

Kamus Dan *Learning Center* Bahasa Jepang Untuk Persiapan *Japanese Language Proficiency Test (JLPT)* Menggunakan Algoritma Aho Corassick Dan Hu-Moments Dan L1-Metric

Japanese Language Dictionary and Learning Center for Japanese Language Proficiency Test (JLPT) Preparation Using Aho-Corasick Algorithm, Hu Moments, and L1 Metric

Muhammad Rivaldi Hamzah Lubis¹, Budi Triandi²

^{1,2}Program Studi Informatika Universitas Potensi Utama, Jl.K.L Yos Sudarso KM 6.5 Tj.Mulia, Medan, 20241, Indonesia
E-mail: rivaldilubis05@gmail.com¹, Budi Triandi²

ABSTRAK

Bahasa Jepang merupakan salah satu bahasa asing yang paling kompleks dan menantang untuk dipelajari, dengan *Japanese Language Proficiency Test (JLPT)* digunakan sebagai standar internasional untuk mengukur tingkat kemahiran berbahasa Jepang. Penelitian ini bertujuan untuk mengembangkan sebuah prototipe aplikasi pembelajaran bahasa Jepang yang dirancang untuk mendukung persiapan menghadapi ujian *JLPT*. Aplikasi ini mengintegrasikan kamus kosakata (*goi*) dan kamus kanji, dengan penerapan algoritma Aho-Corasick untuk pencarian string yang efisien dan akurat. Selain itu, aplikasi ini menyediakan fitur pelatihan menulis karakter Jepang, menggunakan teknik pencocokan citra berbasis algoritma Hu Moment Invariant untuk membandingkan hasil tulisan pengguna dengan karakter referensi. Sebagai tambahan, disediakan pula modul simulasi ujian berbasis *Computer Assisted Test (CAT)* guna meningkatkan keterampilan pengguna dalam menghadapi format ujian yang sesungguhnya. Melalui penggabungan berbagai fitur ini, prototipe yang dikembangkan diharapkan dapat meningkatkan efektivitas pembelajaran, memperkuat keterampilan berbahasa Jepang, serta mendukung keberhasilan pengguna dalam menghadapi ujian *JLPT*.

Kata Kunci: Bahasa Jepang, *JLPT*, Pembelajaran Bahasa, Aho-Corasick, Hu Moment Invariant, *Computer Assisted Test*

ABSTRACT

The Japanese language is recognized as one of the most complex and challenging foreign languages to master, with the *Japanese Language Proficiency Test (JLPT)* serving as an international standard for assessing language proficiency. This research aims to develop a prototype of a Japanese language learning application designed to assist users in preparing for the *JLPT* examination. The application integrates a vocabulary (*goi*) dictionary and a Kanji dictionary, utilizing the Aho-Corasick algorithm for efficient and accurate string-based search. Additionally, the application offers a character writing training feature, employing image matching techniques based on the Hu Moment Invariant algorithm to compare user input with reference characters. Furthermore, a *Computer Assisted Test (CAT)*-based simulation module is provided to enhance users' familiarity with the actual exam format. Through the integration of these features, the developed prototype is expected to improve learning effectiveness, strengthen Japanese language skills, and support users' success in passing the *JLPT*.

Keywords: Japanese Language, *JLPT*, Language Learning, Aho-Corasick, Hu Moment Invariant, *Computer Assisted Test*

Info Artikel :

Disubmit: 2 Juni 2025

Direview: 25 Juli 2025

Diterima : 11 Agustus 2025

Copyright © 2025 - Journal UPU. All rights reserved.

1. PENDAHULUAN

Bahasa Jepang merupakan bahasa terbanyak ke-13 di seluruh dunia, dengan jumlah penutur bahasa sedikitnya 125,3 juta orang. Selain dari banyak nya penutur, bahasa Jepang juga menjadi semakin populer di seluruh dunia melalui *Anime*, *Japan-Drama (J-Drama)*, dan lagu-lagu Jepang (*J-Songs*). Lebih lagi,

sejak beberapa tahun terakhir, mengingat semakin menurunnya populasi asli dari bangsa Jepang dengan cukup ekstrem, Negara Jepang mulai membuka banyak peluang untuk warga negara asing untuk bekerja di Jepang, dengan syarat memiliki kemampuan dasar bahasa Jepang yang diukur menggunakan instrumen kemahiran bahasa Jepang, yang dikenal dengan nama *Japanese Language Proficiency Test* (JLPT). Umumnya, perusahaan mensyaratkan pekerja luar negeri Jepang untuk setidaknya lulus JLPT level N4. Tulisan Jepang sendiri terdiri dari 4 jenis huruf utama, yaitu *Hiragana*, *Katakana*, *Kanji*, *Romaji*. Masing-masing *Hiragana* dan *Katakana* terdiri dari 46 huruf dasar yang memiliki pelafalan yang saling paralel, di mana setiap huruf merupakan 1 silabel. Kemudian, pada *hiragana* dan *katakana* sendiri terdapat tambahan sistem penulisan diakritik *dakuten* (*ten-ten*), dan *handakuten* (*maru*), yang digunakan untuk merubah vokalisasi konsonan pada huruf-huruf tertentu. Sehingga secara total, terdapat 104 silabel untuk masing-masing *Hiragana* dan *Katakana*. (Hadamitzky, 2012)

Huruf *Kanji* merupakan huruf yang diserap dari sistem penulisan bahasa Cina. Setiap huruf *kanji* umumnya memiliki 2 cara baca yang disebut dengan *onyomi* dan *kunyomi*. *Onyomi* adalah pelafalan yang diambil dari cara bahasa Cina untuk huruf tersebut yang disesuaikan dengan gaya bahasa Jepang, sementara *kunyomi* adalah cara baca asli *kanji* tersebut dalam bahasa Jepang. Walaupun sudah memiliki 2 cara baca, namun tidak jarang 1 huruf *kanji* memiliki lebih dari 1 cara baca *onyomi*, dan lebih dari 1 baca *kunyomi*. Misalnya *kanji* lahir/kehidupan 生 memiliki 17 cara baca umum, yaitu 3 *onyomi* (*sei*, *shou*, *jou*) dan sisanya *kunyomi* (*i*, *u*, *ha*, *o*, *nama*, *na*, dll). Namun demikian, walaupun memiliki banyak cara baca, umumnya setiap *kanji* ini akan memiliki pelafalan yang tetap sesuai dengan pasangan karakter sebelum ataupun sesudahnya, baik *hiragana* ataupun *kanji*. (Tsujimura, 1996)

