

---

## APLIKASI PENGENALAN TULISAN TANGAN HURUF ARAB MENGGUNAKAN KOMBINASI FITUR DAN TTS UNTUK ANAK USIA DINI

Fitriyatul Qomariyah<sup>1</sup>, Luthfatun Nisa<sup>1</sup>

<sup>1</sup>Pendidikan Agama Islam, Fakultas Tarbiyah, Institut Agama Islam Negeri Madura

<sup>2</sup> Pendidikan Islam Anak Usia Dini, Fakultas Tarbiyah, Institut Agama Islam Negeri Madura

<sup>1</sup>Fitriyatulqomariyah@iainmadura.ac.id , <sup>2</sup>luthfatunnisa@iainmadura.ac.id

### ABSTRAK

Pengenalan tulisan tangan huruf arab menjadi masalah yang cukup sulit dipecahkan. Beberapa huruf dengan bentuk yang sama menjadi kendala dalam proses pengenalan. Gaya, bentuk, dan ukuran tulisan setiap orang yang beranekaragam juga menjadi masalah yang kompleks dalam kasus pengenalan tulisan tangan huruf arab ini. Pemilihan fitur yang cocok sangat dibutuhkan dalam kasus ini untuk mendapatkan hasil yang baik. Pada penelitian ini, metode ekstraksi fitur diusulkan untuk pengenalan tulisan tangan huruf arab. Ekstraksi fitur dilakukan dengan cara mengambil nilai fitur yang sama dari beberapa fitur huruf (huruf yang sama) untuk dijadikan sebagai fitur baru dari huruf tersebut. Kemudian membandingkan setiap fitur data latih terhadap semua fitur data sampel untuk mengklasifikasikan setiap huruf menggunakan City\_Block. Aplikasi ini diharapkan dapat membantu orang tua untuk mengenalkan huruf arab kepada anak.

**Kata kunci:** Pengenalan huruf, tulisan tangan huruf arab, kombinasi fitur, TTS, ekstraksi fitur, City\_Block.

### ABSTRACT

*Handwriting recognition of Arabic letters is a problem that is quite difficult to solve. Several letters with the same shape become an obstacle in the recognition process. The different styles, shapes, and sizes of people's writings are also a complex problem in the case of handwriting recognition of Arabic letters. Selection of suitable features is needed in this case to get good results. In this study, a feature extraction method is proposed for handwriting recognition of Arabic letters. Feature extraction is done by taking the same feature value from several letter features (the same letter) to be used as a new feature of the letter. Then compare each training data feature against all sample data features to classify each letter using City\_Block. This application is expected to help parents to introduce Arabic letters to children.*

**Keywords:** Character Recognition, Handwriting of Arabic Character, Feature Combination, TTS, Feature Extraction, City\_Block.

### 1. PENDAHULUAN

*Optical Character Recognition (OCR)* adalah mesin elektronik untuk menerjemahkan tulisan tangan, menjadi teks yang dapat dibaca dan disunting oleh komputer [1]. Tujuan utama dari sistem Optical Character Recognition (OCR) adalah untuk menirukan kemampuan manusia dalam hal membaca. Oleh sebab itu, sistem OCR dianggap sebagai cabang dari kecerdasan buatan dan sebuah cabang dari visi komputer [1]. Pengenalan tulisan tangan huruf Arab pada Optical Character Recognition (OCR) menjadi focus penelitian akhir-akhir ini. Meskipun demikian, pengenalan tulisan tangan huruf arab memiliki tingkat kesulitan yang lebih dibandingkan pengenalan huruf abjad lainnya. Bentuk dan ukuran setiap karakter dari tulisan orang yang berbeda menjadi masalah utama di dalam pengenalan tulisan tangan huruf arab [2].

Banyak metode yang digunakan dalam proses pengenalan tulisan tangan huruf arab salah satunya adalah dengan menggunakan ekstraksi fitur [3]. Pemilihan fitur yang cocok akan sangat berpengaruh terhadap proses pengenalan khususnya dalam pengenalan tulisan tangan huruf arab [4]. Bentuk huruf arab yang hampir sama pada beberapa hurufnya dan gaya tulisan setiap orang yang berbeda-beda menjadi

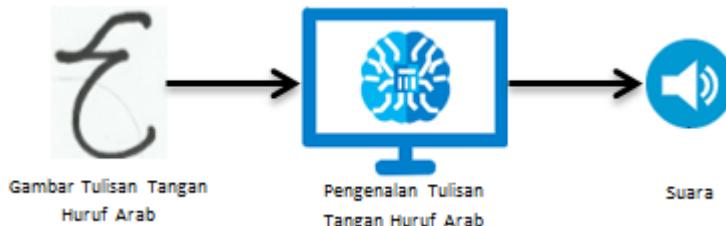
sebuah permasalahan yang kompleks didalam pemilihan fitur yang cocok dalam proses pengenalan untuk mendapatkan hasil yang baik.

