

Implementasi Algoritma Freeman Chain Code Dan Support Vector Machine (SVM) Pada Identifikasi Aksara Arab Melayu

by Sufa Atin

Submission date: 04-Mar-2024 01:53PM (UTC+0700)

Submission ID: 2311132627

File name: artikel_B.A.25.pdf (1.09M)

Word count: 3957

Character count: 22820

Implementasi Algoritma Freeman Chain Code Dan Support Vector Machine (SVM) Pada Identifikasi Aksara Arab Melayu

Irawan Afrianto¹, Raju Riyanda², Sufa atin³

Teknik Informatika – Fakultas Teknik dan Ilmu Komputer
Universitas Komputer Indonesia
Bandung, Indonesia

¹irawan.afrianto@email.unikom.ac.id, ²r4ju_riy4nd4@yahoo.co.id, ³sufaatin@email.unikom.ac.id

Abstrak— Aksara arab melayu merupakan salah satu bentuk penulisan yang ditulis menggunakan aksara Arab Hijaiyah. Meskipun menggunakan aksara Arab, pembacaan dan penulisannya menggunakan tata bahasa Indonesia. Penulisan aksara arab melayu dilengkapi dengan konsonan abjad Indonesia yang ditulis dengan aksara Arab yang telah dimodifikasi. Penelitian ini bertujuan melakukan identifikasi tulisan tangan aksara arab menggunakan algoritma freeman chain code (FCC) sebagai ekstraksi ciri dan support vector machine (SVM) sebagai metode pengklasifikasiannya. FCC bekerja dengan cara menelusuri piksel-piksel objek yang terhubung menggunakan panduan 8 arah mata angin. Sedangkan SVM adalah proses pengklasifikasian terhadap hasil ekstraksi ciri yang dihasilkan oleh FCC. Keluaran dari aplikasi adalah identifikasi huruf arab melayu serta pelafalannya dalam bentuk huruf latin. Hasil penelitian menunjukkan bahwa algoritma FCC dan SVM dapat bekerja dengan baik untuk melakukan pengenalan tulisan tangan aksara arab arab melayu dengan akurasi mencapai 78% dari pengujian yang dilakukan.

Kata kunci—identifikasi; aksara arab melayu; freeman chain code (FCC); support vector machine (SVM); pengenalan pola; pengenalan tulisan tangan;

I. PENDAHULUAN

Aksara arab melayu adalah salah satu warisan budaya yang harus tetap dilestarikan. Aksara arab melayu merupakan salah satu bentuk penulisan yang ditulis menggunakan aksara arab hijaiyah. Meskipun menggunakan aksara arab, pembacaannya menggunakan tata bahasa Indonesia karena huruf arab melayu berasal dari bahasa melayu [1].

Handwriting recognition merupakan bagian dari Optical Character Recognition (OCR), dimana karakter masukan berasal dari tulisan tangan manusia yang kemudian diubah secara digitalisasi oleh komputer menjadi sebuah citra [2]. OCR adalah sebuah sistem komputer yang dapat membaca huruf, baik yang berasal dari teks tercetak maupun yang berasal dari tulisan tangan. OCR adalah aplikasi yang menerjemahkan gambar karakter (image character) menjadi bentuk teks dengan cara menyesuaikan pola karakter per baris dengan pola yang telah tersimpan di dalam aplikasi. Hasil dari proses OCR adalah berupa teks sesuai dengan gambar output scanner

dimana tingkat keakuratan penerjemahan karakter tergantung dari tingkat kejelasan gambar dan metode yang digunakan [3].

Tujuan Identifikasi tulisan tangan aksara arab melayu sama halnya dengan pengenalan bentuk atau pengenalan pola. Secara umum terdapat dua hal yang dapat mempengaruhi proses pengenalan aksara arab melayu yaitu mekanisme ekstraksi ciri dan mekanisme klasifikasi. Pengenalan pola pada aksara arab melayu bertujuan untuk melakukan proses terhadap suatu bentuk huruf arab melayu ke dalam salah satu kelas tertentu. Hal yang mempengaruhi pengenalan pola pada aksara arab melayu adalah bentuk huruf, jumlah titik, dan posisi titik.

Freeman chain code digunakan untuk melakukan mengekstraksi ciri-ciri pada suatu karakter dengan cara merepresentasikan bentuk kontur objek dan merepresentasikan piksel-piksel yang terhubung dengan melakukan penelusuran piksel-piksel objek menggunakan panduan arah mata angin dengan 8 arah atau 4 arah mata angin [4]. Penelitian H.Izackian dkk, menunjukkan bahwa chain code digunakan untuk mengidentifikasi aksara arab dengan mempertimbangkan jumlah dan lokasi titik serta bagian tambahan dari aksara arab tersebut [5]. Sementara penelitian A.B. Salem dkk, menunjukkan bahwa simpul-simpul dari suatu objek dapat dijadikan sebagai chain code untuk dapat dijadikan sebagai kode guna mengidentifikasi bentuk tersebut [6]. Penelitian lainnya dilakukan oleh H.I. Khan dkk, dengan mengombinasikan metode chain code untuk proses ekstraksi ciri dan K-Nearest Neighbor sebagai metode pengklasifikasiannya pada karakter vokal aksara Kannada [7]. Penelitian freeman chain code lainnya dilakukan oleh V.D.T Wijaya dkk yang melakukan deteksi pada aksara arab, namun belum menggunakan metode lain sebagai mekanisme untuk pengklasifikasiannya[8].

