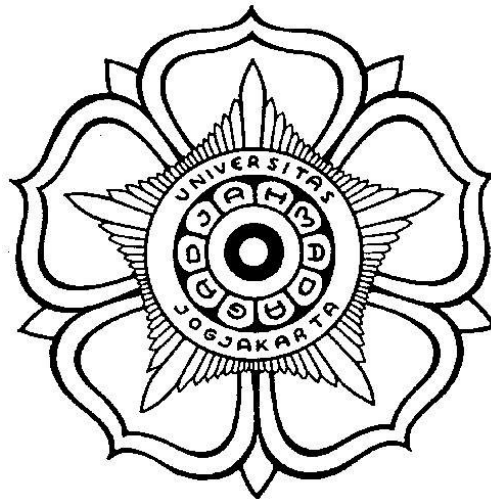


SKRIPSI

**PENGEMBANGAN INPUT METHOD EDITOR
BAHASA JEPANG BERBASIS WEB**

**Agro Rachmatullah
03/165399/PA/09374**

Sebagai salah satu syarat untuk memperoleh derajat S1
Program Studi Ilmu Komputer pada Jurusan Matematika



**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS GADJAH MADA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
YOGYAKARTA
2007**

SKRIPSI

PENGEMBANGAN INPUT METHOD EDITOR BAHASA JEPANG BERBASIS WEB

Agro Rachmatullah
03/165399/PA/09374

Telah dipertahankan dan dinyatakan lulus
oleh Tim Penguji Program Studi Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Gadjah Mada
pada tanggal
22 November 2007

Tim Penguji

Nama	Tanda Tangan
1. Moh. Edi Wibowo, S.Kom., M.Kom (Dosen Pembimbing)	1. _____
2. Khabib Mustofa, Dr., M.Kom (Dosen Penguji)	2. _____
3. Anifuddin Azis, S.Si., M.Kom (Dosen Penguji)	3. _____

**Karya ini dipersembahkan untuk
Bapak dan ibuku
Budi Prakoso dan Imastini Dinuriah
Terima kasih atas segalanya**

KATA PENGANTAR

Segala puji penulis ucapkan bagi ﷻ SWT yang selalu melimpahkan rahmat, hidayah, dan petunjuk-Nya sehingga skripsi ini bisa diselesaikan. Salawat dan salam juga penulis curahkan bagi Nabi Muhammad SAW yang telah mengemban amanah mulia memberi penerangan pada peradaban manusia.

Skripsi ini adalah salah satu syarat untuk mendapatkan derajat S1 pada Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada. Penulis memilih topik pemrosesan teks bahasa Jepang karena selama dua tahun terakhir penulis memang sedang asyik mempelajari bahasa yang sangat unik ini.

Program yang dikembangkan pada awalnya bertujuan untuk memecahkan masalah nyata, yaitu bagaimana cara memasukkan teks bahasa Jepang pada komputer yang terkoneksi internet namun tidak memiliki IME di sistem operasinya. Kasus tersebut akan banyak ditemui pelajar bahasa Jepang di Indonesia dan negara luar Jepang lainnya saat menggunakan komputer umum seperti komputer warnet. Di tengah maraknya ledakan aplikasi web saat ini, IME berbasis web secara alami muncul sebagai solusi yang dipikirkan penulis.

Namun belakangan (setelah topik disetujui dosen pembimbing) ditemukan bahwa sudah ada IME bahasa Jepang berbasis web yang dibuat oleh Taka Kudo dari NTT Communication Science Laboratories sekitar 1 tahun sebelumnya. Pada akhirnya IME yang dibuat penulis kemampuannya masih di bawah IME Taka Kudo tersebut, terutama karena IME Taka Kudo memiliki komponen pemecah

kalimat sehingga bisa mentransliterasi kalimat kompleks seperti この文章は難しいでしょう (dengan IME penulis, kalimat masukan tersebut harus dibagi menjadi 2 bagian yaitu 'konobunshouha' dan 'muzukashiideshou'). IME Taka Kudo bisa diakses di <http://ajaxime.chasen.org>.

Karena masalah yang sebetulnya ingin dipecahkan ternyata sudah ada solusinya, maka tujuan pengembangan program ini pada akhirnya hanyalah sebagai tantangan bagi diri penulis untuk membuat sistem sejenis tanpa mengintip sedikitpun kode IME web dari Taka Kudo maupun dari IME-IME desktop open source seperti SCIM.