Beberapa penelitian telah dilakukan. Diantaranya dilakukan oleh Lawgali and Bouridane. Mereka melakukan penelitian untuk mengenali tulisan tangan huruf arab menggunakan ekstraksi fitur *Discrete Cosine Transform* (DCT) dan *Discrete Wavelet Transform* (DWT)[5]. Hasil akurasi yang didapat pada penelitian ini sebesar 94,87% untuk ekstraksi fitur menggunakan DCT dan 59,81% untuk DWT pada gambar dengan ukuran 64x64. M. A. Abdullah dan H. H. Al-Fraidi et al juga melakukan penelitian untuk mengenali tulisan tangan huruf arab[6]. Pada penelitiannya, pengenalan huruf arab dilakukan dengan beberapa tahap, diantaranya adalah dengan mengekstraksi bentuk dan fitur dari huruf tersebut. Dengan metode tersebut, akurasi yang dicapai sebesar 81%. Penelitian yang lain juga dilakukan oleh Sahlol dan Suen, mereka melakukan penelitian untuk mengenali tulisan tangan huruf arab menggunakan integral proyeksi (vertical dan horizontal proyeksi) dan beberapa fitur lainnya dengan akurasi sebesar 88% [7].

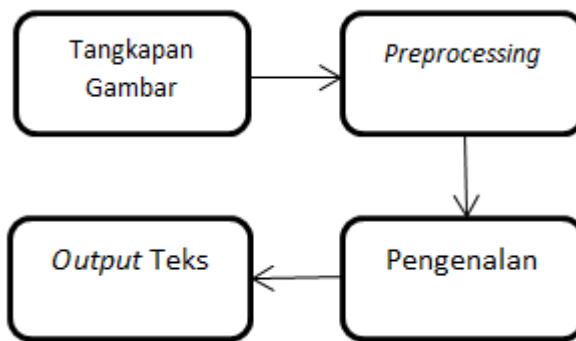
Pada penelitian ini, peneliti mengusulkan sebuah metode ekstraksi fitur untuk proses pengenalan tulisan tangan huruf arab. Pengambilan fitur dilakukan dengan cara menggabungkan beberapa fitur huruf yang sama dari beberapa tulisan tangan orang yang berbeda menjadi satu fitur baru. Fitur baru tersebut kemudian dijadikan sebagai fitur dari sebuah huruf tertentu. Data yang digunakan pada penelitian ini adalah 113 bentuk huruf dengan 10 sampel tulisan tangan yang berbeda pada setiap hurufnya, sehingga jumlah total sampel yang digunakan adalah sebanyak 1130. Pada penelitian sebelumnya, peneliti melakukan proses pengenalan tulisan tangan huruf arab dalam bentuk teks. Pada penelitian ini, penulis mengembangkannya menjadi aplikasi dapat mengeluarkan suara sebagai terjemahan hasil teks, yang tidak hanya menghasilkan teks seperti penelitian sebelumnya. Penelitian ini difokuskan pada pengenalan tulisan tangan huruf arab.

## 2. METODOLOGI PENELITIAN

Aplikasi Pengenalan Tulisan Tangan Huruf Arab ini dibangun dengan menggunakan 113 bentuk huruf dengan 10 sampel tulisan tangan yang berbeda pada setiap hurufnya. Usulan tahapan umum sistem pengenalan ditunjukkan pada Gambar 1. Huruf arab yang akan dikenali pada aplikasi ini merupakan tulisan tangan huruf arab.

