

Implementasi Algoritma Freeman Chain Code dan Algoritma K-Nearest Neighbor Dalam Pengenalan Huruf Mandarin

Ericson, Tedy Matius Surya Mulyana*

Fakultas Teknologi Dan Desain, Teknik Informatika, Universitas Bunda Mulia, Jakarta, Indonesia

Email: ¹ericsonwen123@gmail.com, ²*tmulyana@bundamulia.ac.id

Email Penulis Korespondensi: tmulyana@bundamulia.ac.id

Submitted 06-07-2022; Accepted 16-08-2022; Published 30-08-2022

Abstrak

Bahasa mandarin adalah bahasa internasional kedua yang digunakan oleh penduduk dunia dan merupakan bahasa yang paling banyak dipelajari oleh pelajar di Indonesia. Penelitian ini dilakukan untuk mengetahui tingkat keberhasilan algoritma Freeman Chain Code dan algoritma K-Nearest Neighbor dalam pengenalan huruf mandarin. Dalam melakukan pengenalan huruf Mandarin, terdapat beberapa tahapan yang harus dilalui, yaitu pre-processing, ekstraksi ciri, dan pengenalan huruf. Tahap pre-processing menggunakan grayscale, Gaussian Blur, binerisasi, dan thinning. Tahap ekstraksi ciri menggunakan Algoritma Freeman Chain Code dan Algoritma Depth First Search (DFS). Tahap klasifikasi menggunakan Algoritma K-Nearest Neighbor dan Algoritma L1-Metric (Manhattan Distance). Pada penelitian ini, terdapat 10 kelas huruf dengan tiap huruf memiliki 100 citra sampel. Rasio pembagian dari penelitian ini adalah 70% data training dan 30% data testing. Penelitian ini menghasilkan aplikasi yang mampu mengenali huruf Mandarin. Tingkat keberhasilan yang dihasilkan dari penelitian ini adalah 72% dengan 216 dari 300 citra berhasil dikenali.

Kata kunci: Freeman Chain Code; K-Nearest Neighbor; Ekstraksi Ciri; Huruf Mandarin; Klasifikasi; Pre-processing

Abstract

Mandarin is the second international language used by the world's population and is the language most studied by students in Indonesia. This research was conducted to determine the success rate of the Freeman Chain Code algorithm and the K-Nearest Neighbor algorithm in mandarin letter recognition. In recognizing Chinese characters, there are several stages that must be passed, namely pre-processing, feature extraction, and letter recognition. The pre-processing stage uses grayscale, Gaussian Blur, binaryization, and thinning. The feature extraction stage uses the Freeman Chain Code Algorithm and the Depth First Search (DFS) Algorithm. The classification stage uses the K-Nearest Neighbor Algorithm and the L1-Metric Algorithm (Manhattan Distance). In this study, there were 10 letter classes with each letter having 100 sample images. The distribution ratio of this research is 70% training data and 30% testing data. This research produces an application that is able to recognize Chinese characters. The success rate resulting from this study was 72% with 216 of the 300 images successfully recognized.

Keywords: Freeman Chain Code; K-Nearest Neighbor; Feature Extraction; Chinese Characters; Classification; Pre-processing

1. PENDAHULUAN

Bahasa mandarin adalah bahasa internasional kedua yang digunakan oleh penduduk dunia dan merupakan bahasa yang paling banyak dipelajari oleh pelajar di Indonesia [1]. Aksara Han atau huruf Mandarin adalah simbol yang digunakan dalam bahasa tulisan untuk memvisualisasikan bahasa Mandarin. Aksara Han dapat dibedakan dalam kategori tunggal dan gabungan [2]. Dalam penelitian ini, aksara Han yang digunakan merupakan aksara Han yang masuk ke dalam kategori tunggal.

Pelajar sekarang mengalami kemalasan dalam hal menghafal aksara Bahasa Mandarin dengan goresan-goresan yang rumit dan susah dipahami. Itu lah penyebab pelajar malas belajar Bahasa Mandarin [1]. Tujuan dari penelitian ini adalah untuk mengetahui tingkat keberhasilan Algoritma Freeman Chain Code dan Algoritma K-Nearest Neighbor dalam pengenalan huruf Mandarin. Selain itu, penelitian ini juga bertujuan untuk mengembangkan aplikasi pengenalan huruf Mandarin yang mengimplementasikan Algoritma Freeman Chain Code dan Algoritma K-Nearest Neighbor.

Penelitian sebelumnya yang mirip dengan penelitian ini adalah penelitian yang berjudul Yoruba Handwritten Character Recognition using Freeman Chain Code and K-Nearest Neighbor Classifier oleh Jumoke Falilat Ajao, David Olufemi Olawuyi, dan Odetunji O. Odejobi [3]. Penelitian ini menggunakan 37 alfabet Yoruba. Tiap alfabet memiliki 61 sampel karakter untuk setiap jenis karakter. Dari 61 sampel tersebut, 1 sampel karakter digunakan sebagai data training dan 60 sampel karakter sebagai data testing. Citra yang akan dikenali pada penelitian ini adalah karakter Yoruba yang ditulis tangan. Setelah ditulis tangan, gambar dari karakter Yoruba dipindai menggunakan HP scanner dengan resolusi 300 dpi. Hasil pindai merupakan citra dengan format PNG. Pada penelitian ini, citra diolah terlebih dahulu melalui tahap pre-processing. Tahap pre-processing meliputi pemotongan image, penghilangan noise, grayscale, binerisasi, dan normalisasi. Setelah itu, terdapat ekstraksi ciri menggunakan 8-arah mata angin Algoritma Freeman Chain Code. Kemudian, klasifikasi dilakukan menggunakan Algoritma K-Nearest Neighbor. Penelitian ini menghasilkan rata-rata tingkat keberhasilan pengenalan karakter Yoruba sebesar 87,7%.

