

Optical Character Recognition Citra Kata Kanji Menggunakan Ekstraksi Fitur Algoritma Chain Code Dan Algoritma L1-Metric

¹Kevin Laurence H., ²Karman Surya, ³Halim Agung

^{1,2,3}Fakultas Teknologi dan Desain, Universita Bunda Mulia

^{1,2,3}Jl. Lodan Raya No. 2 Jakarta Utara, DKI Jakarta

¹vinz_lrn@gmail.com, ²suryakarman@gmail.com, ³hagung@bundamulia.ac.id

Abstract – *Kanji is a language that originated from Japan. Japanese itself has spread in various countries especially in Indonesia. However, Japanese is not easy to learn because Japanese is not an international language. Therefore we need a system that can read Japanese character, especially kanji. This research will focus on designing an optical character recognition using chain code feature extraction to perform pattern recognition on kanji characters image and calculating the distance with manhattan distance (L1 Metric). The application will be built using pascal with Lazarus IDE and database system integration. The process in this research consisted of 5 stages, that is pre-processing, segmentation, linear spatial filtering, chain code feature extraction, and distance calculation. The distance of tested kanji image will compered with the kanji image that has been trained in the database to find the smallest distance. The test result with the chain code algorithm and manhattan distance (L1 Metric) showed that 21.82% of the kanji handwritten images were successfully recognized and 78.18 failed to recognize the images. The failure was caused by not using the thinning method so that the kanji image with different thickness and also the varied size of the object (kanji) would produce different feature values.*

Keyword: *Chain Code, Feature Extraction, Kanji, L1 Metric, Manhattan Distance, OCR.*

Abstrak - *Kanji merupakan salah satu bahasa yang berasal dari negara Jepang. Bahasa Jepang sendiri telah menyebar di berbagai negara terutama di Indonesia. Namun dikarenakan bahasa Jepang bukanlah bahasa yang mudah dipelajari karena bahasa Jepang tidak termasuk kedalam bahasa Internasional. Oleh karena itu diperlukan sistem yang bisa membaca bahasa aksara bahasa Jepang khususnya kanji. Penelitian ini akan difokuskan pada perancangan aplikasi pengenalan karakter optik dari aksara kanji menggunakan ekstraksi fitur chain code untuk melakukan pengenalan pola dari citra aksara kanji dan perhitungan jarak manhattan distance (L1 Metric). Dalam membangun aplikasi digunakan bahasa pemrograman pascal menggunakan Lazarus IDE dan integrasi sistem basis data. Proses dalam penelitian ini terdiri dari 5 tahap yaitu pre-processing, segmentasi, filtering spasial linier, ekstraksi fitur chain code, dan perhitungan jarak nilai fitur. Jarak citra kanji yang diuji akan dibandingkan dengan citra kanji yang sudah dilatih di basis data untuk mencari jarak terkecil. Hasil pengujian yang dilakukan dengan algoritma chain code dan manhattan distance (L1 Metric) menunjukkan bahwa sebesar 21.82% citra tulisan tangan kanji berhasil dikenali dan 78.18% mengalami kegagalan dalam mengenali citra yang diuji.*

Kata Kunci: *Chain Code, Ekstraksi Fitur, Kanji, L1 Metric, Manhattan Distance, OCR.*

I. PENDAHULUAN

Karakter merupakan sesuatu yang sangat penting dimana bisa tersimpan informasi didalamnya. Penggunaan karakter sendiri berbeda-beda diberbagai tempat contohnya di Jepang. Jepang sendiri tidak memiliki huruf abjad, oleh karena itu Jepang menggunakan hiragana, katakana, dan kanji. Kanji adalah salah satu sitem penulisan huruf Jepang yang paling banyak digunakan karena kanji berfungsi untuk menulis kata asli Jepang yang memiliki bunyi dan arti.

Optical Character Recognition (OCR) merupakan proses pengenalan karakter yang digunakan dengan tujuan untuk mengkonversi gambar atau citra huruf menjadi karakter ASCII yang dikenali oleh komputer. Gambar atau citra huruf yang dimaksud dapat berupa hasil scan dokumen, print-screen atau screenshot, dan lain-lain [1].