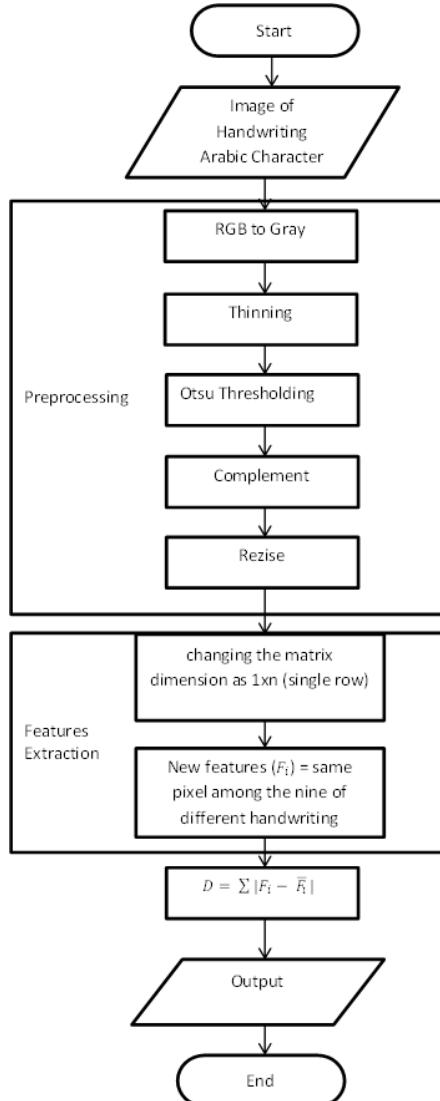


Gambar 1. Tahapan Umum Sistem



Gambar 2. Blok Diagram Pengenalan Tulisan Tangan Huruf Arab

Dari hasil tangkapan gambar, gambar diproses untuk proses pengenalan. Diagram blok pengenalan dapat dilihat pada Gambar 2. Sedangkan untuk proses pengenalan dapat dilihat pada Gambar 3. proses ini dilakukan untuk memproses gambar menjadi teks huruf arab. Pertama, sistem melakukan proses preprocessing, kedua, proses ekstraksi fitur , dan terakhir proses pengenalan.



**Gambar 3. Design System Pengenalan Tulisan Tangan Huruf Arab**

Preprocessing adalah proses awal perbaikan sebuah citra untuk menghilangkan noise. Seperti yang dikemukakan oleh Bahri dan Maliki, preprocessing adalah proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada citra input untuk diproses lebih lanjut [8]. preprocessing pada penelitian ini melalui beberapa proses yaitu *RGB to Gray*, *Otsu Thresholding*, *Complement*, *Thinning*, dan *Resize* gambar. Preprocessing adalah untuk memudahkan peneliti mengidentifikasi setiap karakter tulisan tangan arab.

Gambar RGB adalah gambar dimana setiap pikselnya terdiri dari 3 nilai warna (merah, hijau, dan biru) [9]. RGB to Gray adalah proses mengkonversi nilai RGB (merah, hijau, dan biru) menjadi grayscale.

Luminance adalah algoritma standar yang digunakan untuk memproses gambar. Hal ini digunakan oleh MATLAB dengan fungsi "rgb2gray" yang sering digunakan dalam visi komputer. Luminance [10] dirancang untuk mencocokkan persepsi kecerahan manusia dengan menggunakan kombinasi bobot dari RGB:

$$G_{\text{Luminance}} = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

Dimana R adalah nilai warna merah, G adalah nilai warna hijau, dan B adalah nilai warna biru.

*Otsu Thresholding* adalah sebuah metode untuk membedakan antara *foreground* dan *background* dengan menggunakan nilai *threshold*. Akan tetapi, pencerahan yang kurang baik dan warna *background* yang tidak sama akan menyulitkan didalam menentukan nilai *threshold*. Misalnya diasumsikan bahwa C1 adalah sebuah citra asli yang memiliki daerah terang dan daerah gelap, Sedangkan P(s) adalah probabilitas dari setiap daerah. maka dapat dirumuskan sebagai berikut [11]:

$$P_1(s) = \sum_{i=0}^s P_i \quad \text{for } C_1[s > 0] \quad (2)$$

$$P_2(s) = \sum_{i=s+1}^{L-1} P_i = 1 - P_1(s) \quad \text{for } C_2[s + 1, L - 1] \quad (3)$$

Dimana P\_1(s) dan P\_2(s) adalah probabilitas dari daerah gelap dan daerah terang pada daerah C\_1 dan C\_2, sedangkan P\_i adalah histogram normal dari keseluruhan ukuran citra. Dalam menentukan nilai *threshold* k, *bimodal histogram, statistics stationary, hence, adaptive and uniform illumination area* dapat dimodifikasi untuk mendapatkan nilai *threshold* dengan mengikuti persamaan :

$$k = \frac{\sigma^2_B}{\sigma^2_G} \quad (4)$$



**Gambar 4. Gambar RGB (a) dan Gambar Biner (b)**

Dimana k adalah nilai *threshold*,  $\sigma^2_B$  adalah keragaman global dari keseluruhan gambar, sedangkan  $\sigma^2_G$  adalah keragaman antar kelas.