Walaupun tidak secanggih IME Taka Kudo, IME yang dikembangkan penulis juga bisa digunakan untuk menulis teks bahasa Jepang. Karena itu, silakan mencoba produk domestik ini di <http://agro.web.ugm.ac.id/ime>. Penulis juga berharap bahwa skripsi ini bisa menjadi bacaan yang berguna bagi orang Indonesia yang tertarik dengan topik yang penulis pilih.

Pada kesempatan ini penulis juga ingin mengucapkan terima kasih kepada:

1. Kedua orang tua penulis yang membesarkan penulis dengan penuh sabar dan kasih sayang.
2. Adik-adikku Agri, Cadika, Duta, Kika, dan Uchang yang membuat suasana keluarga senantiasa ramai dan ceria.
3. Eyang Kakung dan Eyang Bandung yang selalu mendoakan cucu-cucunya.
4. Om Bagus, Tante Eni, Mas Alvin dan adik-adikaknya yang menjadi *benefactor* penulis selama kuliah dan memberikan banyak dukungan lainnya.

5. Bapak Agus Sihabuddin, S.Si., M.Kom, sebagai dosen wali penulis yang telah membimbing selama 4 tahun lebih.
6. Bapak Moh. Edi Wibowo, S.Kom., M.Kom yang telah meluangkan waktunya sebagai dosen pembimbing penulisan skripsi ini
7. Bapak Khabib Mustofa, Dr., M.Kom dan Bapak Anifuddin Azis, S.Si., M.Kom selaku dosen penguji yang telah memberikan berbagai saran untuk perbaikan skripsi ini.
8. Teman-teman satu angkatan, secara alfabetis, Arin, Awang, Bram, Dian, Endro, Fuad, Karnan, Tri, Tri¹, Wijaya, dan tentu saja seluruh anak 03 lainnya.
9. Teman-teman di YIC, Mas Ibond, Tegar, Mas Jiyuu, Mba Shinda, Gustra, Mba Hacchan, Mas Erik, Hyu, Welldan, dan masih banyak lagi. Kalian adalah salah satu bagian penting hitam dan putihnya kehidupan penulis selama di Jogja.

皆様、本当にどうもありがとうございました！

Tentu saja karya ini jauh dari sempurna. Oleh karenanya, silakan kirim kritik dan saran ke agro1986@gmail.com.

Yogyakarta, 22 November 2007
Penulis,

Agro Rachmatullah

1 bukan typo, memang ada 2 'Tri'

Daftar Isi

KATA PENGANTAR.....	iv
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiv
INTISARI.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	5
1.6 Tinjauan Pustaka.....	5
1.7 Metodologi Penelitian.....	6
1.8 Sistematika Penulisan.....	7
BAB II LANDASAN TEORI.....	9
2.1 Sistem Tulis Bahasa Jepang.....	9
2.1.1 Hiragana.....	9
2.1.1.1 Simbol-Simbol Hiragana.....	10
2.1.1.2 Penggunaan Hiragana.....	15
2.1.2 Katakana.....	16
2.1.2.1 Simbol-Simbol Katakana.....	16
2.1.2.2 Penggunaan Katakana.....	19
2.1.3 Kanji.....	21
2.1.3.1 Bacaan on.....	22
2.1.3.2 Bacaan kun.....	23
2.1.3.3 Bacaan nanori.....	24
2.1.3.4 Bacaan yang digunakan.....	24
2.2 Transliterasi Bahasa Jepang menggunakan Huruf Latin.....	26
2.3 Modifikasi Kata Dasar pada Bahasa Jepang.....	30
2.4 Partikel dan Gobi.....	33
2.5 Input Method Editor (IME).....	33
2.5.1 IME Fonetis Windows XP.....	36
2.6 Himpunan Karakter dan Pengkodean Bahasa Jepang di Komputer.....	38
2.6.1 Himpunan Karakter dan Pengkodean.....	38
2.6.2 Himpunan Karakter Bahasa Jepang.....	39
2.6.2.1 Hiragana pada Unicode.....	40
2.6.2.2 Katakana pada Unicode.....	40
2.6.2.3 Kanji pada Unicode.....	42
2.7 Breadth-first Search (BFS).....	43
2.8 EDICT.....	43
2.8.1 Format.....	44
2.9 SQLite.....	48
2.10 Framework .NET, C#, dan ASP.NET.....	49

