

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

Alexander Setiawan<sup>(1)</sup>, Rolly Intan<sup>(2)</sup>, Rikko Filiano<sup>(3)</sup>

Fakultas Teknologi Industri – Jurusan Teknik Informatika Universitas Kristen Petra

Jl. Siwalankerto 121-131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

[alexander@peter.petra.ac.id](mailto:alexander@peter.petra.ac.id)<sup>(1)</sup>, [rintan@peter.petra.ac.id](mailto:rintan@peter.petra.ac.id)<sup>(2)</sup>, [pyokola@gmail.com](mailto:pyokola@gmail.com)<sup>(3)</sup>

---

## Abstrak

Bahasa Jepang memiliki ribuan karakter untuk sistem penulisannya. Dibutuhkan sebuah sistem bernama *Input Method* untuk memungkinkan 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.

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

---

## 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.*

**Keyword:** JavaScript, PHP, *Input Method*, *MeCab*, *Longest Match Method*

---

## 1. Pendahuluan

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:

- a. Aplikasi *desktop* paling tidak harus dilakukan proses instalasi.
- b. *AjaxIME* –Taka Kudo (*web application*) [1] tidak dapat digunakan untuk *Rich Text Editor*.
- c. *GownFull* (*web application*) [2] hanya terfokus pada bahasa Mandarin.
- d. *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 permasalahan yang akan dipecahkan pada penelitian ini adalah:

- a. Bagaimana cara membuat *web application* yang dapat melakukan transliterasi huruf latin (*romaji*) menjadi karakter *hiragana* - *katakana* maupun transliterasi karakter *hiragana* menjadi karakter *kanji*?
- b. Bagaimana cara melakukan analisa *morphology* untuk frase atau kalimat bahasa Jepang sehingga mendapatkan karakter *kanji* yang sesuai?
- c. Bagaimana merancang dan mendesain aplikasi sehingga dapat digunakan pada *Rich Text Editor*?
- d. Bagaimana merancang dan mendesain aplikasi sehingga dapat digunakan lintas *domain* (*cross domain*)?

Tujuan dari penelitian ini adalah merancang dan membuat sebuah *web application* dengan 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.

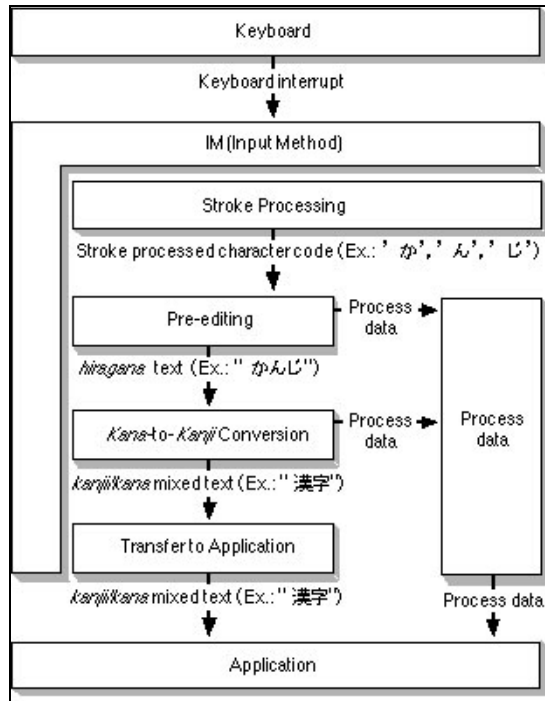
Manfaat dari aplikasi ini adalah memberikan keluasaan dalam penggunaan *web application* dengan sistem *Input Method* pada halaman-halaman *web* yang ada dan pada fitur-

fitur khusus untuk *web Content Management System* (*Rich Text Editor*).

## 2. Sistem *Input Method*

Pada bahasa Asia timur, seperti bahasa Jepang, bahasa Cina, bahasa 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 memungkinkan 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 masukkan karakter yang diinginkan 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 masukkan, seperti *text input* atau kotak *text*. Diagram alir sistem *Input Method* untuk bahasa Jepang dapat dilihat pada Gambar 1.