Proses *complement image* yaitu proses mengubah nilai piksel 0 menjadi 1 dan sebaliknya seperti yang ditunjukkan pada Gambar 5.



**Gambar 5. Gambar Biner (a) dan Gambar Complement (b)**

Algoritma Thinning ini adalah algoritma untuk menipiskan piksel dari objek pada gambar. Algoritma ini dilakukan dengan melakukan perulangan terhadap lapisan piksel sampai tidak ada piksel yang dapat dihapus lagi. Biasanya proses pengecekan dilakukan sebanyak dua kali, jika tidak ada perubahan maka proses perulangan berhenti [12]. Algoritma ZS merupakan algoritma thinning yang diusulkan oleh Zhang dan Suen pada tahun 1984 dengan menggunakan 3x3 ketetanggaan seperti yang ditunjukkan pada Gambar 6 [13].

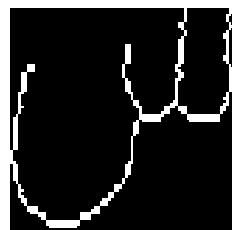
P8	P2	P3
P7	P1	P4
P6	P1	P5

Gambar 5. ZS 3x3 *neighborhood*



Gambar 6. Gambar *Complement* (a) dan Gambar *Thinning* (b)

Resize gambar merupakan preprocessing yang terakhir. Proses ini mengubah ukuran piksel setiap huruf arab menjadi 81x81 piksel. Proses ini dilakukan untuk memudahkan sistem dalam mengolah gambar tersebut pada tahap selanjutnya.

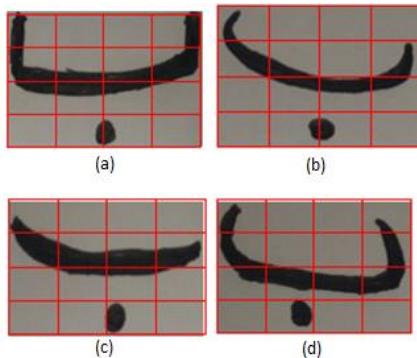


Gambar 7. Seen 81x81 Piksel

Setelah semua proses preprocessing telah dilakukan, proses selanjutnya adalah proses ekstraksi fitur. Pada proses pengenalan huruf arab di bidang computer vision, Bentuk beberapa huruf yang sama menjadi masalah yang cukup sulit untuk dipecahkan. Selain itu, masalah yang kompleks juga terjadi pada kasus tulisan tangan huruf arab dimana setiap orang memiliki bentuk dan ukuran tulisan yang berbeda. Oleh karena itu, pemilihan fitur yang baik sangat dibutuhkan dalam proses pengenalan untuk mendapatkan hasil yang baik.

Pada penelitian ini, suatu ekstraksi fitur baru diusulkan untuk proses pengenalan tulisan tangan huruf arab. Data sampel yang digunakan sebanyak 102 bentuk huruf dimana setiap huruf ditulis oleh 10 orang berbeda (total data sampel adalah 102x10). Pertama-tama, dilakukan pengkodean data latih citra huruf arab dengan ukuran 15x30 menjadi kode biner (bernilai 0 dan 1) dimana foreground bernilai 1 dan background bernilai 0 seperti contoh pada Gambar 4 dan Gambar 5. Kami menggunakan 10 bentuk tulisan huruf yang berbeda di dalam satu karakter.

Kemudian dilakukan pengambilan fitur dengan cara mengubah dimensi matrik menjadi 1 x n (satu baris) seperti pada Gambar 10. Setelah itu, langkah selanjutnya adalah mencari nilai piksel yang sama dari kesembilan bentuk tulisan huruf untuk dijadikan sebagai fitur dari karakter tersebut seperti yang digambarkan pada Gambar 11 dimana piksel yang diberi warna pink merupakan nilai piksel yang sama.