2.10.1 Framework .NET.....	49
2.10.2 C#.....	50
2.10.3 ASP.NET.....	51
2.11 JavaScript dan AJAX.....	52
2.11.1 JavaScript.....	52
2.11.2 AJAX.....	52
2.12 Pustaka Event Dean Edwards.....	55
BAB III ANALISIS, PERANCANGAN, DAN IMPLEMENTASI.....	56
3.1 Gambaran Umum Sistem.....	56
3.2 Komponen Pembangun Basis Data: EdictDbBuilder.exe.....	58
3.2.1 Tinjauan.....	58
3.2.2 Analisis Permasalahan.....	58
3.2.3 Analisis Kebutuhan Sistem.....	59
3.2.4 Tabel edict.....	59
3.2.4.1 Field kana.....	60
3.2.4.2 Field kanji.....	60
3.2.4.3 Contoh Pengisian (kana, kanji).....	60
3.2.4.4 (kana, kanji) sebagai Kunci Primer.....	62
3.2.4.5 Field tags, edict_allowed_tags.txt, dan edict_tags_transform.txt.....	63
3.2.4.6 Field pop.....	67
3.2.4.7 Field uk.....	70
3.2.5 Implementasi.....	70
3.2.6 Kelas EdictEntry.....	71
3.2.6.1 Tinjauan.....	71
3.2.6.2 Perancangan Kelas.....	71
3.2.6.3 Metode Statik string EdictEntry.KatakanaToHiragana(string input).....	72
3.2.6.4 Metode void EdictEntry.TryAddTag(List<string> tagList, string tag).....	74
3.2.6.5 Kontruktor EdictEntry().....	74
3.2.7 Kelas BuildDb.....	77
3.2.7.1 Tinjauan.....	77
3.2.7.2 Perancangan Kelas.....	77
3.2.7.3 Metode Statik string BuildDb.JoinTags(string[] tags).....	78
3.2.7.4 Metode Statik void BuildDb.Help().....	78
3.2.7.5 Metode Statik void BuildDb.PopulateDb(SQLiteConnection conn, List<EdictEntry> entries).....	78
3.2.7.6 Metode Statik int BuildDb.Main(string[] args).....	79
3.3 Komponen Pentransliterasi Kanji: GamaIme.dll.....	80
3.3.1 Tinjauan.....	80
3.3.2 Analisis Permasalahan.....	80
3.3.3 Analisis Kebutuhan Sistem.....	80
3.3.4 Mesin Transliterasi Kanji.....	81
3.3.4.1 Tinjauan.....	81
3.3.4.2 Penggunaan Romaji.....	84
3.3.4.3 Romanisasi yang Digunakan.....	85