Meskipun penelitian tersebut menggunakan algoritma yang sama dengan penelitian yang dilakukan oleh peneliti, tetapi terdapat perbedaan dalam pengimplementasian algoritma. Dalam penelitian yang dilakukan oleh peneliti, peneliti menggunakan 5 jenis jumlah tetangga dalam tahap klasifikasi, yaitu k-1, k-3, k-5, k-7, dan k-9. Alasan penggunaan 5 jenis jumlah tetangga dalam tahap klasifikasi dikarenakan peneliti ingin menghilangkan ketergantungan pengenalan karakter hanya pada 1 jenis jumlah tetangga saja. Selain itu, penelitian ini juga menggunakan Algoritma L1-Metric dalam

melakukan pengukuran jarak. Sedangkan penelitian oleh Jumoke Falilat Ajao, David Olufemi Olawuyi, dan Odetunji O. Odejobi menggunakan Euclidean Distance dalam melakukan pengukuran jarak.

Selain itu, terdapat penelitian yang berjudul Identifikasi Tulisan Tangan Huruf Katakana Jepang Dengan Metode Euclidean oleh Imam Riadi, Abdul Fadlil, dan Putri Annisa [4]. Citra yang dikenali pada penelitian ini adalah 5 jenis huruf katakana dasar. Masing-masing jenis huruf terdapat 50 hasil tulis tangan berupa citra digital. Penelitian ini memiliki 50 citra yang akan dibagi menjadi 20 data training dan 30 data testing dengan rasio pembagian 40% data training dan 60% data testing. Pada penelitian ini, citra akan diolah terlebih dahulu melalui tahap pre-processing. Pada tahap ini, citra akan melewati proses grayscale dan binerisasi. Setelah itu, terdapat tahap ekstraksi ciri menggunakan chain code. Kemudian, citra akan dikenali menggunakan Euclidean Distance. Penelitian ini menghasilkan tingkat keberhasilan pengenalan huruf katakana sekitar 78%.

Penelitian yang dilakukan oleh peneliti menggunakan grayscale, Gaussian Blur, Binerisasi, dan Thinning untuk pre-processing, Algoritma Freeman Chain Code dan Algoritma Depth First Search (DFS) untuk ekstraksi ciri, dan Algoritma L1-Metric (Manhattan Distance) dan Algoritma K-Nearest Neighbor (KNN) untuk klasifikasi. Pre-processing menghasilkan skeleton dari citra. Sedangkan penelitian oleh Imam Riadi, Abdul Fadlil, dan Putri Annisa menggunakan grayscale dan binerisasi untuk pre-processing, chain code untuk ekstraksi ciri, dan Euclidean Distance untuk klasifikasi. Pre-processing menghasilkan citra biner.

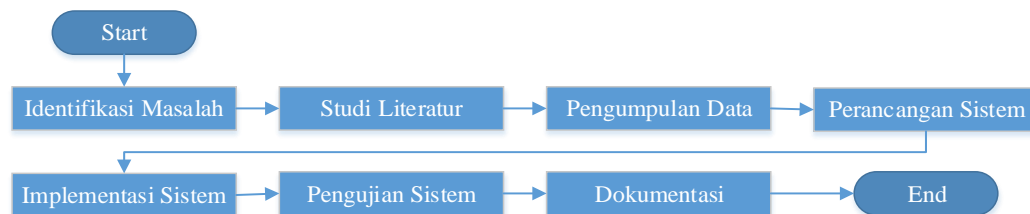
Penelitian ini merupakan kelanjutan dari penelitian sebelumnya mengenai pengenalan karakter kanji. Pada penelitian tersebut digunakan fitur matriks populasi piksel.[5] Pengembangan yang dilakukan pada penelitian kali ini dilakukan dengan menggunakan kode rantai Freeman. Selain itu penelitian ini juga merupakan bagian dari penelitian mengenai pengenalan karakter aksara jawa menggunakan kode rantai dan *levenshtein distance*. [6] dimana penelitian tersebut juga merupakan pengembangan dari penelitian OCR huruf kanji menggunakan fitur matriks populasi piksel.. Pada penelitian ini akan diteliti OCR menggunakan kode rantai freeman untuk mengenali huruf kanji yang diperkitakan sesuai dengan karakteristik huruf kanji, sedangkan pada penelitian pengenalan aksara jawa menggunakan *levenshtein distance* dan kode rantai yang sesuai dengan bentuk rangka dari aksara jawa.