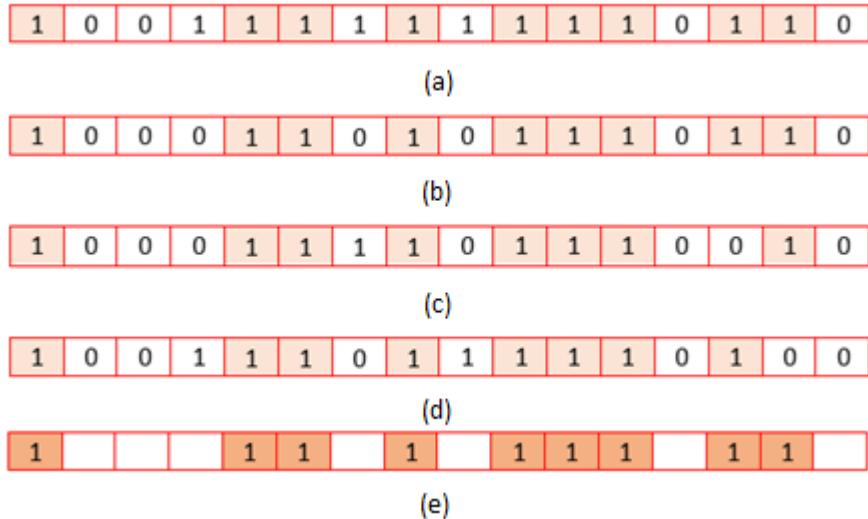
**Gambar 8. Contoh Gambar Huruf Baa**

<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	0	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	0	0	1	1	0	1	0	1	1	1	0	1	1	0
1	0	0	1																														
1	1	1	1																														
1	1	1	1																														
0	1	1	0																														
1	0	0	0																														
1	1	0	1																														
0	1	1	1																														
0	1	1	0																														
<p style="text-align: center;">(a)</p>	<p style="text-align: center;">(b)</p>																																
<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	0	0	1	1	1	1	0	1	1	1	0	0	1	0	<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	1	0	0	1	1	1	0	1	1	1	1	1	0	1	0	0
1	0	0	0																														
1	1	1	1																														
0	1	1	1																														
0	0	1	0																														
1	0	0	1																														
1	1	0	1																														
1	1	1	1																														
0	1	0	0																														
<p style="text-align: center;">(c)</p>	<p style="text-align: center;">(d)</p>																																

**Gambar 9. Gambar Biner Huruf Baa**

<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	0	<p style="text-align: center;">(a)</p>
1	0	0	1	1	1	1	1	1	1	1	1	0	1	1	0		
<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr></table>	1	0	0	0	1	1	0	1	0	1	1	1	0	1	1	0	<p style="text-align: center;">(b)</p>
1	0	0	0	1	1	0	1	0	1	1	1	0	1	1	0		
<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table>	1	0	0	0	1	1	1	1	0	1	1	1	0	0	1	0	<p style="text-align: center;">(c)</p>
1	0	0	0	1	1	1	1	0	1	1	1	0	0	1	0		
<table border="1"><tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>	1	0	0	1	1	1	0	1	1	1	1	1	0	1	0	0	<p style="text-align: center;">(d)</p>
1	0	0	1	1	1	0	1	1	1	1	1	0	1	0	0		

**Gambar 10. 1xn Gambar Biner**



**Gambar 11. Piksel yang Sama pada 1xn Gambar Biner**

Setelah semua fitur data sampel dari proses sebelumnya didapatkan, maka langkah selanjutnya adalah membandingkan setiap fitur data latih dengan semua data sampel dengan menggunakan *City\_Block*. *City\_Block* merupakan algoritma yang digunakan untuk menghitung jarak atau tingkat kemiripan dua objek. Rumus *City\_Block* dapat dilihat pada persamaan berikut [14]:

$$D = \sum |F_i - \bar{F}_i| \quad (5)$$

Dimana:

$D$ =tingkat perbedaan kedua objek,

$F_i$ =Objek input, dan

$\bar{F}_i$  = Objek pembanding/output

Hasil dari proses pengenalan yang berupa teks selanjutnya dikonversi menjadi suara. Proses ini menggunakan metode *Text To Speech* (TTS). TTS merupakan metode untuk menciptakan sinyal suara yang berasal dari masukan teks [15]. Adapun keunggulan metode ini ialah memori yang dibutuhkan sedikit serta mudah diatur[16]. Ilustrasi konversi menggunakan TTS digambarkan pada Gambar 12. Pertama buatlah File suara dalam format wav dengan jumlah total 29 file kemudian simpan dalam database. Banyak file tersebut didasarkan pada huruf arab dari Alif sampai Ya (Hamzah dihitung sebagai huruf tersendiri)[17]. Untuk menentukan kesamaan antara teks dengan file wav, lakukan komparasi antara fonem teks dengan file wav. Terakhir, suara akan dimainkan dengan penggabungan antara ekstraksi fonem teks dengan file wav yang sesuai.