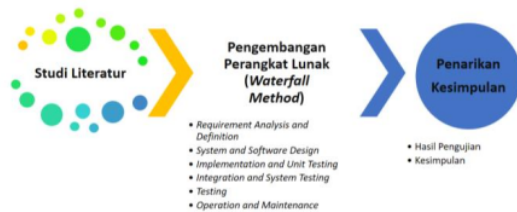
Support Vector Machine (SVM) adalah suatu metode klasifikasi learning machine yang berusaha menemukan hyperplane yang terbaik antar dua buah class pada input space SVM dapat digunakan untuk klasifikasi yang dapat diterapkan pada deteksi tulisan tangan, pengenalan obyek, identifikasi suara dan lain-lain [9]. SVM saat pertama kali diperkenalkan oleh Vapnik, hanya dapat mengklasifikasikan data ke dalam dua kelas (klasifikasi biner). Namun, penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi

data yang memiliki lebih dari dua kelas, terus dilakukan. Ada dua pilihan untuk mengimplementasikan multi class SVM yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data yang terdiri dari beberapa kelas ke dalam sebuah bentuk permasalahan optimasi [10].

Berdasarkan hal tersebut, penelitian ini bermaksud untuk mengembangkan suatu aplikasi yang mampu mengidentifikasi aksara arab melayu dan pelafalannya ke dalam huruf latin dengan memanfaatkan teknologi komputer, khususnya dalam pengolahan citra untuk membantu dalam pengenalan aksara arab melayu dan menulis aksara arab melayu dengan mengimplementasikan algoritma freeman chain code dan support vector machine. Pada penelitian ini, metode freeman chain code digunakan untuk mengekstrak ciri-ciri huruf arab melayu dengan menelusuri piksel-piksel pembentuk huruf menggunakan 4 atau 8 arah mata angin. Sedangkan metode support vector machine (svm) digunakan sebagai klasifikasi pengenalan pola yang bertujuan menemukan hyperplane yang terbaik pada ciri-ciri pembentuk huruf arab melayu.

II. METODE PENELITIAN

Metode penelitian yang dilakukan mencakup 3 tahapan yaitu : studi literatur dimana pengumpulan data dilakukan dengan cara mengumpulkan literatur, jurnal, paper yang ada kaitannya dengan judul penelitian. Pengembangan perangkat lunak menggunakan metode *waterfall*. Metode pengujian menggunakan metode black box dan uji statistik untuk melihat akurasi pengenalan aksara arab melayu.



Gambar 1. Metode penelitian

III. HASIL DAN PEMBAHASAN

A. Aksara Arab Melayu

Arab-Melayu adalah tulisan yang menggunakan aksara/huruf Arab (hijaiyah) dengan bahasa Melayu. Tulisan Arab-Melayu itu muncul bersamaan dengan penyebaran Islam ke tanah Melayu, yaitu sejak masa Kerajaan Samudera Pasai di Aceh [4] menyebar ke Kerajaan Melayu-Islam. Karakteristik aksara arab melayu ditulis dan dibaca dari kanan ke kiri seperti dalam tulisan arab asli, jumlah huruf arab-melayu terdapat 36 varian huruf yang terdiri 32 huruf arab (hijaiyah) [4] 4 huruf tambahan, semua huruf arab adalah konsonan. Agar dapat dibaca, maka huruf arab membutuhkan tanda baca harakat yang sebagiannya berfungsi sebagai kononon (a.i.u). Aksara arab melayu umumnya tidak menggunakan tanda harakat, sehingga terlihat sebagai arab gundul. Sebagai ganti harakat,

terkadang digunakan huruf 'illat (alif, wau dan ya') sebagai vokal (a-i-u) [11].

| LATIN | ARAB-MELAYU | LATIN | ARAB-MELAYU | LATIN | ARAB-MELAYU |
|-------|-------------|-------|-------------|-----------|-------------|
| A | ا | KH | خ | TH | ط |
| B | ب | L | ل | W | و |
| C | چ | M | م | Y | ي |
| D | د | N | ن | Z | ز |
| DL | طس | P | ف | DZ | ذ |
| F | ف | Q | ق | ZH | ظ |
| G | گ | R | ر | ZH | ظ |
| GH | غ | S | س | 'ain | ع |
| H | ح | TS | ث | hamzah | ء |
| H | هـ | SY | ش | Ta' | ة |
| J | ج | SH | ص | Marbuthah | ة |
| K | ك | T | ت | NG | ڠ |
| | | | | NY | ن - ي |

Gambar 2. Aksara arab melayu [11]