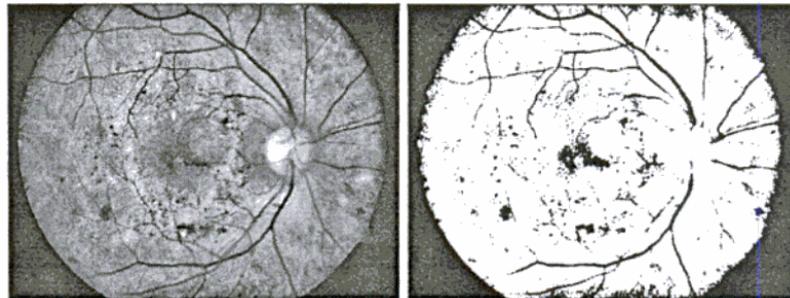
OCR sendiri memiliki beberapa metode dalam mendeteksi karakter, yaitu : Offline dan Online Character Recognition, Template Matching, Zoning, Projection Profile, Chaincode, dan Histogram of Oriented Gradient (HOG) [2]. Dengan memanfaatkan metode chain code, gambar atau citra huruf kanji akan dideteksi polanya dan disimpan kedalam basis data.

Pada penelitian ini, penulis hendak menggunakan algoritma chain code untuk melakukan pengenalan pola (pattern recognition) dari suatu gambar atau citra kanji yang mana hasil dari penggunaan algoritma chain code akan digunakan untuk optical character recognition.

II. LANDASAN TEORI DAN METODE

1. Segmentasi Citra

Segmentasi adalah proses pengolahan citra yang bertujuan untuk memisahkan objek dan latar. Nilai objek biasanya diisi dengan 0 dan latar diisi dengan 1 atau sebaliknya. Tujuan pemisahan objek dan latar adalah agar objek bisa dianalisa dengan mudah dalam rangka untuk mengenali objek [2].



Gambar 1. Contoh Segmentasi *Thresholding* Global [2]

Segmentasi *thresholding* global adalah metode segmentasi yang mengkonversikan semua piksel dengan menggunakan rumus dibawah.

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) \leq T \end{cases}$$

Keterangan :

g : nilai thresholding

f : nilai piksel

x : koordinat x

y : koordinat y

T : nilai ambang

2. Filtering Spasial Linier

Filtering spasial linier adalah proses perbaikan citra yang dilakukan dengan operasi linier berupa konvolusi atau korelasi dengan menggunakan operasi kernel [3]. Filtering spasial linier dibagi menjadi 2, yaitu high pass filtering dan low pass filtering. High pass filtering dengan koefisien 0 akan menghasilkan kontur pada objek sehingga objek dapat dianalisa dengan mudah.

$$\begin{matrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{matrix}$$

Gambar 2. Matriks 3x3 *High Pass Filtering* Koefisien 0 [3]

Cara kerja high pass filtering koefisien 0 adalah dengan menjumlahkan piksel dengan kernel matriks 3x3. Pada saat piksel dijumlahkan dengan kernel maka total penjumlahan adalah 0. Berikut rumus dari persamaan konvolusi.

$$h(x, y) = f(x, y) \times g(x, y) = \sum_{k=1}^M \sum_{l=1}^N f(k, l) \cdot g(x - (k - m), y - (l - n))$$

Keterangan :

x,y,k,l : koordinat piksel

h(x,y) : hasil konvolusi pada koordinat x dan y

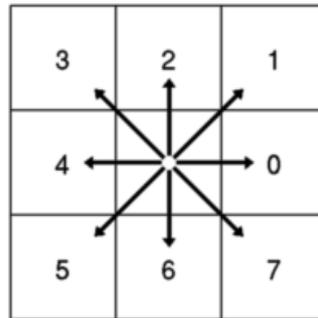
f(x,y) : fungsi f yang mengolah piksel x dan y berikut tetangganya

g(x,y) : fungsi *filter* untuk mengolah piksel x dan y

M,N: batas titik tetangga yang mempengaruhi titik yang sedang diolah

3. Metode Chain Code

Chain code atau kode rantai merupakan salah satu algoritma ekstraksi ciri kontur yang nilainya tidak berubah terhadap perlekukan rotasi, translasi, pencerminan, dan penskalaan. Pada metode ini dihasilkan delapan nilai yang menunjukkan arah piksel penyusun objek. Arah piksel dari *chain code 8-connected* ditunjukkan pada gambar di bawah ini.