Gambar 12. Ilustrasi TTS

### 3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan 101 sampel huruf arab yang masing-masing ditulis oleh 15 orang dengan tulisan tangan yang berbeda (total sampel adalah 101x15). Data yang digunakan dalam penelitian ini dilakukan preprocessing terlebih dahulu, preprocessing yang dilakukan meliputi konversi elemen RGB (merah, hijau, dan biru) menjadi grayscale menggunakan persamaan (1), membedakan foreground dengan background menggunakan nilai threshold (2-4) , pengubahan nilai 0 menjadi 1 atau sebaliknya menggunakan algoritma komplemen, redaman piksel citra menggunakan algoritma thinning , yang terakhir dilakukan preprocessing resize data citra tulisan tangan karakter arab dengan ukuran yang sama yaitu 81x81. Contoh data yang digunakan dalam penelitian ini ditunjukkan pada Tabel 1.

Tabel 1. Data Tulisan Tangan Huruf Arab

No	Character	First	Middle	End	Isolated
1	Alif	-		ج	ا
2	Baa	ا	ا	ا	ا
3	Ta	ت	ت	ت	ت
4	Tha	ث	ث	ث	ث
5	Gheem	ج	ج	ج	ج
6	Ha'a	ح	ح	ح	ح
7	Kh'a	خ	خ	خ	خ
8	Dal	-	-	د	د
9	Dhal	-	-	ذ	ذ

10	Ra	-	-	ر	ر
11	Zeen	-	-	ز	ز
12	Seen	س	س	س	س
13	Sheen	ش	ش	ش	ش
14	Sad	ص	ص	ص	ص
15	Dad	ض	ض	ض	ض
16	Tah	ط	ط	ط	ط
17	Dhad	ظ	ظ	ظ	ظ
18	Aen	ع	ع	ع	ع
19	Ghen	غ	غ	غ	غ
20	Fa	ف	ف	ف	ف
21	Qaf	ق	ق	ق	ق
22	Kaf	ك	ك	ك	ك
23	Lam	ل	ل	ل	ل
24	Meem	م	م	م	م
25	Noon	ن	ن	ن	ن
26	Ha	ه	ه	ه	ه
27	Waw	-	-	و	و
28	Ya	ي	ي	ي	ي

Tabel di atas merupakan karakter arab dengan berbagai bentuk diantaranya bentuk ketika huruf berdiri sendiri, ketika di awal, di tengah dan di akhir kata. Huruf Arab yang digunakan sebagai sampel adalah gambar tulisan tangan Huruf Arab.

pertama, kita lakukan preprocessing pada semua sampel tulisan tangan Arab. Setelah preprocessing, proses selanjutnya adalah ekstraksi fitur dan pengenalan menggunakan *City\_Block* (5). Hasil tulisan tangan karakter Arab disajikan dengan menggunakan perhitungan akurasi sebagai berikut:

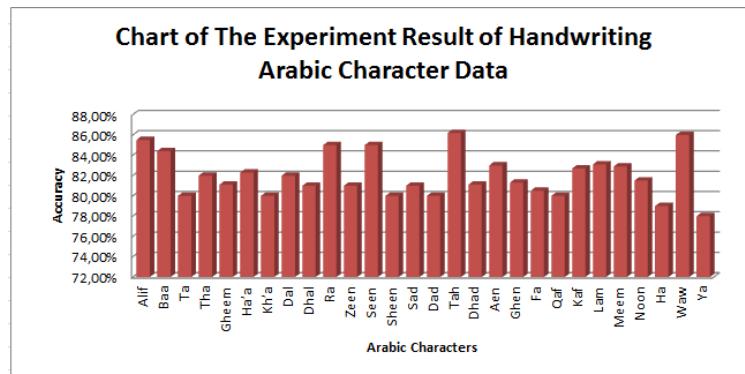
$$\text{System Performance} = \frac{\text{The Same Pixels of input and fitur}}{\text{All Pixels of input and fitur}} \times 100\%$$

Dari hasil perhitungan diatas, didapatkan nilai akurasi setiap sampel huruf yang digunakan pada penelitian ini. Hasil akurasi dapat dilihat pada Tabel 2.