B. Deskripsi Sistem

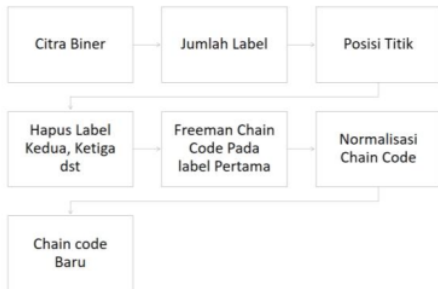
Sistem yang akan dibangun untuk pengenalan aksara arab melayu menggunakan algoritma freeman chain code sebagai salah satu ekstraksi ciri pada huruf arab melayu dan untuk pengklasifikasian berdasarkan ekstraksi ciri agar aksara arab melayu dapat dikenali dengan benar menggunakan metode support vector machine (svm) dapat dilihat pada gambar 2. Inputan karakter yang berasal dari optical mouse yang digerakkan oleh pengguna ke aplikasi. Setiap huruf arab melayu memiliki perbedaan ciri-ciri dengan yang huruf lainnya, seperti memiliki titik atau tidak memiliki titik dan posisi titik. Menurut paper H. Izakian, S. A. Monadjemi, B. Tork Ladani, dan K. Zamanifar [4], bahwasanya badan utama karakter akan dianggap sebagai objek utama, sedangkan bagian lain seperti titik akan dianggap sebagai objek tambahan. Objek utama pada karakter akan diekstraksi cirinya dengan memeriksa bentuk kontur objeknya menggunakan algoritma freeman chain code. Untuk memisahkan objek lainnya dari objek utama menggunakan metode connected component labelling (pemberian label pada komponen yang terhubung), sehingga sistem dapat menganalisis dan membedakan objek utama dengan objek tambahan. Kemudian hasil ekstraksi ciri karakter arab melayu diklasifikasikan menggunakan klasifikasi support vector machine (svm). Tujuan klasifikasi menggunakan support vector machine (svm) adalah agar aksara arab melayu dapat dikenali dengan benar dan memberikan output berupa pelafalan dalam bentuk aksara latin.



Gambar 2. Deskripsi sistem yang dibangun

C. Analisis Ekstraksi Ciri (Feature Extraction)

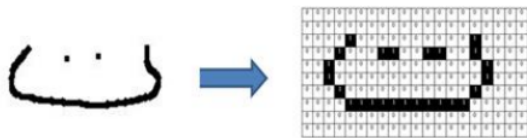
Karakter di dalam citra dapat dikenali berdasarkan ciri-ciri khususnya. Setiap objek karakter pasti mempunyai ciri-ciri yang berbeda dengan karakter yang lain. Ciri-ciri berguna untuk membedakan antara pola yang satu dengan yang lain. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat menghasilkan keakuratan yang tinggi. Ekstraksi ciri adalah proses pengambilan ciri-ciri dari suatu objek di dalam citra untuk membedakan objek yang satu dengan yang lain. Berikut alur ekstraksi ciri ditunjukkan pada gambar 3.



Gambar 3. Alur proses ekstraksi ciri

1) Citra Biner Aplikasi

Citra biner terdiri dari foreground berwarna hitam dan background berwarna putih, dimana foreground bernilai 1 dan background bernilai 0 seperti yang ditunjukkan oleh gambar 4..

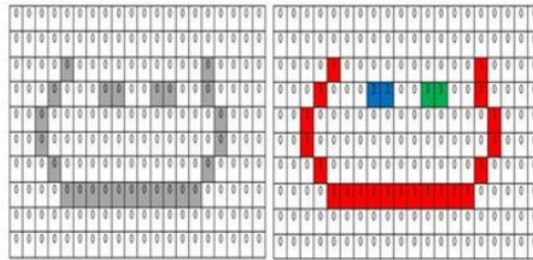


Gambar 4. Citra biner aplikasi

2) Connected Component Labelling

Beberapa karakter huruf arab melayu memiliki titik lebih dari satu atau karakter tambahan seperti huruf kaf (ك). Agar mengetahui sebuah karakter memiliki titik atau karakter tambahan, maka dilakukan proses perhitungan jumlah label yang saling terhubung pada piksel hitam. Jumlah label pada karakter arab melayu berbeda dengan yang satu dengan yang lainnya karena karakter arab melayu memiliki jumlah titik yang berbeda. Berikut hasil pelabelan dengan 8- connected

neighbors komponen yang terhubung ditunjukkan pada gambar 5.



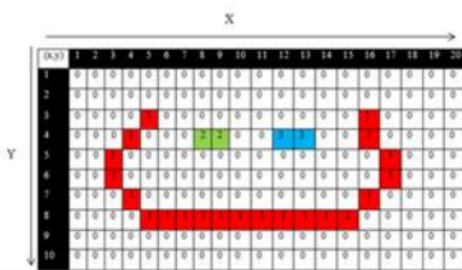
Gambar 5. Hasil pelabelan dengan 8- connected neighbors

3) Posisi Titik Pada Huruf Arab Melayu

Pada karakter arab melayu memiliki jumlah komponen terhubung yang sama dengan karakter lainnya. Maka diperlukan cara menentukan posisi titik pada karakter arab melayu. Posisi titik pada karakter arab melayu terbagi dua yaitu posisi atas dan bawah. Karena karakter arab melayu memiliki titik maka setiap posisi titik diberi nilai, untuk karakter yang tidak memiliki titik maka diberi nilai 0, karakter yang memiliki posisi titik di atas diberi nilai 1 dan karakter yang memiliki posisi titik di bawah diberi nilai 2. Posisi titik ditentukan dengan membandingkan posisi koordinat y pada setiap label dengan cara mencari nilai minimum dan maksimum dari koordinat y pada label pertama kemudian membandingkan nilai maksimum label kedua. Pada penelitian ini hanya label kedua yang dibandingkan dengan label pertama karena label kedua yang merupakan titik sebagai acuan untuk menentukan posisi titik.