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

Pada umumnya *Input Method* menyediakan berbagai alternatif metode untuk menerima masukan, antara lain adalah:

- Masukkan berbasis fonetis, pengguna memberikan masukan dari papan ketik lalu *Input Method* akan mengolah masukan tersebut.
- 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.
- Pengenalan suara, pengguna memberikan masukan dalam bentuk suara (dengan mengucapkan kata) melalui mikrofon lalu *Input Method* akan mendeteksi suara tersebut dan mengolahnya.
- 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.

### 3. METODOLOGI PENELITIAN

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

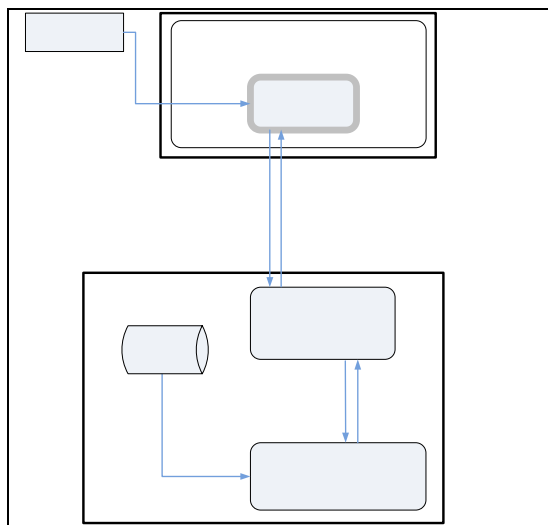
- Studi Literatur  
Dalam studi literatur ini, mempelajari berbagai literatur bahasa Jepang. Pemakaian *grammar* dan sintaks untuk frase atau kalimat bahasa Jepang.
- Pengumpulan Data  
Melakukan survei kamus kata bahasa Jepang yang akan digunakan, penggunaan *Object Orientation Programming* untuk JavaScript dan PHP.
- Perancangan dan Pembuatan Sistem  
Dalam tahapan ini, dilakukan:
  - Pembuatan diagram alir sistem aplikasi
  - Pembuatan *Database* menggunakan MySQL.
  - Pembuatan sistem aplikasi (JavaScript dan PHP) dengan menggunakan Adobe Dreamweaver CS3.
- Pengujian dan Analisis Perangkat Lunak  
Melakukan pengujian beberapa halaman *web* yang sering diakses oleh masyarakat.
- Pembuatan Laporan  
Laporan lengkap mengenai penelitian yang telah dibuat, mulai dari pendahuluan sampai dengan kesimpulan.

Sebelum melakukan desain sistem, akan dilakukan studi literatur *grammar* dan sintaks dari struktur kalimat bahasa Jepang. Hasil studi literature tersebut digunakan sebagai pondoman untuk mendesain sistem analisa *morphology* yang digunakan oleh sistem aplikasi untuk mendapatkan kata yang tepat dalam konversi karakter *hiragana* menjadi karakter *kanji*.

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

Pada aplikasi sisi klien, aplikasi yang telah diaktifkan melalui *link* ataupun *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 / katakana* sesuai dengan masukan dari pengguna. Sedangkan aplikasi pada sisi *server* hanya akan aktif saat pengguna melakukan permintaan untuk melakukan 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.