Gambar 3. *Chain Code 8-Connected* [4]

Cara kerja algoritma *chain code* adalah dengan mencari posisi titik mulai. Posisi titik mulai adalah posisi piksel hitam yang pertama kali ditemukan dan selanjutnya akan mencari piksel hitam yang ada di sekitarnya. Untuk mendapatkan nilai fitur dari *chai code* dapat dilakukan dengan menggunakan rumus dibawah.

$$f_i = \frac{(\text{Jumlah angka 1}) + \dots + (\text{Jumlah angka 8})}{\text{Total kode rantai}}$$

Keterangan :

f_i : nilai fitur

jumlah angka 1 : jumlah angka 1 yang ada pada *chain code* citra

jumlah angka 8 : jumlah angka 8 yang ada pada *chain code* citra

total kode rantai : total dari seluruh *chain code* citra mulai dari angka 1, 2, sampai 8

4. Manhattan Distance (L1-metric)

L1-Metric melakukan pengukuran jarak antara fitur-fitur yang dimiliki dua buah citra, dimana jarak kedua buah citra ini yang nantinya akan dipertimbangkan sebagai kemiripan antara dua buah citra. Semakin kecil nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin mirip, semakin besar nilai jarak yang dihasilkan maka kedua citra akan dianggap semakin berbeda.

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

Keterangan :

d : jarak antara x dan y

x : data *training*

y : data yang diuji

n : jumlah data

i : atribut individu antara 1 sampai dengan n

x_i : data *training* ke i

y_i : data yang diuji ke i

5. Penelitian Sejenis

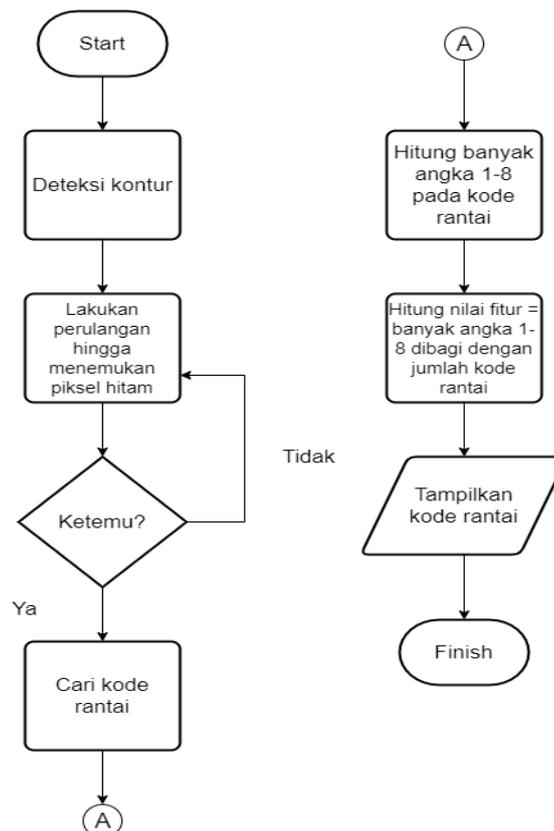
Berikut adalah penelitian sejenis yang mengambil topik *optical character recognition* tetapi menggunakan objek dan metode yang berbeda.