Ujian JLPT menguji lima bagian utama: Moji (kosakata dalam kanji), Go-I (penggunaan kata dan kolokasi), Bunpou (tata bahasa), Dokkai (membaca), dan Choukai (mendengarkan). Tidak seperti IELTS, JLPT tidak menguji menulis dan berbicara. JLPT dibagi menjadi lima level: N5 (paling dasar) sampai N1 (paling sulit), dan setara dengan level CEFR dari A1 hingga C1. Semakin tinggi levelnya, semakin banyak huruf kanji, kosakata sulit, dan pola kalimat kompleks yang digunakan. Di level rendah, kata-kata sulit biasanya ditulis dalam *hiragana* atau dibantu dengan *furigana* (bacaan huruf kecil di atas kanji). Tapi di level tinggi, *furigana* jarang digunakan, sehingga jika peserta lupa bacaan atau arti kanji, mereka bisa kesulitan memahami soal. (JEES, 2009)

Supaya dapat memberikan pengalaman belajar yang baik, sehingga calon peserta JLPT dapat menguasai materi, kosakata, dan huruf *kanji* dengan sangat baik, perlu untuk dibuatnya suatu aplikasi yang berisikan kamus (*Go-I*) dan *Kanji* yang telah di grup dengan baik (sesuai panduan buku *kanji* bahasa Jepang), yang memungkinkan pelajar tersebut dapat mencari kosakata Bahasa Jepang secara cepat dan efisien, baik menggunakan huruf *kanji*, *romaji*, ataupun *hiragana*, dan *katakana*. Selain itu perlu juga ditambahkan fitur pencarian huruf *kanji* menggunakan suatu kanvas digital, di mana pelajar dapat melakukan pencarian huruf *kanji*, dengan cara menggambarkan huruf *kanji* tersebut pada kanvas tersebut. Selanjutnya kanvas ini juga dapat digunakan sebagai media latihan menulis huruf *kanji* tertentu, sehingga diharapkan pelajar tersebut dapat memiliki pemahaman komprehensif terhadap huruf dan kosakata Bahasa Jepang. Oleh karena itu, penulis mengangkat judul: **Kamus dan Learning Center Bahasa Jepang Untuk Persiapan Japanese Language Proficiency Test (JLPT) Menggunakan Algoritma Aho-Corassick (untuk Dictionary dan Hu-Moment dan L1-Metric (Untuk Pengenalan Huruf Kanji).**

2. METODE

Pada tahap ini dilakukan dengan mempelajari teori dasar yang mendukung penelitian, pencarian dan pengumpulan data-data yang dibutuhkan, serta perancangan sistem yang akan dihasilkan sebagai bentuk *output* dari penelitian ini.

A. Metode Pengumpulan Data

Penelitian ini memerlukan data-data berikut, yaitu: daftar kosakata bahasa Jepang, dataset gambar huruf-huruf Jepang, materi JLPT, dan soal-soal JLPT. Sehingga untuk mendapatkan data-data tersebut maka pada penelitian ini akan dilakukan sejumlah metode pengumpulan sebagai berikut

1. Studi Literatur

Studi literatur akan digunakan untuk bisa mendapatkan daftar kosakata bahasa Jepang, daftar huruf-huruf, dari materi JLPT.

2. Dokumentasi

Teknik dokumentasi akan digunakan pada penelitian ini untuk mengumpulkan soal-soal dari berbagai referensi soal JLPT dari berbagai situs yang dapat ditemukan melalui pencarian Google, dan juga dari Youtube pembelajaran JLPT. Selain itu dokumentasi juga digunakan untuk mengumpulkan dataset gambar dari huruf-huruf Jepang dari berbagai sumber guna mendapatkan gambaran variasi dari cara penulisan huruf tersebut.

B. Alur Penelitian

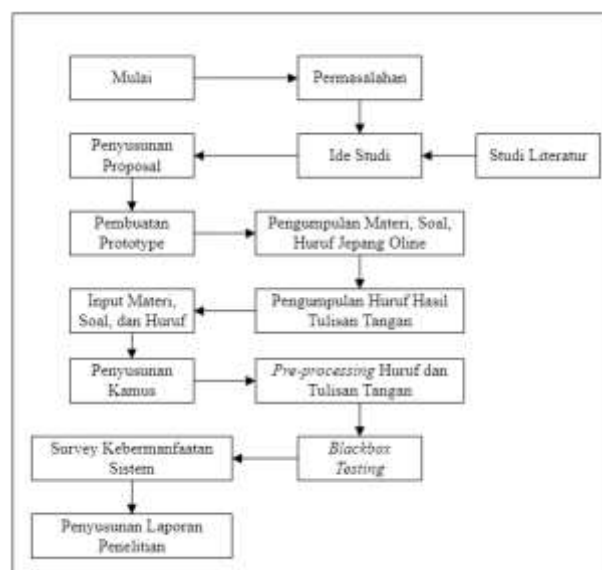


Diagram 1. Alur Penelitian

Diagram di atas menggambarkan alur utama dari penelitian ini. Penelitian dimulai dengan mengidentifikasi permasalahan, yaitu kesulitan dalam persiapan ujian JLPT yang disebabkan oleh kompleksitas bahasa Jepang. Dari situ, dirumuskan hipotesis dan metode penelitian, dengan gagasan utama berupa prototype yang menyediakan fasilitas learning center dan kamus bahasa Jepang yang dilengkapi flash card huruf dan kosakata. Untuk mematangkan gagasan ini, dilakukan studi literatur terkait metode pembelajaran bahasa Jepang. Selanjutnya disusun proposal penelitian sebagai dasar pengembangan prototype. Setelah prototype dibangun, dikumpulkan data yang dibutuhkan seperti materi belajar, daftar huruf, kosakata, dan soal-soal. Karena penelitian ini juga mencakup pelatihan dan pencarian karakter huruf Jepang melalui algoritma image-matching, maka dikumpulkan berbagai bentuk tulisan dari internet dan tulisan tangan. Data huruf yang telah dikumpulkan kemudian diproses dan dijadikan dataset untuk pelatihan. Kosakata yang telah dikompilasi disusun menjadi kamus berbasis flash card. Prototype kemudian diuji menggunakan metode black-box testing, dilanjutkan dengan survei kepada pengguna untuk menilai efektivitas aplikasi dalam mendukung persiapan ujian JLPT. Hasil dari pengujian dan survei diolah untuk mendapatkan nilai akhir penelitian, dan penelitian ini ditutup dengan penyusunan laporan akhir.

C. Metode Pengujian Sistem

Berikut adalah skema pengolahan dan pengujian data pada penelitian ini:

1. Data penelitian berupa image yang didapatkan masing-masing akan disimpan ke database dengan terlebih dahulu melalui tahap pre-processing mengikuti algoritma.
2. Untuk masing-masing data hasil pre-processing dan soal-soal yang ditemukan kemudian akan dipastikan value tujuannya / kunci jawabannya.
3. Output dari White-Box testing untuk string-matching akan dihitung 1 (jika sesuai), dan 0 (jika tidak sesuai). Sehingga akan diperoleh overall score dari string-matching dalam bentuk persentase.
4. Output dari White-Box Testing untuk image-matching, akan dihitung 2 nilai. Pertama, akan dihitung overall score seperti pada poin 3 di atas. Kedua akan dihitung akurasi maksimal untuk masing-masing kasus uji, di mana kemudian akan dihitung rata-rata akurasi kecocokan image pada kasus uji.

