJUAN

***WEB SERVICE APPLICATION PROGRAMMING INTERFACE* (API)**
TRANSLITERSI AKSARA JAWA KE AKSARA LATIN
DENGAN *BACKTRACKING* SEBAGAI
METODE PENYUSUNAN KATA

SKRIPSI

Oleh :
SILVA AHMAD ZAKY
NIM. 1565033

Telah Diperiksa dan Disetujui untuk Diuji

Pada Tanggal : 29 November 2019

Dosen Pembimbing I	Dosen Pembimbing II
A'la Syauqi, M. Kom	M. Imamudin, Lc., M.A
NIP. 19771201 200801 1 007	NIP. 19740602 200901 1 010

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

LEMBAR PENGESAHAN

***WEB SERVICE APPLICATION PROGRAMMING INTERFACE* (API)**
TRANSLITERSI AKSARA JAWA KE AKSARA LATIN
DENGAN *BACKTRACKING* SEBAGAI
METODE PENYUSUNAN KATA

SKRIPSI

Oleh :
SILVA AHMAD ZAKY
NIM. 1565033

Telah Dipertahankan di Depan Dewan Penguji
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Pada Tanggal 11 Desember 2019

Susunan Dewan Penguji			**Tanda Tangan**
1. Penguji Utama	:	Ajib Hanani, M.T NIDT. 19840731 20160801 1 076	( )
2. Ketua Penguji	:	Fajar Rohman Hariri, M.Kom NIP. 19890515 201801 1 001	( )
3. Sekretaris Penguji	:	A'la Syauqi, M. Kom NIP. 19771201 200801 1 007	( )
4. Anggota Penguji	:	M. Imamudin, Lc., M.A NIP. 19740602 200901 1 010	( )

Mengetahui,
Ketua Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang

Dr. Cahyo Crysdian
NIP. 19740424 200901 1 008

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini:

Nama : Silva Ahmad Zaky Zamani

NIM : 15650133

Fakultas/Jurusan : Sains dan Teknologi/Teknik Infomatika

Judul Skripsi : *Web Service Application Programming Interface* (API) Transliterasi Aksara Jawa ke Aksara Latin dengan *Backtracking* Sebagai Metode Penyusunan Kata

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila dikemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 10 November 2019
Yang membuat pernyataan,

METERAI TEMPEL
BBE9FAHF200674767
6000
ENAM RIBU RUPIAH

Silva Ahmad Zaky
NIM. 15650133

HALAMAN MOTTO

“*Jika mimpimu belum ditertawakan orang lain, berarti mimpimu masih terlalu kecil.*”

KATA PENGANTAR

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

Puji dan syukur penulis panjatkan ke hadirat Allah subhanahu wa ta'ala yang telah melimpahkan rahmat dan hidayahNya kepada kita, sehingga penulis bisa menyelesaikan skripsi dengan tepat waktu, yang kami beri judul "*Web Service Application Programming Interface* (API) Transliterasi Aksara Jawa ke Aksara Latin dengan *Backtracking* sebagai Metode Penyusunan Kata". Tujuan dari penyusunan skripsi ini guna memenuhi salah satu syarat untuk bisa menempuh ujian sarjana komputer pada Fakultas Sains dan Teknologi (FSAINTEK) Program Studi Teknik Informatika di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang. Didalam pengerjaan skripsi ini telah melibatkan banyak pihak yang sangat membantu dalam banyak hal. Oleh sebab itu, disini penulis sampaikan rasa terima kasih sedalam-dalamnya kepada:

1. Prof. Dr. Abdul Haris, M.Ag selaku Rektor Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini,M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdian, Selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
4. A'la Syauqi, M. Kom, selaku Dosen Pembimbing I yang telah membimbing dalam penyusunan skripsi ini hingga selesai.
5. M. Imamuddin, Lc., MA, selaku Dosen Pembimbing II yang telah membimbing dalam penyusunan skripsi ini hingga selesai.

6. Dr. Cahyo Crysdian, selaku Dosen Wali yang senantiasa memberikan banyak motivasi dan saran untuk kebaikan penulis.
7. Para staff laboran Fakultas Sains dan Teknologi yang telah bersedia memberikan data.
8. Orang tua tercinta yang telah banyak memberikan doa dan dukungan kepada penulis secara moril maupun materil hingga skripsi ini dapat terselesaikan.
9. Kakak tercinta juga anggota keluarga dan kerabat yang senatiasa memberikan doa dan dukungan semangat kepada penulis.
10. Sahabat-sahabat seperjuangan yang tiada henti memberi dukungan dan motivasi kepada penulis serta target bersama untuk lulus skripsi dan wisuda bersama
11. Rekan-rekan interface yang selalu memberikan semangat dan doa kepada penulis.
12. Semua pihak yang telah banyak membantu dalam penyusunan skripsi ini yang tidak bisa penulis sebutkan semuanya.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat kekurangan dan penulis berharap semoga skripsi ini bisa memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi.

Malang, 15 Desember 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Zaky, Silva Ahmad. 2019. ***Web Service Application Programming Interface (API) Transliterasi Aksara Jawa ke Aksara Latin dengan Backtracking sebagai Metode Penyusunan Kata.***

Pembimbing : (I) A'la Syauqi, M. Kom. (II) M. Imamudin, Lc., MA

Kata Kunci : *Bakctracking*, *Web Service*, API, Segmentasi, Transliterasi

Aksara Jawa yang merupakan salah satu aksara tradisional Nusantara yang perlu dilestarikan. Aksara Jawa yang penulisannya ditulis berkesinambungan tanpa ada pemisah (spasi) dalam setiap katanya menimbulkan permasalahan saat dilakukan transliterasi ke aksara Latin. Huruf latin hasil transliterasi akan bersambung tanpa spasi layaknya aksara Jawa menjadikan teks hasil transliterasi sulit dibaca.

Penelitian ini menggunakan metode *backtracking* untuk memisahkan masing masing kata tanpa spasi tersebut agar terpisah sesuai kaidah penulisannya. Metode ini dipilih karena akurasinya yang tinggi dalam menangani beberapa kasus yang berkaitan dengan segmentasi kata. Hasil transliterasi kemudian ditampilkan dalam bentuk API menggunakan *web service* agar dapat diakses oleh platform web maupun android.

Pengujian dilakukan menggunakan 50 kalimat aksara Jawa yang bersumber dari penelitian sebelumnya yang berjudul Aplikasi Penterjemah Kalimat Bahasa Indonesia ke Bahasa Jawa disertai Transliterasi Aksara Jawa. Diketahui dari hasil pengujian bahwa akurasi sistem untuk transliterasi dari aksara Jawa ke aksara Latin adalah sebesar 96%. Kemudian penyusunan kata menggunakan *backtracking* dengan mengabaikan aksara "ꦩ" pada kata yang memiliki awalan (a, i, u ,e, dan o) memiliki akurasi sebesar 86% dan akurasi sebesar 58% untuk penyusunan kata menggunakan *backtracking* dengan menggunakan aksara "ꦩ" pada kata yang memiliki awalan (a, i, u ,e, dan o). Waktu yang diperlukan sistem dalam menyusun kata, berbanding lurus dengan panjang karakter atau kalimat yang akan disusun.

ABSTRACT

Zaky, Silva Ahmad. 2019. ***Web Service Application Programming Interface (API) Transliterasi Aksara Jawa ke Aksara Latin dengan Backtracking sebagai Metode Penyusunan Kata.***

Supervisor : (I) A'la Syauqi, M. Kom. (II) M. Imamudin, Lc., MA.

Keyword : *Bakctracking, Web Service,* API, *Segmentatation, Transliteration*

Javanese script is one of Indonesia's traditional scripts that need to be preserved. Javanese scripts are written continuously without separating (spaces) in each word causing problems when transliterating from Javanese to Latin. Latin characters resulting from transliteration will continue without spaces such as Javanese script which makes transliteration texts difficult to read.

This research uses a backtracking method to separate each word without spaces so that it is separated according to the writing conventions. This method was chosen because of its high accuracy in handling several cases related to word segmentation. The transliteration results are then displayed in the form of an API using a web service so that it can be accessed by the web and android platforms.

Tests carried out by using 50 sentences Javanese script derived from previous research titled Sentences Indonesian Translators Application to the Java Language Accompanied by English transliteration Java. From the test results known that the accuracy of the transliteration system of the Java language to Latin is 96%. Then, the preparation of words using backtracking by ignoring the letters "ꦩ" in words that have prefixes (a, i, u, e, and o) have an accuracy of 86% and 58% accuracy for preparation of words using backward by using the character "ꦩ" in words that have a prefix (a, i, u, e, and o). The time it takes the system to construct a sentence, is directly proportional to the length of a character or a sentence to be regulated.

مستخلص البحث

زكي، سيلفا أحمد. 2019. *خدمة ويب تطبيق بالبرمجة الواجهة (API) ترجمة الخط الجاوي إلى اللاتيني بالروجوعي كمنهجية تأليف الكلمات*

المشرف الأول: أعلى شوقي الماجستير، والمشرف الثاني: محمد إمام الدين الماجستير

الكلمة الأساسية: الرجوعي، خدمة الويب، الترجمة

الخط الجاوي من أحد خطوط تقليدية بنوسانتارا التي تحتاج إلى محافظته. الخط الجاوي الذي كتابته مكتوب متصل بدون المسافة كل كلماته تسبب المشكلات عندما تعمله الترجمة إلى الخط اللاتيني. الخط اللاتيني الحاصل من الترجمة المتصل بدون المسافة كالخط الجاوي تسبب النصوص الحاصلة من الترجمة صعوبة للمقروء.

المنهج المستخدم في هذا البحث هو المنهج الرجوعي لانفصال كل الكلمات بدون تلك المسافة لكي تفصل وفقا بالقواعد الإملائية. يفضل هذا المنهج لأنه لدى الدقة العالية على مواجهة بعض الأحوال التي تتعلق بقطع الكلمات. وأما نتائج الترجمة ستعرض على شكل *API* التي تستخدم بخدمة الويب لكي تعمل منصة الويب أو أندرويد.

يقام هذا الاختبار باستخدام خمسين كلمات في الخط الجاوي المصدر من الدراسة السابقة تحت الموضوع تطبيق ترجمة الكلمات الإندونيسية إلى الجاوية مصاحبة ترجمة الخط الجاوي. معلوما من نتائج الاختبار أن دقة نظام ترجمة الخط الجاوي إلى اللاتيني تبلغ إلى ٩٦ %. ثم تأليف الكلمات المستخدمة إلى الرجوعي مع إهمال خط "ꦁ" في الكلمات التي تبدأ بأحرف الصائتة الجاوية (a, i, u, e, o) لديها دقة تبلغ إلى ٨٦ % و58 % لتأليف الكلمات المستخدمة إلى الرجوعي مع استخدام خط "ꦁ" في الكلمات التي تبدأ بأحرف الصائتة الجاوية (a, i, u, e, o). أما الوقت المحتاج بالنظام في تأليف الكلمات، فمتوقف على الأحرف أو الكلمات المؤلفة.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Transliterasi adalah representasi dari kata yang diberikan dalam bahasa sumber menjadi sebuah kata dalam bahasa target, di mana abjad dari bahasa target hanya perlu digunakan tanpa mengubah fonem kata. Transliterasi adalah kombinasi dari dua kata: Trans + Littera, di mana Trans berarti mengubah dan Littera adalah kata Latin yang berarti surat. Arti dari transliterasi secara istilah adalah perubahan huruf / kata-kata menjadi karakter yang sesuai pada abjad / bahasa lain (Dhindsa dan Sharma, 2017). Transliterasi dapat dilakukan terhadap berbagai macam aksara, mulai dari Arab, Cina, Sunda, sampai dengan aksara Jawa. Meskipun begitu, penulis akan memfokuskan penelitian ini terhadap aksara Jawa yang mana aksara tersebut merupakan salah satu warisan budaya yang perlu dilestarikan keberadaannya.

Aksara Jawa yang dikenal juga sebagai Hanacaraka dan Carakan merupakan salah satu aksara tradisional Nusantara yang digunakan untuk menulis bahasa Jawa dan sejumlah bahasa daerah Indonesia lainnya, seperti bahasa Sunda dan bahasa Sasak (Yohanes, 2017). Walaupun sekarang ini masih digunakan, aksara Hanacaraka sudah hampir ditinggalkan seiring dengan diperkenalkannya penulisan aksara Latin yang menjadi standar dunia internasional (Atina, Palgunadi, & Widiarto, 2016). Penggunaan Aksara Hanacaraka secara umum pun juga sangat terbatas, misalnya pada papan penunjuk jalan, papan nama, dan beberapa artikel yang ada pada koran serta majalah (Sulaiman, t.t.). Padahal kesadaran masyarakat akan pentingnya melestarikan aksara Jawa sebagai salah satu peninggalan budaya

sangat berpengaruh terhadap eksistensi aksara Jawa tersebut di dunia yang serba modern ini. Terlebih lagi para generasi muda saat ini lebih tertarik untuk mengenal dan mempelajari kebudayaan dari negara lain yang mereka anggap populer, akibatnya kebudayaan sendiri mulai terlupakan (Astuti, 2016).

Allah Swt. telah berfirman di dalam Alqur'an surat al-Hujuraat/49:13, seperti berikut ini :

يَا أَيُّهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاكُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا ۚ إِنَّ أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتْقَاكُمْ ۚ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ

Artinya : "Hai manusia, sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa-bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. Sesungguhnya orang yang paling mulia di antara kamu di sisi Allah ialah orang yang paling takwa di antara kamu. Sesungguhnya Allah Maha Mengetahui lagi Maha Mengenal". (QS. Al-Hujuraat/49:13).

Pada Alqur'an surat al-Hujuraat ayat 13 di atas, Allah Swt. menjelaskan bahwa kita diciptakan oleh Allah dari berbagai kelompok etnis untuk saling mengenal. Maksud dari saling mengenal ini adalah, kita tidak dapat memilih untuk dilahirkan dari rahim seorang ibu dari agama apa, dari keluarga mana atau tinggal dimana keluarga kita. Meskipun begitu, kita tetap dapat saling mengenal dan mengetahui keragaman budaya satu sama lain. Dengan bekal saling mengenal keragaman budaya satu sama lain, kita akan lebih toleran terhadap pandangan orang lain. Dan dengan saling mengenal itu pula tercipatalah hubungan yang lebih di sekeliling kita. Dengan kondisi tersebut, penulis sebagai generasi penerus bangsa, selayaknya mempertahankan dan melestarikan warisan budaya tersebut agar terjaga generasi-generasi berikutnya. Salah satu cara melestarikan adalah mengenalkan. Mengenal di sini adalah membagikan pengetahuan mengenai aksara Jawa melalui sebuah aplikasi Web Service Application Programming Interface (API) yang dapat

digunakan untuk mentransliterasi aksara Jawa ke aksara Latin agar aksara tersebut tetap dikenal dan dapat dimanfaatkan orang lain baik itu dibaca maupun dikembangkan menjadi aplikasi aplikasi yang lebih interaktif nantinya.

Application Programming Interface (API) adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu (Saputro, t.t.). Dengan API, tambahan platform baru pada sebuah sistem, tidak akan mempengaruhi kinerja platform yang sudah ada. Hal ini dikarenakan API yang digunakan sudah mempunyai standar dan aturan yang baku dalam melayani permintaan dari *multiplatform*. KelebihanAPI yang lain adalah fitur keamanan dan hak akses yang dapat diatur agar terbuka bagi pengembang lainnya secara publik. Sehingga, developer lain dapat ikut berkontribusi dalam mengembangkan API untuk transliterasi aksara Jawa ke aksara Latin yang dibuat penulis ini sebagai wujud kepedulian mereka terhadap warisan budaya daerah. Meskipun demikian, API tidak dapat serta merta dapat diakses secara publik. Agar API tersebut dapat diakses secara publik, maka dibutuhkan *web wervice* sebagai layanan untuk mengakses API tersebut.

Web service adalah sistem terdistribusi yang memiliki komponen yang dapat digunakan dan diakses menggunakan protokol HTTP (*Hyper Text Transport Protocol*) dan HTTPS (*HTTP Secure*). Layanan *web* ini dapat diprogram dalam berbagai bahasa pemrograman yang ada. Klien yang mengakses layanan dari server web dapat melalui desktop / PC atau seluler (Yusrizal, 2017). Protokol untuk mengatur layanan *web* tersebut yang saat ini populer adalah *Representational State Transfer* (REST). Pada REST, terdapat API yang berfungsi mengintegrasikan aplikasi yang berbeda secara bersamaan. Tujuan menggunakan *web service* dengan

API adalah untuk mempercepat proses pengembangan dengan menyediakan fungsi secara terpisah sehingga pengembang tidak perlu membuat fitur serupa. Penggunaan *web service* akan sangat terasa jika fitur yang diinginkan cukup kompleks. Misalnya untuk penelitian tentang transliterasi aksara Jawa menjadi aksara latin ini. Tentu saja butuh waktu lagi untuk membuat sesuatu yang mirip dengannya dengan platform yang berbeda-beda.

