APLIKASI *INPUT METHOD EDITOR* (IME) JEPANG BERBASIS WEB

Alexander Setiawan Rolly Intan Rikko Filiano
Fakultas Teknologi Industri, Jurusan Teknik Informatika, Universitas Kristen Petra
Email: alexander@peter.petra.ac.id, rintan@peter.petra.ac.id, pyokola@gmail.com

ABSTRACT *Japanese language uses thousands of characters for its writing system. A system called Input Method is used to enable the inputs of those characters using a standard keyboard. Some Input Method applications were already developed, from the desktop application to the web-based application. Unfortunately, those applications are not flawless. In this research, an application using the Input Method system has been developed to cover the weaknesses of those applications that already developed before. This Input Method application system uses standard web technology. Such as JavaScript on the client side. PHP on the server side. MeCab: Yet Another Part-of-Speech and Morphological Analyze dictionary is used for its dictionary. The core algorithm uses Longest Match Method for word searching.*

Keywords: *JavaScript, PHP, Input Method, MeCaB, Longest Match Method*

ABSTRAK *Bahasa Jepang memiliki ribuan karakter untuk sistem penulisannya. Dibutuhkan sebuah sistem bernama Input Method untuk memungkin penulisan karakter-karakter tersebut dengan menggunakan papan ketik standar. Beberapa aplikasi Input Method telah dikembangkan sebelumnya, baik aplikasi untuk desktop maupun aplikasi berbasis web. Namun aplikasi-aplikasi tersebut memiliki kelemahan masing-masing. Dari penelitian ini, dikembangkan sebuah aplikasi baru dengan menggunakan sistem Input Method yang menutupi kelemahan-kelemahan pada aplikasi-aplikasi Input Method yang telah dikembangkan sebelumnya. Pembuatan aplikasi tersebut menggunakan teknologi umum untuk web, JavaScript untuk sisi klien dan PHP untuk sisi server, kamus kata bahasa Jepang menggunakan kamus kata dari aplikasi MeCab: Yet Another Part-of-Speech and Morphological Analyze. Algoritma inti aplikasi sistem Input Method ini menggunakan Longest Match Method untuk pencarian kata.*

Kata Kunci: *JavaScript, PHP, Input Method, MeCaB, Longest Match Method*

Dalam langkah awal di era globalisasi dan langkah awal menuju gerbang *Free Trade* dewasa ini, telah banyak mengundang partisipan-partisipan datang dari negara maju ke negara berkembang maupun sebaliknya, dari negara berkembang ke negara maju, untuk melakukan kerjasama dalam bidang ekonomi, politik, kesehatan, pendidikan dan sebagainya.

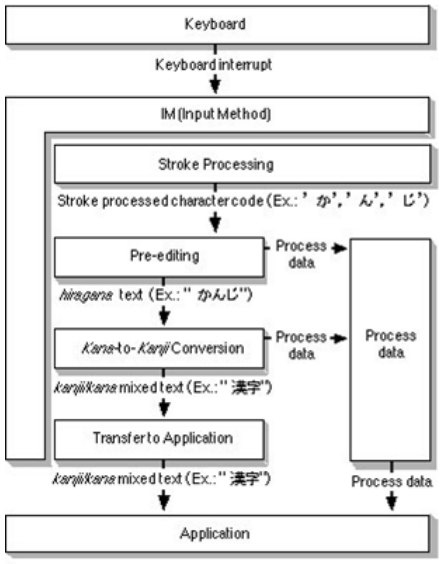
Beberapa kendala ditemukan akibat dari perpindahan ke negara lain yang memiliki bahasa yang berbeda, terutama pada negara-negara yang memiliki tata karakter yang bukan alphabet. Ambil contoh, Jepang. Jepang memiliki 3 (tiga) tata karakter yang berbeda dan bukan dalam bentuk alphabet. Kendala seperti ini menyulitkan bagi partisipan dalam membuat sebuah laporan hasil kerja atau hasil penelitian dengan menggunakan *web application* seperti *Google Docs* atau *Google Spreadsheet*, menulis surat-surat elektronik yang ditujukan ke kolega atau keluarga yang berada di negara asal partisipan tersebut, mencari data referensi dengan bahasa asal partisipan tersebut atau bahkan menulis sebuah *blog* dengan bahasa asal partisipan tersebut, karena negara-negara yang memiliki tata karakter penulisan alphabet hanya menyediakan papan ketik berjenis QWERTY.