D. Diagram Perancangan Sistem

1. Use Case Diagram

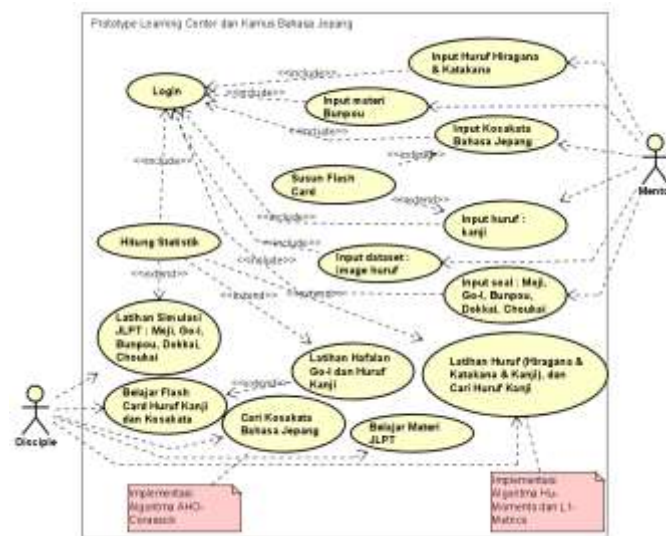


Diagram.2 Use Case Diagram

Aplikasi ini menyediakan dua jenis fitur utama untuk pengguna Disciple: fitur input data dan fitur pembelajaran. Fitur input digunakan untuk memasukkan huruf Jepang (Hiragana, Katakana, Kanji), gambar dataset, kosakata, materi grammar, serta soal-soal latihan JLPT ke dalam sistem. Sementara itu, fitur pembelajaran memungkinkan pengguna mempelajari materi JLPT, kosakata, dan huruf Kanji melalui flash card, serta berlatih dengan soal-soal secara acak. Pengguna juga dapat melatih kemampuan menulis huruf Jepang dan mencari karakter Kanji dengan menggambar langsung, menggunakan algoritma pencocokan citra. Untuk pencarian kosakata berbasis teks, digunakan algoritma Aho-Corasick. Di akhir, pengguna dapat melihat perkembangan belajarnya melalui fitur statistik.

2. Activity Diagram

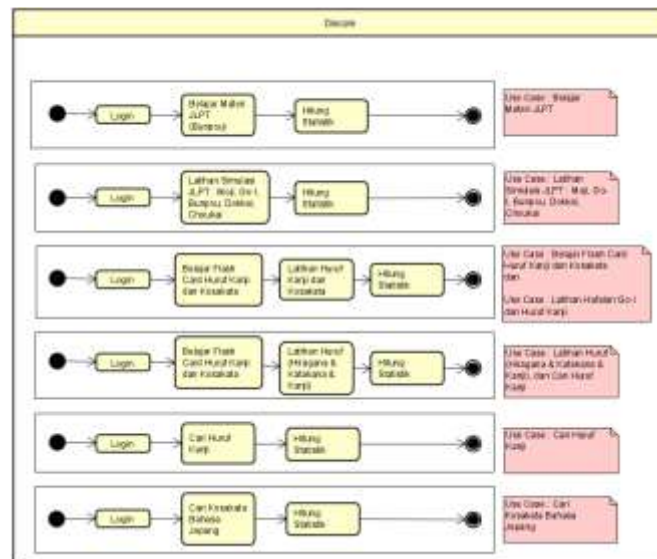


Diagram 3. Activity Diagram

Activity Diagram dengan aktor Disciple menggambarkan secara runtut perilaku yang dilakukan oleh pengguna dalam aplikasi. Untuk dapat mengakses seluruh menu yang tersedia, aktor terlebih dahulu harus melakukan login ke dalam sistem. Setelah berhasil masuk, aktor memiliki beberapa pilihan aktivitas, yaitu mempelajari materi JLPT, belajar melalui flash card, mencari kosakata, serta mencari huruf kanji. Setelah mempelajari materi JLPT, aktor dapat melanjutkan dengan latihan simulasi JLPT. Jika aktor memilih belajar melalui flash card, maka ia dapat melanjutkan dengan melatih penulisan huruf yang telah dipelajari serta menghafal kosakata. Di akhir proses pembelajaran, aktor dapat mengakses menu untuk melihat statistik perkembangan selama menggunakan aplikasi.

3. Class Diagram

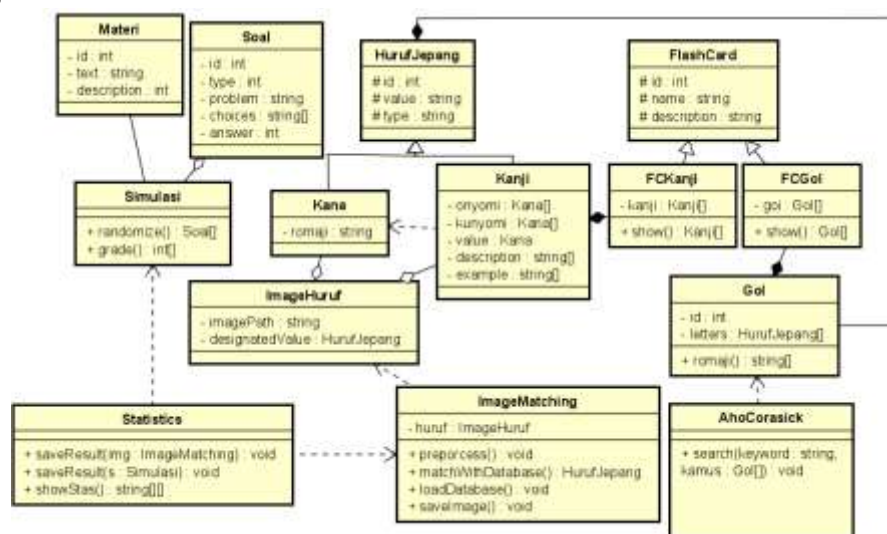


Diagram 4. Class Diagram

Prototype aplikasi ini terdiri dari sejumlah class yang dibentuk berdasarkan hasil analisis permasalahan. Class utama adalah HurufJepang, sebagai superclass dari Kana dan Kanji, yang merepresentasikan huruf dalam bahasa Jepang. Class Kana digunakan untuk huruf hiragana dan katakana, sedangkan Kanji menyimpan huruf kanji beserta daftar onyomi dan kunyomi yang terbentuk dari kumpulan Kana. Class Gol merepresentasikan kosakata Jepang yang terdiri dari urutan HurufJepang (Kanji atau Kana). ImageHuruf menyimpan gambar huruf untuk kebutuhan image-