Tabel 1 Penelitian Sejenis

No.	Peneliti	Judul	Hasil
1	Theresia, Chris Simon	Pengenalan Karakter Menggunakan Pendekatan Algoritma Berbasis Chain Code[5]	Pengenalan karakter aksara hiragana berbasis <i>font</i> komputer dengan menggunakan metode <i>chain code</i> memiliki persentase keberhasilan sebesar 52,945%, tetapi program masih belum bisa mendeteksi huruf besar dan huruf kecil dan pembacaan karakter yang berdekatan kurang baik.
2	Ade Setiawan, Kristien Margi Suryaningrum	<i>Optical Character Recognition</i> Jepang Menggunakan Matriks Populasi Piksel Dan <i>LI-Metric</i> [6]	Pengenalan karakter aksara Jepang dengan jenis <i>font</i> komputer yang berbeda-beda dengan menggunakan matriks populasi piksel dan <i>LI Metric</i> memiliki persentase keakuratan sebesar 82,61% yang membuktikan matriks populasi piksel layak digunakan dalam pengenalan karakter
3	Soumendu Das, Sreeparna Banerjee	<i>An Algorithm for Japanese Character Recognition</i> [7]	Pengenalan karakter hiragana yang ditulis menggunakan tangan menggunakan algoritma <i>Size Translation Rotation Invariant Character Recognition and Feature vector Based (STRICR-FB)</i> memiliki persentase keberhasilan 92,22%. Algoritma STRICR-FB layak digunakan dalam pengenalan karakter khususnya karakter yang ditulis dengan tangan

III. PEMBAHASAN

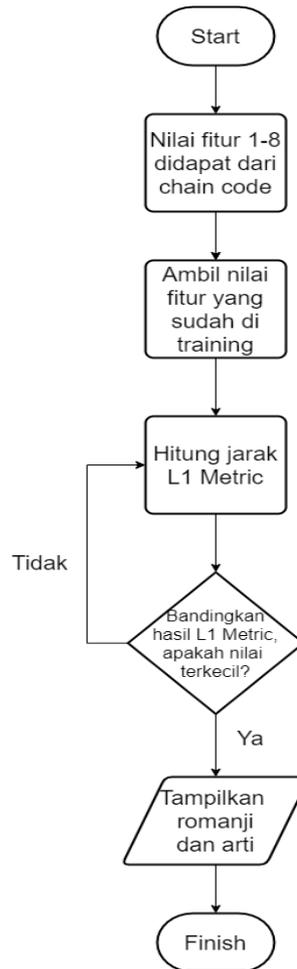
Pada gambar 4 akan ditampilkan *Flow Chart* proses pencarian *chain code*.



Gambar 4. Flowchart Algoritma Chain Code (A)

Gambar atau citra akan dideteksi bentuk kontur untuk mencari posisi titik hitam pertama dan selanjutnya akan dicari adalah kode rantai dari citra. Jika kode rantai sudah didapatkan, maka akan dilakukan perhitungan untuk mencari

nilai fitur dari masing-masing rantai 1 hingga 8. Hasil dari perhitungan nilai fitur akan digunakan untuk mencari nilai jarak terkecil menggunakan *L1-Metric*.



Gambar 5. Flowchart Algoritma Manhattan (*L1-Metric*)

Pada gambar 5, akan dilakukan pengambilan nilai fitur dari basis data dan kemudian akan dilakukan perhitungan *L1 Metric* dengan nilai fitur yang sudah didapatkan pada proses sebelumnya. Dengan mencari nilai terkecil yang pada saat nilai terkecil ditemukan, maka *romaji* dan arti dari *kanji* akan ditampilkan.

Tabel 2 Citra *Kanji* Yang Diuji

No	<i>Kanji</i>	<i>Romaji</i>	Arti
1	一	<i>ichi</i>	satu / 1
2	右	<i>migi</i>	kanan
3	雨	<i>ame</i>	hujan
4	円	<i>en</i>	yen
5	王	<i>ou</i>	raja
6	音	<i>oto</i>	suara
7	下	<i>shita</i>	bawah
8	火	<i>hi</i>	api
9	花	<i>hana</i>	bunga
10	貝	<i>kai</i>	kerang
11	学	<i>gaku</i>	belajar
12	気	<i>ki</i>	suasana hati
13	九	<i>kyuu</i>	sembilan / 9

No	<i>Kanji</i>	<i>Romaji</i>	Arti
14	休	<i>yasumu</i>	istirahat
15	毛	<i>ke</i>	rambut
16	金	<i>kane</i>	uang
17	空	<i>sora</i>	langit
18	月	<i>tsuki</i>	bulan
19	犬	<i>inu</i>	anjing
20	見	<i>miru</i>	melihat
21	五	<i>go</i>	lima / 5
22	口	<i>kuchi</i>	mulut
23	交	<i>kou</i>	asosiasi
24	左	<i>hidari</i>	kiri
25	山	<i>yama</i>	gunung

Pengujian akan dilakukan dengan citra *kanji* berbasis tulisan tangan sebanyak 11 versi yang masing-masing versi terdapat 25 karakter *kanji*.