**Tabel 2. Akurasi Tulisan Tangan Huruf Arab**

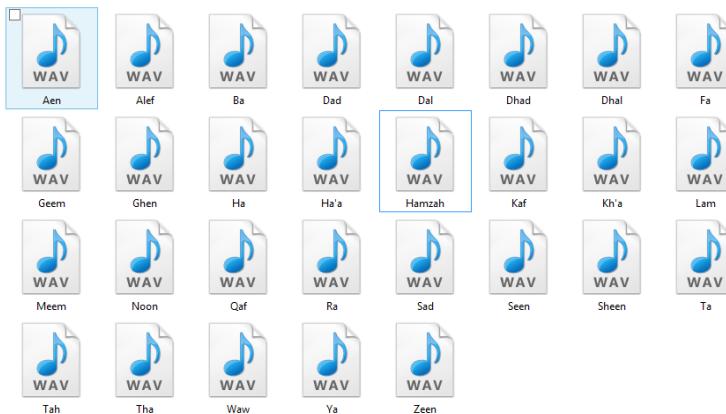
No	Huruf	Akurasi
1	Alif	85,51%
2	Baa	84,41%
3	Ta	80%
4	Tha	82%
5	Gheem	81,1%
6	Ha'a	82,30%
7	Kh'a	80%
8	Dal	82%
9	Dhal	81%
10	Ra	85%
11	Zeen	81%
12	Seen	85%
13	Sheen	80%
14	Sad	81%
15	Dad	80%

16	Tah	86,2%
17	Dhad	81,1%
18	Aen	83%
19	Ghen	81,3%
20	Fa	80,5%
21	Qaf	80%
22	Kaf	82,7%
23	Lam	83,1%
24	Meem	82,9%
25	Noon	81,5%
26	Ha	79%
27	Waw	86%
28	Ya	78%



**Gambar 13. Diagram Hasil Akuarasi Tulisan Tangan Arab**

Dari hasil yang diperoleh, dengan menggunakan Fitur Kombinasi, huruf Ya dan Ha memiliki nilai akurasi paling kecil dibandingkan huruf lainnya, yaitu sekitar 78% dan 79%. Hal ini dikarenakan bentuk huruf yang hampir mirip dengan huruf lainnya, tingkat kemiringan tulisan, posisi titik-titik yang tidak sama dari 10 sampel, serta gaya penulisan yang berbeda. Jadi, kemiripan bentuk tulisan, kemiringan, dan ukuran tulisan juga sangat mempengaruhi terhadap tingkat akurasi. Hasil dari proses pengealan yang berupa teks Hijaiyyah dai Alef sampai Ya selanjutnya dikonversi menjadi luaran suara. Pada ini, langkah pertama membuat database file suara dari keseluruhan huruf seperti pada Gambar 14. Teks luaran dari proses pengenalan kemudian dikomparasikan dengan file wav yang telah tersedia dalam database. Tabel 3 menunjukkan hasil akurasi pengenalan dan konversi suara.



**Gambar 14. Suara setiap Huruf Hijaiyyah dalam Bentuk Wav**

**Tabel 2. Hasil Akurasi Pengenalan Dan Konversi Suara**

No	Huruf	Akurasi Pengenalan	Akurasi Suara
1	Alif	85,51%	85,51%
2	Baa	84,41%	84,41%
3	Ta	80%	80%
4	Tha	82%	82%
5	Gheem	81,1%	81,1%
6	Ha'a	82,30%	82,30%
7	Kh'a	80%	80%
8	Dal	82%	82%
9	Dhal	81%	81%
10	Ra	85%	85%
11	Zeen	81%	81%
12	Seen	85%	85%
13	Sheen	80%	80%
14	Sad	81%	81%
15	Dad	80%	80%

16	Tah	86,2%	86,2%
17	Dhad	81,1%	81,1%
18	Aen	83%	83%
19	Ghen	81,3%	81,3%
20	Fa	80,5%	80,5%
21	Qaf	80%	80%
22	Kaf	82,7%	82,7%
23	Lam	83,1%	83,1%
24	Meem	82,9%	82,9%
25	Noon	81,5%	81,5%
26	Ha	79%	79%
27	Waw	86%	86%
28	Ya	78%	78%

Tabel 3 menunjukkan bahwa nilai akurasi dari pengenalan yang luarannya berbentuk teks dan nilai akurasi dari hasil pengenalan dari teks dan dikonversi ke suara sama. Hal ini dikarenakan output dalam proses pengenalan yang masih berupa teks akan selalu megeluarkan suara sesuai dengan output teks dari proses pengenalan.