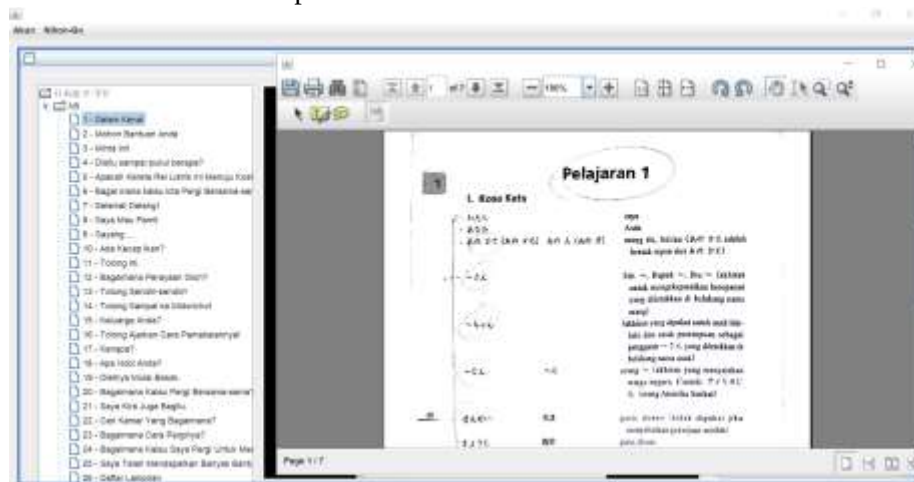
matching, dengan satu huruf Jepang yang ditugaskan oleh Mentor. Simulasi menyimpan latihan soal JLPT dalam bentuk instance dari class Soal. FlashCard merupakan superclass dari FCKanji dan FCCGoI, yang berisi grup Kanji atau kosakata berdasarkan kategori tertentu. Selain itu, terdapat class ImageMatching yang mengimplementasikan algoritma pencocokan gambar, AhoCorasick untuk pencocokan string, dan Materi untuk menyimpan materi JLPT.

3. HASIL DAN PEMBAHASAN

3.1. Hasil

Berikut adalah tampilan aplikasi sesuai dengan rancangan dan analisis sebelumnya.

1. Tampilan Halaman Materi Bunpou



Gambar 1. Halaman Materi Bunpou

Halaman Materi Bunpou ini adalah salah satu menu yang dapat diakses oleh *user* dengan otoritas akses sebagai *Disciple*. Materi disusun dalam bentuk *tree*. Kemudian materi yang diklik akan tampil ke dalam halaman tersebut. Sesuai pada *Form Input Materi Bunpou* di mana materi dapat berupa gambar, PDF, audio, video, maupun *web page*, maka pada halaman ini, materi-materi tersebut dikelompokkan sesuai dengan jenis *file* nya. Untuk materi berupa gambar akan langsung ditampilkan di bagian tengah halaman, sementara untuk jenis yang lain pada *proto-type* ini disediakan kotak dialog tersendiri yang dapat menampilkan isi dari konten tersebut. Kotak dialog ada sejumlah 5 macam yang saling berbeda sesuai dengan jenis *file* yang sedang ditampilkan.

2. Tampilan Halaman Flash Card Goi



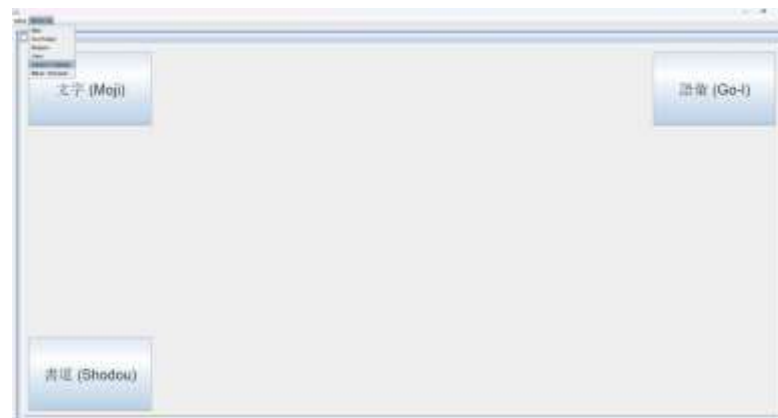
Gambar 2 Halaman Flash Card Goi

Halaman Flash Card Goi ini adalah salah satu menu yang dapat diakses oleh *user* dengan otoritas akses sebagai *Disciple*. *Flash Card* yang ditampilkan merupakan hasil pengelompokan yang dilakukan oleh *Mentor* pada *Form Pembentukan Flash Card* untuk kategori *Goi*. Setiap *flash card* yang diklik, akan memunculkan semua *Goi* yang dikelompokkan pada *flash card* tersebut dalam suatu kotak dialog. Kemudian setiap *Goi* yang muncul pada kotak dialog tersebut dapat diklik untuk memunculkan detil dari kosakata tersebut. Detil dari kosakata yang diklik akan ditampilkan dalam bentuk kotak dialog. Berikut adalah tampilan kotak dialog dari rincian kosakata tersebut.