Di sisi lain, aksara Jawa yang bersifat silabik dan penulisannya ditulis berkesinambungan tanpa ada pemisah (spasi) dalam setiap katanya menimbulkan permasalahan baru saat dilakukan transliterasi. Huruf latin hasil transliterasi akan bersambung tanpa spasi layaknya aksara Jawa. Hal ini tentu saja akan menjadikan teks hasil alih aksara tadi menjadi sulit dibaca dan API hasil transliterasi menjadi kurang tepat.

Salah satu algoritma yang mampu memecahkan permasalahan tersebut adalah algoritma *backtracking.* Algoritma yang berbasis pada *depth first search* ini mampu bekerja secara rekursif untuk melakukan pencarian kata pada kamus dan digunakan sebagai pembanding untuk memisahkan kata hasil transliterasi (Hasani dan Mustafidah, 2016). Pencarian dengan metode ini dilakukan dari node awal secara mendalam hingga yang paling akhir (*dead-end*) atau sampai ditemukan (Utama, 2016). Sehingga apabila metode ini diterapkan, pencarian kata pada kamus akan dilakukan dengan mengunjungi cabang kata yang paling dekat terlebih dahulu sampai tiba di simpul kata yang paling akhir dan kembali lagi mengulang kunjungan ke simpul awal sampai semua kombinasi kata ditemukan.

Kelebihan lain dari metode ini adalah memiliki akurasi yang tinggi (Huang, 2013). Alasannya metode ini dapat memanggil lagi hasil segmentasi kata yang telah

berlalu. Sehingga, hasil penerapan metode runut balik pada penelitian tersebut menghasilkan akurasi yang cukup tinggi yakni 89,6%. Kemudian pada penelitian lain tentang segmentasi kata Cina, menyebutkan bahwa penggunaan metode runut balik cukup efektif dalam meningkatkan akurasi untuk segmentasi kata Cina yang tidak memiliki spasi pada aksaranya (Liu, 2014).

Kemudian kelebihan lainnya yaitu metode runut balik atau *backtracking* efisien dalam pencarian sebuah informasi. Algoritma ini tidak memeriksa semua kemungkinan solusi yang ada, melainkan ini hanya melakukan pencarian yang mengarah ke solusi saja. Langkah yang diambil lebih sedikit, sehingga waktunya pun akan lebih cepat dibanding mencoba keseluruhan kemungkina atau *brute-force* (Ilmania, 2017).

1.2 Identifikasi Masalah

Seberapa akurat penggunaan algoritma *backtracking* sebagai metode penyusunan kata pada aplikasi *web service* API transliterasi aksara jawa ke aksara latin?

1.3 Batasan Masalah

Agar penelitian ini dapat fokus pada permasalahan dan pembahasan tidak meluas, maka penlitian ini dibatasi dengan beberapa batasan masalah di antaranya:

a. Kata yang diproses merupakan kata bersifat umum bersumber dari kamus Basusastra Jawa.
b. Kalimat yang dipisah mengabaikan kata ganti kepemilikan dan preposisi.
c. Penelitian berfokus pada transliterasi aksara Jawa ke latin dan tidak sebaliknya.

d. Implementasi dari API hanya dibatasi pada protokol *Representational State Transfer* (REST).

e. Aplikasi *client* untuk pengujian API dibatasi hanya pada platform Android.

f. API yang dihasilkan ditampilkan dalam format *json*.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun *web service* API transliterasi aksara jawa ke aksara latin yang teruji keakurasiannya.

1.5 Manfaat Penelitian

Penelitian ini diharapkan memberi manfaat bagi akademisi maupun masyarakat pada umumnya.

1.5.1 Manfaat bagi akademisi

a. Untuk mengetahui akurasi dari metode *backtracking* dalam penyusunan kata.

b. Penelitian ini dapat memberikan informasi di bidang filologi, khususnya yang berkaitan dengan aksara Jawa.

c. API untuk transliterasi dapat digunakan sebagai data penunjang dan masukan dalam melakukan penelitian serupa bagi peneliti lain

1.5.2 Manfaat bagi masyarakat

a. Ikut serta dalam melestarikan budaya Jawa khususnya tradisi tulis-menulis menggunakan aksara Jawa.

b. Membantu masyarakat mengalihkan aksara Jawa ke latin melaui sistem secara otomatis sehingga memudahkan dalam memahami isinya.

c. Menjadi motivasi siswa atau masyarakat pada umumnya untuk meningkatkan kemampuan menulis aksara Jawa melalui program transliterasi ortografi otomatis.

1.6 Sistematika Penulisan

Sistematika penulisan laporan penelitian ini terdiri dari lima bab yang terdiri dari pendahuluan, tinjauan pustaka, metode penelitian, pembahasan, dan penutup. Masing-masing bab akan dijelaskan sebagai berikut :

BAB I : PENDAHULUAN

Pendahuluan ini berisi latar belakang masalah yang akan diselidiki, tujuan dan manfaat dari penelitian, kemudian batasan-batasan masalah dalam penelitian, metodologi penelitian, dan sistematika laporan penelitian.

BAB II: TINJAUAN PUSTAKA

Tinjauan pustaka akan membahas mengenai analisis permasalahan-permasalah pada penelitian dan penelitian-penelitian terkait *web service* API untuk transliterasi aksara Jawa ke aksara Latin dengan *backtracking* sebagai metode penyusunan kata.

BAB III : METODE PENELITIAN

Metode penelitian akan membahas mengenai analisis permasalahan penelitian dan penjelasan tentang rancangan struktur program dan antarmuka dari aplikasi *web service* API transliterasi yang dibuat.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab hasil dan pembahasan ini, akan dijelaskan implementasi sistem berdasarkan gambaran antarmuka aplikasi yang sebelumnya telah dibuat dan

pengujian aplikasi terhadap data uji untuk membuktikan bahwa aplikasi dapat berjalan dengan baik. Pada bab ini juga akan membahas hasil pengujian berupa presentase akurasi keberhasilan sistem serta efektivitas dari pengujian tersebut.

BAB V : PENUTUP

Bab terakhir atau penutup bab akan membahas tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang dapat dilakukan untuk mengembangkan penelitian ini di masa mendatang..

BAB II
KAJIAN PUSTAKA

Bab ini menjelaskan beberapa studi pustaka yang digunakan sebagai dasar teori dalam penelitian. Selain itu, bab ini juga membahas tentang penelitian sebelumnya yang terkait dengan penelitian yang akan dilakukan.

2.1 Penelitian Terkait

2.1.1. Transliterasi Aksara Jawa

Program transliterasi antara aksara latin dan aksara jawa dengan metode FSA (Atina dkk., 2016). Pada program tersebut, pengolahan aksara dilakukan dengan cara memenggal kata berdasarkan pola suku kata dan aturan penulisan aksara jawa. Algoritma *Finite State Automata* (FST) yang digunakan pada program tersebut mampu mentransliterasi aksara latin ke aksara jawa maupun sebaliknya. Pengujian pada transliterasi aksara latin ke aksara jawa didapatkan dari dokumen Serat Radyapustaka sepanjang 1 paragraf dengan jumlah suku kata sebanyak 21. Sedangkan untuk transliterasi aksara jawa ke aksara latin diambil dari Serat Rangsang Tuban sepanjang 1 paragraf dengan jumlah kata sebanyak 29. Dari pengujian tersebut, diketahui bahwa inputan dari aksara jawa ke latin sudah memiliki spasi. Padahal, pada dokumen dokumen aksara jawa klasik, setiap kata pada kalimat tidak akan dipisahkan oleh spasi maupun titik melainkan menggunakan pangkon. Sehingga apabila diterapkan pada program ini, program akan mengalami masalah saat membedakan kata per kata karena tidak adanya spasi pada aksara jawa.

Aplikasi transliterasi aksara Latin ke aksara Jawa (Ayumitha, 2014). Penelitian yang dilakukan oleh Ayumitha tersebut dilakukan dengan menerapkan

metode *Decision Tree* dan sifat silabik dari aksara jawa dapat diatasi dengan baik. Keakuratan aplikasi ini juga sangat tinggi, yakni mencapai 90%. Dari penelitian tersebut dapat disimpulkan bahwa aplikasi ini hanya dapat menangani transliterasi dari aksara latin ke aksara jawa namun tidak sebaliknya. Sehingga perlu dibuat penelitian lanjutan untuk transliterasi dari aksara jawa ke aksara latin. Selain itu, penelitian lanjutan dibutuhkan karena penulisan aksara Jawa terkadang melewati batas kanan dari dokumen karena dalam penulisan aksara Jawa tidak mengenal spasi.

Sistem penerjemah bahasa Jawa - aksara Jawa berbasis *Finite State Automata* (Yohanes, 2017). Sistem penerjemah bahasa Jawa ke aksara Jawa dan sebaliknya dapat bekerja dengan baik dengan menggunakan metode *Finite State Automata*. Ditandai dengan hasil penerjemahan bahasa Jawa ke aksara Jawa yang mencapai tingkat keberhasilan 92% untuk penerjemahan bahasa Jawa ke aksara Jawa dan 93.8% untuk penerjemahan dari aksara Jawa ke bahasa Jawa. Meskipun demikian, penerjemahan dari aksara Jawa ke bahasa Jawa masih memiliki kekurangan yakni belum dapat memisahkan setiap kata bila pada aksara Jawa aslinya tertulis tanpa spasi. Seperti contoh kata “NabiNuh”. Tidak ada spasi setelah diterjemahkan dengan system tersebut.

Dictionary-based Word Segmentation for Javanese (Dipta dan Mirna, 2016). Penelitian tersebut membahas tentang segemntasi kata bahasa Jawa menggunakan pendekatan kamus. Algoritma yang digunakan dalam penelitian ini adalah pencocokan maksimum (*Maximal Matching Algorithm*) dengan beberapa modifikasi. Tahap pertama adalah menghasilkan semua kemungkinan kata hasil segmentasi menggunakan pendekatan kamus bahasa Jawa menggunakan. Kata

yang benar dipilih berdasarkan karakter terakhir dalam sebuah kata, dua karakter terakhir pada sebuah kata, perbedaan dari dua kata berturut-turut, dan kata yang sering muncul atau dipakai pada artikel korpus tambahan. Eksperimen ini menunjukkan hasil identifikasi kata menggunakan frekuensi kata yang sering muncul pada artkel korpus tambahan menghasilkan akurasi terbaik yakni 91,08%. Meskipun begitu, metode ini masih terdapat beberapa kelemahan yakni belum dapat memecah kata yang berupa nama, kata yang memiliki affiks, dan kata apapun yang belum tercatat di kamus.

2.1.2. *Web Service* API

Open Chemistry : RESTful *Web* APIs, *JSON*, NW*Chem, and Modern Web Application* (Hanwell et all, 2017). Penelitian ini mengenai sebuah platform *end-to-end* untuk penelitian ilmu kimia telah dikembangkan dengan mengintegrasikan data dari komputasi dan pendekatan eksperimental berbasis *web* modern. Platform ini menawarkan visualisasi interaktif dan lingkungan analitik yang berfungsi dengan baik di perangkat seluler, laptop, dan desktop disertai solusi pragmatis untuk memastikan bahwa kumpulan data yang besar dan kompleks lebih mudah diakses dengan dikembangkan secara terbuka. Yaitu dengan cara semua kode sumber dihosting di platform GitHub agar dapat dengan mudah diakses den dikembangkan orang lain. Platform ini juga dapat disesuaikan dengan situs atau penelitian yang dilakukan. Data tentang ilmu kimia yang berisi struktur dasar yang mendukung perhitungan kimia komputasi disimpan dan ditampilkan menggunakan *json*. Struktur ini dikembangkan untuk memudahkan pemrosesan data pada berbagai platform dan bahasa pemrograman.

Rancang bangun layanan *web* (*web wervice*) untuk aplikasi rekam medis praktik pribadi dokter (Yussrizal dkk, 2017). Pada penelitian tersebut, peneliti bebrencana membuat layanan *web* yang bersifat RESTful bagi rekam medis supaya dapat dipakai oleh berbagai unit layanan kesehatan yang bertujuan membuat pengelolaan dan pengaksesan rekam medis menjadi lebih mudah. Selain itu, pada perancangannya menggunakan RESTful API : Google *Endpoint* menggunakan skema RESTful API untuk digunakan oleh aplikasi Android. Pada Google, *endpoint* data di kirim dalam format *json* dengan tujuan meminimalisir kesalahan pada aplikasi yang dibuat. Penelitian tersebut menghasilkan antara lain :

a. Aplikasi yang sudah dapat digunakan oleh dokter dan pasien
b. Layanan *web* sudah dapat memenuhi kebutuhan pada aplikasi rekam medis praktik pribadi dokter, yaitu sudah dapat mengakses datastore yang tersimpan *Google App Engine* (GAE).
c. Admin menggunakan aplikasi *web* untuk mengelola data klinik serta data pengguna dokter dan perawat yang akan diguakan oleh aplikasi rekam medis praktik pribadi dokter
d. Admin menggunakan aplikasi *web* untuk mengelola data klinik serta data pengguna dokter dan perawat yang akan diguakan oleh aplikasi rekam medis praktik pribadi dokter

2.1.3. Algoritma *Backtracking*

Penelitian tentang *longest backward segmentation* untuk sebuah hubungan kata (Huang, 2013). Makalah ini menyajikan kepada kita tentang metode *longest backward context* untuk segmentasi kata Cina. Metode ini tidak memerlukan label atau informasi semantic. Alur Dari dari metode tersebut

juga tidak perlu mensegmentasi kata secara mekanis, melainkan secara otomatis menemukan kesalahan kata dengan alur *backtracking* secara *real-time*. Dibandingkan dengan algoritma *simple matching*, metode ini meningkatkan kemampuan akurasi dan mengurangi redundansi. Untuk mencapai segmentasi tersebut maka diperlukan empat tahap sebagai berikut:

a. Tahap pra-pemrosesan. Sebelum melakukan segmentasi dengan teks, Anda perlu memproses teks yang merujuk pada *pre-filter* pada sub faktor mengganggu. Tujuan dari preprocessing adalah untuk menyediakan sistem yang hanya mengandung karakter bahasa Cina
b. Tahap struktur kamus. Memuat kamus ke memori, masukkan data dan mengurutkan. Dalam hal ini, metode pencarian biner sesuai dengan kamus, sangat meningkatkan efisiensi algoritma
c. Tahap segmentasi. Jika ada kata bahasa Inggris di dokumen, mengklasifikasikan kata-kata bahasa Inggris ini menjadi sesuai string, begitu juga pada angka, klasifikasikan dengan algoritma *longest backward*.
d. Tahap pasca pemrosesan. Setelah segmentasi, akan ditemukan beberapa kata yang muncul bukan di kamus, ini adalah kata-kata yang tidak dikenal. Pada akhirnya, algoritma menambahkan kata - kata yang tidak dikenal ini secara otomatis ke kamus.

Percobaan di atas menunjukkan ketelitian dan penarikan *longest backward segmentation* lebih tinggi dari itu MMM (*match maximum method*) yakni meningkat hampir 2%. Alasannya, pola *backtracking* terpanjang membuat segmentasi retrospektif (mengingat data di masa lalu), dan metode *match maximum*

tidak. Hasil terakhir *longest backward segmentation* memiliki ketelitian sebesar 89% dan MMM sebesar 87,2%.

Penelitian tentang algoritma *reverse backtracking matching* untuk segmentasi kata Cina (Liu, 2014). Penelitian ini mempelajari algoritma segmentasi kata Cina dan algoritma ambiguitas. Penelititan tersebut ber fokus pada algoritma *reverse backing segmentation*, dan memperkenalkan mekanisme *backtracking*. Jadi dalam pada saat pencocokan kata, tidak hanya berfokus pada pencocokan kata saat itu, tetapi juga mempertimbangkan efek dari pencocokan kata tersebut terhadapa kata selanjutnya sehingga, dapat mengurangi kemungkinan ambiguitas. Eksperimen pada penelitian tersebut menunjukkan bahwa mekanisme *backtracking* yang dimasukkan ke dalam algoritma *maximum segmentation matching*, secara efektif dapat menghilangkan sebagian besar ambiguitas yang mungkin muncul pada algoritma *maximum segmentation matching*. Tetapi, karena penggabungan algoritma tersbut, muncul kesalahan baru yang tidak muncul dalam pencocokan algoritma *maximum segmentation matching.* Dari hasil percobaan keseluruhan, tingkat kesalahan algoritma pencocokan maksimum jauh lebih rendah daripada algoritma *reverse backtracking*. Algoritma *backtracking* secara efektif meningkatkan akurasi segmentasi.

Aplikasi *search engine* dengan metode *depth first search* (Juliasari, 2012). Studi ini membahas pengembangan aplikasi mesin pencari sederhana dengan menggunakan metode *depth first search*. Pencarian pada DFS ini dilakukan dengan menelusuri satu cabang sebuah pohon sampai ke bawah (sampai menemukan solusi) kemudian melakukan *backtracking*. Pencarian dimulai dari akar (level 0) dan pencarian dilanjutkan dengan melacak node yang berada paling kiri.