Adapun tahapan untuk mencari posisi titik yaitu sebagai berikut:

- a) Mencari nilai koordinat (x,y) pada label pertama dan label kedua, ditunjukkan pada gambar 6.
- b) Nilai koordinat label pertama dan label kedua. Karena posisi titik hanya ada dua posisi yaitu posisi atas dan posisi bawah, maka untuk menentukan kedua posisi tersebut hanya menggunakan koordinat y. Untuk label pertama dicari nilai maksimum koordinat y dan nilai minimum koordinat y. Sedangkan untuk label kedua hanya menggunakan nilai maksimum koordinat y. Label koordinat ditunjukkan pada tabel 1.



Gambar 6. Koordinat pada label

TABEL I. KOORDINAT LABEL 1 DAN LABEL 2

| Label | Koordinat (X,Y) | Nilai Minimum Koordinat Y | Nilai Maksimum Koordinat Y |
|---------|---|---------------------------|----------------------------|
| Label-1 | (5,3), (4,4), (3,5), (3,6), (4,7), (5,8), (6,8), (7,8), (8,8), (9,8), (10,8), (11,8), (12,8), (13,8), (14,8), (15,8), (16,7), (17,6), (17,5), (16,4), (16,3), | 3 | 8 |
| Label-2 | (8,4), (9,4) | | 4 |

c) Kemudian hitung perbandingan nilai minimum dan maksimum koordinat y pada label pertama dengan rumus:

$$Label1 = \frac{9+3}{2} = 5,5$$

d) Bandingkan dengan koordinat y maksimum pada label kedua dengan hasil perbandingan label pertama. Jika hasil perbandingan label pertama lebih besar dari label kedua, maka posisi titik berada diatas bagian utama karakter dan diberi nilai 1. Jika perbandingan label pertama lebih kecil dari label kedua, maka posisi titik berada di bawah dan diberi nilai 2. Berdasarkan perhitungan hasil perbandingan label pertama bernilai 5,5 sedangkan label kedua bernilai 4. Karena label pertama lebih besar dari label kedua (5,5 > 4), maka posisi titik berada diatas bagian utama karakter dan diberi nilai 1.

4) Memisahkan Titik Dari Objek Utama Karakter

Proses yang dilakukan selanjutnya adalah memisahkan titik dari objek utama karakter. Tujuannya adalah untuk melakukan ekstraksi ciri menggunakan chain code yang hanya diproses pada objek utama karakter. Beberapa karakter memiliki objek utama karakter yang sama dan yang membedakannya hanya jumlah label dan posisi titiknya. Oleh karena itu, diperlukan penghapusan terhadap titik pada karakter, seperti yang ditunjukkan oleh tabel II.

TABEL II. BAGIAN UTAMA KARAKTER

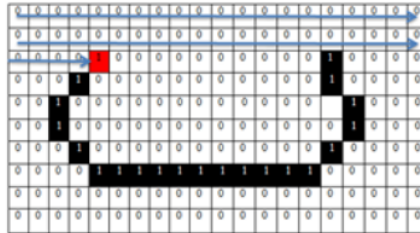
| No | Karakter | Bagian Utama Karakter |
|----|-----------|-----------------------|
| 1 | ب ت ث پ ن | ن |
| 2 | ح خ ج | ح |
| 3 | ذ | ذ |
| 4 | ز | ر |
| 5 | ش | س |
| 6 | ص | ص |
| 7 | ظ | ط |
| 8 | غ | ع |
| 9 | ف ق ك | ف |
| 12 | ك ك | ن |
| 13 | و | و |

5) Analisis Freeman Chain Code

Pada ekstraksi ciri menggunakan algoritma freeman chain code dengan menelusuri piksel-piksel objek menggunakan panduan arah mata angin dengan 8 arah. Dengan mekanisme melakukan penelusuran perpiksel, teknik chain-code dapat digunakan untuk menemukan struktur pembentuk dari suatu objek. Pada karakter yang sudah dipisahkan dari titik atau objek tambahan, maka dilakukan proses ekstraksi ciri menggunakan chain code dengan menggunakan arah mata

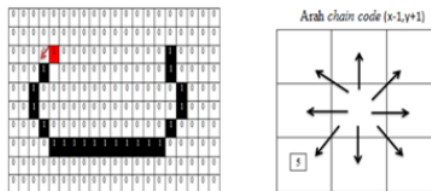
angin dengan 8 arah pada objek utama karakter. Adapun proses penelusuran piksel-piksel pada objek utama karakter menggunakan metode freeman chain code dengan 8 arah mata angin adalah sebagai berikut:

a) Lakukan scanning piksel-piksel terhadap citra foreground mulai sisi atas matrik objek utama yaitu dari kiri ke kanan dan atas ke bawah untuk mencari titik awal penelusuran seperti pada gambar 7.