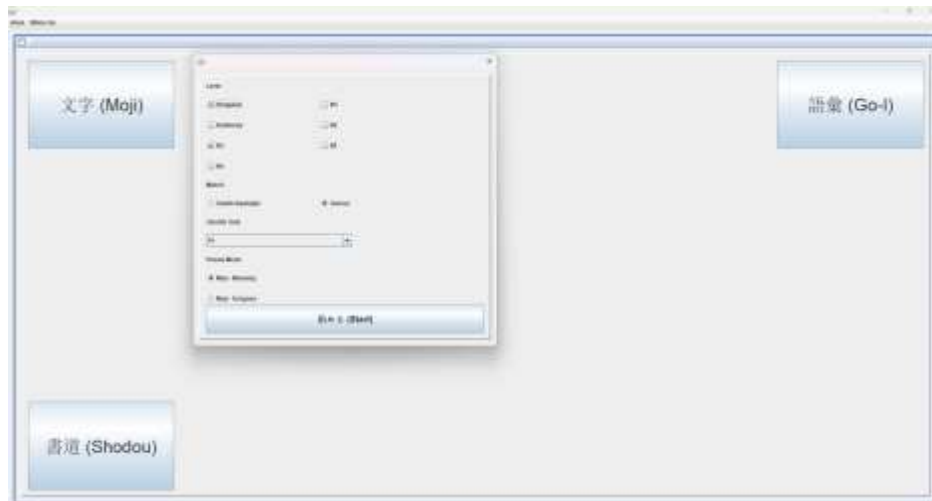
Gambar 3. Halaman Flash Card Goi

3. Tampilan Halaman Latihan



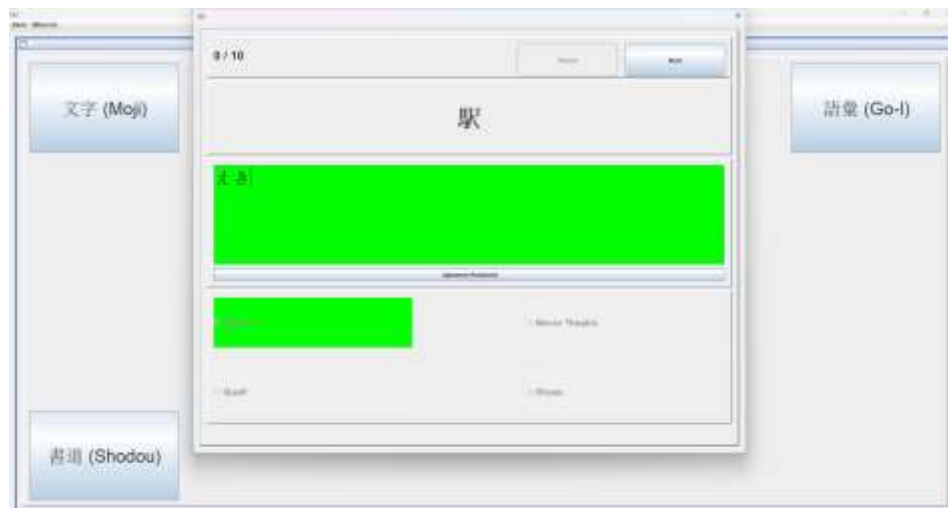
Gambar 4. Halaman Latihan

Halaman Latihan ini adalah salah satu menu yang dapat diakses oleh *user* dengan otoritas akses sebagai *Disciple*. Pada halaman ini terdapat 3 menu utama yang dapat dipilih oleh *Disciple*, yaitu : *Moji*, *Goi*, dan *Shodou*. *Moji* adalah menu yang digunakan untuk menguji hafalan huruf Jepang, termasuk *Hiragana*, *Katakana*, dan *Kanji*. *Goi* adalah menu untuk melatih hafalan kosakata bahasa Jepang. Sementara *Shodou* adalah menu untuk latihan menulis. Untuk memulai latihan *Moji*, *Disciple* hanya perlu untuk memilih menu *Moji* tersebut.



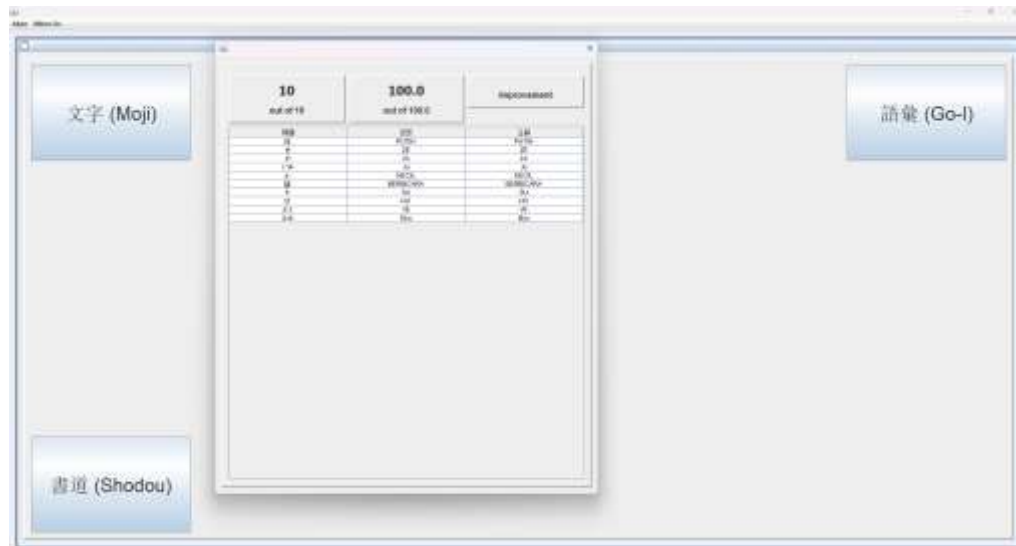
Gambar 5. Kotak Dialog Opsi Latihan

Setelah memilih opsi latihan, maka *Disciple* dapat memulai latihan tersebut. Latihan akan dilakukan pada kotak dialog khusus, di mana setiap huruf akan diujikan satu-per-satu. *Disciple* harus menjawab 1 huruf terlebih dahulu untuk dapat lanjut mencoba huruf yang lain. Setelah dipilih sistem akan menampilkan jawaban yang benar untuk huruf tersebut. Jika huruf tersebut salah, maka *disciple* akan mengulangnya lagi setelah semua huruf yang belum dicoba telah diujikan. Berikut adalah tampilan kotak dialog latihan *Moji*.



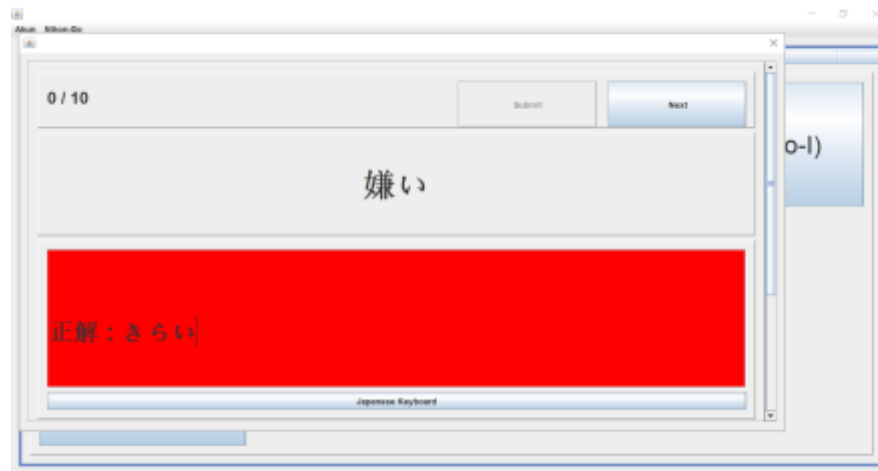
Gambar 6 Kotak Dialog Latihan Hafalan Huruf

Setelah semua huruf telah selesai dikerjakan dengan benar, maka akan muncul raport berbentuk kotak dialog. Perlu diperhatikan bahwa tujuan utama dari halaman hafalan ini adalah agar *Disciple* dapat mengingat semua karakter yang diujikan, sehingga *Disciple* akan dipaksa oleh sistem untuk menyelesaikan latihan tersebut dengan benar terlebih dahulu baru dapat melihat raport latihannya. Pada raport ini karakter huruf yang dihitung benar hanya karakter yang berhasil dijawab benar pada pertama kali menjawab. Berikut adalah tampilan kotak dialog raport hasil latihan.



