

BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisa Pengenalan Karakter Huruf Hijaiyah

Pada tahapan ini akan dilakukan analisa proses untuk pengenalan karakter huruf hijaiyah. Pada tahapan ini akan dilakukan pembahasan mengenai data, proses, hasil dan semua yang berhubungan dengan penelitian. Analisa proses yang dilakukan pada penelitian ini dapat dilihat sesuai dengan Gambar 3.2.


4.1.1 Analisa Data Collection

Analisis *data collection* merupakan proses awal untuk *input* pola karakter huruf hijaiyah yang memiliki 28 karakter. Pola karakter yang dimasukkan pada aplikasi yang dibuat terlihat dapat dilihat pada Tabel 2.1.

4.1.2 Analisa Pre-processing

Setelah pola karakter huruf hijaiyah telah digambarkan di area *canvas* pada aplikasi, maka tahap selanjutnya yang dilakukan adalah tahap *pre-processing*. Analisa proses *pre-processing* yang dilakukan yaitu konversi RGB ke *grayscale* dan konversi *grayscale* ke biner. Selanjutnya akan dilakukan tahap pemotongan objek yang diperlukan dari sebuah citra dengan cara *cropping* dan *resize* untuk mengubah ukuran citra.

1. Konversi RGB ke *Grayscale*

Setelah proses memasukkan pola karakter huruf hijaiyah dengan cara menggambar di area *canvas* pada aplikasi, maka gambar tersebut secara otomatis memiliki ciri warna RGB. Oleh karena itu gambar pola karakter huruf hijaiyah tersebut harus dikonversi ke *grayscale* terlebih dahulu. Sebagai contoh, pola karakter yang dimasukkan pada *data collection* adalah pola karakter huruf tunggal (*isolated*) Ba (). Gambar 4.1 berikut merupakan contoh gambar yang dikonversi RGB ke *grayscale*.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 4.1 Contoh Gambar RGB

Gambar 4.2 adalah contoh gambar dengan ciri warna RGB yang memiliki ukuran 2x2 piksel. Nilai R, G, dan B dari masing-masing piksel tersebut berbeda-beda yaitu dapat dilihat pada Tabel 4.1 di bawah ini:

Tabel 4.1 Nilai RGB

Piksel (1,1)	Piksel (1,2)
R = 104	R = 109
G = 210	G = 220
B = 140	B = 150
Piksel (2,1)	Piksel (2,2)
R = 105	R = 119
G = 222	G = 200
B = 150	B = 151

Dengan menggunakan Persamaan rumus 2.1 (rumus konversi RGB ke *grayscale*), maka akan dilakukan perhitungan sebagai berikut:

$$\begin{aligned} Grayscale\ Pixel_{(1,1)} &= (0,2989 * 104) + (0,5870 * 210) + (0,1141 * 140) \\ &= 170,3296 \end{aligned}$$

$$\begin{aligned} Grayscale\ Pixel_{(1,2)} &= (0,2989 * 109) + (0,5870 * 220) + (0,1140 * 150) \\ &= 178,8351 \end{aligned}$$

$$\begin{aligned} Grayscale\ Pixel_{(2,1)} &= (0,2989 * 105) + (0,5870 * 222) + (0,1141 * 150) \\ &= 178,8135 \end{aligned}$$

$$\begin{aligned} Grayscale\ Pixel_{(2,2)} &= (0,2989 * 119) + (0,5870 * 200) + (0,1141 * 151) \\ &= 170,1982 \end{aligned}$$

Hasil dari perhitungan konversi RGB ke *grayscale* yang telah dilakukan, maka terlihat perbedaan nilai dari masing-masing piksel. Nilai hasil konversi RGB ke *grayscale* tersebut dapat dilihat pada Tabel 4.2 berikut.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Tabel 4.2 Nilai Grayscale

Piksel (1,1)	Piksel (1,2)
170,3296	178,8351
Piksel (2,1)	Piksel (2,2)
178,8135	170,1982

Berikut adalah *pseudocode* untuk konversi pola karakter huruf hijiyah dari ciri warna RGB menjadi *grayscale*:

```

PROGRAM KONVERSIGBKEGRAYSCALE
DEKLARASI
Img;           { gambar yang dimasukkan}
r;             { representasi nilai red (merah) }
g;             { representasi nilai green (hijau) }
b;             { representasi nilai blue (biru) }
gray;          { representasi nilai grayscale yang akan dicari}

ALGORITMA
R (i,:) = (:, :, 1);
g (i,:) = (:, :, 2);
b (i,:) = (:, :, 3);
gray (i,:) = (0.2989 * r) + (0.5870 * g) + (0.1141 * b)
    
```

Gambar 4.2 Pseudocode Konversi RGB Ke Grayscale

2. Konversi *Grayscale* ke Biner

Setelah proses konversi RGB ke *grayscale* selesai dilakukan, maka proses selanjutnya yang akan dilakukan adalah konversi dari *grayscale* ke biner dengan memakai hasil perhitungan yang telah didapat sebelumnya. Dengan menggunakan Persamaan rumus 2.2 yaitu rumus konversi *grayscale* ke biner dan dengan nilai ambang (T)= 128, maka hasil perhitungannya adalah sebagai berikut:

$$Grayscale\ Pixel_{1,1} = 170.3296 \geq T = 1$$

$$Grayscale\ Pixel_{1,2} = 178.8351 \geq T = 1$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$\text{Grayscale Pixel}_{2,1} = 178.8135 \geq T = 1$$

$$\text{Grayscale Pixel}_{2,2} = 170,1982 \geq T = 1$$

Dari hasil perhitungan di atas, didapatkan nilai citra biner seperti yang dapat dilihat pada Tabel 4.3 berikut:

Tabel 4.3 Nilai Citra Biner

1	1
1	1

Berikut adalah *pseudocode* untuk konversi pola karakter huruf hijaiyah dari *grayscale* kebiner:

```

PROGRAM KONVERSIGRAYKEBINER
DEKLARASI
img_gray;           { matriks nilai grayscale }
T;                  { nilai ambang}

ALGORITMA
T= 128;
    If(img_gray (i,:) > = T
        img_bin (i,:) = 0;
    else
        img_bin (i,:) = 1;
    end
    
```

Gambar 4.3 Pseudocode Konversi Grayscale Ke Biner

3. Cropping dan Resize

Proses *cropping* bertujuan untuk mendapatkan objek karakter huruf hijaiyah, yang selanjutnya akan dilakukan proses *resize* untuk mengubah ukuran semula menjadi matriks berukuran $m \times n$ piksel. Kemudian dilakukan *resize* dengan menggunakan matriks 100×100 piksel. Berikut merupakan *pseudocode* proses *cropping* dan *resize*.

- Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

```

PROGRAM CROPPING
DEKLARASI
S1;           {panjang}
S2;           {lebar}
Yi;           {Matriks 1xn total vertical }
Xi;           {Matriks nx1 total horizontal }

ALGORITMA
for i=1 to s1 do
    total ← s2*1
    A ← sum xi
    If A ≠ total
    else remove xi
endfor

for i=1 to s2 do
    total ← s1*1
    B ← sum yi
    If B ≠ total
    else romove yi
  
```

Gambar 4.4 Pseudocode Proses Cropping

Setelah proses *cropping* selesai, maka akan dilakukan *resize* ke dalam matriks 100x100 piksel. Berikut merupakan *pseudocode* untuk tahap *resize*:

```

PROGRAM RESIZE
DEKLARASI
Img           {matriks citra hasil cropping}
Img_baru      {citra hasil resize}

Algoritma
Img_baru= imresize (img,[100 100]);
  
```

Gambar 4.5 Pseudocode Proses Resize

Hak Cipta Dilindungi Undang-Undang

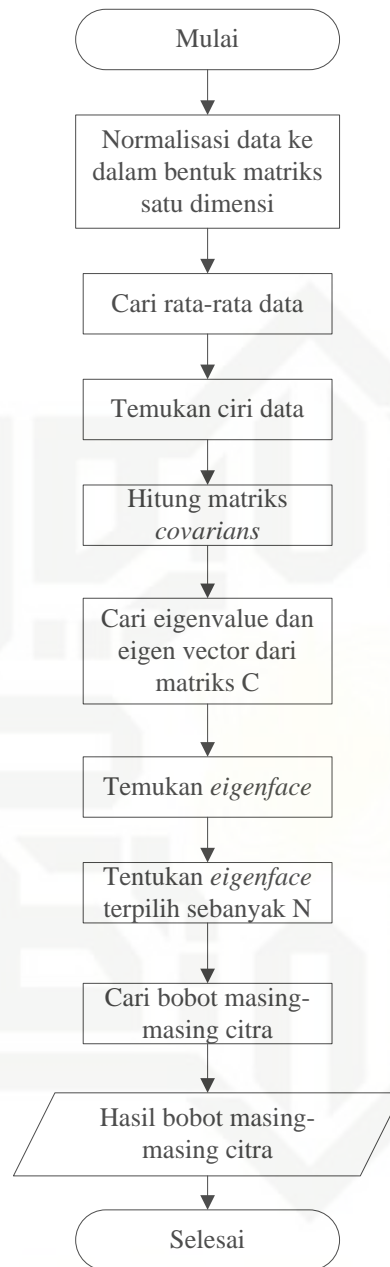
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4.1.3 Analisa Processing

Setelah proses *cropping* dan *resize* selesai dilakukan, maka selanjutnya akan dilakukan tahapan *processing* pada pola karakter huruf hijaiyah, yaitu melakukan tahap *feature extraction* (ekstraksi ciri) dengan tujuan untuk mencari sebuah vektor dalam sebuah image. Tahap ini disebut juga dengan proses pencarian nilai ciri citra huruf hijaiyah. Tahapan *processing* dilakukan dengan menggunakan metode *Principal Component Analysis* (PCA), yang nantinya dari proses ini akan dihasilkan nilai bobot citra huruf hijaiyah. Berikut ini merupakan alur proses ekstraksi ciri citra referensi dengan menggunakan algoritma *eigenface*:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 4.6 Alur Proses Ekstraksi Ciri Citra Referensi

Untuk lebih mudah memahami perhitungan rumus pada PCA, berikut diberikan contoh perhitungan yang lebih sederhana dengan mengambil 5 buah citra huruf hijaiyah dengan ukuran matriks 3×3 . Perhitungan ini dibuat hanya untuk gambaran umum dari tiap-tiap rumus yang ada dan bukan merupakan perhitungan sebenarnya. Karena perhitungan sebenarnya menggunakan citra 100×100 . Sehingga jika dijabarkan perhitungannya akan terlalu besar.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Matriks 3x3 diasumsikan sebagai suatu citra sederhana yang setiap elemennya dianggap sebagai nilai piksel dari citra. Hal ini dilakukan untuk menggambarkan metode *eigenface* secara sederhana. Contohnya terdapat citra A, citra B, citra C, citra D, dan citra E sebagai citra referensi. $M=5$. Dalam hal ini M merepresentasikan jumlah citra referensi.

$$\begin{array}{l}
 \begin{array}{|c|} \hline 1 \\ \hline \end{array} = \begin{array}{ccc} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{array} \quad \begin{array}{|c|} \hline 2 \\ \hline \end{array} = \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{array} \quad \begin{array}{|c|} \hline 3 \\ \hline \end{array} = \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{array} \\
 \\
 \begin{array}{|c|} \hline 4 \\ \hline \end{array} = \begin{array}{ccc} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{array} \quad \begin{array}{|c|} \hline 5 \\ \hline \end{array} = \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{array}
 \end{array}$$

Kemudian akan dilakukan proses perhitungan PCA, dengan Langkah-langkah sebagai berikut:

1. Masing-masing citra akan dirubah menjadi matriks baris dengan ukuran satu baris.

$$\begin{array}{l}
 \begin{array}{|c|} \hline 1 \\ \hline \end{array} = 1 \quad \begin{array}{|c|} \hline 2 \\ \hline \end{array} = 0 \\
 \begin{array}{|c|} \hline 3 \\ \hline \end{array} = 0 \\
 \begin{array}{|c|} \hline 4 \\ \hline \end{array} = 1 \\
 \begin{array}{|c|} \hline 5 \\ \hline \end{array} = 0
 \end{array}
 \begin{array}{cccccccc}
 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
 \end{array} \rightarrow \text{Im}$$

2. Langkah berikutnya adalah mencari rata-rata citra dengan menggunakan Persamaan rumus (2.3). Setiap kolom dari matriks Im akan dihitung rata-ratanya.

$$\begin{array}{l}
 \text{Kolom 1} = \frac{1}{0} = \frac{1}{5} (1+1) = \frac{2}{5} = 0.4 \\
 \text{Kolom 2} = \frac{1}{0} = \frac{1}{5} (1+1) = \frac{2}{5} = 0.4 \\
 \text{Kolom 3} = \frac{0}{0} = \frac{1}{5} (1) = \frac{1}{5} = 0.2 \\
 \text{Kolom 4} = \frac{1}{0} = \frac{1}{5} (1+1) = \frac{2}{5} = 0.4
 \end{array}$$

$$\text{Kolom 5} = \begin{matrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} = \frac{1}{5}(1) = \frac{1}{5} = 0.2$$

$$\text{Kolom 6} = \begin{matrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} = \frac{1}{5}(1) = \frac{1}{5} = 0.2$$

$$\text{Kolom 7} = \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix} = \frac{1}{5}(1+1+1+1) = \frac{4}{5} = 0.8$$

$$\text{Kolom 8} = \begin{matrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{matrix} = \frac{1}{5}(1+1+1) = \frac{3}{5} = 0.6$$

$$\text{Kolom 9} = \begin{matrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{matrix} = \frac{1}{5}(1+1+1) = \frac{3}{5} = 0.6$$

Hasilnya adalah matriks baris sebagai berikut:

$$\text{Mean} = [0.4 \ 0.4 \ 0.2 \ 0.4 \ 0.2 \ 0.2 \ 0.8 \ 0.6 \ 0.6]$$

3. Selanjutnya mencari ciri (nilai *fitur*) setiap citra referensi dengan cara mengurangi data (T) dengan rata-ratanya, yaitu dengan menggunakan Persamaan rumus (2.4).