Sebuah sistem bernama *Input Method* dikembangkan untuk mengatasi masalah diatas, dan beberapa aplikasi telah mengimplementasikan sistem *Input Method* tersebut seperti IME, SCIM, NJStar untuk *desktop* dan *Ajax* IME Taka Kudo, *GownFull*, *GamaIME* Agro Rachmatullah untuk *web application*.

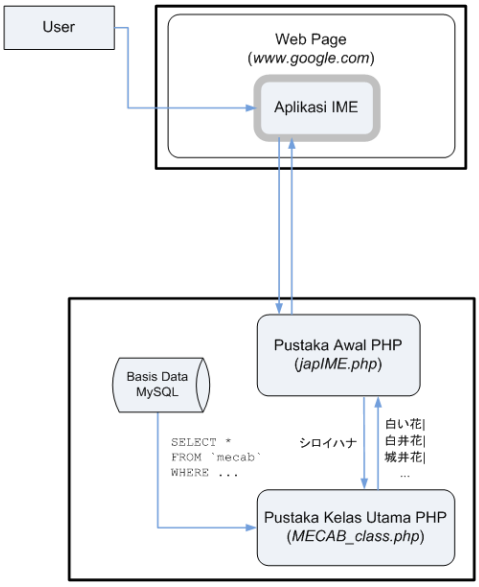
Hanya saja aplikasi *Input Method* tersebut memiliki beberapa kelemahan, antara lain: aplikasi *desktop* paling tidak harus dilakukan proses instalasi, AjaxIME Taka Kudo (*web application*) [1] tidak dapat digunakan untuk *Rich Text Editor*, GownFull (*web application*) [2] hanya terfokus pada bahasa Mandarin dan GamaIME Agro Rachmatullah (*web application*) hanya dapat digunakan pada halaman *web* yang telah disediakan.

Penelitian ini akan difokuskan untuk merancang dan membuat aplikasi sistem *Input Method* untuk bahasa Jepang berbasis *web* dengan menggunakan teknologi JavaScript dan PHP. Secara khusus ada beberapa permasalahan yang akan dipecahkan pada penelitian ini. Permasalahan yang pertama adalah bagaimana cara membuat *web application* yang dapat melakukan transliterasi huruf latin (*romaji*) menjadi karakter *hiragana - katakana* maupun transliterasi karakter *hiragana* menjadi karakter *kanji*. Bagaimana cara melakukan analisa *morphology* untuk frase atau kalimat bahasa Jepang sehingga mendapatkan karakter *kanji* yang sesuai juga menjadi perhatian peneliti. Masalah ketiga dan keempat yang akan diselesaikan adalah bagaimana merancang dan mendesain aplikasi sehingga dapat digunakan pada *Rich Text Editor* dan dapat digunakan untuk lintas *domain (cross domain)*.

Tujuan dari penelitian adalah merancang dan membuat sebuah *web application* menggunakan sistem *Input Method* untuk bahasa Jepang sebagai pilihan alternatif dari *web application* yang telah dikembangkan sebelumnya disaat aplikasi sistem *Input Method* untuk *desktop* tidak dapat digunakan. Manfaat dari aplikasi adalah memberikan keluasaan dalam penggunaan *web application* dengan sistem *Input Method* pada halaman-halaman *web* yang ada dan pada fitur-fitur khusus untuk *web Content Management System (Rich Text Editor)*.



Gambar 1: Diagram alir sistem *Input Method* untuk bahasa Jepang [4]



Gambar 2: Gambaran umum sistem

SISTEM INPUT METHOD

Pada bahasa Asia timur, seperti bahasa Jepang, Cina, atau Korea, jumlah karakter pada bahasa-bahasa tersebut dapat mencapai puluhan ribu sehingga tidak memungkinkan untuk menetapkan model korespondensi *one-to-one* pada papan ketik [3]. Untuk memungkin hal tersebut, dibutuhkan suatu servis khusus yang disebut sebagai *Input Method*. *Input Method* sendiri adalah komponen dalam sistem operasi yang dapat menerima masukan dari papan ketik dan mengolahnya menjadi karakter yang berbeda, bahkan dapat mengolah kembali sebuah karakter atau kumpulan karakter hasil olahannya menjadi karakter baru yang lain. Beberapa *Input Method* telah dikembangkan lebih dalam lagi, seperti *Input Method Editor (IME)* milik Windows atau SCIM milik Linux.