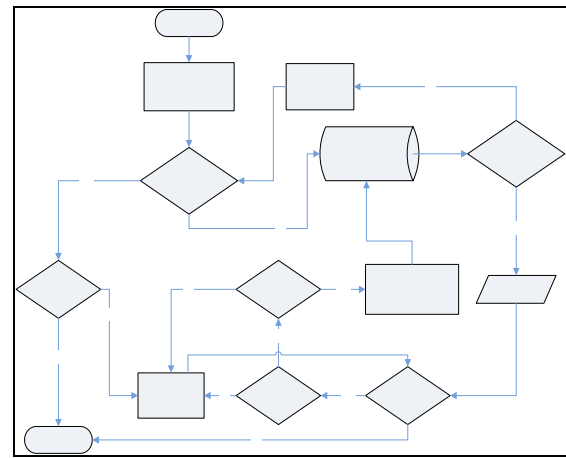


Gambar 2. Gambaran umum sistem

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

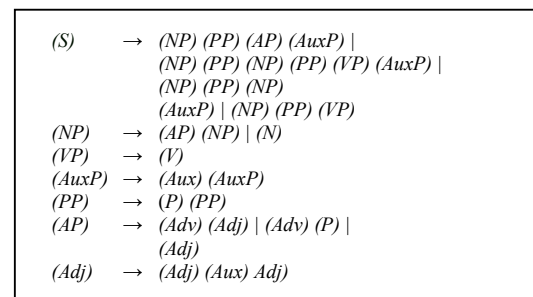
Untuk analisa *morphology*, sistem aplikasi ini menggunakan dua metode, yaitu: *Longest Match Method* dan *Rules Method*. *Longest Match Method* adalah metode pencarian kata pada kamus kata dari kata yang

terpanjang setelah pemotongan satu karakter dari belakang [3]. Diagram alir untuk *Longest Match Method* dapat dilihat pada Gambar 3.



Gambar 3. Diagram alir *Longest Match Method*

Sedangkan *Rules Method* adalah metode pencarian kata berdasarkan aturan *grammar* dan sintaks yang telah ditetapkan sebelumnya. Aturan *grammar* dan sintaks untuk frase atau kalimat bahasa Jepang dirancang menggunakan *Context Free Grammar* yang dapat dilihat pada Gambar 4.



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

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 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*

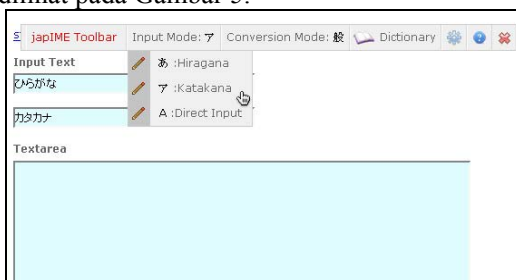
*type*, *conjugation form* dan lain-lain dari kata yang dicari tersebut. Sedangkan pada fasilitas *Rules Learner*, sistem dapat mempelajari aturan *morphology* dari frase atau kalimat yang dimasukkan pengguna dan dapat menyimpan aturan baru *morphology* tersebut kedalam *database*.

## 5. PENGUJIAN SISTEM

Dalam pengujian aplikasi, akan melakukan pengujian dengan dua cara, yang pertama pengujian pada halaman *web Demo* yang telah disediakan sebelumnya. Yang kedua adalah 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>.

### 5.1 Pengujian pada halaman *web Demo*

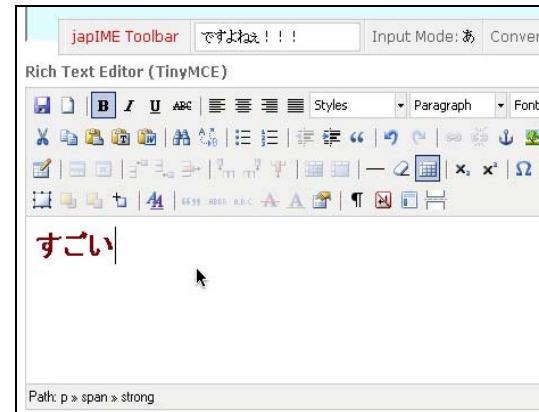
Setelah pengguna mengaktifkan aplikasi melalui *link* atau *bookmark*, aplikasi akan aktif dan memodifikasi seluruh *input text widget* dan *textarea* pada halaman *web* tersebut sehingga dapat dilakukan konversi huruf latin (*romaji*) menjadi karakter *hiragana* atau *katakana*. Hasil pengujian konversi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



Gambar 5. Hasil pengujian konversi huruf latin menjadi karakter *hiragana* atau *katakana*