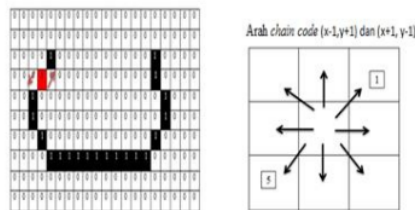


Gambar 7. Scanning piksel pada citra foreground

b) Jika ditemukan piksel foreground pertama sebagai titik awal penelusuran, kemudian periksa nilai chain code dengan arah mata angin 8 arah untuk seluruh piksel foreground. Nilai chain code pada koordinat (x-1,y+1) pada gambar 8. Nilai chain code pada koordinat (x-1,y+1) dan (x+1,y-1) pada gambar 9.



Gambar 8. Chain code pada koordinat (x-1,y+1)



Gambar 9. Chain code pada koordinat (x-1,y+1) dan (x+1,y-1)

c) Setelah selesai menelusuri piksel-piksel pada objek utama karakter menggunakan metode freeman chain code, maka di dapatkan nilai rantai kode hasil penelusuran seperti pada gambar 10.

5516172730304040404040404040414152536276

Gambar 10. Hasil chain code

1) Normalisasi Freeman Chain Code

Hasil akhir dari proses ekstraksi ciri berbasis chain code yang dilakukan adalah sebuah vektor ciri yang berisi informasi urutan kode chain-code pembentuk huruf. Dari mekanisme yang dilakukan oleh chain-code maka urutan chain-code yang dihasilkan untuk setiap huruf dapat memiliki panjang yang berbeda, sehingga diperlukan sebuah mekanisme normalisasi untuk menyamakan panjang chain code agar dapat digunakan sebagai input pada proses klasifikasi. Langkah-langkah normalisasi yang dilakukan yaitu merubah chain code menjadi matriks berisi nilai dan frekuensinya, menghapus nilai dengan frekuensi 1, menerapkan rumus normalisasi untuk mencari frekuensi sesuai yang diinginkan, dan membangun ulang chain code sesuai frekuensi yang baru [4]. Nilai frekuensi baru kode ke-i didapatkan dengan rumus normalisasi yang diterapkan persamaan 1. Frekuensi yang baru yang dihasilkan pada persamaan 1 adalah 10, yang artinya panjang chain code adalah 10 yang ditunjukkan pada tabel III.

$$F_i^n = \frac{F_i}{\sum F_i} \cdot 10 \quad (1)$$

Dimana:

- F_i^n = Nilai chain code yang baru
- F_i = Frekuensi kode ke-i
- $\sum F_i$ = total semua frekuensi

TABEL III. HASIL NORMALISASI CHAIN CODE

| | |
|----------------------------|------------|
| Nilai chain code yang baru | 5167230044 |
|----------------------------|------------|

D. Analisis Support Vector Machine (SVM)

Pada tahapan klasifikasi, aplikasi menggunakan LIBSVM sebagai library klasifikasi untuk karakter arab melayu. Gambar 11 menunjukkan alur klasifikasi menggunakan support vector machine.



Gambar 11. Alur proses klasifikasi

1) Representasi Data

Format input dalam penelitian ini adalah 1 1#1 2#0 3#6 4#6 5#6 6#6 7#6 8#6 9#6 10#2 11#2 12#2. Masukan pertama angka 1 merupakan kelas dan angka kedua menyatakan dimensi (row id) dan angka ketiga setelah karakter ('#') menyatakan hasil ekstraksi ciri. Setiap ekstraksi ciri dipisahkan oleh spasi. Adapun pada dimensi (row id) pertama dan dimensi kedua setelah karakter ('#') merupakan hasil jumlah label dan posisi titik, sedangkan pada dimensi ketiga sampai terakhir adalah hasil proses dari freeman chain code. Huruf arab melayu memiliki 34 huruf, kemudian setiap huruf diberi nilai kelas, maka huruf arab melayu memiliki 34 kelas.

2) Proses Training

Sebelum memasuki proses training pada LIBSVM diperlukan inisialisasi untuk objek parameter dan problem. Objek problem merupakan semua nilai \vec{x} dan y,

dimana nilai \vec{x} merupakan vektor hasil ekstraksi ciri dan nilai y merupakan vektor target. Sedangkan objek parameter berisi parameter-parameter yaitu tipe SVM, jenis kernel, nilai C, dan nilai gamma. Data \vec{x} merupakan parameter problem yang berisi nilai-nilai hasil ekstraksi ciri dalam bentuk vektor. Data \vec{x} berisi nilai vektor berdimensi ($L_1 \times L_2$) dimana L_1 merupakan dimensi baris pada ekstraksi ciri setiap kelas dan L_2 dimensi kolom pada ekstraksi ciri setiap kelas. Sedangkan nilai Y merupakan pelatihan untuk kelas pertama dimana kelas pertama pada karakter huruf arab melayu nilai targetnya adalah 1 sedangkan kelas yang lainnya menjadi adalah -1 dan kelas kedua nilai targetnya adalah 1 sedangkan kelas yang lainnya nilai targetnya adalah -1. Demikian seterusnya hingga kelas ke-N. Berikut contoh objek problem yang berisi nilai \vec{x} dan y ditunjukkan pada tabel IV untuk kelas pertama dengan nilai target Y adalah 1.