Input Method berada di antara aplikasi dengan pengguna. Karena sifatnya yang menjadi sebuah perantara, secara umum pengguna dapat memberikan masukan karakter yang diingini pada berbagai macam aplikasi tanpa perlu memodifikasi apapun di aplikasi-aplikasi tersebut. Lebih jelasnya, *Input Method* dapat dijalankan jika fokus suatu aplikasi berada pada komponen yang menerima masukan, seperti *text input* atau kotak text. Diagram alir sistem *Input Method* untuk bahasa Jepang dapat dilihat pada Gambar 1.

Pada umumnya *Input Method* menyediakan berbagai alternatif metode untuk menerima masukan. Metode masukan berbasis fonetis, pengguna memberikan masukan dari papan ketik lalu *Input Method* akan mengolah masukkan tersebut. Metode pengenalan tulisan tangan, dengan menggunakan tetikus atau *mouse* pengguna menggambar karakter yang diinginkan pada kanvas yang telah disediakan lalu *Input Method* akan mendeteksi gambar tersebut dan mengolahnya. Metode pengenalan suara, pengguna memberikan masukan dalam bentuk suara (dengan mengucapkan kata) melalui mikrofon lalu *Input Method* akan mendeteksi suara tersebut dan mengolahnya. Metode pencarian berdasarkan kriteria, pengguna memberikan masukan berbentuk kriteria tertentu pada jendela atau *window* yang telah disediakan, misalkan memberikan kriteria jumlah goresan yang diinginkan lalu *Input Method* akan melakukan pencarian karakter berdasarkan jumlah goresan dan menampilkan hasil pencariannya kepada pengguna. Diantara metode-metode diatas, yang paling sering digunakan adalah metode fonetis yang penjelasannya seperti diatas.

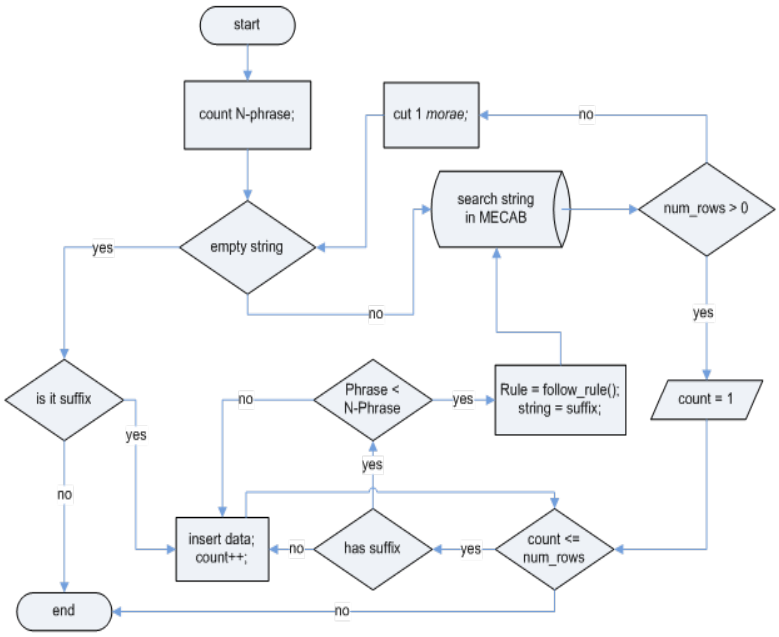
METODOLOGI PENELITIAN

Langkah-langkah yang dilakukan dalam penelitian ini antara lain studi literatur, pengumpulan data, perancangan dan pembuatan sistem, pengujian dan analisis perangkat lunak serta pembuatan laporan. Bagian ini akan menjelaskan secara rinci tahapan yang dilakukan dalam penelitian ini.