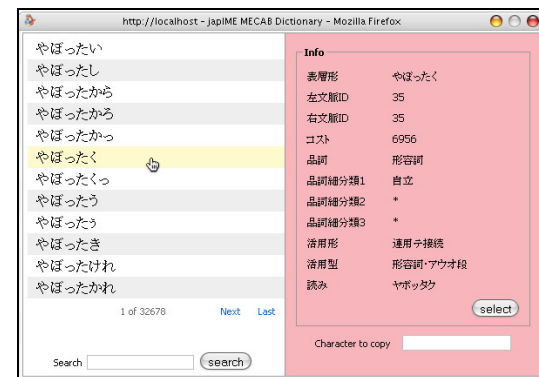
Untuk penggunaan aplikasi pada *Rich Text Editor*, pengguna harus mengaktifkan fasilitas *Rich Text Editor* sebelum melakukan konversi huruf latin menjadi karakter *hiragana* atau *katakana* pada *Rich Text Editor*. Saat fasilitas *Rich Text Editor* aktif, aplikasi akan memunculkan satu *input text widget* baru

sebagai *pre-editing* sebelum dikembalikan kepada *Rich Text Editor*. Hasil pengujian pada *Rich Text Editor* dapat dilihat pada Gambar 6.



Gambar 6. Hasil pengujian pada *Rich Text Editor*

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



Gambar 7. Jendela fasilitas *Dictionary*

Untuk mengaktifkan fasilitas *Rules Learner*, pengguna dapat mengaktifkan fasilitas tersebut dengan memilih opsi *Dictionary* dan memilih opsi *Rules Learner* pada *toolbar* aplikasi. Gambar fasilitas *Rules Learner* aplikasi dapat dilihat pada Gambar 8.



Gambar 8. Jendela fasilitas *Rules Learner*

Saat pengguna telah memasukkan suatu frase atau kalimat bahasa Jepang seperti pada contoh Gambar 8 diatas dan menekan tombol *Analyze*, 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.



Gambar 9. Hasil pengujian analisa frase atau kalimat bahasa Jepang pada fasilitas *Rules Learner*

Agar sistem menyimpan aturan *morphology* baru dari hasil analisa, pengguna dapat menekan tombol *Learn*.

## 5.2 Pengujian pada halaman-halaman *web* lain

Hasil pengujian aplikasi pada halaman *web* mesin pencari *Google Search* dengan alamat <http://www.google.com> dapat dilihat pada Gambar 10.



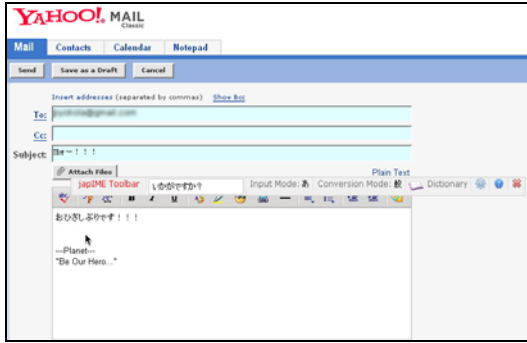
Gambar 10. Hasil pengujian aplikasi pada halaman *Google Search*

Hasil pengujian aplikasi pada halaman *web* mesin pencari *Yahoo Search* dengan alamat <http://www.yahoo.com> dapat dilihat pada Gambar 11.



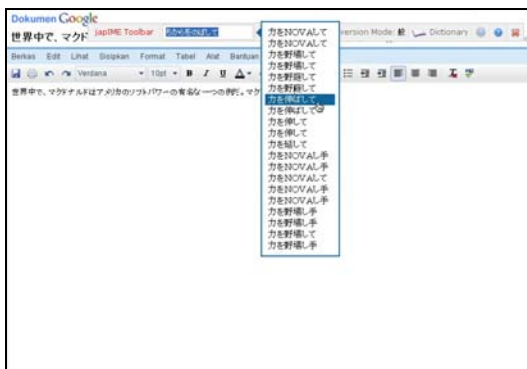
Gambar 11. Hasil pengujian aplikasi pada halaman *Yahoo Search*

Hasil pengujian aplikasi pada halaman *web* mesin pencari *Yahoo Mail* dengan alamat <http://mail.yahoo.com> dapat dilihat pada Gambar 12.