Penelitian ini menghasilkan *search engine* yang dibangun dari semua indeks halaman *web* yang ada. Basis data indeks halaman *web* juga akan diperbarui oleh bantuan *web-crawler*. *Web-crawler* digunakan untuk tujuan membuat salinan dari semua halaman yang telah dikunjungi untuk diproses. *Web-crawler* adalah jenis bot atau agen perangkat lunak. Secara umum, *web-crawler* mwmulai pekerjaannya dengan daftar URL yang akan dikunjungi atau biasa disebut *seeds*. *Web-crawler* mengidentifikasi semua hyperlink di halaman *web* dan menambahkannya ke daftar URL yang akan dikunjungi sehingga aplikasi ini mudah digunakan untuk mengumpulkan url, kata, atau kemunculan sebuah kata.

Algoritma *backtracking* untuk penyelesaian puzzle gambar bendera (Hasani, 2016). Sistem ini dibangun dengan metode *backtracking* karena dinilai efektif dalam menyelesaikan masalah *puzzle* gambar bendera dan efisien dalam masalah pengkodean. *Puzzle* sendiri adalah permainan untuk menyatukan pecahan keping untuk membentuk sebuah pola yang ditentukan. Sistem yang dibangun pada penelitian ini bertujuan untuk memberi pelajaran kepada pemain dengan cara mengenalkan bendera suatu negara dan mengetahui letak wilayah negara pada sebuah benua dan memicu minat pemain untuk belajar tentang bendera sebuah negara. Kelemahan dari sistem ini yaitu belum terdistribusi jadi penyimpanan data masih local dan belum dapat terhubung dengan user lain untuk menyelesaikan *puzzle* bendera tersebut.

Implementasi algoritma *backtracking* dalam sistem informasi perpustakaan untuk pencarian judul buku (Rifqo, 2017). Penelitian ini menganalisis tentang implementasi algoritma *backtracking* untuk pencarian judul buku pada Perpustakaan UPT Universitas Muhammadiyah Yogyakarta Bengkulu. Metode

backtracking yang digunakan pada penelitian ini ternyata mampu menemukan semua solusi pada pencarian data judul buku. Kelebihan dari algorima *backtracking* yaitu mudah merunut balik suatu langkah sehingga apabila menemui langkah terakhir maka dapat kembali pada posisi sebelumnya yang mempunyai langkah solusi sehingga dapat menemukan solusi lebih cepat. Dengan adanya aplikasi tersebut, peneliti berharap pencarian judul buku akan lebih cepat, tepat dan akurat.

Perbedaan penelitian ini dengan penelitian-penelitian terkait yang sudah dilakukan sebelumnya tentang transliterasi aksara Jawa diantaranya yaitu:

a. Penelitian tentang program transliterasi antara aksara latin dan aksara jawa dengan metode FSA (Atina dkk., 2016) diketahui bahwa inputan dari aksara jawa ke latin sudah memiliki spasi. Padahal, pada dokumen dokumen aksara jawa klasik, setiap kata pada kalimat tidak akan dipisahkan oleh spasi maupun titik melainkan menggunakan pangkon. Sehingga apabila diterapkan pada program ini, program akan mengalami masalah saat membedakan kata per kata karena tidak adanya spasi pada aksara jawa.
b. Penelitian mengenai sistem penerjemah bahasa Jawa - aksara Jawa berbasis *Finite State Automata* (Yohanes, 2017) membuktikan bahwa aplikasi memiliki kekurangan yakni belum dapat memisahkan setiap kata bila pada aksara Jawa aslinya tertulis tanpa spasi. Seperti contoh kata “NabiNuh”. Tidak ada spasi setelah diterjemahkan dengan sistem tersebut.
c. Dan penelitian terakhir yang saya jadikan referensi yakni tentang Dictionary-based Word Segmentation for Javanese (Dipta dan Mirna, 2016) hampir sama dengan penelitian ini yaitu menggunakan kamus sebagai acuan. Hanya saja sumber kata yang dijadikan acuan berasal dari artikel korpus dan metode yang

dipakai bukanlah metode *backtracking*. Kemudian dari segi fungsional, metode yang dipakai pada penelitian ini masih terdapat beberapa kelemahan yakni belum dapat memecah kata yang berupa nama dan kata yang memiliki affiks.

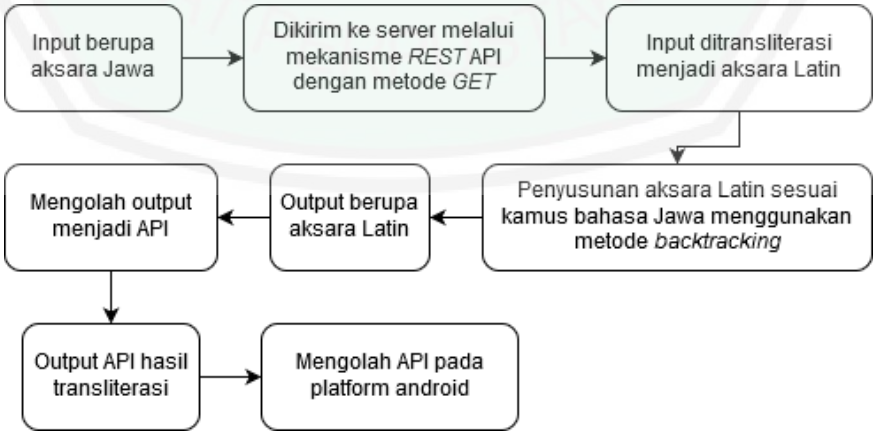
BAB III
METODOLOGI PENELITIAN

3.1 Desain Sistem

Pada subbab ini akan dijelaskan elemen-elemen yang terkait dalam pembuatan *web service* API transliterasi aksara jawa ke aksara latin dengan *backtracking* sebagai metode penyusunnya.

3.1.1 Deskripsi Umum Sistem

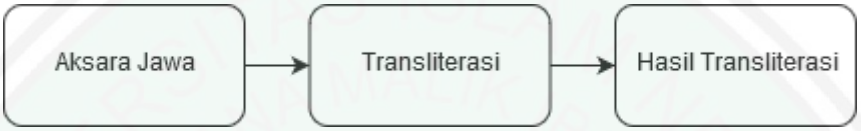
Web service API transliterasi aksara jawa ke aksara latin merupakan sistem yang akan menterjemahkan kalimat aksara Jawa ke dalam tulisan latin yang datanya disajikan dalam bentuk API atau *Application Programming Interface*. Sebelum melakukan penyusunan kata dengan menggunakan metode *backtrackcing*, dilakukan proses penerjemahan kata dari input untuk kemudian dicocokkan dengan database sehingga didapatkan hasil terjemahan kata latin berbahasa jawa yang masih belum terpisah sesuai kaidah. Proses selanjutnya yaitu menampilkan output hasil transliterasi ke dalam bentuk API menggunakan *web service* yang nantinya API tersebut dapat dimanfaatkan oleh platform Web maupun Android.



Gambar 3. 1 Gambaran umum sistem

3.2 Tahap Penerjemah

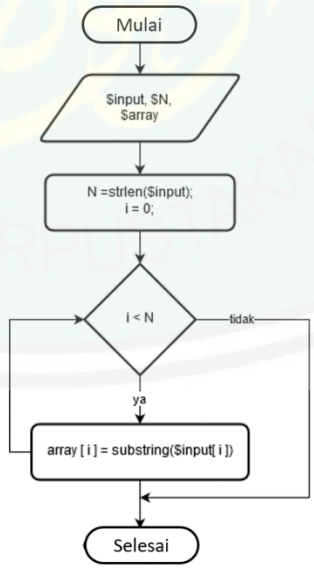
Tahapan ini membahas tentang tahap terjemah kata/kalimat beraksara Jawa kedalam aksara latin yaitu tahap dimana kata/kalimat inputan pengguna akan dialih aksara dangan kata latin dari kamus. Hasil dari tahap terjemah nantinya akan diproses ke tahap selanjutnya. Adapun tahapan dari terjemah adalah seperti digambarkan pada **Gambar 3.2** berikut.



Gambar 3. 2 Tahapan transliterasi

3.2.1 Tokenizing / Parsing data

Pada tahapan ini dilakukan parsing data sederhana yaitu memcah sebuah teks menjadi kumpulan kata-kata tanpa memperhatikan keterkaitan dan peran atau kedudukan antar kata. Prosesnya digambarkan pada flowchart berikut.



Gambar 3. 3 algoritma parsing data

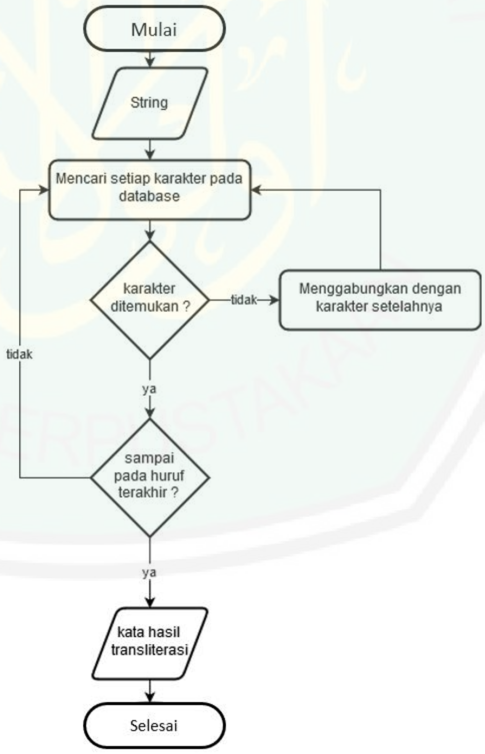
Kemudian dari proses parsing data tadi akan menghasilkan karakter-karakter yang sepenuhnya terpisah seperti pada contoh **Tabel 3.1** berikut ini.

Tabel 3. 1 Parsing data

No	**Input**	**Hasil**
1	[illegible]	[illegible]
2	[illegible]	[illegible]

3.2.2 Transliterasi atau alih aksara

Pada sub bab transliterasi atau alih aksara jawa ini disusun kedalam beberapa tahap dimana kalimat input aksara jawa yang terdiri dari aksara pasangan, aksara sandangan, aksara angka, aksara murda, aksara swara serta aksara pokok (aksra nglegena) akan diterjemahkan ke dalam aksara latin.



Gambar 3. 4 Flowchart transliterasi

Prosesnya adalah, setiap karakter akan akan dicek satu per satu sampai membentuk suatu suku kata aksara jawa. Kemudian sudah menjadi satu suku kata

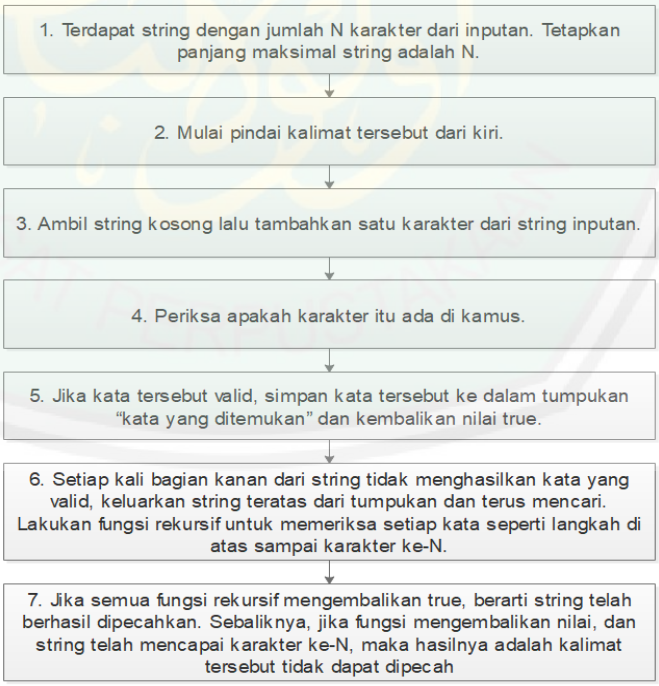
maka satu suku kata tersebut ditransliterasi menjadi aksara latin. Berikut adalah contoh hasil transliterasi dari aksara Jawa ke aksara latin.

Tabel 3. 2 Contoh hasil transliterasi

No	**Input**	**Hasil**
1	ꦄꦢꦶꦏꦸꦭꦒꦶꦠꦸꦫꦸ	Adhikulagituru
2	ꦗꦏꦩꦕꦏꦺꦴꦫꦤ꧀	Jakamacakoran

3.3 Penyusunan Kata dengan Metode *Backtracking*

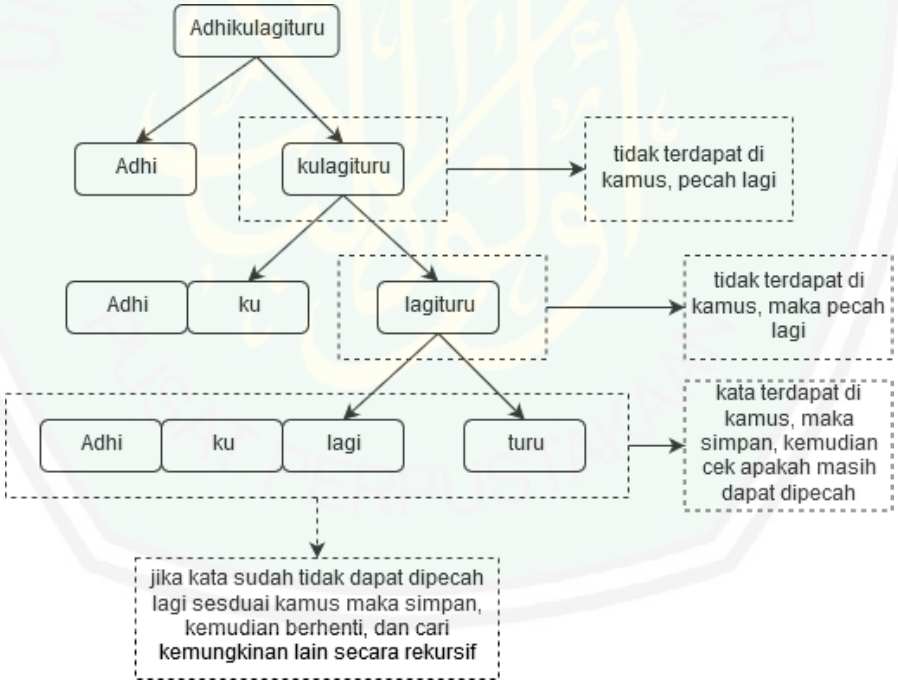
Proses selanjutnya yaitu penyusunan kata menggunakan algoritma *backtracking*. Penyusunan kata dilakukan karena beberapa bahasa ditulis tanpa spasi ataupun garis pemisah seperti pada aksara Cina, Jepang, Thailand, maupun aksara Jawa. Ide desain dari algoritma tersebut adalah seperti berikut:



Gambar 3. 5 Algoritma penyusunan kata

Penyusunan kata juga menggunakan kamus sebagai pembanding kata apa saja yang dimaksudkan dari hasil penerjemahan kata di atas. Hanya saja, uang perlu

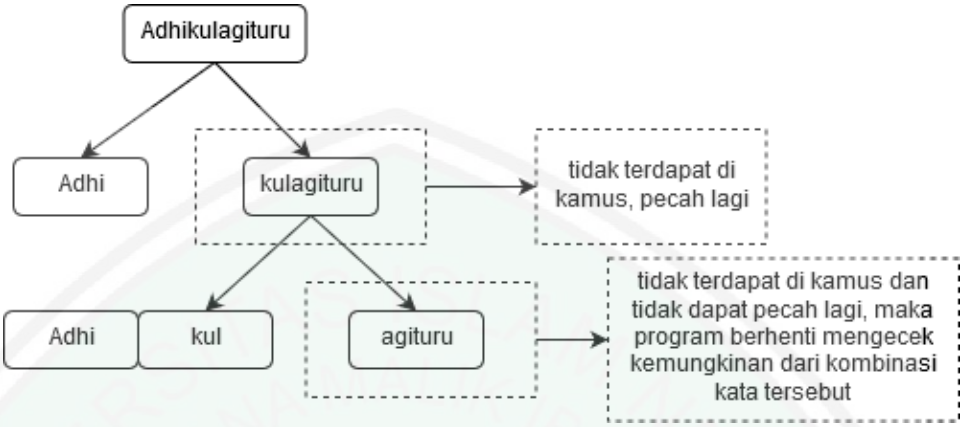
diperhatikan adalah saat menemukan kata yang valid, perlu dieriksa lagi apakah sisa kalimat dapat membuat kata yang valid atau tidak, karena dalam beberapa situasi kata yang ditemukan pertama dari sisi kiri dapat meninggalkan bagian yang tersisa yang tidak dapat dipisahkan lebih lanjut. Jadi, dalam hal ini, kita harus kembali dan meninggalkan kata yang ditemukan saat ini dan terus mencari kata berikutnya. Dan proses ini bersifat rekursif karena untuk mengetahui apakah bagian yang tepat dapat dipisahkan atau tidak. Kemudian untuk melacak kata-kata yang sudah ditemukan, kita akan menggunakan tumpukan. Di saat bagian kanan string tidak membuat kata-kata yang valid, maka keluarkan string teratas dari tumpukan dan terus mencari. Gambaran proses dari alurnya akan seperti **Gambar 3. 6**.