$$\text{Fitur} = \begin{bmatrix} 1-0.4 & 1-0.4 & 0-0.2 & 1-0.4 & 1-0.2 & 0-0.2 & 1-0.8 & 1-0.6 & 0-0.6 \\ 0-0.4 & 1-0.4 & 0-0.2 & 0-0.4 & 0-0.2 & 0-0.2 & 1-0.8 & 0-0.6 & 1-0.6 \\ 0-0.4 & 0-0.4 & 0-0.2 & 0-0.4 & 0-0.2 & 0-0.2 & 1-0.8 & 1-0.6 & 1-0.6 \\ 1-0.4 & 0-0.4 & 1-0.2 & 1-0.4 & 0-0.2 & 1-0.2 & 1-0.8 & 0-0.6 & 1-0.6 \\ 0-0.4 & 0-0.4 & 0-0.2 & 0-0.4 & 0-0.2 & 0-0.2 & 0-0.8 & 1-0.6 & 0-0.6 \end{bmatrix}$$

$$= \begin{bmatrix} 0.6 & 0.6 & -0.2 & 0.6 & 0.8 & -0.2 & 0.2 & 0.4 & -0.6 \\ -0.4 & 0.6 & -0.2 & -0.4 & -0.2 & -0.2 & 0.2 & -0.6 & 0.4 \\ -0.4 & -0.4 & -0.2 & -0.4 & -0.2 & -0.2 & 0.2 & 0.4 & 0.4 \\ -0.4 & -0.4 & 0.8 & 0.6 & -0.2 & 0.8 & 0.2 & -0.6 & 0.4 \\ -0.4 & -0.4 & -0.2 & -0.4 & -0.2 & -0.2 & -0.8 & 0.4 & -0.6 \end{bmatrix}$$

Untuk menghemat komputasi, maka nilai nilai yang diambil adalah bilangan bulat positif saja. Maka matriks fitur yang dihasilkan adalah sebagai berikut:

$$\text{Fitur} = \begin{bmatrix} 0.6 & 0.6 & 0 & 0.6 & 0.8 & 0 & 0.2 & 0.4 & 0 \\ 0 & 0.6 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.4 & 0.4 \\ 0 & 0 & 0.8 & 0.6 & 0 & 0.8 & 0.2 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0 \end{bmatrix}$$

4. Berikutnya adalah mencari matriks kovarian berdasarkan Persamaan rumus (2.5).

$$\text{Cov} = \text{Fitur} \times \text{Fitur}^T$$

- Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$\begin{bmatrix} 0.6 & 0.6 & 0 & 0.6 & 0.8 & 0 & 0.2 & 0.4 & 0 \\ 0 & 0.6 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.4 & 0.4 \\ 0 & 0 & 0.8 & 0.6 & 0 & 0.8 & 0.2 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0 \end{bmatrix} \times \begin{bmatrix} 0.6 & 0 & 0 & 0 & 0 \\ 0.6 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 \\ 0.6 & 0 & 0 & 0.6 & 0 \\ 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0 \\ 0.4 & 0 & 0.4 & 0 & 0.4 \\ 0 & 0.4 & 0.4 & 0.4 & 0 \end{bmatrix}$$

$$\text{Cov} = \begin{bmatrix} 1.92 & 0.4 & 0.2 & 0.56 & 0.16 \\ 0.4 & 0.56 & 0.2 & 0.2 & 0 \\ 0.2 & 0.2 & 0.36 & 0.2 & 0.16 \\ 0.01 & 0.2 & 0.2 & 1.84 & 0 \\ 0.16 & 0 & 0.16 & 0 & 0.16 \end{bmatrix}$$

5. Setelah mendapatkan hasil matriks kovarian, selanjutnya akan dicari nilai *eigen* dan *vector eigen*. Untuk mencari nilai *eigen* (λ) dan vektor *eigen* digunakan Persamaan rumus (2.6). Nilai *eigen* dari matriks kovarian didapatkan dengan Persamaan rumus (2.7) berikut:

$$\text{Det} [\lambda I - \text{Cov}] = 0$$

$$\text{Det} = \begin{bmatrix} \lambda - 1.92 & 0.4 & 0.2 & 0.56 & 0.16 \\ 0.4 & \lambda - 0.56 & 0.2 & 0.2 & 0 \\ 0.2 & 0.2 & \lambda - 0.36 & 0.2 & 0.16 \\ 0.01 & 0.2 & 0.2 & \lambda - 1.84 & 0 \\ 0.16 & 0 & 0.16 & 0 & \lambda - 0.16 \end{bmatrix} = 0$$

$$\begin{bmatrix} \lambda - 1.92 & 0.4 & 0.2 & 0.56 & 0.16 \\ 0.4 & \lambda - 0.56 & 0.2 & 0.2 & 0 \\ 0.2 & 0.2 & \lambda - 0.36 & 0.2 & 0.16 \\ 0.01 & 0.2 & 0.2 & \lambda - 1.84 & 0 \\ 0.16 & 0 & 0.16 & 0 & \lambda - 0.16 \end{bmatrix} \begin{bmatrix} \lambda - 1.92 & 0.4 & 0.2 & 0.56 \\ 0.4 & \lambda - 0.56 & 0.2 & 0.2 \\ 0.2 & 0.2 & \lambda - 0.36 & 0.2 \\ 0.2 & 0.2 & 0.2 & \lambda - 1.84 \\ 0.16 & 0 & 0.16 & 0 \end{bmatrix}$$

Maka didapatkan nilai *eigen*nya adalah $\lambda_1 = 2.2326$, $\lambda_2 = 1.7090$, $\lambda_3 = 0.0397$, $\lambda_4 = 0.5263$ dan $\lambda_5 = 0.3324$. Dengan demikian nilai *eigen* dari matriks *covarians* adalah sebagai berikut:

$$\text{Nilai eigen} = \begin{bmatrix} 2.2326 & 0 & 0 & 0 & 0 \\ 0 & 1.7090 & 0 & 0 & 0 \\ 0 & 0 & 0.0397 & 0 & 0 \\ 0 & 0 & 0 & 0.5263 & 0 \\ 0 & 0 & 0 & 0 & 0.3324 \end{bmatrix}$$

Kemudian *vector eigen* akan dicari berdasarkan Persamaan rumus (2.8) sebagai berikut:

$$\begin{bmatrix} \lambda - 1.92 & 0.4 & 0.2 & 0.56 & 0.16 \\ 0.4 & \lambda - 0.56 & 0.2 & 0.2 & 0 \\ 0.2 & 0.2 & \lambda - 0.36 & 0.2 & 0.16 \\ 0.01 & 0.2 & 0.2 & \lambda - 1.84 & 0 \\ 0.16 & 0 & 0.16 & 0 & \lambda - 0.16 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Maka didapatkan nilai *eigen vectornya* adalah sebagai berikut:

$$\text{Vektor eigen} = \begin{bmatrix} -0.9159 & -0.8197 & -0.0745 & -0.235 & 0.0038 \\ -0.2668 & -0.2106 & 0.2493 & 0.7741 & -0.5387 \\ -0.159 & -0.0872 & -0.5313 & 0.5975 & 0.6157 \\ -0.2403 & -0.5171 & -0.0317 & -0.1979 & 0.0102 \\ -0.0830 & -0.0937 & -0.8056 & 0.1321 & 0.575 \end{bmatrix}$$