Sebelum melakukan perancangan sistem, akan dilakukan studi literatur dalam pemakaian *grammar* dan sintaks untuk frase atau kalimat bahasa Jepang [4, 5, 6]. Hasil studi literatur tersebut digunakan sebagai pedoman untuk mendesain sistem analisa *morphology* yang digunakan oleh sistem aplikasi untuk mendapatkan kata yang tepat dalam konversi karakter *hiragana* menjadi karakter *kanji*.

Proses pengumpulan data dilakukan dengan cara mensurvei kamus kata bahasa Jepang yang akan digunakan dan penggunaan *Object Orientation Programming* untuk Java Script dan PHP [7]. Dalam perancangan dan pembuatan sistem dilakukan: pembuatan diagram alir sistem aplikasi, pembuatan *Database* menggunakan MySQL, pembuatan sistem aplikasi (JavaScript dan PHP) dengan menggunakan Adobe Dreamweaver CS3. Dalam pengujian dan analisis perangkat Lunak, dilakukan pengujian beberapa halaman *web* yang sering diakses oleh masyarakat. Laporan lengkap mengenai penelitian yang telah dilakukan dibuat pada tahap pembuatan laporan akhir, mulai dari pendahuluan sampai dengan kesimpulan.

Secara garis besar, aplikasi *Input Method* ini terbagi dalam dua bagian, yaitu: sisi klien dan sisi server.



Gambar 3: Diagram alir *Longest Match Method*

Sisi klien menggunakan Java Script dan sisi server menggunakan PHP. Pada aplikasi sisi klien, aplikasi yang telah diaktifkan melalui *link* atau-pun *bookmark / favorites* akan memodifikasi setiap *input text widget* dan *textarea widget* yang terdapat pada halaman dimana pengguna mengaktifkan aplikasi. Hasil modifikasi tersebut mengijinkan aplikasi untuk melakukan konversi huruf latin (*romaji*) menjadi karakter *hiragana / kata-kana* sesuai dengan masukan dari pengguna. Sedangkan aplikasi pada sisi *server* hanya akan aktif saat pengguna melakukan permintaan untuk konversi karakter *hiragana* menjadi karakter *kanji*. Pada aplikasi sisi *server* tersebut akan melakukan proses analisa *morphology* untuk menemukan kata atau karakter *kanji* yang tepat dari permintaan konversi oleh pengguna dan mengembalikan hasil konversi *hiragana* menjadi *kanji* dari analisa *morphology*.

Gambaran umum sistem dapat dilihat pada Gambar 2. Gambaran umum pada Gambar 2 mirip dengan gambaran umum sistem aplikasi dari *GamaIME* milik Agro Rachmatullah, karena penelitian ini menggunakan referensi dari sistem aplikasi *GamaIME* milik Agro Rachmatullah. Perbedaannya adalah jika sistem *GamaIME* hanya dapat digunakan pada halaman *web* yang telah disediakan sistem aplikasi ini dapat digunakan pada halaman-halaman *web* yang lain. Sistem *GamaIME* menggunakan ASP .NET pada sisi *server* sementara aplikasi ini menggunakan PHP pada sisi *server*nya.

Untuk analisa *morphology*, sistem aplikasi ini menggunakan dua metode, yaitu: *Longest Match Method* dan *Rules Method*. *Longest Match Method* adalah metode pencarian kata pada kamus kata dari kata yang terpanjang setelah pemotongan satu karakter dari belakang [3]. Diagram alir untuk Longest Match Method dapat dilihat pada Gambar 3. Sedangkan *Rules Method* adalah metode pencarian kata berdasarkan aturan *grammar* dan sintaks yang telah

(S)	→	*(NP) (PP) (AP) (AuxP)* |
		(NP) (PP) (NP) (PP) (VP) (AuxP) |
		(NP) (PP) (NP)
		(AuxP) | *(NP) (PP) (VP)*
(NP)	→	*(AP) (NP)* | *(N)*
(VP)	→	*(V)*
(AuxP)	→	*(Aux) (AuxP)*
(PP)	→	*(P) (PP)*
(AP)	→	*(Adv) (Adj)* | *(Adv) (P)* |
		(Adj)
(Adj)	→	*(Adj) (Aux) Adj)*