Gambar 3. 6 Proses penyusunan kata dengan *backtracking*.

Kata yang di sebelah kiri adalah tumpukan kata yang disimpan di memori karena kata tersebut telah ditemukan di kamus. Setelah sampai karakter terakhir, secara rekursif program akan menelusuri kemungkinan kemungkinan kata lain yang

terdapat di kamus. Kemungkinan lain dari sebuah string yang disusun kembali tersebut adalah seperti berikut.



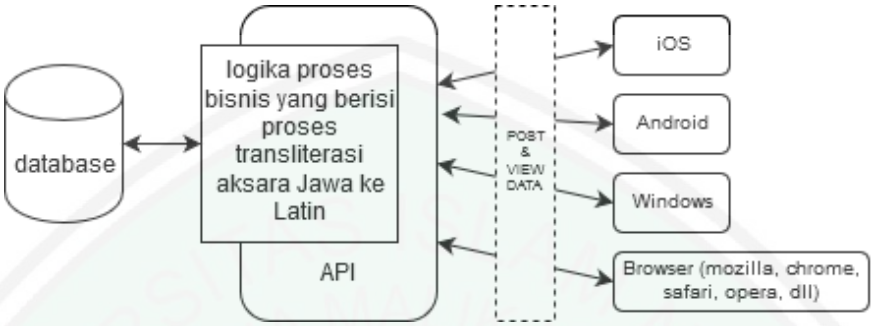
Gambar 3. 7 kemungkinan kata lain yang tidak sesuai

Namun hasil tersebut tidak dipakai dikarenakan kata setelah "kul" tidak dapat dikenali kamus. Apabila sudah tidak tidak terdapat lagi kemungkinan lain, maka proses penelusuran dan penyusann kata akan berhenti dan kalimat yang akan disimpan adalah kalimat yang sepenuhnya terdapat dikamus. Sehingga didapatkan kemungkinan terbaik sesuai dengan kamus kata Basusastra Jawa yaitu kalimat : *Adhi ku lagi turu*.

3.4 Pembuatan API dari *Web Service*

Arsitektur *web service* umumnya terdiri dari database, logika proses bisnis, output yang berupa API, dan platform yang mengakses (*web browser* atau *mobile application*). Algoritma *backtracking* yang telah dibuat adalah seperti *business logic* atau proses logika yang berlangsung pada sistem. Logika bisnis ini erat kaitannya dengan database karena data kamus tersimpan di database dan saat mentransliterasi diperlukan pencocokan kata dari kamus di database. Alurnya yaitu String inputan yang berupa aksara Jawa akan dikirimkan dengan metode *post* untuk kemudian diproses dan dilakukan transliterasi. Output hasil transliterasi itulah yang

kemudian ditampilkan dalam bentuk API dan kemudian diakses oleh aplikasi *client*. Proses untuk mengirim dan menerim data dari API tersebut yang dinamakan mekanisme REST API. Gambaran umum arsitekturnya seperti **Gambar 3. 8**.



Gambar 3. 8 Arsitektur *web wervice* dengan API

Kemudian format API yang akan dibuat nantinya adalah format *json*. Jenis format ini dipilih karena saat ini format tersebut paling banyak dipakai. Bentuk *json* yang akan dibuat nantinya akan seperti berikut.

Method* : *GET
Parameter yang dikirim:

```
{
  "aksara_jawa": "ꦲꦢꦶꦏꦸꦭꦒꦶꦠꦸꦫꦸ"
}
```

Respon saat sukses:

```
{
    "result": "success",
    "status_code": 200,
    "item": [
            "adhi ku lagi turu"
    ]
}
```

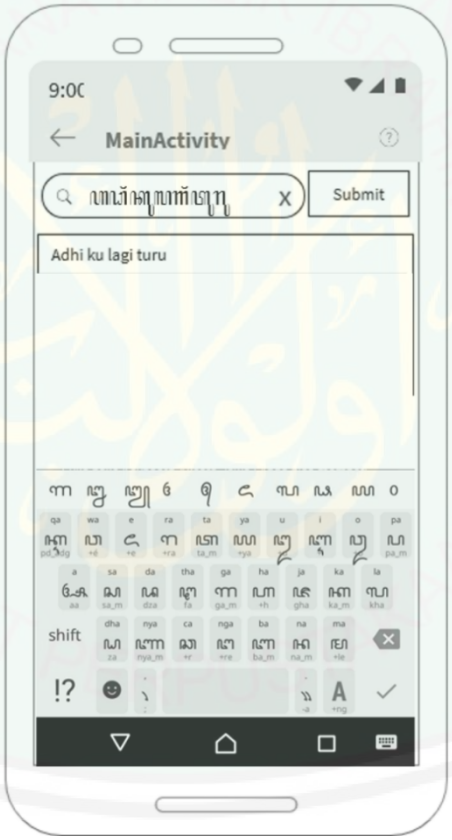
Respon saat gagal:

```
{
    "result": "failure",
    "statusCode": 401,
    "message": "Kata tidak dapat diterjemahkan"
    "request" : "adhikulagituru"
}
```

Format di atas terdapat status code untuk mengetahui status request dan data yang merupakan isi dari API tersebut sesuai dengan hasil transliterasi dengan metode *backtracking*.

3.5 Desain Antarmuka Aplikasi *Client* untuk Pengujian Sistem

Aplikasi yang nantinya akan dipakai untuk pengujian *web service* API untuk transliterasi aksara Jawa ke aksara latin ini memiliki antarmuka seperti pada **Gambar 3. 9**.



Gambar 3. 9 Desain antarmuka aplikasi *client* dengan platform Android.

Berikut penjelasan dari **Gambar 3. 9** :

a. *SearchView*

SearchView ini berfungsi sebagai tempat untuk memasukkan teks aksara Jawa yang akan ditransliterasikan ke dalam huruf latin.

b. Tombol Alih Aksara (Submit)

Tombol Alih Aksara ini berfungsi untuk mentransliterasikan teks dengan aksara Jawa ke huruf Latin yang mana menggunakan metode *backtracking* sebagai penyusun kata.

c. Tombol Reset (X)

Tombol Reset ini berfungsi untuk mengosongkan SearchView seperti keadaan semula.

d. Daftar Output

Daftar Output ini adalah tempat ditampilkannya deretan keluaran atau hasil dari transliterasi berupa teks kalimat dengan aksara Latin.

e. Keyboard Aksara Jawa

Merupakan aplikasi tambahan yang diinstal pada perangkat Android untuk dapat menulis aksara Jawa di perangkat tersebut.

3.6 Pengujian Terhadap Sistem

Perhitungan performansi sistem pada penelitian ini yaitu dengan menguji tingkat akurasi pengenalan karakter atau *Character Accuracy* (Cacc) (Utami, Nurhayati, & Martono, 2016). *Character Accuracy* nantinya akan digunakan untuk mengukur teks hasil pengenalan terhadap hasil yang diharapkan. Pengujian dilakukan untuk mengetahui kualitas sistem berdasarkan tingkat akurasi dalam kemampuannya menangani transliterasi aksara Jawa ke huruf latin.

Untuk menghitung seberapa besar tingkat akurasi aplikasi ini menerjemahkan suatu kalimat dilakukan dengan cara subyektif, dimana kalimat yang dianggap benar dari total kalimat yang diujikan akan menjadi parameter dalam penghitungan tersebut. Nilai akurasi menggambarkan seberapa akurat sistem dapat

menghasilkan output secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang benar dibandingkan dengan keseluruhan data. Nilai akurasi dirumuskan dengan persamaan berikut :

$$Keakuratan\ (\%) = \frac{Jumlah\ Kalimat\ yang\ dianggap\ benar}{Jumlah\ kalimat\ total\ yang\ diujikan}\ x\ 100\%$$

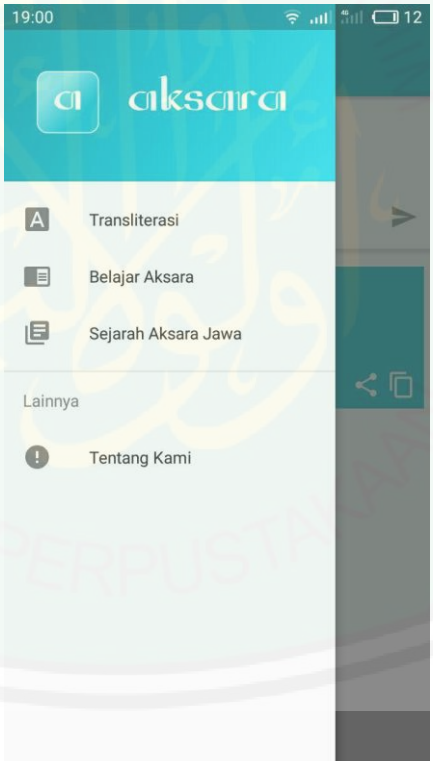
Maksud dari kalimat dan kata yang dianggap benar disini adalah sebagai berikut :

a. Kalimat uji secara benar menurut peneliti dari segi struktur kalimat dan batasan masalah yang ada.
b. Kalimat yang memiliki struktur yang benar dalam penulisan hurufnya sesuai dengan kamus Basusastra Jawa, Purwadarminta.
c. Kata yang tidak beridiom dan berambigu.

BAB IV
PEMBAHASAN

4.1 Implementasi Antarmuka

Di dalam implementasi antar muka, dijelaskan tentang menu-menu yang ada pada antar muka aplikasi *client* dengan platform Android untuk transliterasi aksara Jawa ke aksara latin. Menu yang ada aplikasi ini terdiri dari dua bagian yakni menu utama dan menu pendukung. Menu utama berisi menu transliterasi sedangkan menu pendukung berisi halaman tentang sejarah aksara jawa, pemebelajaran jenis-jeni aksara Jawa dan halaman tentang.



Gambar 4. 1 Antarmuka menu pada aplikasi

Berikut adalah penjelasan untuk masing-masing halaman.

4.1.1 Menu Utama

Menu utama berisi halaman translitersi dan halaman rekomendasi kata. Tampilan dari halaman transliterasi bisa dilihat pada **Gambar 4.1**.

Gambar 4. 2 Tampilan saat tida ada ada proses

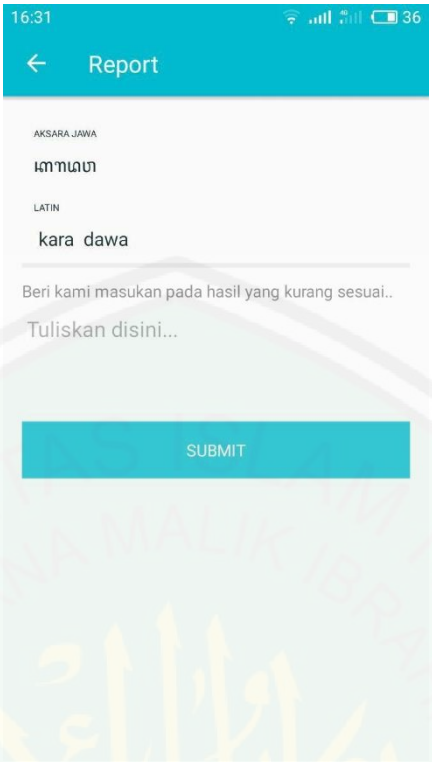
Pada halaman ini terdapat 3 komponen utama yaitu *field* untuk teks yang akan ditrasnliterasi, *textView* hasil transliterasi yang belum tersusun secara baik, dan *textView* hasil transliterasi yang sudah tersusun dengan metode *backtracking*. Tombol *submit* berbentuk panah berfungsi untuk memproses inputan aksara Jawa dari user yang ditulis di *field* pencarian baik melalui *keyboard* aksara Jawa yang sudah terinstall di perangkat pengguna maupun dari sumber lain. Sebelum pengguna mengisi *field* dan menekan tombol *submit* maka tampilan dari *textView* aplikasi berupa teks kosong.

Pada saat user sudah memasukkan teks aksara Jawa yang ingin ditransliterasi dan menekan tombol *submit* maka hasil transliterssi akan ditampilkan pada *textView* hasil transliterasi seperti pada **Gambar 4.2.**

Gambar 4. 3 Tampilan hasil transliterasi

Teks tersebut merupakan hasil akhir yang sudah melalui semua proses dari transliterasi sampai pada API dan ditampilkan pada perangkat Android. Pengguna dapat melakukan beberapa aktivitas pada hasil transliterasi tersebut. Aktivitas tersebut di antaranya membagikan hasil transliterasi pada orang lain melalui aplikasi social yang sudah terinstal pada *handphone* pengguna (whatsapp, facebook, line, dll), menyalin teks hasil transliterasi, dan memberikan rekomendasi kata.

Untuk fitur rekomendasi kata, pengguna nantinya akan diarahkan pada halaman *report* untuk memberikan rekomendasi berupa kata atau kalimat yang lebih sesuai dengan hasil transliterasi yang nantinya akan disimpan di database untuk melengkapi kata-kata yang belum terdapat pada kamus Basusastra Jawa. Antarmuka halaman rekomendasi kalimat oleh pengguna nampak seperti berikut.



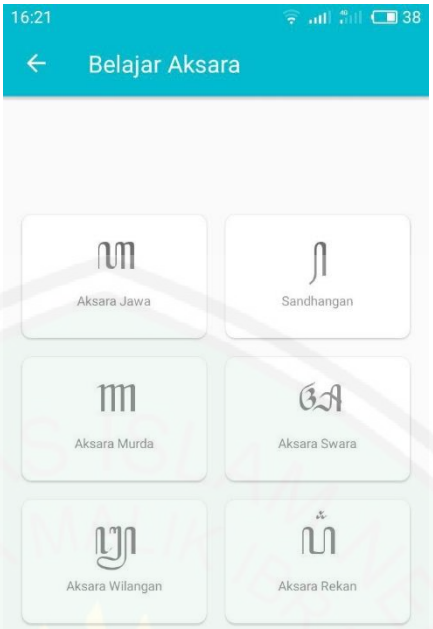
Gambar 4. 4 Antarmuka rekomendasi kata oleh pengguna

4.1.2 Menu Pendukung

Halaman yang termasuk di dalam menu pendukung pada aplikasi ini seperti yang telah dijelaskan sebelumnya yaitu :

a. Halaman Belajar Aksara

Halaman ini berisi penjelasan-penjelasan mengenai jenis-jenis aksara Jawa yang terdiri dari aksara Jawa, aksara Murda, aksara Rekan, sandhangan, aksara wilangan, dan aksara disertai contoh-contohnya.



Gambar 4. 5 Antarmuka halaman belajar aksara

Gambar 4. 6 Antarmuka detail halaman belajar aksara

Contoh-contoh pada masing-masing aksara seperti pada **Gambar 4. 5** tersebut akan ditampilkan pada saat pengguna menekan salah satu aksara di

Gambar 4. **6**. Contoh tersebut akan tampil sesuai dengan jenis aksara mana yang dipilih oleh pengguna.

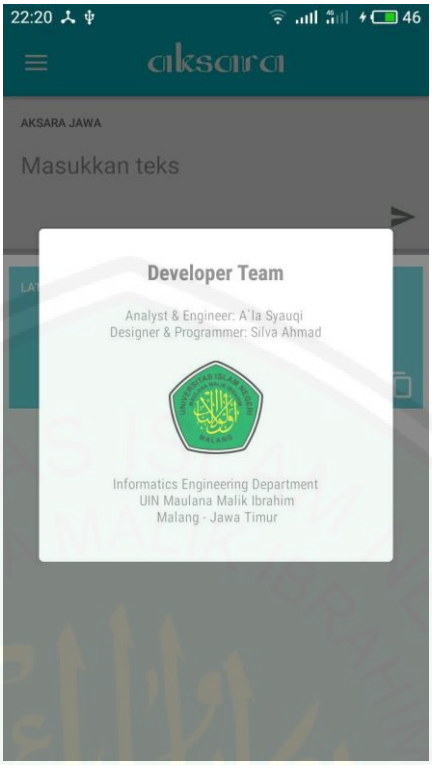
b. Halaman Sejarah

Halaman ini berisi tentang penjelasan dari apa yang dimaksud aksara itu sendiri dan bagaimana sejarahnya kemunculan aksara Jawa di Indonesia hingga perubahan-perubahan apa saja yang terjadi pada penulisan aksara Jawa pada masa ini. Tampilannya seperti pada **Gambar 4. 7** berikut.

Gambar 4. 7 Halaman sejarah aksara Jawa

c. Halaman Tentang

Halaman ini berisi informasi tentang pengembang aplikasi transliterasi aksara Jawa tersebut. Antarmuka halaman tentang dapat dilihat pada **Gambar 4. 8** berikut ini.