Banyak peneliti telah menggunakan Freeman Chain Code untuk ekstraksi fitur karakter atau angka karena kesederhanaannya dan kemampuan untuk kebutuhan memori yang kecil. K-Nearest Neighbors (KNN) juga digunakan karena kesederhanaannya, pengurangan waktu pelatihan dan kinerja yang baik [3]. Dikarenakan oleh penelitian sebelumnya, maka peneliti ingin melakukan pengujian tingkat keberhasilan Algoritma Freeman Chain Code dan Algoritma K-Nearest Neighbor dalam melakukan pengenalan huruf Mandarin. Diharapkan penelitian ini dapat membantu pelajar dalam mempelajari huruf Mandarin.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Pada penelitian ini, terdapat tahapan-tahapan yang dilakukan oleh peneliti dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Gambar 1 merupakan tahapan-tahapan yang dilakukan oleh peneliti dalam melakukan penelitian. Berdasarkan Gambar 1, terdapat beberapa tahapan-tahapan yang akan dilalui peneliti dalam melakukan penelitian, yaitu identifikasi masalah, studi literatur, pengumpulan data, perancangan sistem, implementasi sistem, pengujian sistem, dan dokumentasi. Tahapan-tahapan tersebut dilakukan oleh peneliti agar penelitian dapat mencapai tujuan yang diinginkan.

2.1.1 Identifikasi Masalah

Identifikasi masalah merupakan tahapan pertama pada penelitian ini. Pada tahapan ini, peneliti merumuskan masalah dari penelitian ini dan menentukan tujuan yang ingin dicapai dan didapatkan dari penelitian ini. Hal ini dilakukan agar penelitian memiliki tujuan akhir yang jelas sehingga tahapan-tahapan yang harus dilakukan untuk mencapai hal tersebut menjadi jelas.

2.1.2 Studi Literatur

Tahapan ini merupakan tahapan pengumpulan, pencarian, dan pembelajaran data referensi seperti buku, jurnal, internet, dan sumber lainnya. Data referensi yang dicari merupakan data-data yang terkait dengan penelitian ini, yaitu penjelasan dan pengertian teoritis mengenai algoritma-algoritma yang akan digunakan pada penelitian ini.

2.1.3 Pengumpulan Data

Tahapan ini merupakan tahapan pengumpulan data-data yang diperlukan dalam melakukan penelitian ini. Data-data yang dikumpulkan berupa citra digital huruf Mandarin. Data-data ini dikumpulkan dengan cara tulis tangan menggunakan aplikasi Clip Studio. Citra huruf Mandarin yang dikumpulkan dibagi menjadi 10 kelas. Tiap kelas huruf memiliki 100 citra dan citra tersebut dibagi dengan rasio 70% data training dan 30% data testing. Berdasarkan pembagian tersebut, dari 1000 citra yang dikumpulkan, 700 citra digunakan sebagai data training dan 300 citra digunakan sebagai data testing.

2.1.4 Perancangan Sistem

Tahapan ini merupakan tahapan peneliti dalam merancang sistem sesuai dengan pengertian dan penjelasan teoritis yang didapatkan dari tahap studi literatur. Pada tahapan ini, pemahaman yang didapatkan dari tahap tersebut diubah menjadi rancangan dari sistem yang berupa rancangan tampilan, proses, dan pengujian yang dapat mengenali huruf Mandarin.

2.1.5 Implementasi Sistem

Tahapan ini merupakan tahapan implementasi rancangan yang dibuat pada tahap perancangan sistem. Pada tahapan ini, rancangan-rancangan yang dihasilkan pada tahap perancangan sistem akan diimplementasikan menjadi sebuah aplikasi. Penelitian ini menggunakan bahasa pemrograman Pascal dan aplikasi Lazarus.

2.1.6 Pengujian Sistem

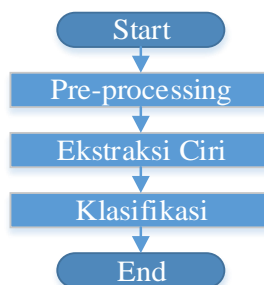
Tahapan ini merupakan tahapan pengujian dari aplikasi yang telah dibangun pada tahap implementasi sistem. Pada tahapan ini, peneliti melakukan dua jenis pengujian, yaitu pengujian data training dan pengujian data testing. Jika pengujian data training menghasilkan tingkat keberhasilan 100%, maka data training sudah konvergen dan dapat dipakai untuk melakukan pengujian data testing. Data testing yang dipakai merupakan data yang berbeda dari data training. Pengujian data testing akan menghasilkan tingkat keberhasilan dari aplikasi.

2.1.7 Dokumentasi

Pada tahapan ini, peneliti mendokumentasi segala hal yang dilakukan dan dicapai dari penelitian ini. Jurnal ini merupakan hasil dokumentasi penelitian ini dari awal sampai akhir pengerjaan penelitian.

2.2 Tahapan Metode

Gambar 2 merupakan tahapan metode dari penelitian ini. Tahapan metode merupakan penyusunan metode-metode yang digunakan dalam penelitian ini secara bertahap.



Gambar 2. Tahapan Metode

Berdasarkan Gambar 2, dalam melakukan pengenalan huruf Mandarin, terdapat 3 tahapan yang harus dilalui agar citra dapat dikenali, yaitu pre-processing, ekstraksi ciri, dan klasifikasi. Dalam tiap tahapan tersebut, terdapat algoritma yang digunakan untuk melakukan tahap tersebut. Pada tahap pre-processing, terdapat Grayscale, Gaussian Blur, Binerisasi, dan Thinning. Setelah pre-processing, terdapat tahap ekstraksi ciri. Tahap ekstraksi ciri menggunakan Algoritma Freeman Chain Code dan Algoritma Depth First Search. Kemudian, terdapat tahap klasifikasi. Tahap ini menggunakan Algoritma L1-Metric (Manhattan Distance) dan Algoritma K-Nearest Neighbor.