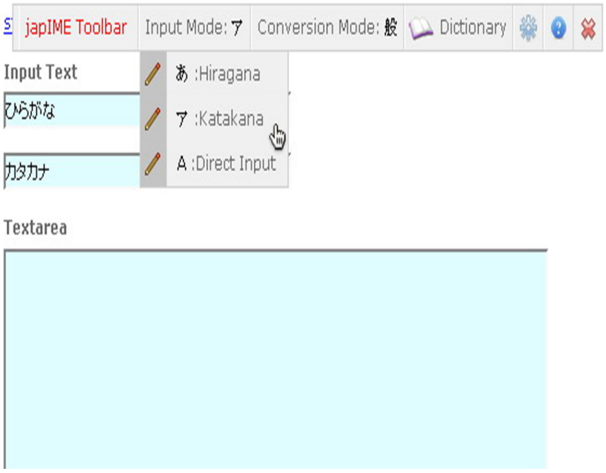
Gambar 4: *Production Rules* untuk aturan *morphology*

ditetapkan sebelumnya. Aturan *grammar* dan sintaks untuk frase atau kalimat bahasa Jepang dirancang menggunakan *Context Free Grammar* yang dapat dilihat pada Gambar 4. Untuk urutan kandidat-kandidat hasil dari analisa *morphology* diurutkan berdasarkan nilai terkecil dari metode *Minimum Cost Method*, dimana nilai yang dihasilkan berdasarkan nilai relasi *Part-of-Speech* dari hasil analisa *morphology*.

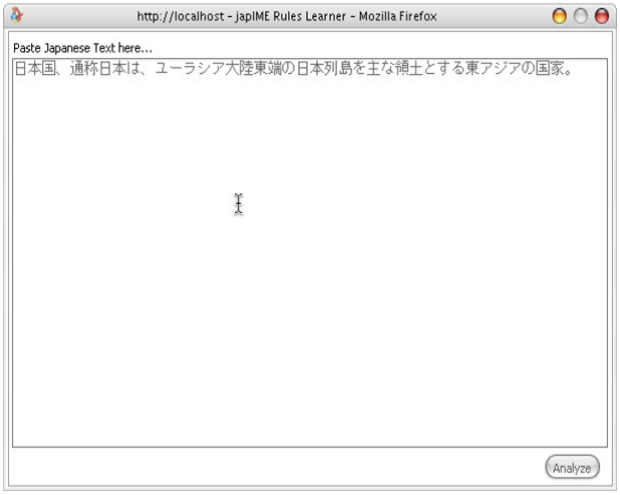
Aplikasi ini juga memberikan fasilitas lainnya, seperti fasilitas *Dictionary* dan fasilitas *Rules Learner*. Pada fasilitas *Dictionary*, pengguna dapat mencari sebuah kata dalam kamus kata MECAB untuk mengetahui jenis *Part-of-speech*, *conjugation type*, *conjugation form* dan lain-lain dari kata yang dicari tersebut. Sedangkan pada fasilitas *Rules Learner*, sistem dapat mempelajari aturan *morphology* dari frase atau kalimat yang dimasukkan pengguna dan dapat menyimpan aturan baru *morphology* tersebut kedalam *database*.

PENGUJIAN SISTEM

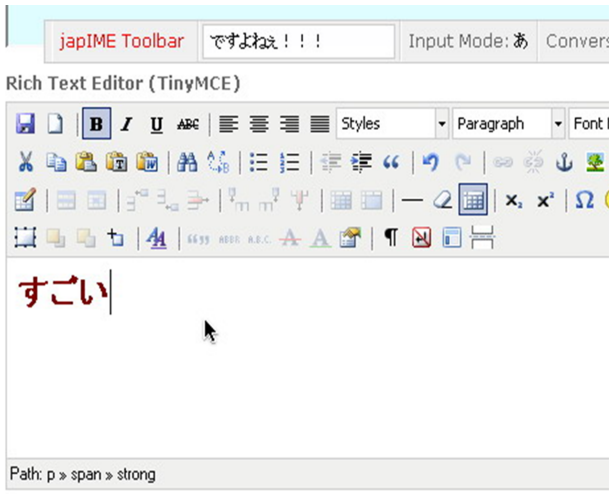
Untuk menguji aplikasi ini, dilakukan proses pengujian dengan dua cara melalui halaman web.