6. Setelah mendapatkan nilai eigen dan vektor eigen, selanjutnya adalah mencari *eigenface* dari citra referensi dengan cara mengalikan $Fitur^T$ dengan *vector eigen* dari matriks kovarian yaitu sesuai dengan Persamaan rumus (2.9).

$$Eig_f = \begin{bmatrix} 0.6 & 0 & 0 & 0 & 0 \\ 0.6 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 \\ 0.6 & 0 & 0 & 0.6 & 0 \\ 0.8 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0 \\ 0.4 & 0 & 0.4 & 0 & 0.4 \\ 0 & 0.4 & 0.4 & 0.4 & 0 \end{bmatrix} \times \begin{bmatrix} -0.9159 & -0.8197 & -0.0745 & -0.235 & 0.0038 \\ -0.2668 & -0.2106 & 0.2493 & 0.7741 & -0.5387 \\ -0.1591 & -0.0872 & -0.5313 & 0.5975 & 0.6157 \\ -0.2403 & -0.5171 & -0.0317 & -0.1979 & 0.0102 \\ -0.0830 & -0.0937 & -0.8056 & 0.1321 & 0.575 \end{bmatrix}$$

$$Eig_f = \begin{bmatrix} -0.5496 & -0.4918 & -0.0447 & -0.141 & 0.0023 \\ -0.7096 & -0.6181 & 0.1049 & 0.3235 & -0.3209 \\ -0.1922 & 0.4137 & 0.0254 & -0.1583 & -0.0082 \\ -0.6937 & -0.1815 & -0.0257 & -0.2597 & -0.0039 \\ -0.7327 & -0.6558 & -0.0596 & -0.1880 & 0.0031 \\ -0.1922 & 0.4137 & 0.0254 & -0.1583 & -0.0082 \\ -0.3164 & -0.1201 & -0.0650 & 0.1757 & 0.0141 \\ -0.4632 & -0.4002 & 0.0799 & 0.1739 & 0.4778 \\ -0.2664 & 0.0878 & -0.1001 & 0.4455 & 0.0267 \end{bmatrix}$$

Dari hasil perhitungan di atas, nilai setiap kolom dari matriks Eig_f adalah nilai *eigenface* masing-masing citra referensi. Nilai *eigenface* terbesar selalu berada di sebelah kanan, semakin ke kiri nilai *eigenface* semakin kecil.

7. Langkah selanjutnya adalah menentukan nilai N. Misalnya N=2, yaitu sebagai berikut:

- Hak Cipta Dilindungi Undang-Undang**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$Eig_fN = \begin{bmatrix} -0.141 & 0.0023 \\ 0.3235 & -0.3209 \\ -0.1583 & -0.0082 \\ -0.2597 & -0.0039 \\ -0.1880 & 0.0031 \\ -0.1583 & -0.0082 \\ 0.1757 & 0.0141 \\ 0.1739 & 0.4778 \\ 0.4455 & 0.0267 \end{bmatrix}$$

8. Setelah menentukan nilai N, selanjutnya adalah mencari bobot dari masing-masing citra referensi, yaitu dengan menggunakan Persamaan rumus (2.10).

$$W = \text{fitur} \times Eig_fN$$

$$W = \begin{bmatrix} 0.6 & 0.6 & 0 & 0.6 & 0.8 & 0 & 0.2 & 0.4 & 0 \\ 0 & 0.6 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.4 & 0.4 \\ 0 & 0 & 0.8 & 0.6 & 0 & 0.8 & 0.2 & 0 & 0.4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0 \end{bmatrix} \times \begin{bmatrix} -0.141 & 0.0023 \\ 0.3235 & -0.3209 \\ -0.1583 & -0.0082 \\ -0.2597 & -0.0039 \\ -0.1880 & 0.0031 \\ -0.1583 & -0.0082 \\ 0.1757 & 0.0141 \\ 0.1739 & 0.4778 \\ 0.4455 & 0.0267 \end{bmatrix}$$

Baris₁ kolom₁

$$[(0.6 \cdot -0.141) + (0.6 \cdot 0.3235) + (0 \cdot -0.1583) + (0.6 \cdot -0.2597) + (0.8 \cdot -0.1880) + (0 \cdot -0.1583) + (0.2 \cdot 0.1757) + (0.4 \cdot 0.1739) + (0 \cdot 0.4455)] \\ = -0.0920$$

Untuk hasil perkalian baris_n kolom_n dilakukan dengan langkah perhitungan yang sama seperti di atas, dengan hasil sebagai berikut:

$$W = \begin{bmatrix} -0.0920 & 0.0029 \\ 0.4074 & -0.1790 \\ 0.2829 & 0.2046 \\ -0.1958 & -0.0020 \\ 0.0696 & 0.1911 \end{bmatrix}$$

Setiap baris dari matriks W adalah bobot dari satu citra referensi, yang urutannya baris pertama merupakan bobot citra referensi A, baris kedua citra referensi B, begitu juga yang lainnya. Dengan ditemukan matriks W ini, maka proses ekstraksi ciri citra referensi selesai dilakukan.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4.1.4 Analisa *Identification* Dengan *Learning Vector Quantization* (LVQ)

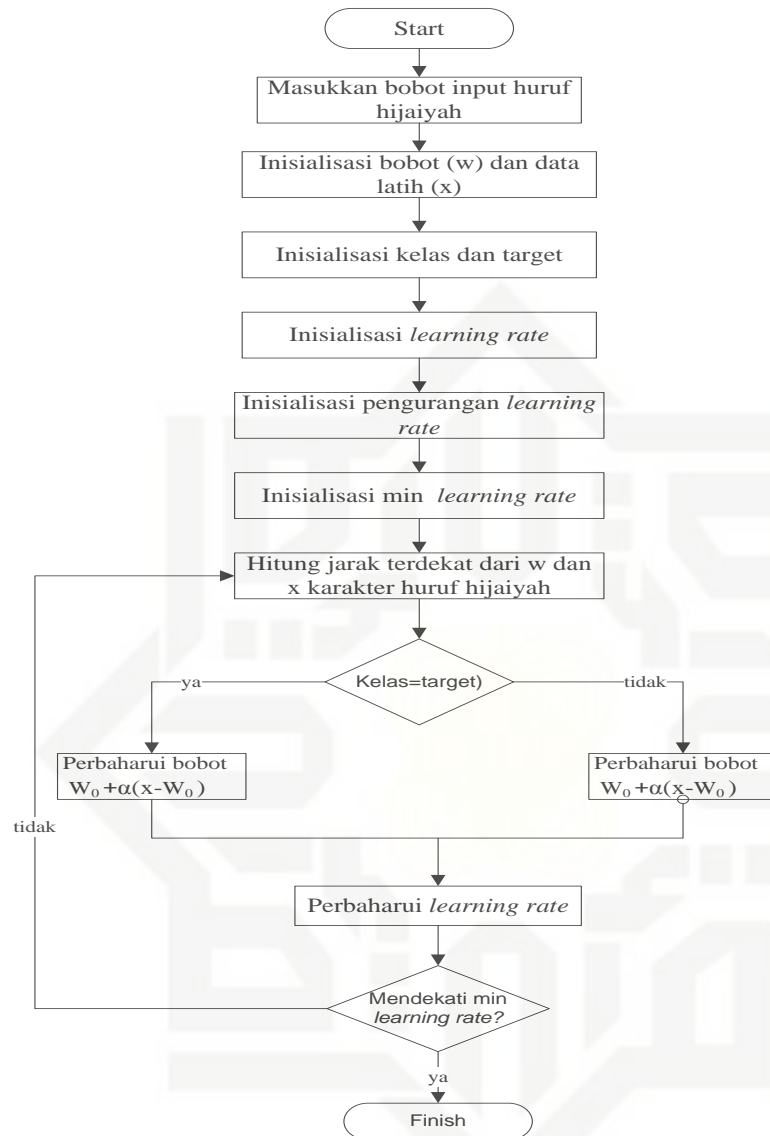
Setelah melalui beberapa tahapan yaitu *data collection*, *pre-processing* dan *processing*, maka tahapan terakhir untuk mendapatkan hasil dari pengenalan karakter huruf hijaiyah adalah *identification*. Tahapan *identification* ini mengklasifikasikan vektor *input* yang telah diperoleh sebelumnya untuk dijadikan data latih (*training*) dan diuji dengan data uji (*testing*), kemudian hasil pelatihan akan dihitung jarak terdekatnya menggunakan *euclidean distance* dan dengan menerapkan jaringan syaraf tiruan yaitu dengan menggunakan metode LVQ.