TABEL IV. OBJEK PROBLEM DATA LATIH

| Kelas | Vektor Hasil Ekstraksi Ciri | Y (target) |
|-------|--|------------|
| 1 | 1#1 2#0 3#6 4#6 5#6 6#6 7#6 8#6 9#6 10#2 11#2 12#2 | 1 |
| 2 | 1#2 2#2 3#6 4#6 5#2 6#2 7#5 8#1 9#7 10#0 11#0 12#3 | -1 |
| 3 | 1#3 2#1 3#6 4#6 5#2 6#2 7#5 8#0 9#0 10#0 11#1 12#4 | -1 |

Adapun langkah-langkah proses pelatihan adalah sebagai berikut:


- Input ekstraksi ciri pada data pelatihan $K(\vec{x}_i, \vec{x}_j)$ ke dalam fungsi kernel RBF pada persamaan 2.17 dengan gamma = 0,02.
- Kemudian hitung kernel RBF.
- Tetapkan nilai C=1000 dan epsilon=0,0001 untuk variabel pada fungsi quadratic programming.
- Hitung solusi quadratic programming untuk mendapatkan nilai α_i dan nilai bias. Berikut hasil α_i dan bias yang didapatkan menggunakan library SVM.

3) Proses Klasifikasi

Setelah melakukan proses training dan didapatkan nilai $\alpha_i > 0$ sebagai support vector dan nilai bias, langkah selanjutnya adalah mendapatkan hasil klasifikasi pada kelas dengan mencari nilai f(x) terbesar. Adapun langkah-langkah untuk untuk proses pengujian adalah sebagai berikut:

- Input ekstraksi ciri pada data pelatihan $K(\vec{x}_i, \vec{x}_d)$ ke dalam fungsi kernel RBF. Dimana \vec{x}_i merupakan support vector yang didapatkan dari proses pelatihan sedangkan \vec{x}_d merupakan data pengujian. Data pengujian untuk klasifikasi sudah dalam bentuk data vector seperti pada tabel V.

TABEL V. DATA PENGUJIAN DALAM BENTUK DATA VEKTOR

| Karakter Uji | Data Vektor |
|---|---|
|  | 1#3 2#1 3#5 4#1 5#6 6#7 7#2 8#3 9#0 10#0 11#4 12#4 |

- b) Hitung $f(x)$.
- c) Lakukan perhitungan pada langkah 1 dan 2 hingga kelas ke-N. Tabel VI adalah hasil perhitungan klasifikasi menggunakan librari SVM.

TABEL VI. HASIL KLASIFIKASI

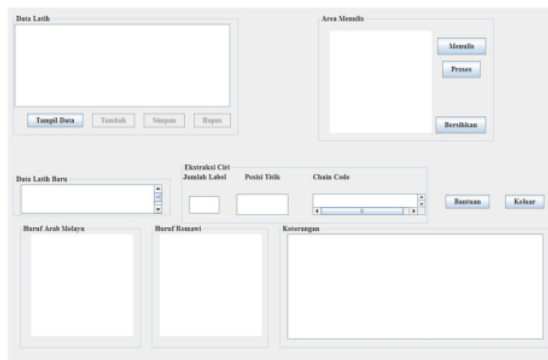
| Kelas | $f(x)$ |
|-------|---------------------|
| 1 | 0.37392553365996606 |
| 2 | 0.24560302312266583 |
| 3 | 0.38047144321736787 |

- d) Cari dan pilih nilai $f(x)$ yang paling maksimal atau nilai yang paling besar pada kelas ke-n.
- e) Kelas ke-n dengan nilai $f(x)$ terbesar merupakan hasil pengenalan karakter. Pada tabel VI terdapat nilai $f(x)$ terbesar pada kelas ke-3. Kelas ke-3 menunjukkan karakter ta (ت). Jadi hasil pengenalan pada citra uji menunjukkan karakter ta (ت).

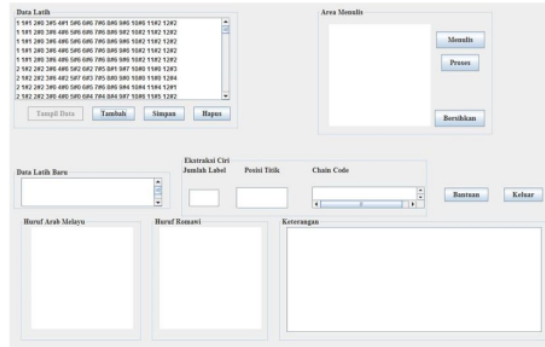
E. Implementasi dan Pengujian

1) Implementasi

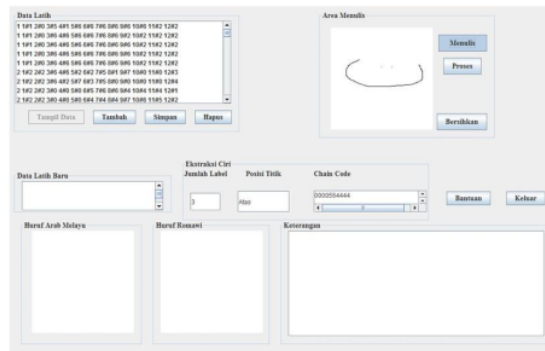
Hasil penelitian berupa aplikasi untuk mengidentifikasi aksara arab melayu seperti yang ditunjukkan oleh gambar 12 yaitu antar muka awal aplikasi, gambar 13 antar muka tampil data latih, gambar 14 antar muka area tulis aksara, gambar 15 antar muka pengenalan aksara, gambar 16 antar muka tambah data latih dan gambar 17 antar muka bantuan aplikasi.