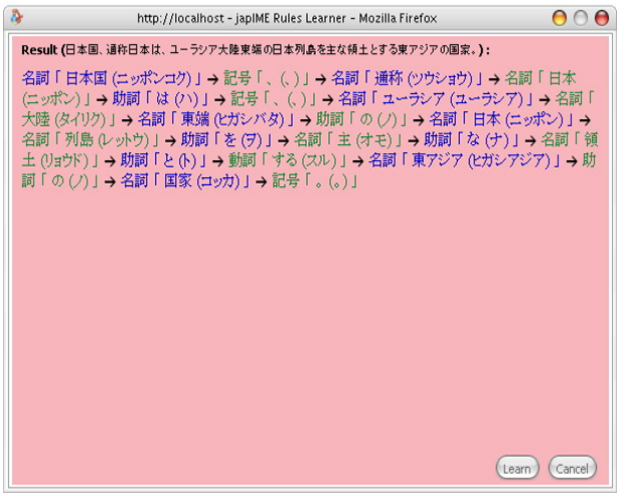
Gambar 5: Konversi huruf latin ke *hiragana-katakana*



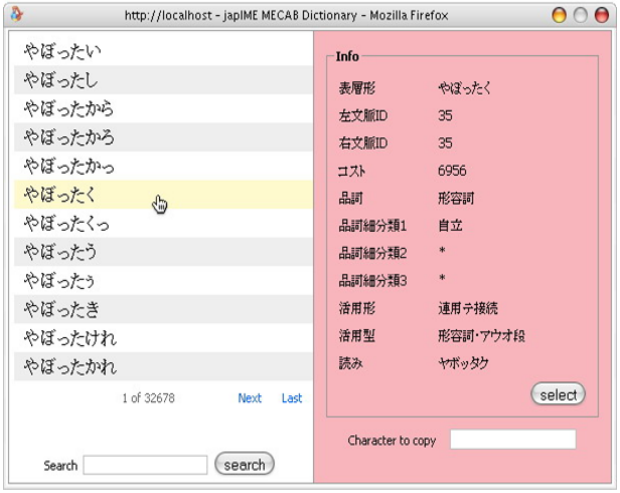
Gambar 8: Jendela fasilitas *Rules Learner*



Gambar 6: Hasil pengujian pada *Rich Text Editor*



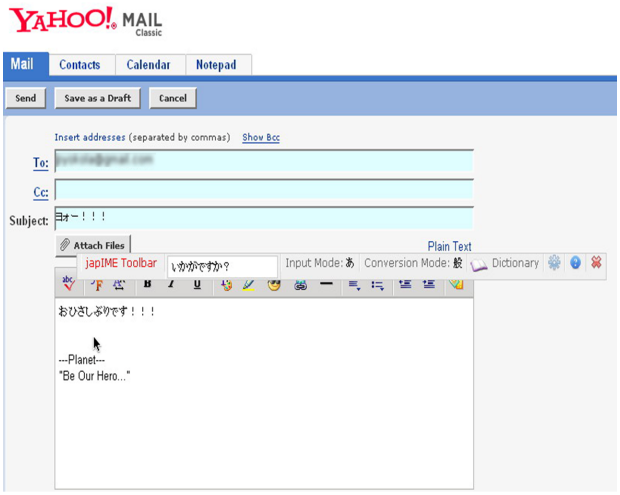
Gambar 9: Pengujian frase Jepang pada *Rules Learner*



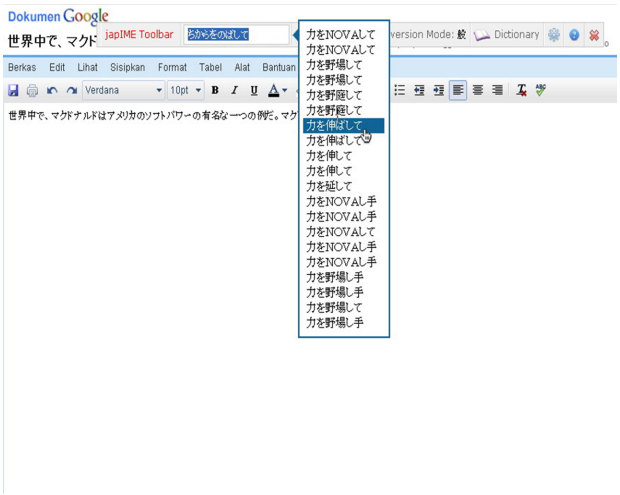
Gambar 7: Jendela fasilitas *Dictionary*