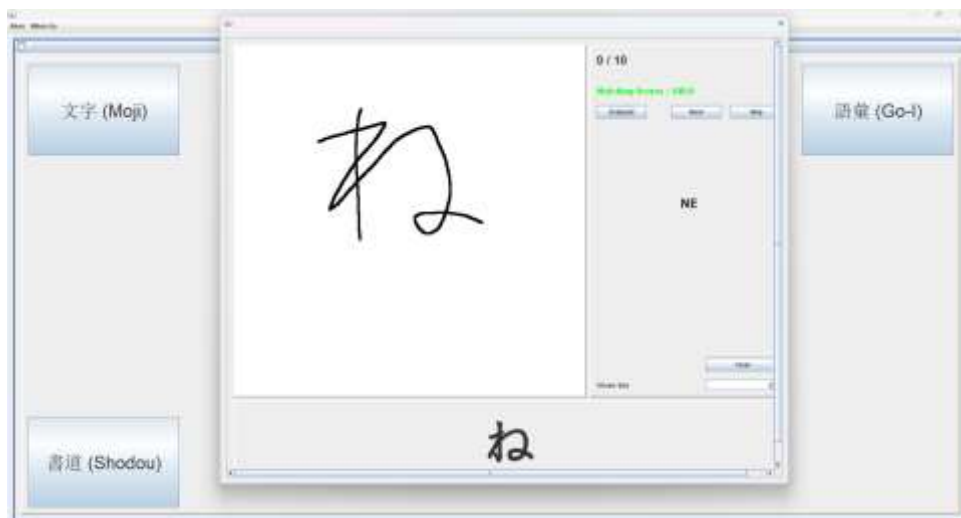
Gambar 7 Kotak Dialog Rapot Latihan

5. Tampilan Latihan Hafalan



Gambar 8 Halaman Latihan Hafalan Goi

Halaman Latihan Hafalan Huruf ini adalah salah satu fitur yang dapat diakses oleh user dengan otoritas akses sebagai *Disciple*. Opsi latihan Goi ini adalah menu yang dapat diakses pada halaman yang sama dengan latihan Moji sebelumnya. Sebagaimana pada latihan hafalan huruf, user akan dihadapkan pada kotak dialog untuk memilih berapa jumlah kosakata yang akan dilatih pada percobaan tersebut. Kemudian kosakata tersebut akan diujikan satu-per-satu.



Gambar 9 Halaman Latihan Menulis Huruf

Halaman *Latihan Menulis Huruf* ini adalah salah satu fitur yang dapat diakses oleh user dengan otoritas akses sebagai *Disciple*. Latihan menulis huruf ini dijalankan dengan mengklik tombol *Shodou* pada halaman latihan. Seperti halnya latihan huruf dan kosakata, *Disciple* akan terlebih dahulu dihadapkan pada sebuah kotak dialog untuk memilih berapa jumlah huruf yang akan dilatih dan teknis latihan menulis seperti apa yang akan diujikan (menjiplak atau mengingat). Kemudian huruf tersebut akan diujikan satu-per-satu. *Disciple* harus menggambarkan huruf yang dimaksud pada kanvas yang disediakan sistem. Sistem selanjutnya akan mengevaluasi seberapa mirip huruf tersebut dengan huruf yang ada pada dataset sistem. *Diciple* harus mendapatkan nilai minimal 80% untuk dapat dianggap telah berhasil menuliskan huruf tersebut dengan benar.

6. Tampilan Halaman Simulasi JLPT



Gambar 13 Halaman Simulasi JLPT

Halaman *Simulasi JLPT* ini adalah salah satu fitur yang dapat diakses oleh user dengan otoritas akses sebagai *Disciple*. Simulasi JLPT ini menggunakan format yang sama seperti ujian asli JLPT di mana *Moji* dan *Goi* diujikan bersamaan dan yang pertama diuji, kemudian dilanjutkan dengan *Bunpou* dan *Dokkai*, dan terakhir *Choukai*. Hasil ujian akan ditampilkan pada suatu *board dialog* yang berisikan skor masing-masing komponen. Perlu diperhatikan bahwa, penilaian pada jlpt dibagi ke dalam 2 kategori untuk N5

dan N4, yaitu *Gengo Chishiki* (Pengetahuan Bahasa), dan *Choukai* (Pendengaran). Sementara untuk N3, N2, dan N1 penilaian dibagi ke dalam 3 kategori, yaitu *Gengo Chishiki* (Pengetahuan Bahasa), *Dokkai* (Membaca), dan *Choukai* (Pendengaran). Sub-kategori *Gengo Chishiki* pada level N5 dan N4 terdiri dari sub-topik *Moji*, *Goi*, *Bunpou*, dan *Dokkai*, di lain sisi pada level N3, N2, dan N1 *Gengo Chishiki* terdiri atas sub-topik *Moji*, *Goi*, dan *Bunpou* saja tanpa *Dokkai*.

7. Tampilan Halaman Kamus



Gambar 14 Halaman Kamus Goi dan String-Matching

Halaman Kamus Goi ini adalah salah satu fitur yang dapat diakses oleh *user* dengan otoritas akses sebagai *Disciple*. Kamus *Goi* ini merupakan bentuk leksikografis dari data yang diinputkan pada *Form Input Goi*. Daftar kosakata diurutkan berdasarkan nomor ASCII dari karakter-karakter penyusun kosata tersebut. Format pada kamus ini menyesuaikan format yang umum digunakan pada kamus-kamus digital bahasa jepang, di mana bagian kamus kosakata ini dilengkapi pencarian teks (*String-Matching*) yang akan dicocokkan ke salah satu dari penulisan asli (kombinasi *kanji* dan *hiragana / katakana*), *furigana*, *romaji*, dan arti dari kosakata tersebut. Pencarian teks ini mengimplementasikan algoritma Aho-Corasick yang mengakomodasikan pencarian menggunakan huruf latin, *hiragana*, *katakana*, dan *kanji*.



Gambar 15 Halaman Kamus Kanji dan Image-Matching

Halaman Kamus Kanji ini adalah salah satu fitur yang dapat diakses oleh user dengan otoritas akses sebagai *Disciple*. Kamus Kanji ini merupakan bentuk leksikografis dari data yang diinputkan pada *Form Input Kanji*. Daftar huruf kanji diurutkan berdasarkan nomor ASCII dari kanji tersebut. Format pada kamus ini menyesuaikan format yang umum digunakan pada kamus-kamus digital bahasa Jepang. Kamus kanji ini dilengkapi pencarian huruf kanji dengan menggambarannya pada suatu kanvas (*image-matching*). Pencarian huruf kanji menggunakan gambar tersebut mengimplementasikan algoritma *Hu-Moment Invariant*.