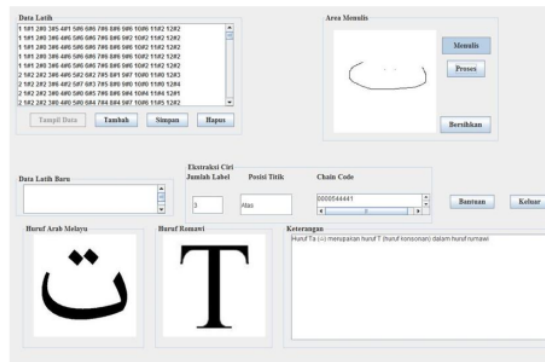
Gambar 12. Antar muka awal aplikasi



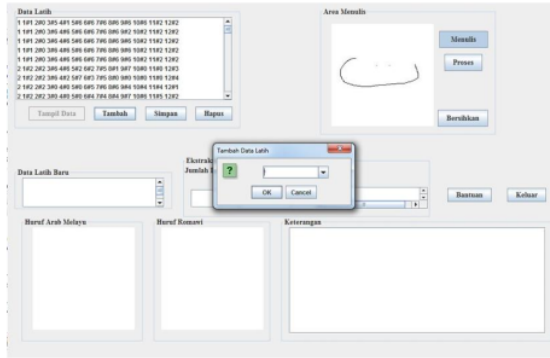
Gambar 13. Antar muka data latih



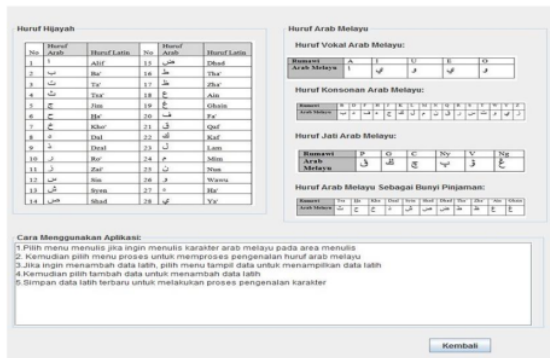
Gambar 14. Antar muka area tulis aksara



Gambar 15. Antar muka pengenalan aksara



Gambar 16. Antar muka tambah data latih



Gambar 17. Antar muka bantuan

3.3 Pengujian aplikasi

Pengujian aplikasi pengenalan aksara arab melayu menggunakan metode pengujian black box dan pengujian akurasi pengenalan aksara. Pengujian black box adalah pengujian aspek fundamental sistem [3] iaitu memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Tabel VII menunjukkan hasil pengujian black box aplikasi yang dibangun.

TABEL VII. PENGUJIAN BLACK BOX APLIKASI

| Butir Uji | Jenis Pengujian | Hasil |
|--|-----------------|----------|
| 1. Proses menulis karakter | Black box | Diterima |
| 2. Proses identifikasi huruf | Black box | Diterima |
| 3. Proses tambah data latih | Black box | Diterima |
| 4. Proses simpan data latih | Black box | Diterima |
| 5. Proses hapus data latih | Black box | Diterima |
| 6. Proses tampil data latih | Black box | Diterima |
| 7. Proses kesalahan tidak ada karakter | Black box | Diterima |
| 8. Proses kesalahan tidak ada data latih | Black box | Diterima |
| 9. Proses kesalahan tidak memilih data latih yang akan dihapus | Black box | Diterima |

Pengujian akurasi aksara arab melayu dilakukan dengan input karakter uji yang sama dengan cara menulis pada canvas

yang sudah disediakan pada aplikasi menggunakan optical mouse, dimana setiap masing-masing karakter di uji memiliki ukuran yang besar, kecil dan miring. Proses pemasukkan data dapat dilihat pada gambar 18, sementara hasil identifikasi aksara arab melayu dapat dilihat pada gambar 19.