Gambar 10: Pengujian aplikasi pada *Google Search*

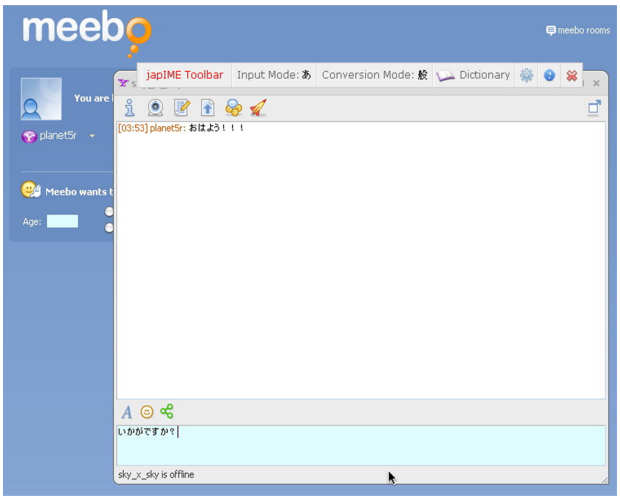
Gambar 11: Pengujian aplikasi pada *Yahoo Search*



Gambar 12: Pengujian aplikasi pada *Yahoo Mail*



Gambar 13: Pengujian aplikasi pada *Google Docs*



Gambar 14: Pengujian aplikasi pada *Meebo*

Mekanisme pengujian yang pertama dilakukan pada halaman *web Demo* yang telah disediakan sebelumnya. Cara yang kedua dilakukan pengujian pada halaman-halaman *web* yang lain, seperti:

- `http://www.google.com`,
- `http://www.yahoo.com`,
- `http://mail.yahoo.com`,
- `http://www.meebo.com`,
- `http://www.wordpress.com`, dan
- `http://docs.google.com`.

Pengujian pada halaman *web* Demo

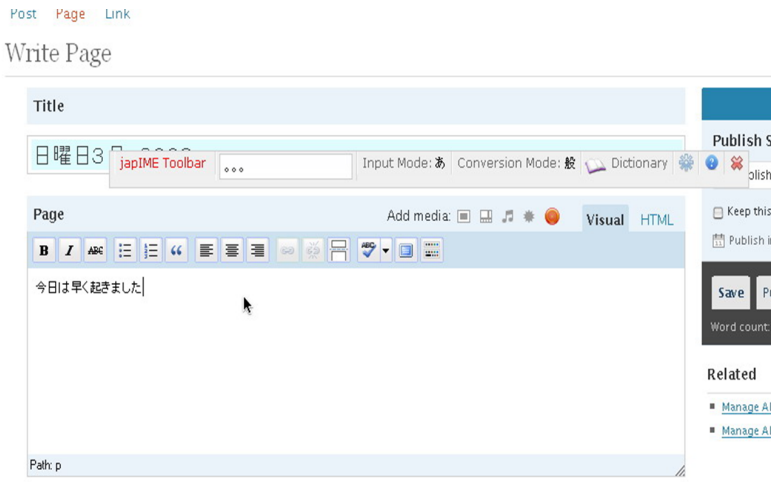
Setelah pengguna mengaktifkan aplikasi melalui *link* atau *bookmark*, aplikasi akan aktif dan memodifikasi seluruh *input text widget* dan *textarea* pada halaman *web* tersebut sehingga dapat dilakukan konversi huruf latin (*romaji*) menjadi karakter *hiragana* atau *katakana*. Hasil pengujian konversi huruf latin menjadi karakter *hiragana* atau *katakana* dapat dilihat pada Gambar 5.