A. **Flowchart *Learning Vector Quantization* (LVQ)**

Flowchart merupakan diagram alir yang menggambarkan alur atau proses yang berjalan. Pada Gambar 4.5 di bawah ini merupakan *flowchart* yang merupakan tahap *training* pada pengenalan karakter huruf Hijaiyah dengan menerapkan metode LVQ.

Hak Cipta Dilindungi Undang-Undang

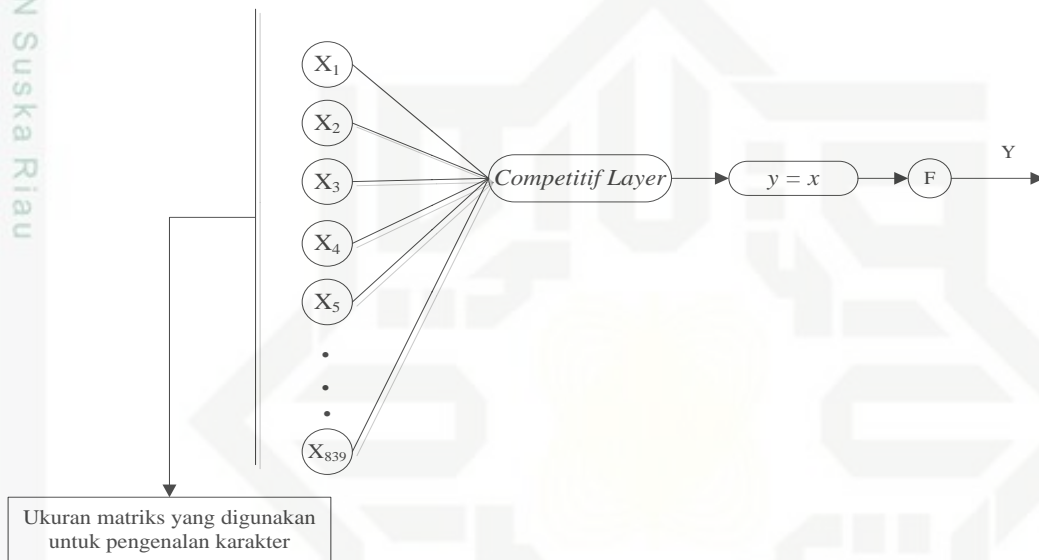
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 4.7 Flowchart LVQ Pengenalan Huruf Hijaiyah

Berdasarkan Gambar 4.7 di atas yang merupakan tahap *training*, proses yang dilakukan pertama adalah inisialisasi bobot (w) dan data latih (x) yang akan dijadikan vektor input karakter huruf hijaiyah yang sebelumnya telah dilakukan proses *processing*. Inisialisasi kelas dan target karakter huruf hijaiyah, inisialisasi *learning rate*, pengurangan *learning rate* dan *learning rate min*. Pada iterasi pertama seluruh input yang telah dinisialisasi melakukan perhitungan jarak menggunakan rumus jarak *euclidean distance*, kemudian ambil nilai indeks jarak terkecil dari perhitungan jarak. Jika telah menemukan nilai indeks jarak terkecil, *update* tiap bobot untuk menemukan bobot baru menggunakan Persamaan (2.11)

dan Persamaan (2.12) untuk memperbaharui bobot pada LVQ. Jika telah menemukan bobot baru, *update learning rate*. Untuk iterasi kedua dan seterusnya, mengikuti perhitungan yang sama seperti iterasi pertama dengan bobot yang telah diperbaharui setiap perhitungannya. Iterasi akan berhenti hingga *learning rate* mencapai atau mendekati *learning rate min*. Gambar 4.8 berikut menjelaskan arsitektur LVQ untuk pengenalan karakter huruf hijaiyah:



Gambar 4.8 Arsitektur LVQ Untuk Pengenalan Karakter Huruf Hijaiyah

Contoh dari *LVQ* yang salah satu inisialisasi kelas *output*-nya yaitu vektor input berupa bobot yang diambil dari hasil perhitungan ekstraksi ciri sebelumnya yaitu dengan menggunakan PCA. Pola vektor bobot tersebut diinisialisasi dengan rumus *LVQ*, maka hasil perhitungannya adalah sebagai berikut:

- Langkah pertama adalah menentukan bobot, menentukan masing-masing kelas *output*, menetapkan *learning rate* (α), pengurangan *Learning rate* dan minimum *learning rate*.

Tabel 4.4 Nilai Bobot Input

No	Input bobot (x)	Kelas output
1	-0.0920 0.0029	1 = ا
2	0.4074 -0.1790	2 = ب
3	0.2829 0.2046	3 = ت
4	-0.1958 -0.0020	1 = ا
5	0.0696 0.1911	3 = ت

Inisialisasi bobot

Data latih

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$\begin{aligned} \text{Learning rate } (\alpha) &= 0.01 \\ \text{Pengurangan } \alpha &= 0.1 \\ \text{Min Learning rate } (\alpha) &= 0.001 \end{aligned}$$

Bandingkan masing-masing *input* dengan masing-masing bobot yang telah ditetapkan dengan melakukan pengukuran jarak antara masing-masing bobot dan input. Pengukuran jarak dihitung menggunakan Persamaan (2.13) yaitu persamaan *euclidean distance*.