#### 4. KESIMPULAN

Dari hasil percobaan dapat disimpulkan bahwa pengenalan tulisan tangan huruf arab menggunakan metode kombinasi fitur sangat baik untuk mengenali huruf arab tulisan tangan. Dalam penelitian ini, kami menggunakan ukuran gambar 81x81 piksel. Kami mendapatkan nilai rata-rata akurasi yang cukup baik yaitu diatas 80%. Kualitas data yang baik akan meningkatkan akurasi. Akurasi dipengaruhi oleh beberapa faktor, seperti bentuk huruf, tingkat kemiringan tulisan, posisi titik-titik, serta gaya penulisan. Penulis juga mengembangkan aplikasi sampai mengeluarkan suara,

---

harapannya aplikasi ini dapat digunakan untuk membantu anak usia dini dalam mengenali huruf Arab. Beberapa rekomendasi untuk mengembangkan aplikasi ini adalah user interface yang menarik sehingga anak-anak lebih suka menggunakan. Perbaikan data tulisan tangan sehingga nilai akurasi semakin baik.

## DAFTAR PUSTAKA

- [1] H. Al-ani, N. Ban, and H. M. Abass, “Journal of Computing::Printed Arabic Character Recognition using Neural Network,” vol. 5, no. 1, pp. 64–66, 2014.
- [2] a Sahloul and C. Suen, “Off-Line System for the Recognition of Handwritten Arabic Character,” *Fourth Int. Conf. Comput. Sci. Inf. Technol. (CCSIT 2014)*, pp. 227–244, 2014.
- [3] C. Science and M. M. Richter, “A NEW FEATURE EXTRACTION METHOD FOR TMNN-BASED ARABIC CHARACTER CLASSIFICATION Khalid Saeed Majida AlBakoor,” vol. 26, pp. 403–420, 2007.
- [4] M. A. Abuzaraida and A. M. Zeki, “Feature Extraction Techniques of Online Handwriting Arabic Text Recognition,” ... *Muslim World (ICT4M), 2013* ..., 2013.
- [5] M. A. Mohammed, M. R. Kumar, and R. Pradeep, “Text Line Segmentation of Arabic Handwritten Documents using Line Height Method,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 11, pp. 170–174, 2014.
- [6] M. A. Abdullah, L. M. Al-Harigy, H. H. Al-Fraidi, M. a Abdullah, L. M. Al-harigy, and H. H. Al-fraidi, “Off-Line Arabic Handwriting Character Recognition Using Word Segmentation,” *J. Comput.*, vol. 4, no. 3, pp. 40–44, 2012.
- [7] A. Sahlol and C. Suen, “A Novel Method for the Recognition of Isolated Handwritten Arabic Characters,” *arXiv Prepr. arXiv1402.6650*, 2014.
- [8] R. S. Bahri and I. Maliki, “Perbandingan Algoritma Template Matching Dan Feature Extraction Pada Optical Character Recognition,” *J. Komput. dan Inform.*, vol. 1, no. 1, pp. 29–35, 2012.
- [9] F. Qomariyah, F. Utaminingrum, and W. F. Mahmudy, “The segmentation of printed Arabic characters based on interest point,” *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2–8, pp. 19–24, 2017.
- [10] M. Balsky, “Luminance values extraction from digital images,” *Proc. 2016 IEEE Light. Conf. Visegr. Countries, Lumen V4 2016*, pp. 1–3, 2016.
- [11] J. Yoo, I. Karpov, S. Lee, J. Jung, H. S. Kim, and H. Hwang, “Threshold Voltage Drift in Te-Based Ovonic Threshold Switch Devices under Various Operation Conditions,” *IEEE Electron Device Lett.*, vol. 41, no. 1, pp. 191–194, 2020.
- [12] M. Wang, L. Zhenglin, S. Fuyuan, and G. Lei, “An Improved Image Thinning Algorithm and Its Application in Chinese Character Image Refining,” no. Itnec, pp. 1870–1874, 2019.
- [13] R. M. Haralick, “A Fast Parallel Algorithm for Thinning Digital Patterns,” vol. 27, no. 3, pp. 236–239, 1984.
- [14] T. Vardoulaki, M. Vakalopoulou, and K. Karantzalos, “Per city-block, density estimation at build-up areas from aerial RGB imagery with deep learning,” *2017 Jt. Urban Remote Sens. Event, JURSE 2017*, pp. 0–3, 2017.
- [15] S. Sawant and M. Deshpande, *English text to speech synthesizer using concatenation technique*, vol. 905. Springer Singapore, 2018.
- [16] A. Jacob and P. Mythili, “Developing a Child Friendly Text-to-Speech System,” *Adv. Human-Computer Interact.*, vol. 2008, pp. 1–6, 2008.
- [17] R. I. O. Ahmed and M. E. M. Musa, “Preprocessing Phase for Offline Arabic Handwritten Character Recognition,” *Int. J. Comput. Appl. Technol. Res.*, vol. 5, no. 12, pp. 760–763, 2016.