Untuk penggunaan aplikasi pada Rich Text Editor, pengguna harus mengaktifkan fasilitas *Rich Text Editor* sebelum melakukan konversi huruf latin menjadi karakter *hiragana* atau *katakana* pada *Rich Text Editor*. Saat fasilitas *Rich Text Editor* aktif, aplikasi akan memunculkan satu *input text widget* baru sebagai pre-editing sebelum dikembalikan kepada *Rich Text Editor*. Hasil pengujian pada *Rich Text Editor* dapat dilihat pada Gambar 6.

Untuk mengaktifkan fasilitas *Dictionary*, pengguna dapat mengaktifkan fasilitas tersebut de-ngan memilih opsi *Dictionary* dan memilih opsi MECAB *Dictionary* pada *toolbar* aplikasi. Gambar fasilitas *Dictionary* aplikasi dapat dilihat pada Gambar 7.

Untuk meng-aktifkan fasilitas *Rules Learner*, pengguna dapat meng-aktifkan fasilitas tersebut dengan memilih opsi *Dictionary* dan memilih opsi *Rules Learner* pada *toolbar* aplikasi. Gambar fasilitas *Rules Learner* aplikasi ditunjukkan pada Gambar 8.

Pertama-tam pengguna memasukkan suatu frase atau kalimat bahasa Jepang seperti pada contoh Gambar 8 dan menekan tombol *Analyze*.



Gambar 15: Pengujian aplikasi pada *Wordpress*

Kemudian sistem akan melakukan analisis dari frase atau kalimat bahasa Jepang yang telah dimasukkan oleh pengguna. Hasil dari analisa frase atau kalimat bahasa Jepang dapat dilihat pada Gambar 9. Pengguna dapat menekan tombol *Learn* agar sistem dapat menyimpan aturan *morphology* baru dari hasil analisa.

Pengujian pada halaman-halaman *web* lain

Hasil pengujian pada *web* mesin pencari:
- Gambar 10 untuk *Google Search*
- Gambar 11 untuk *Yahoo Search*
- Gambar 12 untuk *Yahoo Mail*
- Gambar 13 untuk *Google Docs*
- Gambar 14 untuk *Meebo*
- Gambar 15 untuk *WordPress*

SIMPULAN

Pada akhir perancangan dan pambuatan aplikasi ini dapat ditarik beberapa kesimpulan. Aplikasi ini dapat melakukan konversi dari huruf latin (romaji) menjadi karakter *hiragana* atau *katakana* dan konversi dari karakter *hiragana* menjadi karakter *kanji*. Aplikasi ini juga dapat melakukan analisa *morphology* untuk menemukan kata atau karakter *kanji* yang tepat dalam konversi karakter *hiragana* menjadi karakter *kanji*. Kemudian aplikasi dapat digunakan pada halaman-halaman *web* lain, namun aplikasi tidak dapat digunakan pada halaman *web* yang di desain secara khusus seperti *Google Mail*. Selain itu kecepatan konversi karakter *hiragana* menjadi karakter *kanji* dipengaruhi panjang dari kata, frase, maupun kalimat bahasa Jepang yang dimasukkan oleh pengguna. Kecepatan koneksi *internet* yang digunakan juga mempengaruhi kecepatan *transfer* data dari *client* ke *server* dan sebaliknya.

DAFTAR PUSTAKA

[1] unknown: *Ajax IME: Web-based Japanese Input Method*. Tersedia di `http://ajaxime.chasen.org` (August 2008)

[2] unknown: *GownFull Ajax IME*. Tersedia di `http://www.ajaxime.com` (September 2008)

[3] unknown: *What is an IME (Input Method Editor) and How do I Use It*. Tersedia di `http://www.microsoft.com/globaldev/handson/user/ime_paper.mspx` (September 2008)

[4] unknown: *An Introduction to Japanese Input System*. Tersedia di `http://tronweb.super-nova.co.jp/jpnimintro.html` (September 2008)

[5] Tsujimura, N.: *An Introduction to Japanese Linguistic*. United Kingdom, Blackwell Publishers Inc (2006)

[6] unknown: *A Japanese Guide to Japanese Grammar*. Tersedia di `http://www.guidetojapanese.org` (September 2008)

[7] unknown: *WrenSoft-Zoom Search Engine: Benchmarking PHP vs. ASP vs Javascript vs Binary CGI (C++)*. Tersedia di `http://www.wrensoft.com/zoom/benchmarks.html` (September 2008)