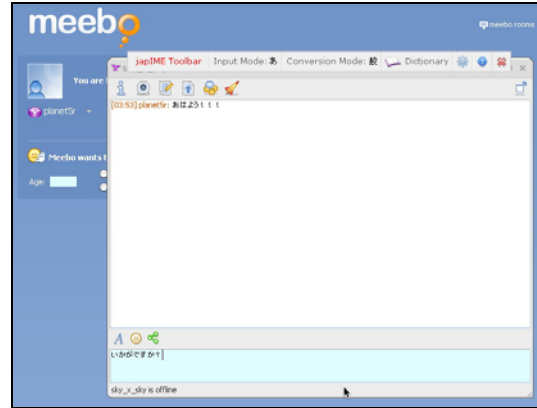
Gambar 12. Hasil pengujian aplikasi pada halaman *Yahoo Mail*

Hasil pengujian aplikasi pada halaman *web* mesin pencari *Google Docs* dengan alamat <http://docs.google.com> dapat dilihat pada Gambar 13.



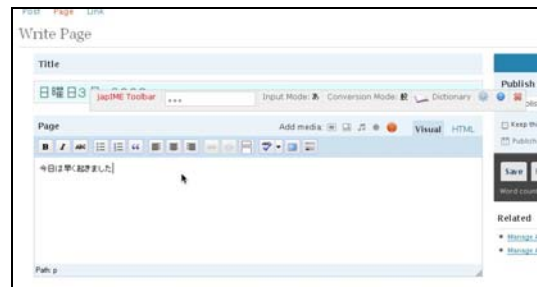
Gambar 13. Hasil pengujian aplikasi pada halaman *Google Docs*

Hasil pengujian aplikasi pada halaman *web* mesin pencari *Meebo* dengan alamat <http://www.meebo.com> dapat dilihat pada Gambar 14.



Gambar 14. Hasil pengujian aplikasi pada halaman *Meebo*

Hasil pengujian aplikasi pada halaman *web* mesin pencari *Meebo* dengan alamat <http://www.wordpress.com> dapat dilihat pada Gambar 15.



Gambar 15. Hasil pengujian aplikasi pada halaman *Wordpress*

## 6. KESIMPULAN

Pada akhir perancangan dan pembuatan aplikasi ini dapat ditarik beberapa kesimpulan, yaitu:

- Aplikasi ini dapat melakukan konversi dari huruf latin (*romaji*) menjadi karakter *hiragana* atau *katakana* dan konversi dari karakter *hiragana* menjadi karakter *kanji*.
- Aplikasi ini dapat melakukan analisa *morphology* untuk menemukan kata atau karakter *kanji* yang tepat dalam konversi karakter *hiragana* menjadi karakter *kanji*.
- Aplikasi dapat digunakan pada halaman-halaman *web* lain, namun aplikasi tidak dapat digunakan pada halaman *web* yang di desain secara khusus seperti *Google Mail*.



- 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 mempengaruhi kecepatan *transfer* data dari *client* ke *server* dan sebaliknya.

## 7. DAFTAR PUSTAKA

- [1] *Ajax IME: Web-based Japanese Input Method*. Last viewed: August 30, 2008. <<http://ajaxime.chasen.org>>
- [2] *GownFull- an Ajax IME*. Last viewed: September 18, 2008. <<http://www.ajaxime.com>>
- [3] *What is an IME (Input Method Editor) and how do I use it*. Last viewed: September 05, 2008. <[http://www.microsoft.com/globaldev/handson/user/ime\\_paper.mspx](http://www.microsoft.com/globaldev/handson/user/ime_paper.mspx)>
- [4] *An Introduction to Japanese Input System*. Last viewed: September 18, 2008. <<http://tronweb.super-nova.co.jp/jpnimintro.html>>
- [5] WrenSoft - Zoom Search Engine –Benchmarking PHP vs. ASP vs Javascript vs Binary CGI (C++). Last viewed: November 20, 2008. <<http://www.wrensoft.com/zoom/benchmarks.html>>
- [6] Tsujimura, Natsuko. 2006, *an introduction to Japanese Linguistic*. United Kingdom, Blackwell Publishers Inc.
- [7] A Japanese guide to Japanese grammar. Last viewed: September 17, 2008. <<http://www.guidetojapanese.org>>