Gambar 4. 8 Antarmuka tentang developer aplikasi

4.2 Implementasi Sistem

4.2.1 REST API

4.2.1.1 Mengirim data dari Android ke *server* melalui API

Langkah pertama yang dilakukan sistem saat pengguna menekan tombol *submit* adalah mengirim inputan yang berisi string aksara Jawa dari *editText* ke *server* untuk kemudian diproses menggunakan protokol *Representational State Transfer* (REST) seperti yang ditunjukkan oleh **Gambar 4. 9** dan **Gambar 4. 10**.

```
String kata = editText.getText().toString();
if (!TextUtils.isEmpty(kata)) {
    transliterasi(kata);
}
```

Gambar 4. 9 Menerima dan mengirim data dari editText

```
public static void transliterasi(String message,final
 DataResponse.DataResponseCallback callback) {
    String requestUrl = BASE_URL + message.trim();
    AndroidNetworking.get(requestUrl.trim())
            .setPriority(Priority.MEDIUM)
            .build()
            .getAsJSONObject(new
                 JSONObjectRequestListener() {

                @Override
                public void onResponse(JSONObject response) {
                   ...
                   ...
                }

                @Override
                public void onError(ANError anError) {
                   ...
                }
            });
}
```

Gambar 4. 10 Mengirim data ke *server* dengan metode *GET*

Source code pada **Gambar 4. 9** berfungsi untuk mengambil teks dari inputan pengguna untuk kemudian dikirimkan ke fungsi *transliterasi()* pada **Gambar 4. 10** yang berguna sebagai jembatan untuk mengirim data dari perangkat Android ke *server* dengan parameter *message* dari teks tersebut. Kemudian dengan menggunakan *library* dari *Fast Android Networking* teks aksara Jawa tadi kemudian dikirim ke *server* dengan metode *get* dan *url* yang telah ditentukan sebelumnya untuk kemudian diproses di *server*. Proses mengirim *server* bisa saja mengalami kegagalan. Maka dari itu nilai kembalian saat mengirim data tadi dikembalikan dengan fungsi *onResponse* dan fungsi *onError*. Fungsi *onResponse* terbaca apabila proses mengirim data sukses. Sedangkan apabila terjadi kesalahan atau eror maka *onError* lah yang akan terbaca.

Fungsi *onResponse* memiliki nilai kembalian berupa *JSONObject* yang berarti hasil proses pengiriman ke *server* tadi akan dikembalikan dalam bentuk API dalam format *json* untuk kemuudian diolah lagi pada platform Android sebagai hasil transliterasi aksara Jawa dari inputan pengguna. Sedangkan fungsi *onError* memiliki parameter *ANError* yang berfungsi melaporkan detail eror apabila terjadi kegagalan saat mengirim ke data *server*. Error tersebut nantinya akan terlihat pada *log* aktivitas aplikasi saat berjalan, sehingga dengan mudah ditemukan solusi dari permasalahan yang mengakibatkan eror tersebut.

4.2.1.2 Memproses data yang dikirim ke *server*

Terdapat 2 fungsi untuk menerima data yang dikirim ke *server*, yaitu *get* dan *post*. Untuk menangkap data yang dikirim dengan metode *get* maka pada bahasa pemrograman *php* kita harus menggunakan fungsi `$_GET`. Sedangkan apabila data tersebut dikirmkan dengan metode *post* maka fungsi untuk menangkap datanya dengan fungsi `$_POST`.

Penelitian ini menerapakan fungsi *get* untuk menerima data. Sehingga di saat data yang berupa teks aksara Jawa berhasil dikirimkan ke *server*, maka bahasa *server* (*php*) yang kemudian akan menerima data tersebut dengan fungsi `$_GET` dan menyimpannya ke dalam sebuah variabel. Kode sumbernya terlihat pada **Gambar 4. 11**.

```
$aksara = $_GET["aksara"];
```

Gambar 4. 11 Menerima data dan disimpan dalam variabel.

Data pada variabel $aksara yang berisi kalimat aksara Jawa yang dikirimkan dengan parameter aksara juga. Nantinya data tersebut akan digunakan sebagai variabel utama pada penelitian ini.

4.2.2 Proses Penerjemahan

4.2.2.1 Tahap Parsing

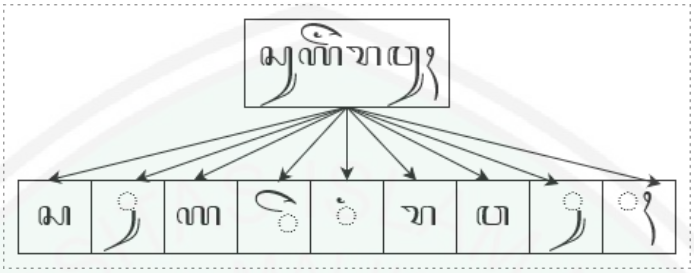
Aksara jawa yang bersifat silabik atau kesuku-kataan membuat setiap karakter dari aksara input tadi harus dipisah sesuai kaidah suku kata dalam bahasa jawa. Namun untuk memisahkan agar sesuai dengan suku kata tersebut langkah awal yang dilakukan adalah adalah memparsing data secara data sederhana. Yaitu memcah sebuah teks menjadi kumpulan kata tanpa memperhatikan keterkaitan antar kata dan peran dalam kalimat. Karakter yang diterima adalah seluruh karakter pada aksara Jawa, baik itu aksara angka, huruf, murda, nglegana, rekaan, pengganti, pasangan, dan aksara sandhangan. Untuk karakter yang berupa titik dan karakter pembuka kalimat akan diabaikan. Berikut cuplikan *source code*-nya.

```
$panjangtext = mb_strlen($aksara);
$karakter = array();
for ($x = 0; $x < $panjangtext; $x++) {
    $karakter[$x] = mb_substr($aksara,$x,1,"utf-8");
}
```

Gambar 4. 12 Kode program untuk parsing

Fungsi *mb_strlen* pada **Gambar 4. 12** adalah untuk menghitung jumlah keseluruhan karakter yang ada pada variabel teks sebelumnya. Kemudian setelah didapatkan total karakternya dilakukan perulangan untuk memisahkan karakter pada teks tersebut satu per satu dengan menggunkan fungsi *substring* dan parameter indeks karakter yang akan dipisah dengan keterangan *Unicode* “utf-8”. Hasilnya

nanti akan ditampung di dalam variabel *array* dengan isi *array* sesuai dengan masing-masing karakter yang ditampung sebelumnya. Contoh hasil pemisahan karakter adalah seperti pada **Gambar 4. 13** berikut ini.



Gambar 4. 13 Parsing data menjadi beberapa karakter.

Array yang sebelumnya telah dideklarasikan akan terisi dengan semua karakter hasil pemisahan untuk nantinya digabungkan kembali sehingga terbagi menjadi beberapa suku kata yang memiliki karakter sesuai dengan penulisan aksara Jawa.

4.2.2.2 Tahap Transliterasi

Pada tahap ini, setiap karakter terpisah yang sudah ditampung di dalam *array*, masing-masing itu akan dicocokkan dengan aturan silabik pada aksara Jawa. Misalnya aksara “ꦛ” yang artinya “ta”. Karena karakter tersebut merupakan satu suku kata, maka kata itu langsung dilakukan transliterasi kemudian hasilnya akan disimpan di dalam variabel yang bersifat sementara untuk nantinya dilakukan pengecekan kembali.

Berbeda apabila karakter di atas termasuk aksara sandhangan atau aksara pasangan. Misalnya karakter “ ꦸ ” yang digunakan sebagai penanda huruf vokal “u”, karakter pepet “ ” sebagai penanda huruf “ e ”, atau karakter taling tarung yang

mana memiliki makna berbeda saat letaknya berberangan dan tidak berbarengan. Maka dari itu pada tahap ini dilakukan pengecekan apakah satu tersebut sudah mewakiliki satu suku kata. Apabila iya, maka simpan, cek lagi karakter berikutnya. Bila itu merupakan sandhangan atau karakter penghubung lainnya, maka gabungkan dengan kata yang telah disimpan sebelumnya. Apabila cocok, maka simpan kata tersebut pada variabel. Begitu seterusnya diulangi satu per satu sampai pada karakter yang paling akhir. Berikut cuplikan kode programnya.

```
for ($i=0; $i < $panjangtext; $i++) {
//--------------kemungkian pertama------------
    if(array_key_exists($i+2, $karakter)){
        $temp = $karakter[$i].$karakter[$i+1].$karakter[$i+2];
        $query = "select * from semua where jawa='".$temp."' order by
latin asc ";
        $result = mysqli_query($con, $query);
        $search = "";
        while($row = $result->fetch_array()){
            $search = $row['latin'];
        }
        if (strlen($search)>0){
            array_push($hasil,$search);
            $i = $i + 2;
            continue;
        }
    }
//--------------kemungkian kedua------------
    if(array_key_exists($i+1, $karakter)) {
        $temp = $karakter[$i].$karakter[$i+1];
        $query = "select * from semua where jawa='".$temp."' order by
latin asc";
        $result = mysqli_query($con, $query);
        $search = "";
        while($row = $result->fetch_array()){
            $search = $row['latin'];
        }
        if (strlen($search)>0){
            array_push($hasil,$search);
            $i = $i + 1;
```

```
                continue;
            }
        }
//--------------kemungkian ketiga------------
    if(array_key_exists($i, $karakter)) {
        $temp = $karakter[$i];
        $query = "select * from semua where jawa='".$temp."' order by
latin asc";
        $result = mysqli_query($con, $query);
        $search = "";
        while($row = $result->fetch_array()) {
            $search = $row['latin'];
        }
        if (strlen($search)>0){
            array_push($hasil,$search);
        }
    }
}
```

Gambar 4. 14 Cuplikan *source code* untuk transliterasi

Pada kode program terdapat keterangan kemungkinan pertama, kemungkian kedua, dan kemungkinan ketiga. Kode program pada kemungkinan pertama akan dijalankan apabila total karakter dari teks aksara yang diproses lebih dari 3. Kode program ini berguna untuk mengecek apakah suatu karakter itu memiliki 3 karakter pada suku kata seperti “ ”, “”, dan “”. Bila digabungkan menjadi “ ” dibaca “ko”. Aksara ini merupakan satu suku kata, maka program akan menjalankan kemungkinan pertama untuk dilakukan aliha aksara. Demikian halnya dengan kemungkinan kedua. Kemungkinan kedua akan dilakukan apabila teks aksara dalam satu suku kata memiliki 2 karakter seperti pada contoh kalimat di atas yaitu “” dan “”. Bila digabungkan menjadi “” dibaca “ke”. Apabila aksara yang terdapat setelah setelah aksara ini sudah tidak dapat digabungkan, maka program akan menjalankan kemungkinan kedua pun begitu juga kondisi

kemungkinan kode program yang pertama akan dilakukan apabila hanya memiliki 1 karakter saja. Proses ini digambarkan seperti pada **Gambar 4. 15** dengan contoh aksara "ꦏꦺꦴꦫ꧀ꦮꦤ꧀" yang dibaca "koran".

Gambar 4. 15 Proses transliterasi

Pada perulangan pertama terlihat bahwa karakter pertama tidak memiliki makna bila tidak digabungkan dengan karakter lainnya. Kemudian apabila digabunngkan dengan dengan karakter kedua berarti "ke" dan bila digabungkan lagi dengan karakter ketiga bermakna "ko". Tetapi setelah sampai pada karakter keempat, sudah bukan lagi karakter penghubung dalam satu suku kata, sehingga tidak sesuai bila disambungkan. Maka simpan kata kata "ko" dan perulangan dimulai lagi dari karakter keempat. Begitu seterusnya sampai berhenti pada karakter paling akhir.

Sayangnya, kalimat hasil transliterasi tidak seluruhnya langsung dapat dikenali oleh kamus dikarenakan faktor ambiguitas pada aksara " ꦲ " yang dapat diartikan "a" atupun "ha" tergantung maksud dari sebuah kata tersebut seperti kata "iki" penulisannya sama dengan kata "hiki". Kata "opo" penulisannya sama dengan

kata “hopo” dan sebagainya. Maka di sini penulis berinisiatif melakukan percobaan untuk mengetahui suku kata mana di antara suku kata “ha” dan “a”, “hi” dan “i”, “he” dan “e”, “h~” dan “~”, dan “ho” dan “o” yang lebih banyak dipakai. Dan didapati hasil sesuai pada **Lampiran II** bahwa huruf vocal tanpa awalan “h” lebih banyak digunakan meskipun pada kata-kata tertentu dan kasus-kasus lain penggantian penghapusan karakter “h” pada huruf vocal malah menimbulkan kesalahan pada hasil transliterasi seperti pada kata “tahu” menjadi “ tau ”, kata “luhur” menjadi “luur”, dan suku kata lain yang ditulis menggunakan aksara “ꦲ”. Meskipun begitu, dengan hasil pada **Lampiran II** maka dilakukanlah penggantian pada karakter-karakter berikut supaya hasil transliterasi lebih optimal :

a. karakter “ha“ menjadi “a”.
b. karakter “hi“ menjadi “i”.
c. karakter “hu“ diganti dengan “u”.
d. karakter “he“ diganti dengan “e”.
e. karakter “h~“ diganti dengan “~”.
f. karakter “ho“ diganti dengan “o”.

Cuplikan *source-code* untuk me*reploce* suku kata tersebut adalah seperti pada **Gambar 4. 16** berikut ini.

```
$hasil_awal = str_replace("ha", "a",$hasil_awal);
$hasil_awal = str_replace("hi", "i",$hasil_awal);
$hasil_awal = str_replace("hu", "u",$hasil_awal);
$hasil_awal = str_replace("he", "e",$hasil_awal);
$hasil_awal = str_replace("h~", "~",$hasil_awal);
$hasil_awal = str_replace("ho", "o",$hasil_awal);
```

Gambar 4. 16 *Source code* untuk normalisasi data

Penggantian karakter tersebut dilakukan karena beberapa kata perlu disesuikan dengan kata pada kamus sehingga algoritma untuk penyusunan kata dapat bekerja maksimal. Selain itu, penulisan karakter "~" adalah untuk membedakan karakter "é" dan "e" dikarenakan karakter unicode tidak dapat ditampilkan pada API.

4.2.3 Penyusunan Kata dengan Metode *Backtracking*

Tahap ini adalah proses dimana kata yang sudah ditransliterasi masih berupa deretan karakter yang menjadi satu belum terpisah sesuai kata pada kamus bahasa Jawa. Karakter aksara Jawa yang memiliki keunikan yaitu setiap kalimatnya selalu ditulis bersambung tanpa ada spasi membuat hasil transliterasi sebelumnya harus dipisahkan dengan cara tersendiri. Contohnya seperti kalimat "sugengrawuh" yang harusnya menjadi "sugeng rawuh". Atapun misalnya kalimat "jakamacakoran" menjadi "jaka maca koran" dan lain sebagainya. Berikut adalah cuplikan kode bagaimana metode *backtracking* mampu memisahkan kata sesuai dengan kamus.

```
function wordBreak($input) {
    processInputString($input, strlen($input), "");
}
```

Gambar 4. 17 Fungsi *backtracking* untuk menyusun kata

Untuk memisahkan kata digunakan fungsi *wordBreak* yang di terdapat pada **Gambar 4. 17**. Fungsi ini bekerja secara rekursif karena menganggil fungsi *proccessInputString* pada **Gambar 4. 18** secara terus menerus mencari semua kemungkinan yang ada hingga berhenti pada kemungkinan yang paling akhir. Fungsi ini sangat efektif digunakan karena kita tidak harus tau berapa kemungkinan

yang akan dihasilkan. Maka dengan menggunakan *backtracking* ini semua kemungkinan terbaik akan dicoba.

```
function processInputString($input, $size, $result) {
    for ($i = 1; $i <= $size; $i++) {
        if (existInDictionary(substr($input, 0, $i))) {
            if ($i == $size) {
                array_push($GLOBALS['hasil'], $result." ".$input);
                break;
            } else {
             processInputString(substr($input, $i, $size),
              ($size - $i), $result." ".substr($input, 0, $i)." ");
            }
        }
    }
}
```

Gambar 4. 18 Fungsi rekursif pada *backtracking*

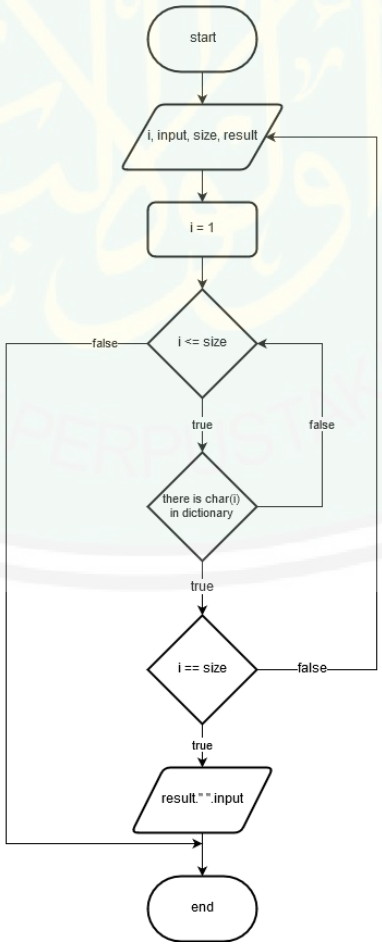
Kemudian pada **Gambar 4. 19** terdapat fungsi *existInDictionary*. yang berperan untuk mengecek apakah data atau teks tersebut terdapat pada kamus. Apabila teks tersebut ada, maka simpan data ke dalam tumpukan *array*. Sebaliknya, bila tidak menemukan kecocokan teks tersebut dengan kamus dari Basusastra Jawa, maka lanjutkan fungsi rekursif untuk mencari kemungkinan kombinasi kata yang lain. Cuplikan kode untuk mengecek kata tersebut ada di kamus atau tidak terdapat pada **Gambar 4. 19** berikut ini

```
function existInDictionary($str) {
    $query = "select teks from jawa where teks = '".$str. "'";
    $dictionary = mysqli_query($con, $query);
    $key = "";
    while ($row = $dictionary -> fetch_array()) {
        $key = $row['teks'];
    }
    if (strlen($key) > 0) {
        return true;
    } else {
```

```
            return false;
        }
    }
```

Gambar 4. 19 Fungsi untuk mengecek kata yang ada di kamus

Metode *backtracking* akan menampilkan karakter yang memiliki suku kata terpanjang di akhir, sehingga hasil terakhirlah yang akan diambil sebagai output. Namun apabila suatu teks saat dipisahkan hanya memiliki satu kemungkinan kombinasi jawaban dari penyusunan karakter, maka otomatis hasil itu yang akan ditampilkan. Seperti pada kalimat "macakoran". Kalimat ini hanya memiliki 1 kemungkinan hasil yaitu "maca koran" dan tidak ada kemungkinan lain. Maka proses penyusunan karakter dengan metode *backtracking* berhenti sampai di sini.