3.2. Pembahasan

1. Black-Box Testing

Pengujian sistem ini digunakan untuk memastikan semua fitur yang dirancang untuk sistem ini telah berhasil diimplementasikan seluruhnya. Metode yang digunakan untuk menguji fungsionalitas *proto-type* aplikasi ini adalah *Blackbox Testing*. Berikut adalah tabel hasil pengujian *Black-Box Testing* sistem. Dari 56 fitur yang dirancang, sistem berhasil mengimplementasikan ke-56 fitur tersebut sesuai dengan yang diharapkan, sehingga keberhasilan implementasi fungsionalitas sistem adalah **100%**.

Tabel 1. Black-Box Testing Fungsionalitas Sistem

No	User	Backlog	Fitur	Hasil
1	Mentor	Login	Login	[✓] Sukses
2	Mentor	Kana	Tambah Kana	[✓] Sukses
			Edit Kana	[✓] Sukses
			Hapus Kana	[✓] Sukses
			Input Dataset Gambar Kana	[✓] Sukses
			Pencarian Kana	[✓] Sukses
3	Mentor	Kanji	Tambah Kanji	[✓] Sukses
			Edit Kanji	[✓] Sukses
			Hapus Kanji	[✓] Sukses
			Input Dataset Gambar Kanji	[✓] Sukses
			Pencarian Kanji	[✓] Sukses
4	Mentor	Goi	Tambah Goi	[✓] Sukses
			Edit Goi	[✓] Sukses
			Hapus Goi	[✓] Sukses
			Pencarian Goi	[✓] Sukses
5	Mentor	Flash Card Moji	Tambah Flash Card Moji	[✓] Sukses
			Edit Flash Card Moji	[✓] Sukses
			Hapus Flash Card Moji	[✓] Sukses
			Pencarian Flash Card Moji	[✓] Sukses
6	Mentor	Flash Card Goi	Tambah Flash Card Moji	[✓] Sukses
			Edit Flash Card Moji	[✓] Sukses
			Hapus Flash Card Goi	[✓] Sukses
			Pencarian Flash Card Goi	[✓] Sukses
7	Mentor	Bunpou	Tambah Bunpou	[✓] Sukses
			Edit Bunpou	[✓] Sukses
			Hapus Bunpou	[✓] Sukses
			Pencarian Bunpou	[✓] Sukses

			Preview Materi Berbentuk Gambar	[✓] Sukses
			Preview Materi Berbentuk Audio	[✓] Sukses
			Preview Materi Berbentuk Video	[✓] Sukses
			Preview Materi Berbentuk PDF	[✓] Sukses
			Preview Materi Berbentuk Web Page	[✓] Sukses
8	Mentor	JLPT	Tambah <i>Problem Batch</i>	[✓] Sukses
			Edit <i>Problem Batch</i>	[✓] Sukses
			Hapus <i>Problem Batch</i>	[✓] Sukses
			Pencarian <i>Problem Batch</i>	[✓] Sukses
			Tambah Pertanyaan	[✓] Sukses
			Edit Pertanyaan	[✓] Sukses
			Hapus Pertanyaan	[✓] Sukses
			Pencarian Pertanyaan	[✓] Sukses
9	Disciple	Login	Login	[✓] Sukses
10	Disciple	Flash Card Moji	Tampil Flash Card Moji	[✓] Sukses
			Tampil Daftar Moji Pada Flash Card	[✓] Sukses
			Tampil Rincian Moji	[✓] Sukses
11	Disciple	Flash Card Goi	Tampil <i>Flash Card Goi</i>	[✓] Sukses
			Tampil Daftar <i>Goi</i> Pada <i>Flash Card</i>	[✓] Sukses
			Tampil Rincian <i>Goi</i>	[✓] Sukses
12	Disciple	Belajar Materi <i>Bunpou</i>	Tampil Materi Gambar	[✓] Sukses
			Tampil Materi PDF	[✓] Sukses
			Tampil Materi Audio	[✓] Sukses
			Tampil Materi Video	[✓] Sukses
			Tampil Materi <i>Web Page</i>	[✓] Sukses
13	Disciple	Latihan	Latihan Hafalan <i>Moji</i>	[✓] Sukses
			Latihan Hafalan <i>Goi</i>	[✓] Sukses
			Latihan Menulis <i>Moji</i>	[✓] Sukses
14	Disciple	JLPT	Simulasi JLPT	[✓] Sukses

2. White-Box Testing

Untuk memastikan sistem telah berhasil menerapkan algoritma dengan benar, maka perlu dilakukan pengujian dengan menerapkan metode *White-Box Testing* dengan menggunakan kasus-kasus uji (*testcases*). Berikut adalah tabel ringkasan hasil pengujian algoritma pencarian string tersebut. Secara keseluruhan dari total 200 kasus uji, sistem berhasil menemukan atau tidak menemukan seluruh 200 string yang dicari sebagaimana seharusnya, sehingga keberhasilan implemementasi algoritma *Aho-Corasick* pada proto-type sistem ini adalah **100%**.

Tabel 2. White-Box Testing Implemementasi Algoritma Aho Corasick

No	Kategori Pengujian	Jumlah Test Case	Jumlah Test Case Berhasil
1	Kata atau romaji dengan huruf latin	30	30
2	Substring huruf latin	30	30
3	Kata berbentuk furigana	30	30

4	Substring furigana	30	30
5	Kata dengan penulisan asli (kombinasi kanji, hiragana, katakana)	30	30
6	Substring dengan kombinasi kanji, hiragana, katakana	30	30
7	Random String	20	20
TOTAL		200	200

Selanjutnya metode White-Box Testing juga diterapkan untuk menguji tingkat kevalidan implementasi algoritma *Hu-Moment Invariant* tersebut pada sistem di penelitian ini. Perhitungan dilakukan dengan menjumlahkan semua mean dari semua huruf termasuk *Hiragana*, *Katakana* dan *Kanji*. Berikut adalah tabel rekapitulasi perhitungan rata-rata dari validitas algoritma *Hu-Moment Invariants* untuk huruf-huruf yang diujikan pada penelitian ini. Dari seluruh pengujian tersebut diperoleh bahwa rata-rata keberhasilan pengenalan huruf-huruf Jepang adalah sebesar **92,38%**.