2.2.1 Pre-Processing

Pre-processing merupakan tahap yang dilakukan untuk memproses citra yang akan dikenali. Citra yang diinput akan diolah terlebih dahulu untuk menyederhanakan informasi yang terdapat pada citra agar mudah dikenali. Tahap ini menggunakan beberapa metode, yaitu Grayscale, Gaussian Blur, Binerisasi, dan Thinning.

a. Grayscale

Grayscale adalah proses mengubah citra berskala keabuan yang dinyatakan dengan intensitas antara 0-255. Nilai 0 adalah warna hitam dan nilai 255 adalah warna putih [7][8]. Citra yang digunakan pada penelitian ini merupakan citra berwarna yang memiliki 3 channel warna. Melalui proses grayscale, 3 channel warna pada citra berwarna tersebut dapat disederhanakan menjadi 1 derajat keabuan pada citra grayscale. Output dari proses grayscale merupakan citra grayscale yang memiliki derajat keabuan. Dalam menghitung derajat keabuan, dapat digunakan persamaan (1).

$$grayscale = 0,299 * R + 0,587 * G + 0,114 * B \quad (1)$$

Keterangan:

grayscale : derajat keabuan

R : channel warna red (merah)

G : channel warna green (hijau)

B : channel warna blue (biru)

b. Gaussian Blur

Mask yang sering pula digunakan untuk penghalusan citra adalah mask penghalusan Gaussian (Gaussian smoothing) [9]. Filter Gaussian yaitu operator konvolusi yang digunakan untuk mengaburkan gambar dan untuk menghilangkan gangguan (noise) [10]. Untuk melakukan Gaussian Blur, digunakan persamaan (2).

$$h(m, n) = \frac{1}{2\pi\sigma^2} e^{-\frac{(m^2+n^2)}{2\sigma^2}} \quad (2)$$

Keterangan:

σ : nilai deviasi standar distribusi normal yang digunakan.

m, n : posisi koordinat mask

π : konstanta dengan nilai 3,14

e : konstanta bilangan natural dengan nilai 2,718281828

c. Binerisasi

Binerisasi merupakan proses untuk mengubah suatu citra menjadi citra yang hanya terdiri dari warna hitam dan putih. Pada proses ini membutuhkan masukan dari user berupa nilai ambang untuk membedakan dan memisahkan area objek dan background [11]. Input dari proses binerisasi berupa citra grayscale. Sedangkan output dari proses ini adalah citra biner.

d. Thinning

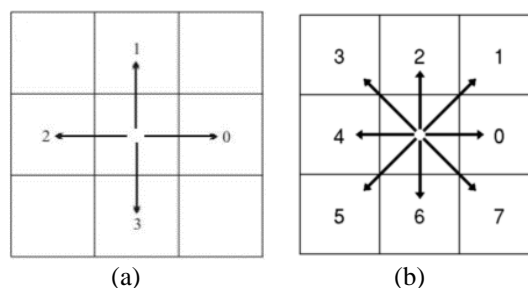
Thinning merupakan salah satu algoritma perubahan citra dalam morphological operation. Definisi image thinning adalah proses morphology image yang merubah bentuk asli binary image menjadi image yang menampilkan batas-batas obyek/foreground hanya setebal satu pixel. Algoritma thinning secara iteratif ‘menghapus’ pixel-pixel pada binary image, dimana transisi dari 0 ke 1 (atau dari 1 ke 0 pada konvensi lain) terjadi sampai dengan terpenuhi suatu keadaan dimana satu himpunan dari lebar per unit (satu pixel) terhubung menjadi suatu garis [12].

2.2.2 Ekstraksi Ciri

Setelah tahap pre-processing telah selesai, citra yang dihasilkan dari tahap tersebut akan digunakan pada tahap ekstraksi ciri. Fitur merepresentasikan ciri dan karakteristik dari objek. Ekstraksi fitur adalah proses mendapatkan ciri dan karakteristik tersebut [7]. Tahapan ini adalah tahapan dimana akan diidentifikasi ciri/karakter pembeda agar objek dapat dibedakan dengan objek yang lain. Ciri yang telah terekstraksi selanjutnya digunakan untuk parameter ataupun nilai masukan pada tahapan klasifikasi [13]. Tahap ini menggunakan beberapa metode dalam melakukan ekstraksi ciri, yaitu Algoritma Freeman Chain Code dan Algoritma Depth First Search.