3.3.4.4 Representasi Aturan.....	88
3.3.4.5 Aplikasi Aturan pada Kata Nyata.....	92
3.3.4.6 Algoritma Perantaraan.....	94
3.3.4.7 Algoritma Breadth-first Search.....	106
3.3.5 Mesin Transliterasi Romaji ke Kana dan Sebaliknya.....	107
3.3.5.1 Tinjauan.....	107
3.3.5.2 Algoritma.....	108
3.3.5.3 Implementasi.....	110
3.3.6 Kelas Transliterasi<S, D>.....	111
3.3.6.1 Tinjauan.....	111
3.3.6.2 Perancangan Kelas.....	111
3.3.6.3 Penggunaan.....	112
3.3.7 Kelas StringTransliterasi.....	116
3.3.7.1 Tinjauan.....	116
3.3.7.2 Perancangan Kelas.....	116
3.3.7.3 Metode Statik ParseRule(string rules).....	117
3.3.8 Kelas RomajitoKanaTransliterasi.....	117
3.3.8.1 Tinjauan.....	117
3.3.8.2 Perancangan Kelas.....	118
3.3.8.3 Contoh Penggunaan.....	118
3.3.9 Kelas HiraganaToRomajiTransliterasi.....	118
3.3.9.1 Tinjauan.....	119
3.3.9.2 Perancangan Kelas.....	119
3.3.9.3 Contoh Penggunaan.....	119
3.3.10 Kelas RuleInstance.....	120
3.3.10.1 Tinjauan.....	120
3.3.10.2 Perancangan Kelas.....	121
3.3.10.3 Ilustrasi Nilai Atribut.....	122
3.3.10.4 Metode string RuleInstance.ToString().....	125
3.3.10.5 Konstruktor RuleInstance().....	126
3.3.10.6 Konstruktor RuleInstance(string perfectMatch).....	126
3.3.10.7 Metode List<string> Replace(string dictionaryWord).....	126
3.3.11 Kelas Rule.....	130
3.3.11.1 Tinjauan.....	130
3.3.11.2 Perancangan Kelas.....	131
3.3.11.3 Metode Rule[] Load(string ruleFile).....	133
3.3.12 Kelas RestrictionChecker.....	133
3.3.12.1 Tinjauan.....	133
3.3.12.2 restriction.txt.....	135
3.3.12.3 Perancangan Kelas.....	137
3.3.12.4 Metode Statik void Load(string fileName).....	137
3.3.12.5 Contoh Penggunaan.....	139
3.3.13 Kelas KanjiTransliterasi.....	139
3.3.13.1 Tinjauan.....	139
3.3.13.2 Perancangan Kelas.....	139
3.3.13.3 Contoh Penggunaan.....	141

3.3.13.4	Konstruktor KanjiTransliterator(string ruleFileName, string restrictionFileName, string tagFileName, string dbFileName).....	141
3.3.13.5	Metode string[] KanjiTransliterator.Transliterate(string inputRomaji).....	142
3.3.13.6	Metode string[] FetchDb(List<List<RuleInstance>> instancesByLevel).....	142
3.4	Komponen Perantara Klien Server: transliterator.aspx.....	143
3.5	Komponen Aplikasi Web IME.....	144
3.5.1	Tinjauan.....	144
3.5.2	Analisis Permasalahan.....	144
3.5.3	Analisis Kebutuhan Sistem.....	145
3.5.4	Perancangan Antarmuka.....	145
3.5.5	Penanganan Keyboard.....	148
3.5.6	Meminta Transliterasi ke Server dengan AJAX.....	150
3.5.7	Implementasi.....	150
3.5.8	misc.js.....	150
3.5.9	transliterator.js.....	152
3.5.10	keyboard.js.....	153
3.5.11	ime.js.....	154
3.5.11.1	Diagram Kelas dan Atribut ime.....	154
3.5.11.2	Diagram Kelas dan Atribut imePopup.....	156
3.5.11.3	Inisialisasi ime.....	158
3.5.11.4	Penanganan Keyboard.....	159
3.5.11.5	Permintaan Transliterasi ke Server dan Caching.....	160
3.5.11.6	Fungsi imePopup.init.....	161
3.5.11.7	Fungsi imePopup.setHighlighted(index: Number).....	161
3.5.11.8	Fungsi imePopup.updateCachePriority().....	162
3.5.11.9	Fungsi imePopup.setConverted(str: String).....	162
3.5.11.10	Fungsi imePopup.setRomaji(str: String).....	163
BAB IV	PEMBAHASAN.....	164
4.1	Antarmuka Aplikasi Web IME.....	164
4.2	Keterbatasan Masukan Transliterasi.....	170
4.3	Kekurangan Algoritma Transliterasi.....	174
4.3.1	Kemampuan Mengurutkan Pilihan yang Terbatas.....	174
4.3.2	Kekurangan Mesin Transliterasi yang Berbasis Romaji.....	174
BAB V	KESIMPULAN DAN SARAN.....	177
5.1	Kesimpulan.....	177
5.2	Saran.....	177
DAFTAR	PUSTAKA.....	179