Gambar 4. 20 Flowchart dari fungsi *backtracking* untuk penyusunan kata.

4.2.4 Pembuatan API dari *Web Service*

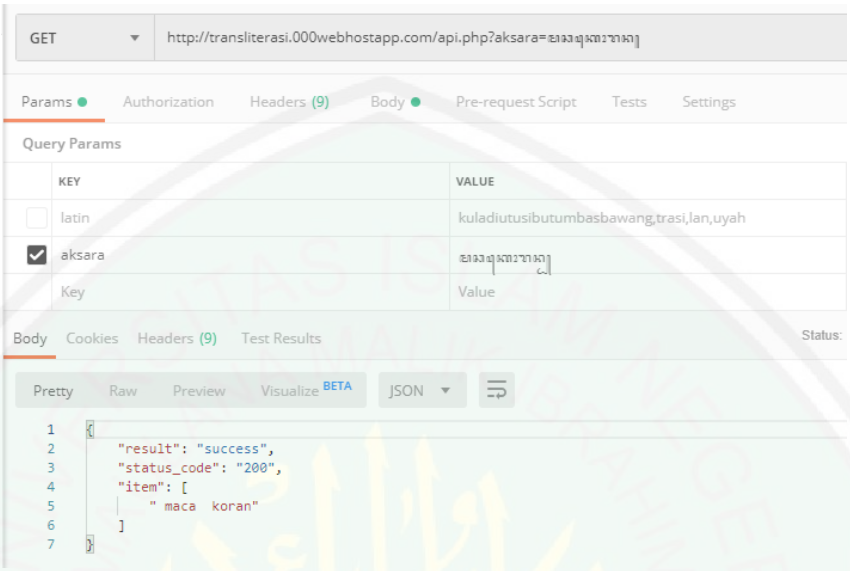
Output dari hasil penyusunan dengan *backtracking* masih disimpan di dalam *array* sehingga platform lain belum dapat mengakses output tersebut. Maka diperlukan beberapa baris kode lagi agar output hasil transliterasi tersebut ditampilkan dalam bentuk API sehingga dapat diakses oleh aplikasi dengan platform yang berbeda-beda. Cuplikan *source code* untuk menampilkan API adalah seperti berikut.

```
if (sizeof($hasil) > 0) {
    $json = $arrayName = array(
            'result' = > 'success',
            'status_code' =>'200',
            'item' =>$hasil
            );
} else {
    $json = $arrayName = array(
            'result' = > 'failure',
            'status_code' =>'401'
             );
}
echo json_encode ($json);
```

Gambar 4. 21 Membuat format *json*

Format *json* dapat kita bentuk sesuai kemauan kita. Meskipun begitu format standar selalu memiliki *status_code* untuk mengetahui status request kita sukses atau tidak. Pada umumnya, kode status yang bernilai di bawah 400 adalah sukses, di atas itu maka dimasukkan kategori gagal. Pada **Gambar 4. 21** tersebut terdapat pengecekan data *array* hasil penyusunan kata dengan *backtracking*. Fungsinya, apabila array tersebut bernilai kosong, maka format *json* akan disertai kode status 200, sedangkan apabila gagal, kode statusnya berisi 401. Di saat sukses, *array* hasil juga ditampilkan bersamaan dengan kode status dan dalam format *json array*.

Hasil *request* untuk melihat format *json* yang telah dibuat dapat dilakukan dengan postman. Hasil seperti berikut.



Gambar 4. 22 Hasil REST API pada aplikasi *postman*

Hasil yang ditampilkan di dalam *postman* akan sama persis dengan apa yang telah dibuat di *source code php* sebelumnya. Kemudian langkah selanjutnya adalah mengolah API tersebut pada platform aplikasi Android.

4.2.5 Mengolah API pada Platform Android

4.2.5.1 Menyiapkan *model*

Pada respon *json* yang telah kita buat sebelumnya, terdapat beberapa entitas atau variabel yang terkandung di dalam respon tersebut. Agar semua entitas tersebut lebih mudah kita olah, maka diperlukan sebuah *model*.

Model merujuk pada data atau objek pada aplikasi yang memiliki entitas-entitas di dalamnya. Pada paradigma pemrograman berorentiasi objek, *model* dapat memberi fleksibilitas yang lebih. Dengan *model*, kita akan lebih mudah dalam mengubah program karena *model* dapat kita bentuk sesuai kebutuhan kita. Selain itu, *model* dapat digunakan dalam skala yang luas dalam membangun perangkat

lunak. Bila kita misalkan keseluruhan respon data dari *json* tadi adalah objek/kelas dan isinya adalah entitas, maka kita akan dapat membuat *model* dari respon tersebut. Contoh kode program untuk membuat *model* adalah seperti berikut.

```
public class DataResponse implements Serializable
{
    @SerializedName("result")
    @Expose
    public String result;
    @SerializedName("request")
    @Expose
    public String request;
    @SerializedName("status_code")
    @Expose
    public int statusCode;
    @SerializedName("item")
    @Expose
    public List<String> item = null;
    public String getResult() {
        return result;
    }
    public void setResult(String result) {
        this.result = result;
    }
    public int getStatusCode() {
        return statusCode;
    }
    public void setStatusCode(int statusCode) {
        this.statusCode = statusCode;
    }
    public List<String> getItem() {
        return item;
    }
    public void setItem(List<String> item) {
        this.item = item;
    }
    public String getRequest() {
        return request;
    }
    public void setRequest(String request) {
        this.request = request;
    }
```

```
    ...
    ...
}
```

Gambar 4. 23 *Model* untuk respon data

Kode program di atas adalah *model* yang digunakan untuk menangani objek berupa respon dari *json*. *Model* biasanya memiliki fungsi *setter* dan *getter*. Kedua fungsi ini berguna sebagai perantar untuk mengolah variabel pada kelas objek tanpa harus memanggil langsung variabel di kelas tersebut. Respon yang diterima nantinya dapat dirubah datanya melalui *method setter* dan *getter* yang telah dibuat.

4.2.5.2 Mengolah dan menampilkan data dengan *model*

Setelah sebelumnya membuat *model* di Android, langkah terakhir pada adalah menampilkan data tersebut. Cuplikan kode program untuk mengolah data json adalah seperti berikut.

```
public class DataResponse implements Serializable
{
    ...
    ...

    public interface DataResponseCallback{
        void onSuccess(DataResponse dataResponse);
        void onFailure(String message);
    }
}
```

Gambar 4. 24 *Interface* untuk *callback* data

Model yang sudah dibuat ini telah mengimplementasikan *Serializable* dari *library gson* yang telah ditambahkan. Artinya *model* ini telah dapat menserialisasi data menjadi *jsonObject* sehingga tidak perlu lagi memilah data dari entitas *json* yang sebelumnya telah dibuat.

Selain itu, terdapat kelas *interface* yang berfungsi sebagai *callback* saat proses request sukses atau gagal. Kelas ini nantinya digunakan sebagai parameter saat request transliterasi ke *server*. Penggunaan kelas *interface* ini terdapat pada **Tabel 4. 13** .

```
APIRequest.transliterasi(message,
   new DataResponse.DataResponseCallback() {
   @Override
   public void onSuccess(DataResponse dataResponse) {
     if (dataResponse.statusCode < 400) {
        String hasil =
             dataResponse.item.get(dataResponse.item.size()-1);
         if (dataResponse.item.size() > 0) {
              textView3.setText(hasil);
          } else {
             textView3.setText(dataResponse.request);
          }
      }
    }
    @Override
    public void onFailure(String message) {
       System.out.println(message);
    }
});
```

Gambar 4. 25 Mengolah API dengan *model*

Kelas *interface* yang telah dibuat ini di dalamnya terdapat *method onSuccess* dan *onFailure*. *Method onSuccess* ini dijalankan saat request ke *server* sukses. *Method* ini akan mengambalikan data berupa respon hasil transliterasi. Bila respon hasil transliterasi tidak kosong, maka akan ditampilkan pada *textView* di Android. Sebaliknya, bila request gagal, maka *method onFailure* yang akan dipanggil bersama nilai kembalian berupa laporan eror yang akan terlihat pada *log* program seperti yang telah dijelaskan pada pembahasan sebelumnya.

4.3 Pengujian Sistem

Untuk mengetahui sejauh mana tingkat keakurasian aplikasi dalam melakukan penterjemahan kalimat bahasa Indonesia ke bahasa jawa dan pengalih aksara dari huruf latin ke dalam aksara Jawa maka perlu dilakukan pengujian. Pengujian dilakukan dengan memberikan 50 kalimat beraksara Jawa yang diambil dari penelitian sebelumnya yang berjudul "*Aplikasi Penterjemah Kalimat Bahasa Indonesia ke Bahasa Jawa disertai Transliterasi Aksara Jawa*" (Santosa, 2016). Yang mana data yang digunakan untuk pengujian diambil dari beberapa sumber sehingga memiliki pola yang berbeda beda.

Dari pengujian yang telah dilakukan pada **Lampiran I, Lampiran II**, dan **Lampiran III**, dapat diketahui bahwa dari 50 aksara Jawa yang sudah selesai diuji, diperoleh persentase sebagai berikut:

a. Transliterasi aksara Jawa ke aksara Latin.

$$\text{Persentase akurasinya} = \frac{48}{50} \times 100$$

$$= 96\%$$

b. Penyusunan kata dengan *backtracking* dengan mengabaikan aksara "ꦩ" pada kata yang memiliki awalan (a, i, u ,e, dan o).

$$\text{Persentase akurasinya} = \frac{43}{50} \times 100$$

$$= 86\%$$

c. Penyusunan kata dengan *backtracking* dengan menggunakan aksara "ꦩ" pada kata yang memiliki awalan (a, i, u ,e, dan o).

$$\text{Persentase akurasinya} = \frac{29}{50} \times 100$$

$$= 58\%$$

4.4 Evaluasi dan Analisa Sistem

Di luar keterbatasan tersebut, tingkat akurasi dari hasil pengujian aplikasi transliterasi aksara jawa ke aksara Latin dengan metode *backtracking* sebagai metode penyusun katanya mencapai akurasi 86% dari 50 kalimat yang diujikan. Terdapat 6 kalimat yang tidak sesuai dengan yang diharapkan, dan 4 kalimat yang tidak dapat disusun hal ini disebabkan karena adanya beberapa huruf vokal yang tidak bisa diproses dengan baik dikarenakan ambiguitas seperti contoh suku kata "a" dan "ha", "i" dan "hi", "e" dan "he", dan "o" dan "ho". Hasil pengujian membuktikan bahwa penggunaan huruf vokal tanpa disertai karakter "h" seperti a, i, u, e, dan o pada seluruh hasil transliterasi menghasilkan akurasi yang lebih tinggi yaitu 86%, dibanding dengan kalimat yang huruf vokalnya diawali karakter "h" seperti ha, hi, hu, he, dan ho yang hanya memiliki akurasi sebesar 58% dikarenakan metode *backtracking* tidak dapat menemukan kata yang sebenarnya.

Selanjutnya kendala dalam penanganan kata yang ambigu seperti kalimat "kreminemati" harusnya menjadi "kremi ne mati" tetapi dari pengujian didapatkan hasil "kremi nem ati". Sedangkan proses mentransliterasikan kata maupun kalimat mempunyai tingkat akurasi kebenaran sebesar 96% denga kendala kesulitan membedakan suku kata (a, i, u, e, dan o) dikarenakan ambiguitas hasil transliterasi aksara "ꦎꦩ" seperti yang telah dijelaskan sebelumnya.

Namun, pengujian ulang metode *backtracking* untuk penyusunan kata dengan inputan hasil transliterasi yang telah dilakukan perbaikan seperti kata "dhaar" menjadi "dhahar", "tau" menjadi "tahu" pada **Lampiran III** ternyata menghasilkan akurasi yang lebih tinggi dibandingkan sebelumnya, yaitu 90% meningkat 4% dari sebelumnya. Hal ini dikarenakan input dari kata ambigu yang

dicocokkan dengan sudah sesuai penulisan karakternya dengan yang ada di kamus. Misalnya kata “dhahar”, hasil transliterasi adalah “dhaar” padahal harusnya “dhahar” sehingga program tidak menemukan kata tersebut di kamus. Akibatnya algoritma untuk penyusunan kata dengan metode *backtracking* ini tidak akan bekerja dengan baik.

4.5 Integrasi Penelitian Tentang Aksara Jawa dengan Islam

Masyarakat menyadari bahwa keragaman adalah hal terus ada dan tidak dapat dihindari. Tetapi diskusi tentang bagaimana merespons keberagaman ini masih sering disalahartikan. Sebagian orang berpendapat bahwa perbedaan seharusnya dihilangkan dan perlu diadakan upaya untuk keseragaman. Ada juga yang setuju bahwa perbedaannya adalah hal yang perlu dipertahankan. Islam memberikan ikhtisar mengenai pandangan Islam tentang wawasan multikultural dalam menyikapi keragaman. Hal tersebut dijelaskan di Surah Al-Hujurat ayat 13 dan ayat-ayat yang membahas konsep keragaman.

Surat al-Hujuraat ayat 13 berbunyi:

يَا أَيُّهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاكُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا ۚ إِنَّ أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتْقَاكُمْ ۚ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ

Artinya :“Hai manusia, sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa-bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. Sesungguhnya orang yang paling mulia di antara kamu di sisi Allah ialah orang yang paling takwa di antara kamu. Sesungguhnya Allah Maha Mengetahui lagi Maha Mengenal”. (QS. Al-Hujuraat/49:13).

1. Penjelasan menurut Tafsir Jalalain :

(*Hai manusia, sesungguhnya Kami menciptakan kalian dari seorang laki-laki dan seorang perempuan*) yakni dari Adam dan Hawa (*dan Kami menjadikan*

kalian berbangsa-bangsa) lafal *Syu'uuban* adalah bentuk jamak dari lafal *Sya'bun*, yang artinya tingkatan nasab keturunan yang paling tinggi (*dan bersuku-suku*) kedudukan suku berada di bawah bangsa, setelah suku atau kabilah disebut *Imarah*, lalu *Bathn*, sesudah *Bathn* adalah *Fakhdz* dan yang paling bawah adalah *Fashilah*. Contohnya ialah Khuzaimah adalah nama suatu bangsa, Kinanah adalah nama suatu kabilah atau suku, Quraisy adalah nama suatu *Imarah,* Qushay adalah nama suatu *Bathn*, *Hasyim* adalah nama suatu *Fakhdz*, dan Abbas adalah nama suatu *Fashilah.* (*supaya kalian saling kenal-mengenal*) lafal *Ta'aarafuu* asalnya adalah *Tata'aarafuu*, kemudian salah satu dari kedua huruf Ta dibuang sehingga jadilah *Ta'aarafuu*; maksudnya supaya sebagian dari kalian saling mengenal sebagian yang lain bukan untuk saling membanggakan ketinggian nasab atau keturunan, karena sesungguhnya kebanggaan itu hanya dinilai dari segi ketakwaan. (*Sesungguhnya orang yang paling mulia di antara kalian di sisi Allah ialah orang yang paling bertakwa. Sesungguhnya Allah Maha Mengetahui*) tentang kalian (*lagi Maha Mengenal*) apa yang tersimpan di dalam batin kalian.