Gambar 18. Proses penulisan aksara arab melayu



Gambar 19. Identifikasi aksara arab melayu

Sementara uji akurasi dilakukan untuk mengetahui seberapa akurat aplikasi yang dibangun dapat mengidentifikasi aksara arab melayu dapat dilihat pada Tabel VIII. Pengujian dilakukan dengan memasukkan tulisan tangan aksara arab melayu sebanyak 6 pola untuk masing-masing huruf

TABEL VIII. PENGUJIAN AKURASI AKSARA ARAB MELAYU

| Huruf | Latin | Akurasi | Huruf | Latin | Akurasi |
|-------|-------|---------|--------|-------|---------|
| Alif | A | 100% | Ain | A' | 33% |
| Ba | B | 100% | Ghain | Gh | 33% |
| Ta | T | 83% | Fa | F | 50% |
| Tsa | Ts | 100% | Qaf | Q | 100% |
| Jim | J | 100% | Kaf | K | 83% |
| Ha | H | 50% | Lam | L | 67% |
| Kha | Kh | 100% | Mim | M | 83% |
| Dal | D | 83% | Nun | N | 83% |
| Dzal | Dz | 67% | Wawu | W | 100% |
| Ra | R | 67% | Hamzah | H | 33% |
| Zai | Z | 100% | Ya | Y | 100% |
| Sin | S | 50% | Nya | Ny | 100% |
| Syen | Sy | 83% | Ca | C | 67% |
| Shad | Sh | 33% | Nga | Ng | 83% |
| Dhad | Dh | 83% | Pa | P | 83% |
| Tha | Th | 67% | Gaf | G | 100% |
| Zha | Zh | 83% | Va | V | 100% |

Dari hasil pengujian dengan 6 pola yang berbeda untuk masing-masing aksara, diperoleh kesimpulan bahwa akurasi aplikasi dalam mengidentifikasi aksara arab melayu adalah sebesar 78%.

IV. KESIMPULAN

3

Hasil yang diperoleh menunjukkan aplikasi identifikasi huruf arab melayu menggunakan algoritma freeman chain code dan support vector machine (SVM), maka dapat disimpulkan sebagai berikut: Metode freeman chain code telah dapat melakukan ekstraksi ciri dari tulisan tangan aksara arab melayu yang menjadi data uji. Sementara penggunaan SVM sudah mampu mengidentifikasi huruf arab melayu dan pelafalannya dalam huruf latin dengan akurasi sebesar 78%, sehingga dapat dikatakan aplikasi ini dapat mempermudah pengguna untuk mengidentifikasi aksara arab melayu. Aplikasi yang dikembangkan baru sebatas pengenalan aksara arab melayu, sehingga perlu dikembangkan penelitian lebih lanjut yaitu mengenali penulisan kalimat dalam aksara arab melayu, karena penulisan aksara arab melayu ketika berada diawal, tengah dan akhir kalimat memiliki bentuk yang lebih beragam.

REFERENSI

- [1] M.I. Shofwani, "Mengenal Tulisan Arab Melayu". Penerbit : Adicita Karya Nusa. Yogyakarta. 2005.
- [2] L. Eikvil, Optical Character Recognition. Norsk Regnesentral, 1-33. 1993.
- [3] R. C. Gonzales dan R. E. Woods, "Digital Image Processing". USA: Prentice-Hall, Inc. 2002.
- [4] M.A. Abed, "Freeman chain code contour processing for handwritten isolated Arabic characters recognition". Alrymook University Magazine, Baghdad. 2012.
- [5] H. Izakian, S.A. Monadjemi, B.T. Ladani, and K. Zamanifar, "Multi-font Farsi/Arabic isolated character recognition using chain codes". World Academy of Science, Engineering and Technology. 2008 Sep:43:67-70.
- [6] A.B. Salem, A.A. Sewisy and U.A. Elyan. "A vertex chain code approach for image recognition," *ICGST International Journal on Graphics, vision and Image processing*, vol. 5, no. 3, pp. 1-8, Mar 2005.
- [7] H.I. Khan, U.V. Smitha, & D.S. Kumar, "Isolated Kannada Character Recognition Using Chain Code Features," *International Journal of Science and Research (IJSR)*, vol. 2, no. 8, pp.67-70, August 2013.
- [8] V.D.T Wijaya, S. Novianto, and U.Rosyidah, "Deteksi Huruf Arab Menggunakan Metode Freeman Chain Code," *Techno. Com*, vol. 13, no. 4, pp. 245-250, 2014.
- [9] A.S. Nugroho, A. B. Witarto, dan D. Handoko, *Support vector machine. Teori dan Aplikasinya dalam Bioinformatika*. Ilmu Komputer.com, Indonesia. 2003.
- [10] E. Mayoraz, E. Alpaydin, "Support vector machines for multi-class classification," in *International Work-Conference on Artificial Neural Networks*, Springer, Berlin, Heidelberg, Jun 2 1999, pp. 833-842.
- [11] Panduan Baca Tulis Arab-Melayu. [Online]. Available: https://admin.kemenag.go.id/files/jambi/file/file/pontren/Panduan_Baca_Tulis_Arab-Melayu_untuk_MDTA.pdf [Accessed Juni 8, 2018].

Implementasi Algoritma Freeman Chain Code Dan Support Vector Machine (SVM) Pada Identifikasi Aksara Arab Melayu

ORIGINALITY REPORT

18%

SIMILARITY INDEX

18%

INTERNET SOURCES

3%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

[docplayer.info](#)

Internet Source

5%

2

[garuda.ristekdikti.go.id](#)

Internet Source

5%

3

[id.123dok.com](#)

Internet Source

4%

4

[dokumen.tips](#)

Internet Source

3%

Exclude quotes On

Exclude bibliography On

Exclude matches < 3%