Daftar Gambar

Gambar 2.1 Kedudukan IME pada proses masukan.....	35
Gambar 2.2 Konversi “mita” pada IME.....	37
Gambar 2.3 Hasil konversi “mita”.....	37
Gambar 2.4 Tampilan kemungkinan “mita”.....	38
Gambar 2.5 Hasil konversi “mita” yang telah dikirim ke aplikasi.....	38
Gambar 2.6 Hello World di C#.....	51
Gambar 2.7 Contoh aplikasi ASP.NET.....	51
Gambar 2.8 Model aplikasi web klasik.....	53
Gambar 2.9 Model aplikasi web menggunakan AJAX.....	54
Gambar 3.1 Hubungan Komponen IME.....	57
Gambar 3.2 edict_allowed_tags.txt.....	65
Gambar 3.3 edict_tags_transform.txt.....	67
Gambar 3.4 edict_popularity_tags.txt.....	69
Gambar 3.5 Diagram Kelas EdictEntry.....	72
Gambar 3.6 Metode KatakanaToHiragana.....	73
Gambar 3.7 Pemisahan Arti pada Konstruktor EdictEntry.....	75
Gambar 3.8 Pengisian Bacaan dan Penulisan pada Konstruktor EdictEntry.....	76
Gambar 3.9 Pengisian Penanda pada Konstruktor EdictEntry.....	77
Gambar 3.10 Diagram Kelas BuildDb.....	78
Gambar 3.11 Penggunaan BEGIN dan QUERY pada PopulateDb.....	79
Gambar 3.12 Contoh Informasi pada suatu Rule.....	82
Gambar 3.13 Contoh Informasi pada suatu RuleInstance.....	82
Gambar 3.14 Contoh Informasi pada suatu RuleInstance yang diperoleh melalui Perantaraan.....	82
Gambar 3.15 Contoh Informasi pada suatu RuleInstance untuk Dirantarkan.....	83
Gambar 3.16 Contoh Informasi pada suatu RuleInstance hasil Perantaraan.....	83
Gambar 3.17 Pohon Perantaraan “tabenakat'ta”.....	84
Gambar 3.18 Algoritma Perantaraan bagian 1.....	95
Gambar 3.19 Ilustrasi Penanda yang tidak Cocok dalam Perantaraan.....	96
Gambar 3.20 Algoritma Perantaraan bagian 2.....	96
Gambar 3.21 Ilustrasi Kasus Perantaraan Kata Kosong yang Ingin Dicegah.....	97
Gambar 3.22 Algoritma Perantaraan bagian 3.....	97
Gambar 3.23 Ilustrasi Kasus Perantaraan yang Depannya Cocok.....	97
Gambar 3.24 Algoritma Perantaraan bagian 4.....	97
Gambar 3.25 Ilustrasi Kasus Perantaraan yang Belakangnya Cocok.....	98
Gambar 3.26 Algoritma Perantaraan bagian 5.....	98
Gambar 3.27 Pemberian Nilai srcTag pada nextInstance.....	98
Gambar 3.28 Algoritma Perantaraan bagian 6.....	98
Gambar 3.29 Penentuan Kata Dasar nextInstance.....	99
Gambar 3.30 Algoritma Perantaraan bagian 7.....	99
Gambar 3.31 Contoh Kasus yang Melanggar Syarat Fonologis Penanda.....	99
Gambar 3.32 Algoritma Perantaraan bagian 8.....	99
Gambar 3.33 Algoritma Perantaraan bagian 9.....	100