Data ke 4= (-0.1958 -0.0020)

$$\text{Bobot 1= } (-0.0920 \ 0.0029)$$

$$\begin{aligned} \text{Jarak} &= \sqrt{(-0.1958 + 0.0920)^2 + (-0.0020 - 0.0029)^2} = \sqrt{0.010633} \\ &= 0.1031 \end{aligned}$$

$$\text{Bobot 2= } (0.4074 \ -0.1790)$$

$$\begin{aligned} \text{Jarak} &= \sqrt{(-0.1958 + 0.4074)^2 + (-0.0020 - -0.1790)^2} = \sqrt{0.395179} \\ &= 0.6286 \end{aligned}$$

$$\text{Bobot 3= } (0.2829 \ 0.2046)$$

$$\begin{aligned} \text{Jarak} &= \sqrt{(-0.1958 + 0.2829)^2 + (-0.0020 - 0.2046)^2} = \sqrt{0.271837} \\ &= 0.5214 \end{aligned}$$

Nilai minimum dari hasil perbandingan akan menentukan kelas dari vektor input dan perubahan bobot dari kelas tersebut menggunakan Persamaan (2.11) atau Persamaan (2.12). Jarak terkecil adalah bobot ke-1 (C=1), target data latih adalah 1 (T=1), sehingga C=T. Maka bobot ke-1 diperbaharui dengan rumus 2.9.

$$w_{1,1} = 0.0029 + 0,01 (-0.002 - 0.0029) = 0,0028$$

$$w_{1,2} = -0.0914 + 0,01 (-0.1958 + 0.0914) = -0,0924$$

$$w_{1(\text{baru})} = [0,0028 \ -0,0924]$$

Data ke 5= (0.1911 0.0696)

$$\text{Bobot 1= } (0.0028 \ -0.0924)$$

$$\begin{aligned} \text{Jarak} &= \sqrt{(0.1911 - 0.0028)^2 + (0.0696 + 0.0924)^2} = \sqrt{0.0617009} \\ &= 0.2484 \end{aligned}$$

$$\text{Bobot 2= } (-0.1790 \ 0.4074)$$

$$\text{Jarak} = \sqrt{(0.1911 + 0.179)^2 + (0.0696 - 0.4074)^2} = \sqrt{0.228652} = 0.1512$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

$$\text{Bobot } 3 = (0.2046 \quad 0.2829)$$

$$\text{Jarak} = \sqrt{(0.1911 - 0.2046)^2 + (0.0696 - 0.2829)^2} = \sqrt{0.0456792} = 0.2137$$

Jarak terkecil adalah bobot ke-2 (C=2), target data latih adalah 3 (T=3), sehingga $C \neq T$. Maka bobot ke-2 diperbaharui dengan rumus 2.10.

$$w_{2,1} = -0.179 + 0,01 (0.1911 - 0.179) = 0,1822$$

$$w_{1,2} = 0.4074 + 0,01 (0.0696 + 0.4074) = 0.4108$$

$$w_{2(\text{baru})} = [0,1822 \quad 0.4108]$$

Lakukan pembaharuan (*update*) pengurangan *learning rate*

$$\alpha_{\text{baru}} = 0.01 - (0.1 * 0.01) = 0,009$$

Lakukan perhitungan LVQ hingga *learning rate min* terpenuhi yaitu hingga 23 iterasi. Hasil pembaharuan bobot hingga 23 iterasi adalah sebagai berikut:

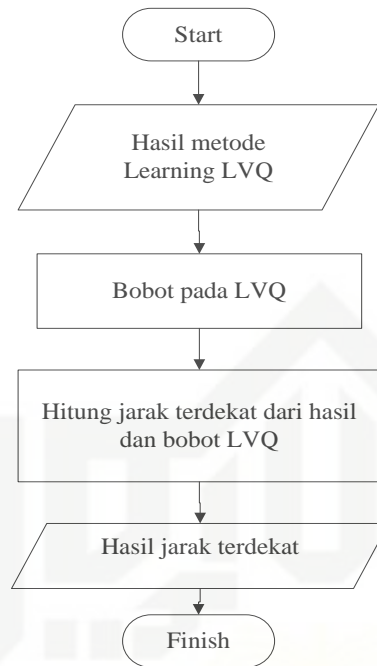
$$w_1 = [-0.1011 \quad 0.0025]$$

$$w_2 = [0.4074 \quad -0.1790]$$

$$w_3 = [0.2634 \quad 0.2034]$$

4.1.5 Euclidean Distance

Tahap *euclidean distance* dilakukan untuk mendapatkan jarak terdekat dari bobot yang telah diperbaharui menggunakan metode LVQ dengan vektor *input*. Jarak terdekat dari hasil perhitungan akan mendapat kesimpulan pola yang menyerupai *input* yang dimasukkan. Pada pengenalan karakter huruf hijaiyah menggunakan metode PCA-LVQ, tahap *euclidean distance* merupakan tahap *testing* pada pengenalan karakter huruf menggunakan PCA-LVQ. *Flowchart* metode *euclidean distance*, adalah sebagai berikut:



Gambar 4.9 Flowchart Euclidean Distance untuk Pengenalan Karakter Huruf Hijaiyah

Hasil dari bobot metode LVQ akan dihitung jaraknya menggunakan rumus jarak *euclidean distance*. Bobot dengan hasil jarak paling kecil dari perhitungan jarak menggunakan rumus *euclidean distance* akan dijadikan hasil pengenalan dari *input* yang dimasukkan. Perhitungan *euclidean distance* yang inisialisasi vektornya diambil dari hasil perhitungan LVQ pada data ke-4 (x) yang merujuk pada Tabel 4.4 dengan menggunakan Persamaan (2.11) yaitu persamaan *euclidean distance* dengan perhitungan sebagai berikut:

$$\begin{aligned} \text{Euclidean distance } (x_4, w_1) &= \sqrt{(-0.1958 - (-0.1011))^2 + (-0.0020 - 0.0025)^2} \\ &= 0.0948 \end{aligned}$$

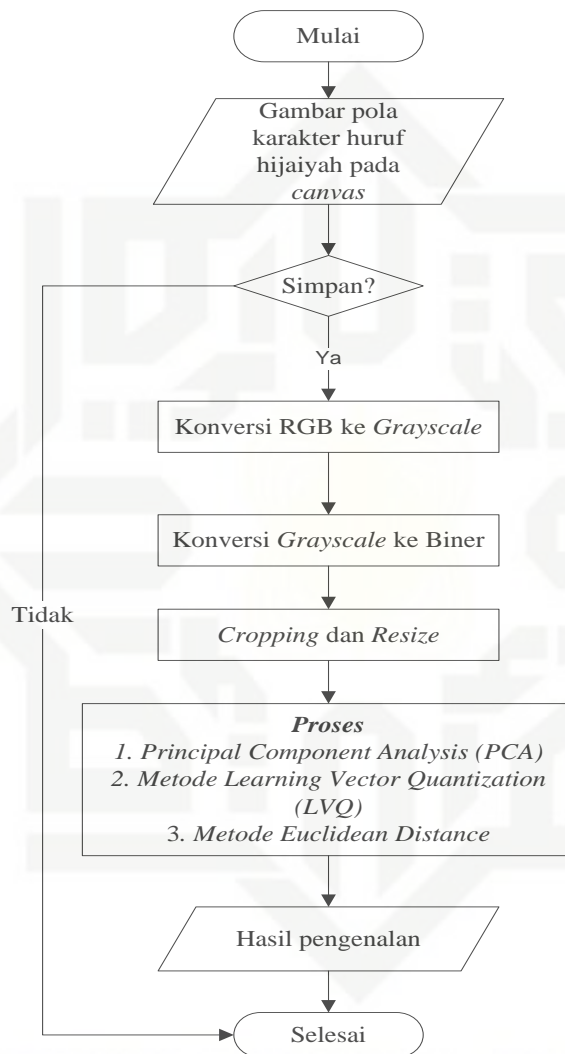
$$\begin{aligned} \text{Euclidean distance } (x_4, w_2) &= \sqrt{(-0.1958 - 0.4074)^2 + (-0.0020 - (-0.1790))^2} \\ &= 0.5677 \end{aligned}$$

$$\begin{aligned} \text{Euclidean distance } (x_4, w_3) &= \sqrt{(-0.1958 - 0.2634)^2 + (-0.0020 - 0.2034)^2} \\ &= 0.5030 \end{aligned}$$

Dari perhitungan *euclidean distance* di atas, dapat disimpulkan bahwa pola *input* yang dimasukkan adalah pola bobot pertama.