a. Algoritma Freeman Chain Code

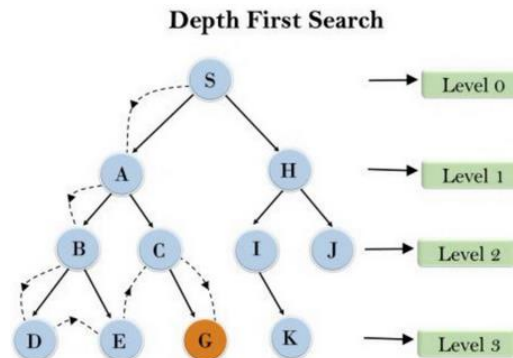
Chain code pertama kali diperkenalkan oleh Freeman pada tahun 1961 yang dikenal dengan nama FCC [14]. Chain code atau kode rantai merupakan salah satu algoritma ekstraksi ciri kontur yang nilainya tidak berubah terhadap perlekukan rotasi, translasi, pencerminan, dan penskalaan [15]. Kode rantai direpresentasikan dengan 4 arah atau 8 arah mata angin. Arah dari suatu mata angin dikodekan dengan menggunakan skema penomoran [16]. Arah setiap segmen dikodekan dengan menggunakan skema penomoran. Kode batas yang dibentuk sebagai urutan nomor arah tersebut disebut sebagai Freeman Chain Code [17]. Chain code mendeskripsikan sebuah obyek dengan segmen garis yang berurutan berdasarkan arah prioritas penelusuran yang telah ditetapkan. Arah dari tiap segmen direpresentasikan dengan angka tertentu [18]. Gambar 3(a) merupakan Algoritma Freeman Chain Code dengan 4 arah mata angin. Sedangkan Gambar 3(b) merupakan Algoritma Freeman Chain Code dengan 8 arah mata angin.



Gambar 3. Algoritma Freeman Chain Code (a) 4-Arah Mata Angin, (b) 8-Arah Mata Angin

b. Algoritma Depth First Search (DFS)

Gambar 4 merupakan gambar yang menggambarkan cara kerja dari algoritma Depth First Search (DFS).



Gambar 4. Algoritma Depth First Search (DFS)

Algoritma Depth First Search (DFS) adalah suatu metode pencarian pada sebuah pohon dengan menelusuri satu cabang sebuah pohon sampai menemukan solusi. Algoritma DFS adalah algoritma yang melakukan penelusuran dengan mengunjungi secara rekursif [19]. Pada pencarian dengan algoritma Depth First Search, pencarian dimulai dari level paling pertama (level 0) kemudian dilanjutkan ke anak paling kiri pada level berikutnya (level 1) demikian seterusnya sampai tidak terdapat anak lagi atau level yang lebih dalam lagi [20]. Pencarian dilakukan pada satu node dalam setiap level dari yang paling kiri dan dilanjutkan pada node sebelah kanan. Jika solusi ditemukan maka tidak diperlukan proses backtracking yaitu penelusuran balik untuk mendapatkan jalur yang diinginkan [21].

2.2.3 Klasifikasi

Klasifikasi adalah proses penemuan model (atau fungsi) yang menggambarkan dan membedakan kelas data atau konsep yang bertujuan agar bisa digunakan untuk memprediksi kelas dari objek yang label kelasnya tidak diketahui [22]. Klasifikasi citra merupakan proses untuk pengelompokan sejumlah pixel atau picture element pada sebuah citra menjadi kelas-kelas pada masing-masing kelas mendiskripsikan suatu entitas yang mempunyai karakter agar dapat dikenali [13]. Tahapan ini merupakan tahapan yang sangat penting karena hasil ekstraksi fitur akan diklasifikasi dan dianalisa tingkat kecocokan antara fitur data training dan fitur data testing [14]. Dalam melakukan klasifikasi, terdapat beberapa metode yang digunakan, yaitu Algoritma L1-Metric (Manhattan Distance) dan Algoritma K-Nearest Neighbor.

a. Algoritma L1-Metric (Manhattan Distance)

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 [15]. Manhattan distance digunakan untuk menghitung perbedaan absolut (mutlak) antara koordinat sepasang objek [23]. Untuk melakukan perhitungan jarak menggunakan Algoritma L1-Metric, dapat menggunakan persamaan (3).

$$d(x, y) = \sum_{i=1}^n |x_i - y_i| \quad (3)$$

Keterangan:

d : jarak antara x dan y

x : data training

y : data testing

n : jumlah data

i : atribut individu antara 1 sampai dengan n

x_i : data training ke i

y_i : data testing ke i

b. Algoritma K-Nearest Neighbor

Algoritma K-Nearest Neighbor (K-NN) adalah sebuah metode melakukan klasifikasi terhadap objek berdasarkan data pembelajaran jaraknya paling dekat dengan objek tersebut [24]. K-Nearest Neighbor adalah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasi sebelumnya dan bertujuan untuk mengklasifikasi objek baru berdasarkan atribut dan training samples [25]. K-NN merupakan algoritma yang hasil nilai instance yang baru akan diklasifikasikan berdasarkan mayoritas atau kelas yang paling banyak muncul yang akan menjadi kelas hasil klasifikasi dari kategori pada K-NN sehingga termasuk dalam algoritma supervised learning [26]. Prinsip kerja K-Nearest Neighbor (KNN) adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K-Nearest Neighbor terdekatnya dalam data pelatihan [27]. Untuk melakukan perhitungan Algoritma K-Nearest Neighbor dapat menggunakan persamaan (4).

$$\text{similarity}(T, S) = \frac{\sum_{i=1}^n f(T_i, S_i) * W_i}{W_i} \quad (4)$$

Keterangan:

T : Kasus baru

S : Kasus yang ada dalam penyimpanan

n : Jumlah atribut dalam masing-masing kasus

i : Atribut individu antara 1 sampai dengan n

W : Bobot yang diberikan pada atribut ke I

3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan beberapa algoritma dalam melakukan pengenalan huruf Mandarin. Algoritma-algoritma tersebut diterapkan sesuai dengan tahapan yang terdapat pada Gambar 2 yang merupakan tahapan metode dari penelitian ini. Pengenalan huruf Mandarin dimulai dengan pemilihan citra yang akan dikenali. Citra tersebut kemudian akan masuk ke dalam tahap pre-processing untuk menyederhanakan informasi-informasi yang terdapat pada citra tersebut. Tahap pre-processing menggunakan grayscale, gaussian blur, binerisasi, dan thinning dalam menyederhanakan citra. Hasil dari tahap ini berupa citra biner yang berisi skeleton dari huruf Mandarin. Kemudian, citra tersebut akan masuk ke dalam tahap ekstraksi ciri yang berfungsi untuk mendapatkan fitur-fitur dari citra tersebut. Dalam melakukan ekstraksi ciri, digunakan algoritma Freeman Chain Code dan Algoritma Depth First Search (DFS). Hasil ekstraksi ciri berupa fitur-fitur dari citra yang akan digunakan pada tahap klasifikasi. Tahap klasifikasi akan mengklasifikasi citra dengan cara membandingkan fitur dari citra yang akan dikenali dengan fitur dari data training. Pengenalan ini dilakukan menggunakan algoritma K-Nearest Neighbor (KNN) dan algoritma L1-Metric. Penelitian ini menggunakan 5 jenis jumlah tetangga dari k , yaitu 1, 3, 5, 7, dan 9. Alasan penggunaan tersebut adalah untuk mengurangi ketergantungan pengenalan citra pada satu jenis jumlah tetangga saja. Pengenalan huruf Mandarin dilakukan dengan melakukan pengukuran jarak antara fitur dari citra yang akan dikenali dan fitur dari data training. Hasil pengukuran jarak tersebut akan diurutkan dari terkecil hingga terbesar sesuai dengan kelas dari data training yang dibandingkan. Dari setiap kelas yang telah diurutkan akan diambil nilai terkecil sesuai dengan setiap jenis jumlah tetangga dari k . Nilai-nilai tersebut akan dijumlahkan kemudian dibagi sesuai dengan banyak nilai yang diambil. Kemudian, nilai-nilai yang dihasilkan dari setiap kelas tersebut akan dibandingkan untuk mendapatkan nilai k terkecil. Setelah itu, nilai k dari setiap jenis jumlah tetangga dari k yang berupa kelas dari hasil pengenalan didapatkan. Akhirnya, kelas dengan frekuensi nilai k terbanyak merupakan hasil pengenalan akhir dari citra tersebut.

Data yang digunakan dalam penelitian ini merupakan citra digital dari huruf mandarin dengan format bmp yang ditulis secara manual menggunakan aplikasi Clip Studio. Pada penelitian ini, terdapat 100 citra digital huruf mandarin untuk tiap kelas huruf mandarin dengan total 1000 citra digital huruf mandarin. Citra tersebut akan dibagi menjadi data training dan data testing dimana 70% citra akan digunakan sebagai data training, yaitu 70 citra untuk setiap kelas huruf mandarin dengan total 700 citra dan 30% citra akan digunakan sebagai data testing, yaitu 30 citra untuk setiap kelas huruf mandarin dengan total 300 citra.

Penelitian ini akan melakukan dua pengujian, yaitu pengujian data training sebagai data testing untuk mengecek apakah data training yang digunakan telah konvergen dan pengujian data testing yang berbeda dari data training untuk mengecek tingkat keberhasilan dalam pengenalan huruf mandarin.

Tabel 1 merupakan tabel hasil pengujian data training sebagai data testing. Pada pengujian ini, 700 citra yang digunakan sebagai data training digunakan kembali sebagai data testing. Pengujian ini dilakukan untuk mengetahui apakah data training konvergen. Berdasarkan Tabel 1, setiap citra berhasil dikenali dan tingkat keberhasilan dalam pengenalan citra mencapai 100% dengan tingkat kegagalan 0%. Dari hasil pengujian ini dapat disimpulkan bahwa data training yang digunakan telah konvergen.

Tabel 1. Tabel Tingkat Keberhasilan Pengujian Data Training

Kelas	Berhasil	Gagal	Tingkat Keberhasilan
女 Nǚ	70	0	100,00%
人 Rén	70	0	100,00%
君 Jūn	70	0	100,00%
年 Nián	70	0	100,00%
木 Mù	70	0	100,00%
左 Zuǒ	70	0	100,00%
生 Shēng	70	0	100,00%
牛 Niú	70	0	100,00%
力 Lì	70	0	100,00%
色 Sè	70	0	100,00%
Total	700	0	100,00%

Tabel 2 merupakan tabel hasil pengujian data testing yang berbeda dari data training. Pengujian ini menggunakan 300 citra yang berbeda dari data training sebagai data testing. Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan program dalam pengenalan citra. Berdasarkan Tabel 2, dapat diketahui bahwa tingkat keberhasilan data testing mencapai 72% dengan 216 citra yang berhasil dikenali dan 84 citra yang gagal dikenali dari 300 citra yang digunakan sebagai data testing. Huruf yang memiliki tingkat keberhasilan tertinggi adalah huruf 人 (Rén) dengan tingkat keberhasilan 100%. Sebaliknya, huruf yang memiliki tingkat keberhasilan terendah adalah huruf 生 (Shēng) dan huruf 牛 (Niú) dengan tingkat keberhasilan 36,67%.