Gambar 3.34 Contoh Perantaian Bagian Depan Kana.....	100
Gambar 3.35 Algoritma Perantaian bagian 10.....	100
Gambar 3.36 Contoh Perantaian Bagian Depan Kanji.....	101
Gambar 3.37 Algoritma Perantaian bagian 11.....	102
Gambar 3.38 Algoritma Perantaian bagian 12.....	102
Gambar 3.39 Algoritma Perantaian bagian 13.....	103
Gambar 3.40 Contoh Penentuan Nilai endOrigKana pada Perantaian.....	103
Gambar 3.41 Algoritma Perantaian bagian 14.....	103
Gambar 3.42 Contoh Penentuan Nilai endNewKana pada Perantaian.....	104
Gambar 3.43 Algoritma Perantaian bagian 15.....	104
Gambar 3.44 Contoh Penentuan Nilai endNewKanji pada Perantaian karena rule	105
Gambar 3.45 Algoritma Perantaian bagian 16.....	105
Gambar 3.46 Contoh Penentuan Nilai endNewKanji pada Perantaian karena instance.....	105
Gambar 3.47 Contoh Pohon Aturan Transliterasi Kana.....	110
Gambar 3.48 Diagram Hubungan berbagai Kelas Pentransliterasi Kana.....	111
Gambar 3.49 Diagram Kelas Transliterator.....	112
Gambar 3.50 Diagram Kelas StringTransliterator.....	116
Gambar 3.51 Diagram Kelas RomajiToKanaTransliterator.....	118
Gambar 3.52 Contoh Penggunaan RomajiToKanaTransliterator.....	118
Gambar 3.53 Diagram Kelas HiraganaToRomajiTransliterator.....	119
Gambar 3.54 Contoh Penggunaan HiraganaToRomajiTransliterator.....	119
Gambar 3.55 Diagram Kelas RuleInstance.....	121
Gambar 3.56 Ilustrasi middle untuk soundMorph false.....	128
Gambar 3.57 Ilustrasi middle untuk soundMorph true.....	129
Gambar 3.58 Ilustrasi hasil infleksi untuk soundMorph false.....	130
Gambar 3.59 Ilustrasi hasil infleksi untuk soundMorph true.....	130
Gambar 3.60 Diagram Kelas Rule.....	131
Gambar 3.61 restriction.txt.....	136
Gambar 3.62 Diagram Kelas RestrictionChecker.....	137
Gambar 3.63 Contoh Penggunaan RestrictionChecker.....	139
Gambar 3.64 Diagram Kelas KanjiTransliterator.....	140
Gambar 3.65 Contoh Penggunaan Kelas KanjiTransliterator.....	141
Gambar 3.66 Perancangan Antarmuka Utama Aplikasi Web IME.....	146
Gambar 3.67 Perancangan Antarmuka Popup IME.....	146
Gambar 3.68 Fungsi next yang mengabaikan elemen teks.....	151
Gambar 3.69 Diagram Kelas ime.....	155
Gambar 3.70 Diagram Kelas imePopup.....	156
Gambar 3.71 Kode writeIme yang Menuliskan IME-nya.....	158
Gambar 3.72 Kode HTML Popup IME.....	159
Gambar 3.73 Pengecekan Cache setelah Respons AJAX diterima.....	161
Gambar 3.74 Perpindahan ke Mode selecting setelah Respons AJAX diterima.	161
Gambar 4.1 Antarmuka Utama Aplikasi Web IME.....	164
Gambar 4.2 Perilaku Kotak Teks saat IME belum Diaktifkan.....	165
Gambar 4.3 IME yang Aktif.....	165

Gambar 4.4 IME Aktif dengan Masukan 'watah'.....	166
Gambar 4.5 Masukan 'watashi' Dimasukkan ke Kotak Teks.....	166
Gambar 4.6 Pilihan Pertama Transliterasi Kanji 'torimasu'	167
Gambar 4.7 Kotak Pilihan Transliterasi Alternatif untuk 'torimasu'.....	168
Gambar 4.8 Masukan 'test' yang Ditransliterasi Paksa Menjadi Huruf Latin Lebar Penuh.....	169
Gambar 4.9 Menyalin Tulisan untuk Aplikasi Lain.....	169

Daftar Tabel

Tabel 2.1 Hiragana Dasar.....	10
Tabel 2.2 Hiragana dengan Dakuten.....	11
Tabel 2.3 Hiragana dengan Handakuten.....	12
Tabel 2.4 Hiragana dengan Youon.....	12
Tabel 2.5 Hiragana untuk Suara Luar.....	14
Tabel 2.6 Katakana Dasar.....	17
Tabel 2.7 Katakana dengan Dakuten.....	17
Tabel 2.8 Katakana dengan Handakuten.....	17
Tabel 2.9 Katakana dengan Youon.....	18
Tabel 2.10 Katakana untuk Suara Luar.....	18
Tabel 2.11 Contoh Bacaan On Beberapa Kanji.....	23
Tabel 2.12 Contoh Bacaan Kun Beberapa Kanji.....	24
Tabel 2.13 Perbedaan konsistensi fonologis Hepburn, Nihon-shiki, dan Kunrei-shiki.....	27
Tabel 2.14 Padanan bahasa Indonesia beberapa infleksi pada bahasa Jepang.....	30
Tabel 2.15 Ilustrasi penulisan “tobu” dan berbagai infleksinya.....	30
Tabel 2.16 Beberapa infleksi “kuru”.....	31
Tabel 2.17 Infleksi lampau berbagai jenis verba.....	32
Tabel 2.18 Hiragana pada Unicode.....	40
Tabel 2.19 Katakana Lebar Penuh pada Unicode.....	42
Tabel 2.20 Katakana Lebar Setengah pada Unicode.....	42
Tabel 2.21 Penanda Kelas Kata EDICT.....	45
Tabel 3.1 Field-Field Tabel edict.....	59
Tabel 3.2 Contoh Masukan dan Keluaran Transliterasi Karakter.....	113
Tabel 3.3 Tabel Perubahan Keadaan Popup IME.....	147
Tabel 3.4 Tombol Konversi Paksa IME.....	148
Tabel 3.5 Objek Tabel Kode Karakter Keyboard pada keyboard.js.....	153
Tabel 4.1 Tombol Navigasi Kotak Pilihan.....	167
Tabel 4.2 Tombol Konversi Paksa IME.....	168
Tabel 4.3 Contoh Pemecahan Kalimat.....	171