4.1.6 Flowchart Aplikasi

Alur proses aplikasi untuk pengenalan karakter pada huruf hijaiyah menggunakan metode PCA-LVQ dapat digambarkan pada *flowchart*. *Flowchart* alur proses aplikasi tersebut adalah sebagai berikut:



Gambar 4.10 Flowchart Aplikasi Pengenalan Karakter Huruf Hijaiyah Menggunakan Metode PCA-LVQ

Berdasarkan alur di atas, proses pengenalan karakter huruf hijaiyah dimulai dengan menggambarkan pola karakter huruf ke area *canvas* pada aplikasi. Setelah itu gambar akan dikonversikan dari RGB ke *grayscale* dan kemudian dilanjutkan lagi dengan konversi *grayscale* ke biner. Setelah itu gambar akan dilakukan proses *cropping* untuk meminimalkan *background* dan *resize* untuk mengubah ukuran matriks menjadi berukuran 100x100 piksel. Dari matriks tersebut akan

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dilakukan *processing* yaitu ekstraksi ciri menggunakan metode PCA. Dari perhitungan PCA akan menghasilkan bobot citra pola huruf hijaiyah. Nilai bobot inilah yang akan dijadikan *neuron input* ke tahap *identification*, yaitu dengan menggunakan metode LVQ dan *euclidean distance*, agar karakter yang dimasukkan dapat dikenali.

4.2 Perancangan

Perancangan merupakan proses yang dilakukan setelah tahap analisa selesai dilakukan. Tujuan dilakukannya perancangan ini adalah untuk dapat dijadikan pedoman dalam pembuatan aplikasi pengenalan karakter huruf hijaiyah. Perancangan yang dilakukan yaitu perancangan tampilan aplikasi (*interface*).

4.2.1 Perancangan Tampilan Aplikasi (*User Interface*)

Perancangan tampilan aplikasi (*Interface*) merupakan faktor penting yang harus dilakukan dalam perancangan sebuah aplikasi. Tujuan dilakukan perancangan interface adalah untuk membuat komunikasi serta interaksi yang baik antara aplikasi dengan pemakai sistem (*user*). Tampilan *interface* dikatakan baik apabila tampilan tersebut mudah dipahami dan konsisten dari segi bahasa.

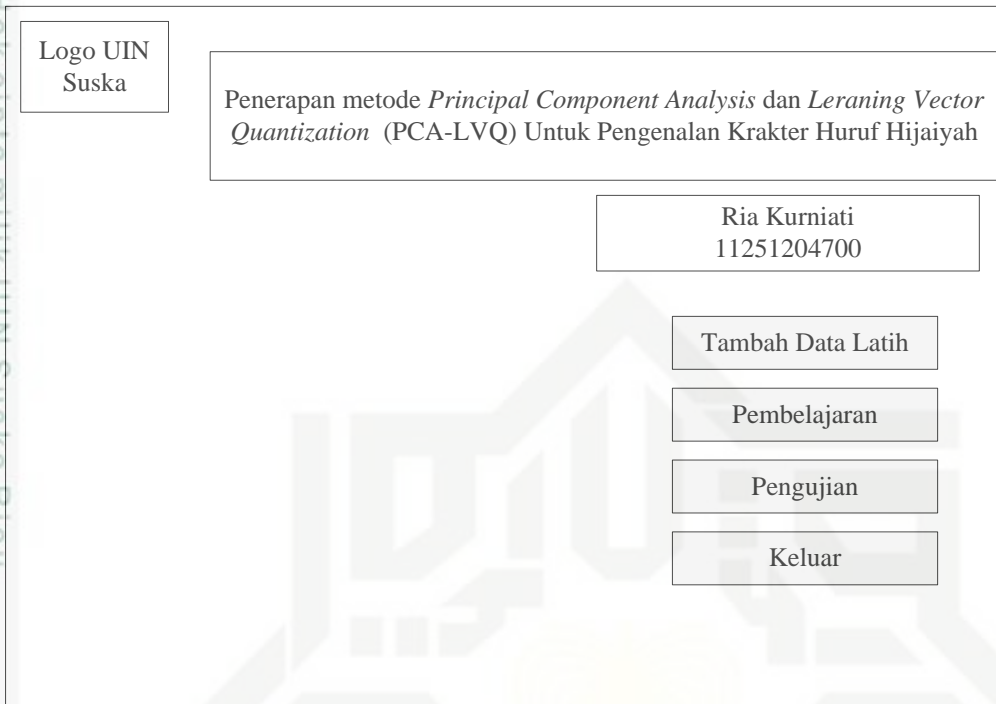
Perancangan *interface* pada aplikasi pengenalan karakter huruf hijaiyah meliputi perancangan tampilan halaman depan yang meliputi perancangan tampilan menu data latih, perancangan tampilan menu pembelajaran, dan perancangan tampilan menu aplikasi pengenalan karakter.

1. Perancangan Tampilan Halaman Depan

Tampilan halaman depan merupakan halaman awal dari aplikasi yang akan dibuat. Pada perancangan halaman depan terdapat menu yang dapat digunakan oleh *user*, diantaranya adalah menu tambah data latih yang berfungsi untuk menambah data latih citra, menu pembelajaran yang berfungsi untuk pembelajaran untuk proses JST, dan menu aplikasi untuk pengenalan karakter. Berikut merupakan tampilan halaman depan yang dapat dilihat pada Gambar 4.11 di bawah ini:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