Tabel 2. Tingkat Keberhasilan Pengujian Data Testing 1

Kelas	Berhasil	Gagal	Tingkat Keberhasilan
女 Nǚ	20	10	66,67%
人 Rén	30	0	100,00%
君 Jūn	22	8	73,33%
年 Nián	20	10	66,67%
木 Mù	27	3	90,00%
左 Zuǒ	29	1	96,67%
生 Shēng	11	19	36,67%
牛 Niú	11	19	36,67%
力 Lì	19	11	63,33%
色 Sè	27	3	90,00%
Total	216	84	72,00%

4. KESIMPULAN

Penelitian ini meneliti implementasi Algoritma Freeman Chain Code dan Algoritma K-Nearest Neighbor dalam pengenalan huruf Mandarin. Penelitian ini terdapat 10 kelas huruf yang tiap hurufnya memiliki 100 citra. Citra tersebut kemudian akan dibagi dengan rasio 70% data training dan 30% data testing. Tingkat keberhasilan pengujian menggunakan gabungan k-1, k-3, k-5, k-7, dan k-9 sebesar 72% dengan 216 dari 300 citra data testing berhasil dikenali.

REFERENCES

- [1] M. Andriani, "Kajian Peningkatan Minat Belajar Bahasa Mandarin Dengan Media Lagu," *VOX EDUKASI J. Ilm. Ilmu Pendidik.*, vol. 10, no. 2, pp. 78–83, 2019, doi: 10.31932/ve.v10i2.466.
- [2] Ina, "STRATEGI PEMBELAJARAN AKSARA HAN PADA PEMBELAJARAN MANDARIN TINGKAT DASAR," *VOX EDUKASI Jurnal Ilm. Ilmu Pendidik.*, vol. 12, no. 1, pp. 81–93, 2021, doi: <https://doi.org/10.31932/ve.v12i1.920>.
- [3] J. F. Ajao, D. O. Olawuyi, and O. O. Odejobi, "Yoruba Handwritten Character Recognition using Freeman Chain Code and K-Nearest Neighbor Classifier," *J. Teknol. dan Sist. Komput.*, vol. 6, no. 4, pp. 129–134, 2018, doi: 10.14710/jtsiskom.6.4.2018.129-134.
- [4] I. Riadi, A. Fadlil, and P. Annisa, "Identifikasi Tulisan Tangan Huruf Katakana Jepang Dengan Metode Euclidean," *J-SAKTI (Jurnal Sains Komput. dan Inform.)*, vol. 4, no. 1, p. 29, 2020, doi: 10.30645/j-sakti.v4i1.184.
- [5] T. M. S. Mulyana and A. Harjoko, "A chinese character recognition method based on population matrix and relational database," *Proc. 2nd Int. Conf. Inf. Commun. Technol. Semin. ICTS 2006*, no. 1, pp. 518–523, 2006.
- [6] T. M. S. Mulyana, D. Widyaningrum, and H. Herlina, "OCR HURUF JAWA DENGAN FITUR KODE RANTAI DAN LEVENSCHTEIN DISTANCE," *Netw. Eng. Res. Oper.*, vol. 6, no. 1, p. 67, Apr. 2021, doi: 10.21107/nero.v6i1.217.
- [7] T. F. Abidin, A. A. AzZuhri, and F. Arnia, "Pengenalan Karakter Plat Nomor Kendaraan Bermotor Menggunakan Zoning dan Fitur Freeman Chain Code," *J. Rekayasa Elektr.*, vol. 14, no. 1, pp. 19–25, Apr. 2018, doi: 10.17529/jre.v14i1.8932.
- [8] T. M. S. Mulyana and Herlina, "Evenly brightening using kurtosis Gaussian pattern to simplify image binarization," *J. Phys. Conf. Ser.*, vol. 1397, no. 1, p. 012076, Dec. 2019, doi: 10.1088/1742-6596/1397/1/012076.
- [9] H. Sunandar, "Perbaikan kualitas Citra Menggunakan Metode Gaussian Filter," *MEANS (Media Inf. Anal. dan Sist.)*, vol. 2, no. 1, pp. 19–22, 2017, doi: 10.54367/means.v2i1.18.
- [10] S. S. Sumijan, A. W. Purnama, and S. Arlis, "Peningkatan Kualitas Citra CT-Scan dengan Penggabungan Metode Filter Gaussian dan Filter Median," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 6, pp. 591–600, 2019, doi: 10.25126/jtiik.201966870.
- [11] A. Septiarini, K. Kunci, and P. Proyeksi, "Segmentasi Karakter Menggunakan Profil Proyeksi," *J. Inform. Mulawarman Ed. Juli*, vol. 7, no. 2, pp. 66–69, 2012, doi: <http://dx.doi.org/10.30872/jim.v7i2.88>.
- [12] Z. Zurnawita and Z. Suar, "Algoritma Image Thinning," *Elektron J. Ilm.*, vol. 1, no. 1, pp. 29–37, 2009, doi: 10.30630/eji.1.1.7.
- [13] A. Herdiansah, R. I. Borman, D. Nurnaningsih, A. A. J. Sinlae, and R. R. Al Hakim, "Klasifikasi Citra Daun Herbal Dengan Menggunakan Backpropagation Neural Networks Berdasarkan Ekstraksi Ciri Bentuk," *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 2, pp. 388–395, 2022, doi: 10.30865/jurikom.v9i1.3846.
- [14] . S., F. Arnia, and R. Muharar, "Pengenalan Aksara Jawi Tulisan Tangan Menggunakan Freeman Chain Code (FCC), Support Vector Machine (SVM) dan Aturan Pengambilan Keputusan," *J. Nas. Tek. Elektro*, vol. 5, no. 1, p. 45, 2016, doi: 10.25077/jnte.v5n1.185.2016.
- [15] K. L. H., K. Surya, and H. A. Agung, "Optical Character Recognition Citra Kata Kanji Menggunakan Ekstraksi Fitur Algoritma Chain Code Dan Algoritma L1-Metric," *J. Ilm. Tek. Inform.*, vol. 9, no. 2, p. 112, 2020, doi:

- <http://dx.doi.org/10.22441/format.2020.v9.i2.002>.
- [16] K. R. R. Wardani, A. Setiawan, and Y. Setiawan, “Penerapan Sistem Pakar Berbasis Aturan dan Metode Freeman Chain Code pada Aplikasi Pengukuran Isi Botol,” *J. Telemat.*, vol. 13, no. 2, pp. 67–72, 2018.
 - [17] S. Mawaddah and N. Suciati, “Pengenalan Karakter Tulisan Tangan Menggunakan Ekstraksi Fitur Bentuk Berbasis Chain Code,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 4, p. 683, Aug. 2020, doi: 10.25126/jtiik.2020742022.
 - [18] S. Rahman and M. Ulfayani, “Perancangan Aplikasi Identifikasi Biometrika Telapak Tangan Menggunakan Metode Freeman Chain Code,” *CESS (Journal Comput. Eng. Syst. Sci.)*, vol. 2, no. 2, pp. 64–73, 2017, doi: <https://doi.org/10.24114/cess.v2i2.6195>.
 - [19] B. Prasetyo and M. R. Hidayah, “Penggunaan Metode Depth First Search (DFS) dan Breadth First Search (BFS) pada Strategi Game Kamen Rider Decade Versi 0.3,” *Sci. J. Informatics*, vol. 1, no. 2, pp. 161–167, 2014, doi: 10.15294/sji.v1i2.4022.
 - [20] M. Rahmasuci, H. H. S, M. Azizah, P. Wulandari, D. A. A, and S. Bukhori, “Strategi Menemukan Jalan Keluar Labirin dengan Waktu Tercepat Menggunakan Metode DFS,” *INFORMAL Informatics J.*, vol. 3, no. 1, pp. 12–16, 2018, doi: 10.19184/isj.v3i1.9852.
 - [21] Rismayani and Ardiansyah, “Mobile-Based Public Transportation Search Application for Makassar City Using Depth First Search Algorithm,” *J. Pekommas*, vol. 18, no. 3, pp. 171–180, 2015, doi: <http://dx.doi.org/10.30818/jpkm.2015.1180303>.
 - [22] M. Laia, R. K. Hondro, and T. Zebua, “Implementasi Pengolahan Citra dengan Menggunakan Metode K-Nearest Neighbor Untuk Mengetahui Daging Ayam Busuk dan Daging Ayam Segar,” *J. Ris. Komputer*, vol. 8, no. 2, pp. 2407–389, 2021, doi: 10.30865/jurikom.v8i2.2818.
 - [23] M. Nishom, “Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square,” *J. Inform. J. Pengemb. IT*, vol. 4, no. 1, pp. 20–24, 2019, doi: 10.30591/jpit.v4i1.1253.
 - [24] M. Hasibuan, A. S. Sembiring, and R. D. Sianturi, “Penerapan Algoritma K-NN Pada Rekrutment Program Magang Keluar Negeri,” *JURIKOM (Jurnal Ris. Komputer)*, vol. 8, no. 4, pp. 120–127, 2021, doi: 10.30865/jurikom.v8i4.3589.
 - [25] J. W. Yodha and A. W. Kurniawan, “Pengenalan Motif Batik Menggunakan Deteksi Tepi Canny Dan K-Nearest Neighbor,” *Techno.COM*, vol. 13, no. 4, November, pp. 251–262, 2014, doi: <https://doi.org/10.33633/tc.v13i4.607>.
 - [26] F. S. Pattihha, “Perbandingan Metode K-NN, Naïve Bayes, Decision Tree untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia,” *JURIKOM (Jurnal Ris. Komputer)*, vol. 9, no. 2, pp. 506–514, 2022, doi: 10.30865/jurikom.v9i2.4016.
 - [27] R. K. Dinata, H. Akbar, and N. Hasdyna, “Algoritma K-Nearest Neighbor dengan Euclidean Distance dan Manhattan Distance untuk Klasifikasi Transportasi Bus,” *Ilk. J. Ilm.*, vol. 12, no. 2, pp. 104–111, 2020, doi: 10.33096/ilkom.v12i2.539.104-111.