Tabel 3. Rekapitulasi White-Box Testing Algoritma Hu-Moment Invariant

No	Jenis Moji	Jumlah Moji	Mean
1	Kanji	110	89,95
2	Hiragana	123	93,28
3	Katakana	123	93,65
Total		356	92,38

4. KESIMPULAN

Adapun kesimpulan dari penelitian Kamus Dan Learning Center Bahasa Jepang Untuk Persiapan *Japanese Language Proficiency Test* (JLPT) Menggunakan Algoritma Aho Corassick Dan Hu-Moments Dan L1-Metric Ini Adalah Sebagai Berikut :

1. Penelitian ini berhasil menghasilkan sebuah *proto-type* aplikasi pembelajaran bahasa Jepang yang dikhususkan untuk melatih kemampuan bahasa Jepang untuk menghadapi ujian JLPT, di mana aplikasi yang dihasilkan dilengkapi dengan kamus *Kanji* dan *Goi* yang memungkinkan user untuk melakukan pencarian teks (*String-Matching*) menggunakan algoritma *Aho-Corasick*, dan pencarian dan latihan menulis huruf *Kanji* pada kanvas (*Image-Matching*) dengan mengimplementasikan algoritma *Hu-Moment Invariants*. Terdapat 56 fitur yang harus terdapat pada sistem ini, di mana berdasarkan pengujian validitas menggunakan metode *Black-Box Testing* ke-56 fitur tersebut berhasil dibuat, sehingga angka keberhasilan implementasi adalah 100%.
2. Algoritma *Aho-Corasick* yang diimplementasikan pada penelitian ini berhasil menemukan 200 kata kunci yang diujikan dengan metode *White-Box Testing* dengan angka keberhasilan validasi sebesar 100%.
3. Algoritma *Hu-Moment Invariants* mampu mengenali 123 karakter huruf *Hiragana*, 123 huruf *Katakana*, dan 110 huruf *Kanji* N5, di mana masing-masing huruf diujikan sebanyak 5 kali dengan metode *White-Box Testing* dengan angka rata-rata validitas sebesar 92,38%.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Universitas Potensi Utama yang telah banyak memberikan masukan dan saran dalam penyelesaian penelitian ini.

REFERENSI

- [1] Alfarobbi, M. B. (2022). *Fundamental Aksara Jepang*. Yogyakarta: Deepublish. hlm. 10. [ISBN 978-623-02-5213-6](#).
- [2] [Aho, Alfred V.](#); [Corasick, Margaret J.](#) (June 1975). "[Efficient string matching: An aid to bibliographic search](#)". *Communications of the ACM*. **18** (6): 333–340. [doi:10.1145/360825.360855](#). [MR 0371172](#). [S2CID 207735784](#)
- [3] [Bowring, Richard](#); [Haruko Uryu Laurie](#) (2004). *[An Introduction to Modern Japanese: Book 1](#)*. United Kingdom: Cambridge University Press. p. 9. [ISBN 978-0521548878](#).
- [4] [Chen, Haijing](#) (2014). "[A Study of Japanese Loanwords in Chinese](#)". [University of Oslo](#). Archived from the original on September 12, 2021
- [5] [Cormen, Thomas H.](#), [Charles E. Leiserson](#), [Ronald L. Rivest](#), and [Clifford Stein](#). *[Introduction to Algorithms](#)*, Third Edition. MIT Press and McGraw-Hill, 2009. [ISBN 0-262-03293-7](#). Chapter 32: String Matching, pp. 985–1013.
- [6] [Douglass, Bruce](#) (2014). *[Real-Time UML Workshop 2nd Edition](#)*. Newnes. [ISBN 978-0-471-29551-8](#).
- [7] [Eells, Walter Crosby](#) (May 1952). "Language Reform in Japan". *The Modern Language Journal*. **36** (5): 210–213. [doi:10.1111/j.1540-4781.1952.tb06122.x](#). [JSTOR 318376](#)
- [8] [Flusser, J.](#) and [T. Suk](#), "[Rotation Moment Invariants for Recognition of Symmetric Objects](#)", *IEEE Trans. Image Proc.*, vol. 15, pp. 3784–3790, 2006.
- [9] [Gonzalez, Rafael](#) (2018). *Digital image processing*. New York, NY: Pearson. [ISBN 978-0-13-335672-4](#). [OCLC 966609831](#).
- [10] [Hadamitzky, Wolfgang](#) and [Spahn, Mark](#) (2012), *[Kanji and Kana: A Complete Guide to the Japanese Writing System](#)*, Third Edition, Rutland, VT: Tuttle Publishing. [ISBN 4805311169](#).
- [11] "[Japanese Language Proficiency Test guidelines, 2006 \(PDF\), page 3](#)" (PDF). JEES and The Japan Foundation. Archived from [the original](#) (PDF) on July 11, 2009. Retrieved February 18, 2009.
- [12] [JLPTSensei](#). JLPT N5 Kanji List. <https://jlptsensei.com/jlpt-n5-kanji-list/>, retrieved on July 21 2024
- [13] [Mathieu](#) (November 19, 2017). "[The History of Kanji 漢字の歴史](#)". *It's Japan Time*. Archived from the original on September 12, 2021.
- [14] [Mehta, Chintan](#); [K Bhavsar, Ankit](#); [Oza, Hetal](#); [Shah, Subhash](#) (15 February 2018). *[MySQL 8 Administrator's Guide: Effective Guide to Administering High-performance MySQL 8 Solutions](#)*. Packt Publishing. p. 32. [ISBN 9781788393843](#).
- [15] [Miller, Roy Andrew](#) (1966) *A Japanese Reader: Graded Lessons in the Modern Language*, Rutland, Vermont: Charles E. Tuttle Company, Tokyo, Japan, p. 28, Lesson 7: Katakana: *a—no*. "Side by side with hiragana, modern Japanese writing makes use of another complete set of similar symbols called the katakana."
- [16] [Myler, Harley R.](#)(1998). "[2.3 Flowcharts](#)". *Fundamentals of Engineering Programming with C and Fortran*. Cambridge University Press. pp. 32–36. [ISBN 978-0-521-62950-8](#).

- [17] Navarro, Gonzalo; Mathieu Raffinot (2008), *Flexible Pattern Matching Strings: Practical On-Line Search Algorithms for Texts and Biological Sequences*, Cambridge University Press, ISBN 978-0-521-03993-2
- [18] Petrović, Maja; Malešević, Branko; Banjac, Bojan; Obradović, Ratko (2014). *Geometry of some taxicab curves*. 4th International Scientific Conference on Geometry and Graphics. Serbian Society for Geometry and Graphics, University of Niš, Srbija. arXiv:1405.7579
- [19] Shibatani, Masayoshi (2008). *The Languages of Japan*. Cambridge University Press. ISBN 978-0521369183.
- [20] Suski, P.M. (2011). *The Phonetics of Japanese Language: With Reference to Japanese Script*. Taylor & Francis. p. 1. ISBN 9780203841808.
- [21] Taylor, Insup; Taylor, Maurice Martin (1995). *Writing and literacy in Chinese, Korean, and Japanese*. Amsterdam: John Benjamins Publishing Company. p. 305. ISBN 90-272-1794-7.