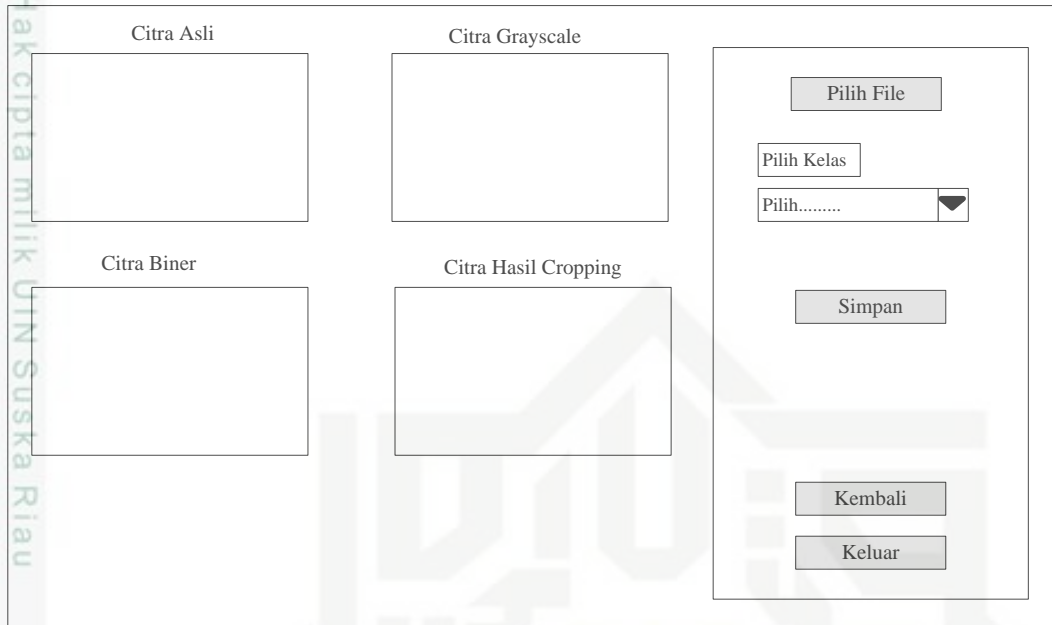


Gambar 4.11 Perancangan Tampilan Halaman Depan

2. Perancangan Tampilan Halaman Tambah Data Latih

Perancangan tampilan menu tambah data latih digunakan untuk menambahkan data latih citra karakter huruf hijaiyah. Perancangan tampilan menu tambah data latih dapat dilihat pada Gambar 4.12 berikut ini:

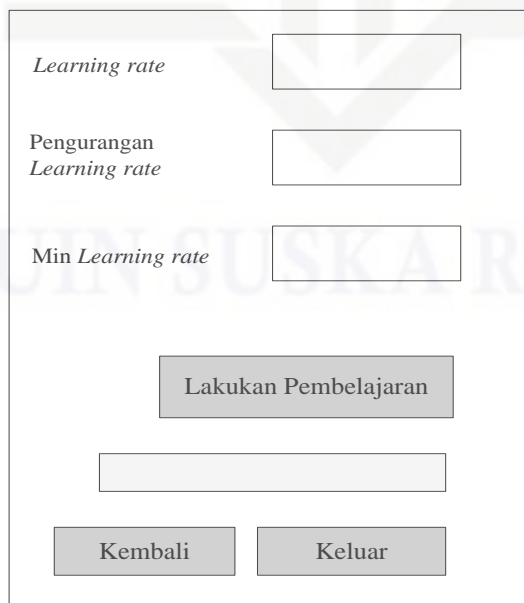
- Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 4.12 Perancangan Tampilan Menu Tambah Data Latih

3. Perancangan Tampilan Menu Pembelajaran

Perancangan tampilan menu pembelajaran ini digunakan untuk inisialisai parameter input yang akan digunakan pada proses LVQ, yaitu berupa parameter *learning rate* (α), pengurangan *learning rate* (α), dan minimum *learning rate* (α). Perancangan tampilan menu pembelajaran yang dapat dilihat pada Gambar 4.13 berikut:



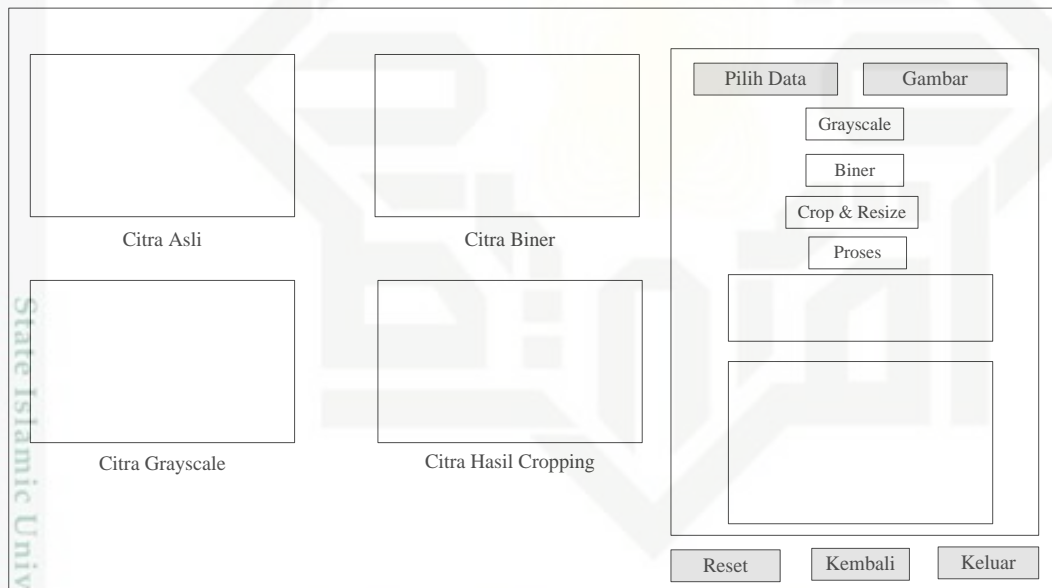
Gambar 4.13 Perancangan Tampilan Menu Pembelajaran

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4. Perancangan Tampilan Menu Pengujian

Perancangan tampilan menu aplikasi merupakan tampilan yang digunakan dalam melakukan pengujian untuk pengenalan karakter huruf hijaiyah. Pada tampilan menu pengujian ini, disediakan beberapa menu di dalamnya, seperti menu ‘pilih data’ dan menu ‘gambar’. Menu ‘pilih data’ berfungsi untuk melakukan identifikasi pada gambar yang telah tersimpan sebelumnya, sedangkan menu ‘gambar’ berfungsi untuk menggambarkan pola karakter huruf hijaiyah yang inputan citranya berasal dari tampilan *canvas* yang kemudian dijadikan gambar/citra. Tampilan *canvas* akan muncul setelah kita melakukan klik pada menu ‘gambar’. Perancangan tampilan menu Pengujian dapat dilihat pada Gambar 4.14 berikut:



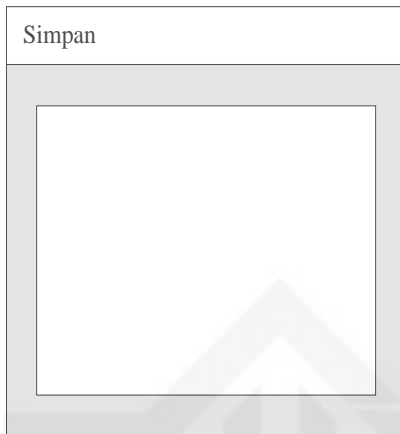
Gambar 4.14 Perancangan Tampilan Menu Pengujian

5. Perancangan Tampilan Canvas

Perancangan tampilan *canvas* merupakan tampilan yang digunakan untuk menggambarkan pola karakter huruf hijaiyah yang akan dikenali. Perancangan tampilan *canvas* dapat dilihat pada Gambar 4.15 berikut:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



Gambar 4.15 Perancangan Tampilan *Canvas*