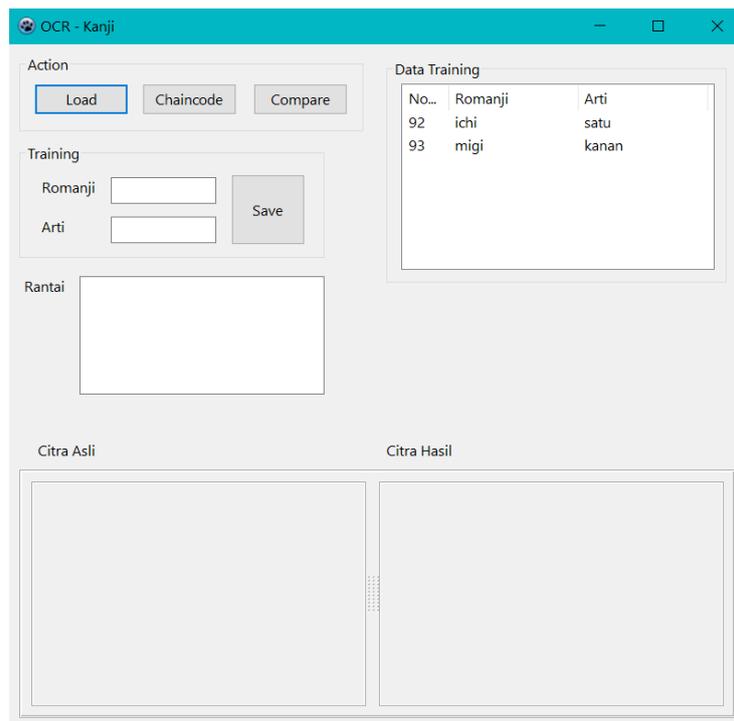
Tabel 3 Rekap Hasil Pengujian

No	Versi Tulisan Tangan	Banyak Karakter	Benar	Salah	Persentase
1	Tuliasn Tangan 1	25	6	19	24%
2	Tuliasn Tangan 2	25	9	16	36%
3	Tuliasn Tangan 3	25	7	18	28%
4	Tuliasn Tangan 4	25	3	22	12%
5	Tuliasn Tangan 5	25	5	20	20%
6	Tuliasn Tangan 6	25	2	23	8%
7	Tuliasn Tangan 7	25	1	24	4%
8	Tuliasn Tangan 8	25	5	20	20%
9	Tuliasn Tangan 9	25	8	17	32%
10	Tuliasn Tangan 10	25	5	20	20%
11	Tuliasn Tangan 11	25	9	16	36%
Rata-rata					21.82%

Hasil rata-rata dari 25 citra *kanji* dari 11 versi tulisan tangan yang berbeda adalah sebesar 21.82%. Hal ini menunjukkan bahwa algoritma *chain code* masih layak digunakan pada *optical character recognition*.