2. Penjelasan menurut Tafsir Al-Misbah

(Hai manusia, sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan). Yakni Adam dan Hawa, atau dari sperma (benih laki-laki) dan ovum (indung telur perempuan), (*serta menjadikan kamu berbangsa-bangsa juga bersuku-suku supaya kamu saling kenal-mengenal)* yang mengantar kamu untuk bantu-membantu serta saling melengkapi, (*sesungguhnya yang paling mulia di antara kamu di sisi Allah ialah yang paling bertakwa di antara kamu. Sesungguhnya Allah Maha Mengetahui lagi Maha Mengenal)* sehingga tidak ada

sesuatu pun yang tersembunyi bagi-Nya, walau detak detik jantung dan niat seseorang (Shihab, 2002:616).

Semakin kuat pengenalan satu pihak kepada selainnya, semakin terbuka peluang untuk saling memberi manfaat. Karena itu, ayat diatas menekankan perlunya saling mengenal. Perkenalan itu dibutuhkan untuk saling menarik pelajaran dan pengalaman pihak lain guna meningkatkan ketakwaan kepada Allah SWT. yang dampaknya tercermin pada kedamaian dan kesejahteraan hidup (Shihab, 2002:617).

Demikan halnya dengan pengenalan terhadap alam raya. Semakin banyak pengenalan terhadapnya, semakin banyak pula rahasia-rahasianya yang terungkap, dan ini adalah peran peneliti dalam melahirkan kemajuan ilmu pengetahuan dan teknologi serta menciptakan kesejahteraan lahir dan batin, dunia dan akhirat.

3. Surat Ar-Ruum ayat 22

وَمِنْ آيَاتِهِ خَلْقُ السَّمَاوَاتِ وَالْأَرْضِ وَاخْتِلَافُ أَلْسِنَتِكُمْ وَأَلْوَانِكُمْ ۚ إِنَّ فِي ذَٰلِكَ لَآيَاتٍ لِلْعَالِمِينَ

Artinya:"dan di antara tanda-tanda kekuasaan-Nya ialah menciptakan langit dan bumi dan berlain-lainan bahasamu dan warna kulitmu. Sesungguhnya pada yang demikan itu benar-benar terdapat tanda-tanda bagi orang-orang yang mengetahui".(QS. Ar-Ruum:22).

Dari penjelasan beberapa tafsir dan ayat di atas dapat diketahui bahwa perbedaan senantiasa ada pada setiap manusia dan sudah jelas bahwa keberagaman merupakan hal yang diakui dalam iIlam. Sedangkan yang dilarang adalah di dalam Islam permusuhan dan perpecahan dalam menyikapi keberagaman. Dengan kata lain, Islam sangat menghargai adanya perbedaan budaya, suku, adat-istiadat, bahasa, dan lain sebagainya, Perbedaan itu bukan untuk disombongkan oleh satu sama lain, tetapi justru untuk dijadikan sebagai alat untuk saling mengenal lebih

dekat. Keragaman cara berpikir dan cara bertindak umat manusia dalam kontek ruang dan waktu akan terus eksis.

Sementara itu, peran teknologi adalah sebagai jembatan dalam keberagaman yang telah ada. Ilmu pengetahuan dan teknologi yang telah dikembangakan, khususnya pada penelitian ini adalah menjadi sarana untuk memahami, mengenali, serta melestarikan budaya yang telah diwariskan secara turun temurun oleh nenek moyang kita. Aplikasi transliterasi aksara Jawa ke aksara Latin ini contohnya. Pengembangan aplikasi ini secara tidak langsung telah mengamalkan nilai-nilai dalam surat al-hujurat ayat 13 yakni mengenali keberagaman budaya di Indonesi khususnya aksara Jawa.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian, implementasi, dan serangkain uji coba yang telah dilakukan dapat disimpulkan bahwa akurasi sistem untuk transliterasi dari aksara Jawa ke aksara Latin adalah sebesar 96%. Kemudian penyusunan kata menggunakan *backtracking* dengan mengabaikan aksara "ꦫ" pada kata yang memiliki awalan (a, i, u ,e, dan o) memiliki akurasi sebesar 86% dan akurasi sebesar 58% untuk penyusunan kata menggunakan *backtracking* dengan menggunakan aksara "ꦫ" pada kata yang memiliki awalan (a, i, u ,e, dan o).

Penyebabnya ketidakakuratan hasil sistem tersebut di antaranya adalah ambiguitas karakter aksara Jawa pada suku kata "a" dan "ha", ambiguitas pada penyusunan kata seperti "kremi ne mati" dan "kremi nem ati", dan disebabkan tidak dikenalnya sebuah kata pada kamus Basusastra Jawa.

5.2 Saran

Aplikasi *web service* API hasil transliterasi aksara Jawa ke aksara latin dengan *backtracking* sebagai metode penyusunan kata ini masih memiliki beberapa kekurangan yang harus diperbaiki agar menjadi lebih baik. Oleh sebab itu, beberapa saran yang dapat dilakukan untuk pengembangan berikutnya di antaranya yaitu :

a. Diperlukan penambahan algoritma tertentu agar dapat membedakan ambiguitas penggunaan suku kata "a" dan "ha".

b. Penambahan *machine learning* untuk mengenali sebuah subjek kata yang dimaksud sehingga dapat mengatasi permasalahan ambiguitas kata dalam pemisahaan kalimat seperti “kremi nem ati” dan “kremi ne mati”.
c. Penambahan algoritma untuk mengenali sebuah subjek kata yang berupa kata ganti kepemilikan dan mengenali perbedaan antara preposisi dan bukan preposisi.
d. Mengolah data rekomendasi dari pengguna sebagai tambahan data training *machine learning* yang akan dibuat, supaya hasil transliterasi sistem menjadi lebih akurat.
e. Penambahan kata-kata baru yang mana belum terdapat pada kamus Basusastra Jawa.
f. Saat ini waktu yang diperlukan sistem dalam menyusun kata, berbanding lurus dengan panjang karakter atau kalimat yang akan disusun. Maka kedepan dapat dilakukan optimasi kode program agar penanganan proses pemisahan kata dilakukan lebih cepat tanpa mengurasi akurasi.

DAFTAR PUSTAKA

Al-Mahalli, Jalaluddin dan As-Suyuthi, Jalaluddin. (2018). *Tafsir Jalalain*. Jakarta: Ummul Qura'.

Astuti, Siti Puji. (2016). *Pengaruh Penggunaan Dadu Aksara Jawa Terhadap Keterampilan Menulis Aksara Jawa di Kelas IV SD Negeri Bangunharjo*. Skripsi. Jurusan Pendidikan Sekolah Dasar Fakultas Ilmu Pendidikan Universitas Negeri Yogyakarta.

Atina, V., Palgunadi, S., & Widiarto, W. (2016). Program Transliterasi Antara Aksara Latin dan Aksara Jawa dengan Metode FSA. *Jurnal Teknologi & Informasi ITSmart, 1*(2), 60. https://doi.org/10.20961/its.v1i2.592

Ayumitha, Fevi Henda. (2014). *Transliterasi Huruf Latin ke Aksara Jawa dengan Menggunakan Decisison Tree.* Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Depag, RI. Tt. *Al Qur''an dan Terjemahnya*. PT.Karya Toha Putra, Semarang.

Dhindsa, Baljeet Kaur dan Sharma, Dharam Veer. (2017). *English to Hindi Transliteration System Using Combination-Based Approach.* International Journal of Advanced Research in Computer Science, Volume 8, No.8, September-October 2017, pp. 609, ISSN: 0976-5697.

Hanwell *et all.* (2017). *Open Chemistry: RESTful Web APIs, JSON, NWChem, and The Modern Web Application*. Journal of Cheminformatics. Published by Springer Open. Doi: 10.1186/s13321-017-0241-z

Hasani, A. K. dan Mustafidah, H. (2016). *Algoritma Backtracking Untuk Penyelesaian Puzzle Bendera*. JUITA Vol. 4, No. 2, ISSN: 2086-9398.

Ilmania, Arfinda. (2017). *Penerapan Algoritma Runut-Balik untuk Menyelesaikan Permainan Pencarian Kata*. Makalah IF2211 Strategi Algoritma.

Juliasar, N.,& Sitompul, J. C. (2013). Aplikasi Search Engine denganMetode Depth First Search(DFS).Jurnal Mahasiswa TISI, 9(1).

L. Zhen and L. Yu-sheng. (2010). *Reverse Backtracking Research of Chinese Segmentation Based on Dictionary of Hash Structure*," 2010 Second International Conference on Information Technology and Computer Science, Kiev, 2010, pp. 265-267. doi: 10.1109/ITCS.2010.71.

Santosa, Bakhtiar Puji. (2016). *Aplikasi Penterjemah Kalimat Bahasa Indonesia ke bahasa Jawa Disertai Transliterasi Aksara Jawa Berbasi Web dengan Metode Analisis Averbia dan Decision Tree.* Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Saputro, D. A. (t.t.). *Penerapan RESTful Web Service dan JSON pada Application Programming Interface (API) Sistem Informasi Perkembangan Ayam Broiler Berbasis Kemitraan*. 28.

Shihab,M.Quraish. 2002. *Terjemahan Tafsir Al-Mishbah*. Vol.12. Lentera Hati, Jakarta.

Sulaiman, A. M. (t.t.). *HANACARAKA: AKSARA JAWA YANG MULAI DITINGGALKAN*. 39.

Tanaya, Dipta and Adriani, Mirna. (2016). *Dictionary-based Word Segmentation for Javanese*. 5th Workshop on Spoken Language Technology for Under-resourced Languages, SLTU, 9-12 May 2016. doi:10.1016/j.procs.2016.04.051

Utama, F dkk. (2016). *Implementasi Backtracking Algorithm Untuk Penyelesaian Permainan Sudoku Pola 9x9*. Jurnal Informatika Mulawarkan Vol. 11, No.1, Februari 2016.

Utami, A. E., Nurhayati, O. D., & Martono, K. T. (2016). *Aplikasi Penerjemah Bahasa Inggris – Indonesia dengan Optical Character Recognition Berbasis Android*. Jurnal Teknologi dan Sistem Komputer, *4*(1), 167. https://doi.org/10.14710/jtsiskom.4.1.2016.167-177

Weichun, Huang & Jianjian, Jiang. (2013). *Research on Longest Backward Segmentation for Context*. 10.2991/ccis-13.2013.95.

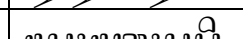
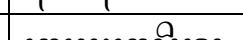
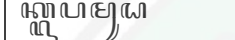
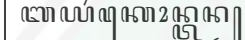
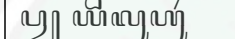
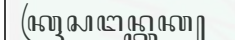
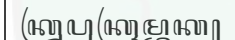
Yohanes, Banu Wirawan dkk. (2017). *Sistem Penerjemah Bahasa Jawa-Aksara Jawa Berbasis Finite State Automata.* JNTETI Vol. 6, No. 2, Mei 2017, ISSN: 2301-4156.

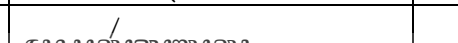
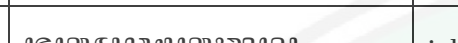
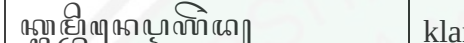
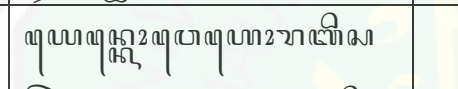
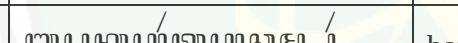
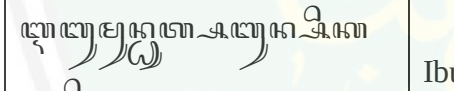
Yusrizal dkk. (2017). *Rancang Bangun Layanan Web (Web Service) Untuk Aplikasi Rekam Medis Praktik Pribadi Dokter*. KITEKRO: Jurnal Online Teknik Elektro, Vol. 2, No. 1, 2017: 1-8, ISSN: 2252-7036.

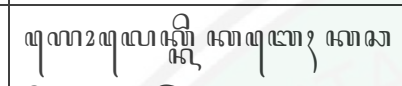
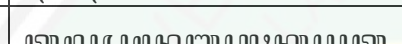
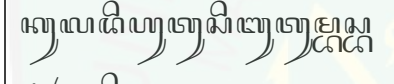
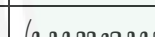
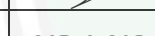
LAMPIRAN I

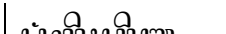
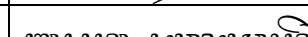
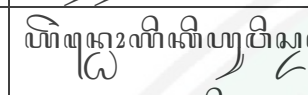
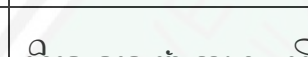
Tabel hasil pengujian sistem menggunakan inputan kalimat beraksara Jawa dan output aksara Latin dengan mengabaikan karakter "h" pada suku kata hasil transliterasi aksara "ꦲ"

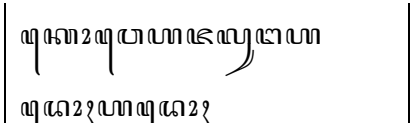
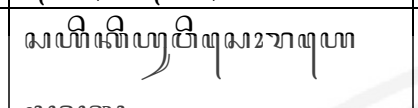
No	Aksara Jawa	Hasil Transliterasi	Keterangan	Penyusunan kata menggunakan *backtracking*	Hasil
1	ꦱꦥꦚꦤ	sapanyana	**sesuai**	sapa nyana	**sesuai**
2	ꦭꦫꦭꦥ	laralapa	**sesuai**	lara lapa	**sesuai**
3	ꦏꦫꦢꦮ	karadawa	**sesuai**	kara dawa	**sesuai**
4	ꦥꦏ꧀ꦒꦸꦫꦸꦁꦔꦼꦤ꧀ꦢꦶꦏ	pakgurungendika	**sesuai**	pak guru ngendika	**sesuai**
5	ꦥꦿꦧꦸꦏ꧀ꦫꦺꦱꦤ	prabukresna	**sesuai**	prabu kresna	**sesuai**
6	ꦱꦸꦒꦼꦔ꧀ꦫꦮꦸꦃ	sugengrawuh	**sesuai**	sugeng rawuh	**sesuai**
7	ꦲꦶꦏꦶꦠꦸꦩ	ikituma	**sesuai**	iki tuma	**sesuai**
8	ꦧꦸꦩꦶꦱꦶꦗꦶ	bumisiji	**sesuai**	bu misi ji	**tidak sesuai**
9	ꦠꦺꦴꦏꦺꦴꦫꦺꦴꦠꦶ	tokoroti	**sesuai**	toko roti	**sesuai**

10		tukutau	**sesuai**	tuku tahu	**sesuai**
11		sotosapi	**sesuai**	soto sapi	**sesuai**
12		uyahasin	**sesuai**	uyah asin	**sesuai**
13		adhikulagituru	**sesuai**	adhi ku lagi turu	**sesuai**
14		kulakwarta	**sesuai**	kulak warta	**sesuai**
15		klapamudha	**sesuai**	klapa mudha	**sesuai**
16		bayarkontan	**sesuai**	bayar kontan	**sesuai**
17		pyayiluur	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
18		krasangantuk	**sesuai**	krasa ngantuk	**sesuai**
19		krupukrambak	**sesuai**	krupuk rambak	**sesuai**
20		kreminémati	**sesuai**	kremi ném ati	**tidak sesuai**
21		lemarinégedhé	**sesuai**	lemari né gedhé	**sesuai**

22		akutukupelemjeruklansalak	**sesuai**	aku tuku pelem jeruk lan salak	**sesuai**
23		masakmartabak	**sesuai**	masak martabak	**sesuai**
24		jakamacakoran	**sesuai**	jaka maca koran	**sesuai**
25		reganélarang	**sesuai**	rega né larang	**sesuai**
26		klambinéreged	**sesuai**	klambi né reged	**sesuai**
27		gulalegi	**sesuai**	gula legi	**sesuai**
28		yénkowéorabisatekangomongosaiki	**sesuai**	yén kowé ora bisa teka ngomongo saiki	**sesuai**
29		bapakdha**a**rtaucampur	**tidak sesuai**	tidak terdeteksi	**tidak sesuai**
30		Ibumundhutsabunsikatlanlinga	**sesuai**	Ibu mundhut sabun sikat lan linga	**sesuai**

31		Akuoramaturmarangsirané	**sesuai**	Aku ora matur marang sira né	**sesuai**
32		pakgurungendikabocahsaikikudusragepsinau	**sesuai**	pak guru ngendika bocah saiki kudu sregep sinau	**sesuai**
33		golékkikabéhkacasingduwétembungwénéhana	**sesuai**	golék ki kabéh kaca sing duwé tembung wénéhana	**sesuai**
34		tuladhanébuahkayata	**sesuai**	tuladha né buah kayata	**sesuai**
35		kuladiutusibutumbasbawangtrasilanuyah	**sesuai**	kula di utus ibu tumbas bawang trasi lan uyah	**sesuai**
36		praulayar	**sesuai**	prau layar	**sesuai**
37		tokosepi	**sesuai**	toko sepi	**sesuai**
38		tindakmenyangpasar	**sesuai**	tindak menyang pasar	**sesuai**