INTISARI

Pengembangan Input Method Editor Bahasa Jepang Berbasis Web

oleh

AGRO RACHMATULLAH
03/165399/PA/09374

Bahasa Jepang menggunakan ribuan karakter untuk sistem menulisnya. Untuk memungkinkan penulisan ribuan karakter tersebut dengan *keyboard* standard, terdapat program yang disebut *Input Method Editor* (IME). Pada umumnya IME berada di tingkat sistem operasi, dan akan memodifikasi masukan yang datang dari *keyboard* sebelum dikirim ke aplikasi yang bersangkutan.

Di Jepang, komputer umum (misal komputer perpustakaan dan warnet) dapat dipastikan memiliki IME. Kemudahan ini tidak berlaku di negara luar, misalnya di Indonesia. Ini menimbulkan kesulitan bagi orang yang berada di luar Jepang namun perlu menulis Jepang di komputer umum. Contohnya adalah orang Jepang yang sedang berkunjung ke luar negeri dan pelajar bahasa Jepang yang ingin berkomunikasi di Internet menggunakan bahasa Jepang.

Untuk mengatasi masalah tersebut dikembangkan suatu IME bahasa Jepang berbasis web. Dengan IME berbasis web, pengguna dapat mengunjungi situsnya dan melakukan pengetikan bahasa Jepang pada aplikasi web tersebut. Di sisi klien, IME menggunakan teknologi-teknologi standard web yaitu HTML, CSS, JavaScript, dan AJAX. Di sisi server, digunakan C#, ASP.NET, SQLite, dan EDICT. Algoritma intinya menggunakan *breadth-first search* untuk merantailkan aturan tata bahasa. Skripsi ini akan menjelaskan arsitektur dan cara kerja IME yang dikembangkan tersebut.

Kata Kunci: Bahasa Jepang, *Input Method Editor*, IME, HTML, CSS, JavaScript, AJAX, C#, .NET, ASP.NET, SQLite, EDICT

ABSTRACT

The Development of a Web-based Japanese Input Method Editor

by

AGRO RACHMATULLAH
03/165399/PA/09374

The Japanese language uses thousands of characters for its writing system. Programs called IME (Input Method Editor) enables the input of those characters using a standard keyboard. Usually, IME runs on the operating system level, modifying input from the keyboard before sending it to user-level applications.

In Japan, public computers (e.g., library, internet cafe) can be assured to have IME installed. This convenience doesn't hold overseas, for example in Indonesia. It presents difficulties for those outside of Japan but needs to write Japanese in public computers. Some examples are a Japanese traveling overseas and a student of the Japanese language wishing to communicate on the internet using Japanese.

To tackle the problem, a web-based Japanese IME is developed. With a web-based IME, user can visit its web site and type Japanese on the web application. On the client side, the IME uses standard web technologies which are HTML, CSS, JavaScript, and AJAX. On the server side, C#, ASP.NET, SQLite, and EDICT are used. The core transliteration algorithm uses breadth-first search to chain grammar rules. This paper will explain the architecture and workings of the developed IME.

Keywords: Japanese Language, Input Method Editor, IME, HTML, CSS, JavaScript, AJAX, C#, .NET, ASP.NET, SQLite, EDICT