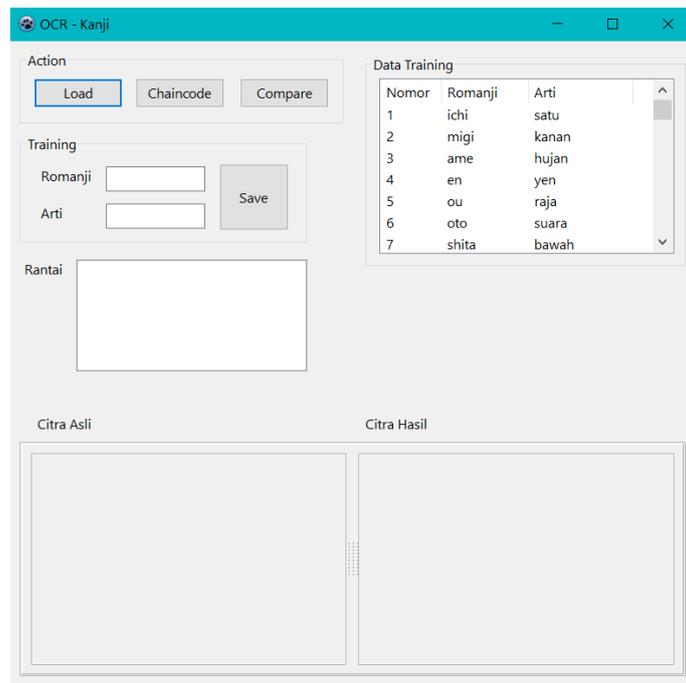
IV. IMPLEMENTASI SISTEM

Berikut akan dijelaskan tahap-tahap penggunaan sistem, dimana Ketika *user* menjalankan program, program akan menampilkan halaman awal seperti pada gambar 6 menunjukkan tampilan halaman awal sistem.



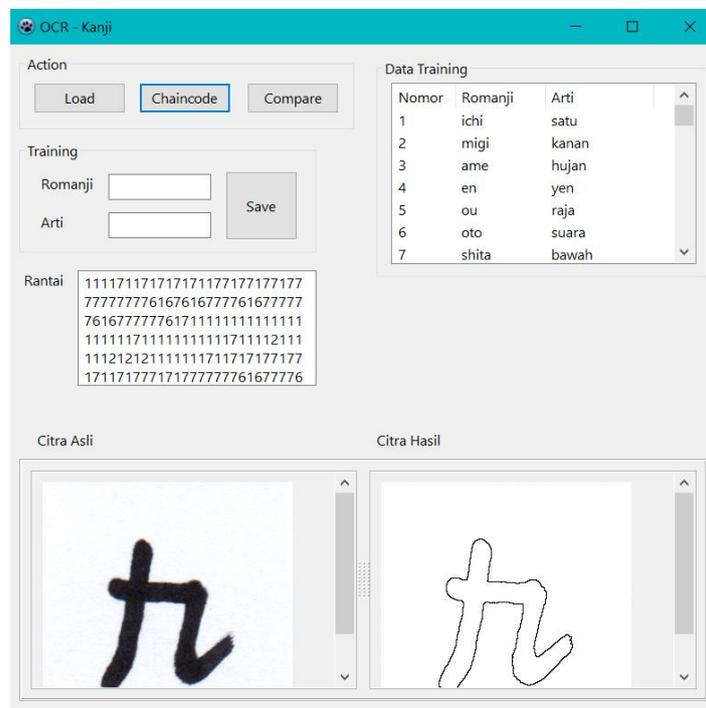
Gambar 6. Implementasi Halaman Awal

Pada gambar 7 *user* akan memasukan gambar atau citra yang akan dikenali dengan melakukan *load* maka gambar atau citra yang dipilih akan ditampilkan dikanvas yang sudah disediakan dibagian bawah.



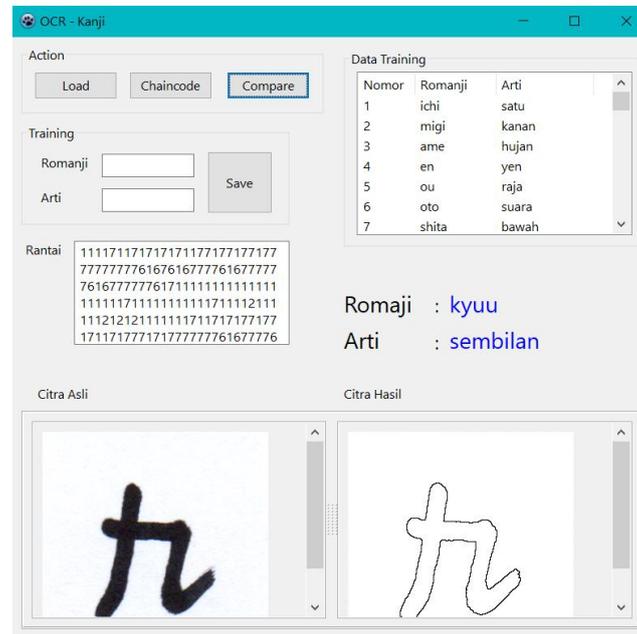
Gambar 7. Implementasi Halaman Awal

Pada gambar 8 menjelaskan tahap proses *chain code*. Hasil dari *chain code* akan ditampilkan kembali di kanvas citra hasil dan nilai rantai akan ditampilkan pada *EditText* Rantai.