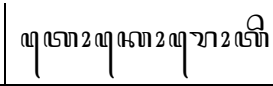
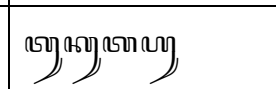
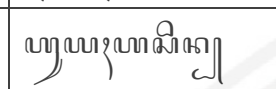
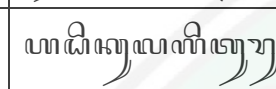
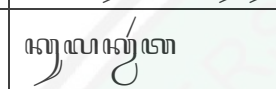
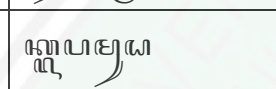
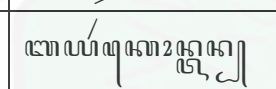
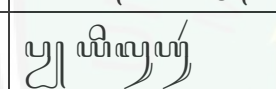
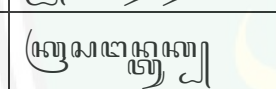
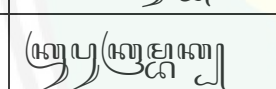
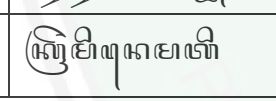
39		panggihibu	**sesuai**	panggih ibu	**sesuai**
40		bapaksaréwontenkamar	**sesuai**	bapak saré wonten kamar	**sesuai**
41		bengiikiibuoraisaturu	**sesuai**	bengi iki ibu ora isa turu	**sesuai**
42		endhogikiuwisdadibocahnékdipangandadilara	**sesuai**	endhog iki uwis dadi bocah nék di pangan dadi lara	**sesuai**
43		pinarakbulenggahwontenngajeng	**sesuai**	pinarak bu lenggah wonten ngajeng	**sesuai**
44		Akutukuklambi	**sesuai**	Aku tuku klambi	**sesuai**
45		bukukuwihisianyar	**sesuai**	bu ku kuwi isih anyar	**tidak sesuai**
46		adhikakutakonsapasingtekaiku	**sesuai**	adhi kaku ta kon sapa sing teka iku	**tidak sesuai**

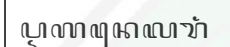
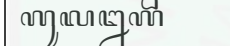
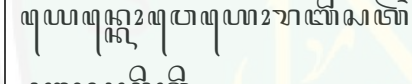
47		kowéajalungaadohadoh	**sesuai**	kowé aja lunga adoh adoh	**sesuai**
48		saikiuwisoraénék	**sesuai**	saiki uwis ora énék	**sesuai**
49		bangsakitauwisduwéundangundangdasar	**sesuai**	bangsa ki ta uwis duwé undang undang dasar	**tidak sesuai**
50		sapasingduwéklambiiki	**sesuai**	sapa sing duwé klambi iki	**sesuai**

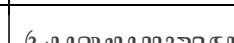
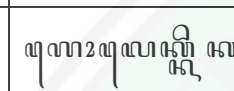
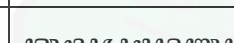
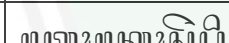
LAMPIRAN II

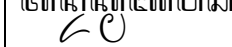
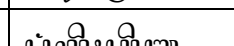
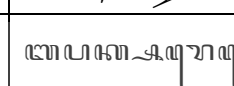
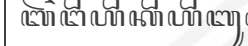
Tabel hasil pengujian sistem menggunakan inputan kalimat beraksara Jawa dan output aksara Latin dengan tidak mengabaikan karakter “h” pada suku kata hasil transliterasi aksara “ꦲ”

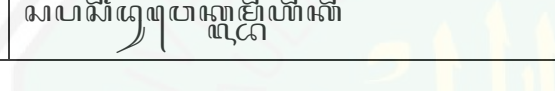
No	Aksara Jawa	Hasil Transliterasi	Keterangan	Penyusunan kata menggunakan *backtracking*	Keterangan
1	ꦱꦥꦚꦤ	sapanyana	**sesuai**	sapa nyana	**sesuai**
2	ꦭꦫꦭꦥ	laralapa	**sesuai**	lara lapa	**sesuai**
3	ꦏꦫꦢꦮ	karadawa	**sesuai**	kara dawa	**sesuai**
4	ꦥꦏ꧀ꦒꦸꦫꦸꦁꦔꦼꦤ꧀ꦢꦶꦏ	pakgurungendika	**sesuai**	pak guru ngendika	**sesuai**
5	ꦥꦿꦧꦸꦏ꧀ꦫꦼꦱꦤ	prabukresna	**sesuai**	prabu kresna	**sesuai**
6	ꦱꦸꦒꦼꦔ꧀ꦫꦮꦸꦃ	sugengrawuh	**sesuai**	sugeng rawuh	**sesuai**
7	ꦲꦶꦏꦶꦠꦸꦩ	hikituma	**tidak sesuai**	**tidak terdeteksi**	**sesuai**
8	ꦨꦸꦩꦶꦱꦶꦗꦶ	bumisiji	**sesuai**	bu misi ji	**tidak sesuai**

9		tokoroti	**sesuai**	toko roti	**sesuai**
10		tukutahu	**sesuai**	tuku tahu	**sesuai**
11		sotosapi	**sesuai**	soto sapi	**sesuai**
12		**hu**yah**ha**sin	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
13		**ha**dhikulagituru	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
14		kulakwarta	**sesuai**	kulak warta	**sesuai**
15		klapamudha	**sesuai**	klapa mudha	**sesuai**
16		bayarkontan	**sesuai**	bayar kontan	**sesuai**
17		pyayiluhur	**sesuai**	pyayi luhur	**sesuai**
18		krasangantuk	**sesuai**	krasa ngantuk	**sesuai**
19		krupukrambak	**sesuai**	krupuk rambak	**sesuai**
20		kreminémati	**sesuai**	kremi ném ati	**tidak sesuai**

21		lemarinégedhé	**sesuai**	lemari né gedhé	**sesuai**
22		akutukupelemjeruklansalak	**sesuai**	aku tuku pelem jeruk lan salak	**sesuai**
23		masakmartabak	**sesuai**	masak martabak	**sesuai**
24		jakamacakoran	**sesuai**	jaka maca koran	**sesuai**
25		reganélarang	**sesuai**	rega né larang	**sesuai**
26		klambinéreged	**sesuai**	klambi né reged	**sesuai**
27		gulalegi	**sesuai**	gula legi	**sesuai**
28		yénkowé**ho**rabisatekangomongosa**hi**ki	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
29		bapakdhahartahucampur	**sesuai**	bapak dhahar tau campur	**sesuai**

30		Ibumundhutsabunsikatlanlinga	**sesuai**	Ibu mundhut sabun sikat lan linga	**sesuai**
31		Aku**ho**ramaturmarangsirané	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
32		pakgurungendikabocahsa**hi**kikudusregepsina**hu**	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
33		golékkikabéhkacasingduwétembungwénéh**ha**na	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
34		tuladhanébu**ha**hkayata	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
35		kuladi**hu**tusibutumbasbawangtrasilanuyah	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
36		pra**hu**layar	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
37		tokosepi	**sesuai**	**tidak terdeteksi**	**tidak sesuai**

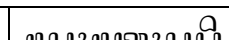
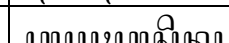
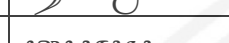
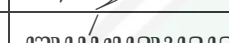
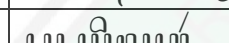
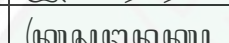
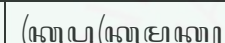
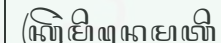
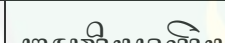
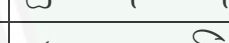
38		tindakmenyangpasar	**sesuai**	tindak menyang pasar	**sesuai**
39		panggih**hi**bu	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
40		bapaksaréwontenkamar	**sesuai**	bapak saré wonten kamar	**sesuai**
41		bengi**hi**ki**hi**buhora**hi**saturu	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
42		**he**ndhogikihuwisdadibocahnékd**i**pang ndadilara	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
43		pinarakbulenggahwontenngajeng	**sesuai**	pinarak bu lenggah wonten ngajeng	**sesuai**
44		**ha**kutukuklambi	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
45		bukukuwi**hi**sihanyar	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
46		adhik**ha**kutakonsapasingteka**hi**ku	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**

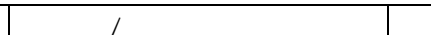
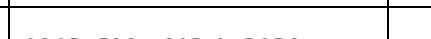
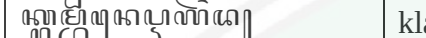
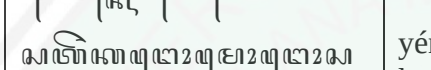
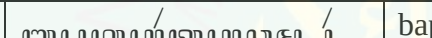
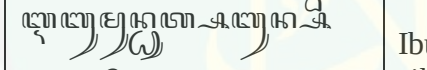
47		kowé**ha**jalunga**ha**do**ha**doh	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
48		sa**hi**ki**hu**wis**ho**ra**hé**nék	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
49		bangsakita**hu**wisduwé**hu**ndang**hu**ndangdasar	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**
50		sapasingduweklambi**hi**ki	**tidak sesuai**	**tidak terdeteksi**	**tidak sesuai**

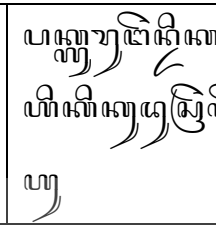
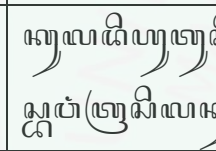
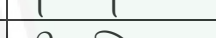
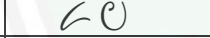
LAMPIRAN III

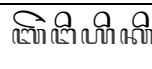
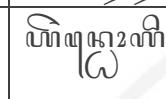
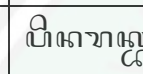
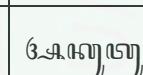
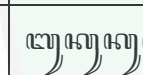
Tabel hasil pengujian sistem dengan melakukan perbaikan pada setiap suku kata hasil transliterasi aksara "ꦩ" yang belum sesuai.

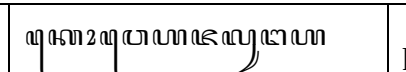
No	Aksara Jawa	Hasil Transliterasi	Keterangan	Perbaikan Kalimat Hasil Transliterasi	Penyusunan kata menggunakan *backtracking*	Keterangan
1	ꦱꦥꦚꦤ	sapanyana	***sesuai***	sapanyana	sapa nyana	**sesuai**
2	ꦭꦫꦭꦥ	laralapa	***sesuai***	laralapa	lara lapa	**sesuai**
3	ꦏꦫꦢꦮ	karadawa	***sesuai***	karadawa	kara dawa	**sesuai**
4	ꦥꦏ꧀ꦒꦸꦫꦸꦁꦔꦼꦤ꧀ꦢꦶꦏ	pakgurungendika	***sesuai***	pakgurungendika	pak guru ngendika	**sesuai**
5	ꦥꦿꦧꦸꦏ꧀ꦫꦼꦱ꧀ꦤ	prabukresna	***sesuai***	prabukresna	prabu kresna	**sesuai**
6	ꦱꦸꦒꦼꦔ꧀ꦫꦮꦃ	sugengrawuh	***sesuai***	sugengrawuh	sugeng rawuh	**sesuai**
7	ꦶꦏꦶꦠꦸꦩ	ikituma	***sesuai***	ikituma	iki tuma	**sesuai**
8	ꦧꦸꦩꦶꦱꦶꦗꦶ	bumisiji	***sesuai***	bumisiji	bu misi ji	**tidak sesuai**
9	ꦠꦺꦏꦺꦴ꦳ꦫꦺꦴ꦳ꦠꦺꦴ꦳ꦫꦶ	tokoroti	***sesuai***	tokoroti	toko roti	**sesuai**
10	ꦠꦸꦏꦸꦠꦮ	tukutau	***sesuai***	tukutau	tuku tahu	**sesuai**

11		sotosapi	***sesuai***	sotosapi	soto sapi	**sesuai**
12		uyahasin	***sesuai***	uyahasin	uyah asin	**sesuai**
13		adhikulagituru	***sesuai***	adhikulagituru	adhi ku lagi turu	**sesuai**
14		kulakwarta	***sesuai***	kulakwarta	kulak warta	**sesuai**
15		klapamudha	***sesuai***	klapamudha	klapa mudha	**sesuai**
16		bayarkontan	***sesuai***	bayarkontan	bayar kontan	**sesuai**
17		pyayiluur	***tidak sesuai***	pyayiluhur	pyayi luhur	**sesuai**
18		krasangantuk	***sesuai***	krasangantuk	krasa ngantuk	**sesuai**
19		krupukrambak	***sesuai***	krupukrambak	krupuk rambak	**sesuai**
20		kreminématí	***sesuai***	kreminématí	kremi ném ati	**tidak sesuai**
21		lemarinégedhé	***sesuai***	lemarinégedhé	lemari né gedhé	**sesuai**
22		akutukupelemjeru klansalak	***sesuai***	akutukupelemjeruk lansalak	aku tuku pelem jeruk lan salak	**sesuai**

23		masakmartabak	***sesuai***	masakmartabak	masak martabak	**sesuai**
24		jakamacakoran	***sesuai***	jakamacakoran	jaka maca koran	**sesuai**
25		reganélarang	***sesuai***	reganélarang	rega né larang	**sesuai**
26		klambinéreged	***sesuai***	klambinéreged	klambi né reged	**sesuai**
27		gulalegi	***sesuai***	gulalegi	gula legi	**sesuai**
28		yénkowéorabisate kangomongosaiki	***sesuai***	yénkowéorabisatek angomongosaiki	yén kowé ora bisa teka ngomongo saiki	**sesuai**
29		bapakdhaartauca mpur	***tidak sesuai***	bapakdhahartauca mpur	bapak dhahar tau campur	**sesuai**
30		Ibumundhutsabun sikatlanlinga	***sesuai***	Ibumundhutsabunsi katlanlinga	Ibu mundhut sabun sikat lan linga	**sesuai**
31		Akuoramaturmara ngsirané	***sesuai***	Akuoramaturmaran gsirané	Aku ora matur marang sira né	**sesuai**

32		pakgurungendikabocahsaikikudusragepsinau	***sesuai***	pakgurungendikabocahsaikikudusragepsinau	pak guru ngendika bocah saiki kudu sregep sinau	**sesuai**
33		golékkikabéhkacasingduwétembungwénéhana	***sesuai***	golékkikabéhkacasingduwétembungwénéhana	golék ki kabéh kaca sing duwé tembung wénéhana	**sesuai**
34		tuladhanébuahkayata	***sesuai***	tuladhanébuahkayata	tuladha né buah kayata	**sesuai**
35		kuladiutusibutumbasbawangtrasilanuyah	***sesuai***	kuladiutusibutumbasbawangtrasilanuyah	kula di utus ibu tumbas bawang trasi lan uyah	**sesuai**
36		praulayar	***sesuai***	praulayar	prau layar	**sesuai**
37		tokosepi	***sesuai***	tokosepi	toko sepi	**sesuai**
38		tindakmenyangpasar	***sesuai***	tindakmenyangpasar	tindak menyang pasar	**sesuai**
39		panggihibu	***sesuai***	panggihibu	panggih ibu	**sesuai**

40		bapaksaréwonten kamar	***sesuai***	bapaksaréwontenkamar	bapak saré wonten kamar	**sesuai**
41		bengiikiibuoraisaturu	***sesuai***	bengiikiibuoraisaturu	bengi iki ibu ora isa turu	**sesuai**
42		endhogikiuwisdadibocahnékdipangandadilara	***sesuai***	endhogikiuwisdadibocahnékdipangandadilara	endhog iki uwis dadi bocah nék di pangan dadi lara	**sesuai**
43		pinarakbulenggahwontenngajeng	***sesuai***	pinarakbulenggahwontenngajeng	pinarak bu lenggah wonten ngajeng	**sesuai**
44		Akutukuklambi	***sesuai***	Akutukuklambi	Aku tuku klambi	**sesuai**
45		bukukuwihisianyar	***sesuai***	bukukuwihisianyar	bu ku kuwi isih anyar	**tidak sesuai**
46		adhikakutakonsapasingtekaiku	***sesuai***	adhikakutakonsapasingtekaiku	adhi kaku ta kon sapa sing teka iku	**tidak sesuai**

47		kowéajalungaadohadoh	***sesuai***	kowéajalungaadohadoh	kowé aja lunga adoh adoh	**sesuai**
48		saikiuwisoraénék	***sesuai***	saikiuwisoraénék	saiki uwis ora énék	**sesuai**
49		bangsakitauwisduwéundangundangdasar	***sesuai***	bangsakitauwisduwéundangundangdasar	bangsa ki ta uwis duwé undang undang dasar	**tidak sesuai**
50		sapasingduwéklambiiki	***sesuai***	sapasingduwéklambiiki	sapa sing duwé klambi iki	**sesuai**