Gambar 8. Implementasi Halaman Awal (Proses *Chain Code*)

Pada gambar 9 merupakan tahap akhir dari dimana pada saat *user* melakukan proses *compare* maka akan ditampilkan hasil *compare* dari citra *kanji* yang diuji dengan citra *kanji* yang sudah tersimpan di basis data yang kemudian hasilnya akan di tampilkan pada label *romaji* dan arti.



Gambar 9. Implementasi Halaman Awal (Proses Compare)

V. KESIMPULAN

1. Kesimpulan

Dari penelitian dan pengujian yang dilakukan maka dapat disimpulkan sebagai berikut :

1. Algoritma *chain code* dapat diterapkan pada *optical image recognition* dan mengenali tulisan tangan yang belum dilatih di basis data dengan rata-rata keakuratan sebesar 21,82%.
2. Ketebalan dan ukuran dari objek (*kanji*) mempengaruhi kode rantai dan nilai fitur.
3. Objek (*kanji*) yang bentuknya hamper mirip dapat mengakibatkan sistem salah atau terbalik dalam mengenali citra yang diuji.

2. Saran

Pada penelitian ini dengan menggunakan algoritma *chain code* dan *manhattan distance (LI Metric)* diharapkan akan menjadi lebih baik dengan beberapa saran yang dapat digunakan :

1. Dapat membaca tulisan tangan dengan lebih baik.
2. Dapat membaca kalimat kanji.
3. Menambahkan metode *thinning* dengan tujuan menambah keakuratan algoritma *chain code*.
4. Menambahkan metode *noise removal* pada citra tulisan tangan.

DAFTAR PUSTAKA

- [1] N. Venkata Rao, A. S. C. S. Sastry, A. S. N. Chakravarthy, and P. Kalyanchakravarthi, "Optical character recognition technique algorithms. ISSN : 1992-8645," *J. Theor. Appl. Inf. Technol.*, vol. 83, no. 2, pp. 275–282, 2016.
- [2] D. Putra, *Pengolahan Citra Digital*. ISBN : 978-979-29-1443-6. Yogyakarta: ANDI, 2010.
- [3] T. M. S. Mulyana, "Efek High Pass Filtering Dengan Koefisien Nol Pada Citra Biner. ISSN : 2579-6402," *J. Muara Sains, Teknol. Kedokt. dan Ilmu Kesehat.*, vol. 1, no. 1, pp. 75–83, 2017, doi: 10.24912/jmstkik.v1i1.394.
- [4] P. Liu, J. Zhang, and K. Guo, "A Parking-Lines Recognition Algorithm Based on Freeman Chain Code. ISBN : 978-1-4799-8645-3," *2015 7th Int. Conf. Intell. Human-Machine Syst. Cybern.*, vol. 2, pp. 349–352, 2015, doi: 10.1109/IHMSC.2015.247.
- [5] Theresia and C. Simon, "Pengenalan Karakter Menggunakan Algoritma Berbasis Chain Node. ISSN : 2477-5894," *Proceeding 2nd Informatics Conf. 2016 June 18, 2016, Univ. Budi Luhur, Jakarta, Indones.*, vol. 2016, pp. 65–91, 2016.
- [6] A. Setiawan and K. M. Suryaningrum, "Optical Character Recognition Jepang Menggunakan Matriks Populasi Pikel Dan L1-," vol. IV, no. 1, pp. 62–72, 2017.
- [7] S. Das and S. Banerjee, "An Algorithm for Japanese Character Recognition," *Int. J. Image, Graph. Signal Process.*, vol. 7, no. 1, pp. 9–15, 2014, doi: 10.5815/ijigsp.2015.01.02.