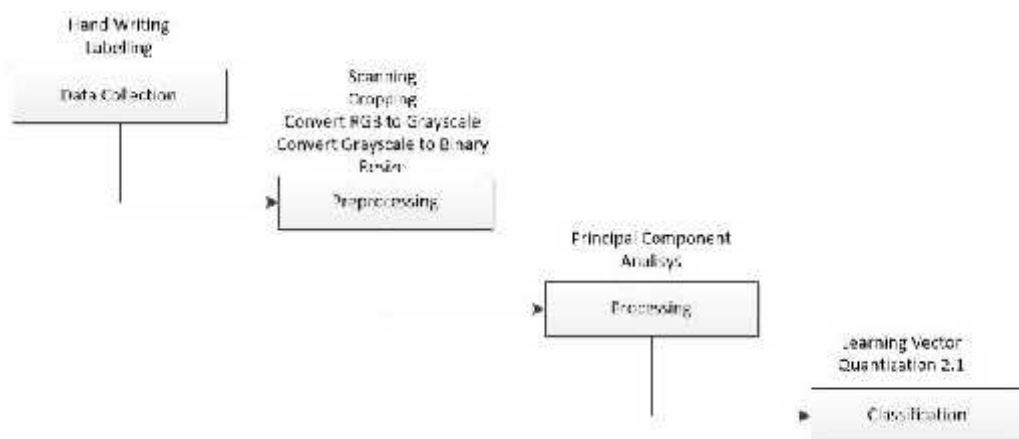


BAB IV

ANALISA DAN PERANCANGAN

Tahapan penting dalam pembuatan suatu system atau aplikasi adalah tahap analisa dan perancangan. Pada tahap ini dilakukan proses kajian yang membahas, memahami dan menspesifikasikan secara rinci apa saja yang menjadi pokok permasalahan dalam penelitian agar memperoleh pemahaman yang tepat mengenai masalah,data,proses,hasil, dan semua hal yang berkaitan dengan penelitian.



Gambar 4.1 Tahapan Pengenalan Karakter Seseorang Berdasarkan Pola Tulisan Huruf t dengan Metode *Learning Vector Quantization 2.1*

4.1 Analisis Pengenalan Karakter

Analisis yang perlu dilakukan pada penelitian ini adalah menganalisis tahapan atau proses yang perlu dilakukan agar aplikasi dapat melakukan pengenalan karakter seseorang berdasarkan pola tulisan huruf t. Analisis proses yang dilakukan adalah data collection, pre-processing, processing dan classification.

4.1.1 Analisis Data Collection

Proses *data collection* merupakan proses pengumpulan data latih dan data uji.

4.1.2 Pembagian Data

Pembagian data untuk proses penentuan karakter seseorang dengan menggunakan metode LVQ 2.1 (*Learning Vector Quantization 2.1*) terdiri dari dua bagian data, yaitu data latih (*training*) dan data uji (*testing*). Untuk penelitian ini digunakan *sample* sebanyak 20 orang yang akan diambil data tulisan huruf t. Adapun data penelitian dapat dilihat pada Lampiran A.1

4.1.2.1 Data Latih

Jumlah data latih yang akan digunakan adalah 360 data latih. Data latih ini akan dinormalisasi dan disimpan dalam *database* dan nantinya akan dijadikan acuan untuk proses penentuan karakter seseorang tersebut. Adapun data latih dapat dilihat pada Lampiran A.2.1 dan A.2.2

4.1.2.2 Data Uji

Pengujian dilakukan dengan tujuan untuk menentukan tingkat akurasi proses klasifikasi penentuan karakter seseorang. Jumlah data uji yang digunakan adalah 40 Data uji citra huruf t dari data tulisan 2 orang juga akan melalui proses normalisasi dengan cara yang sama dengan data latih, selanjutnya menguji dengan metode LVQ 2.1 untuk menentukan kelas data uji sehingga ditemukan akurasi dari hasil pengujian. Adapun data uji dapat dilihat pada Lampiran A.3.1 dan A.3.2

4.1.3 Analisis Pre-Processing

Setelah citra pola tulisan huruf t telah diinputkan pada aplikasi, maka tahap selanjutnya yang dilakukan adalah tahap pre-processing yang merupakan tahap pemisahan objek citra tulisan huruf t dengan area background dengan cara cropping, kemudian melakukan perubahan ciri warna yaitu konversi RGB ke grayscale, kemudian dari ciri warna grayscale dikonversi lagi ke biner, setelah itu gambar akan di-resize ke piksel yang lebih kecil.

4.1.3.1 Scanning

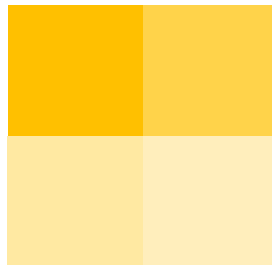
Scanning dilakukan untuk mengakuisisi data tulisan huruf t menjadi citra digital.

4.1.3.2 Cropping

Cropping dilakukan adalah untuk meminimalisir background pada citra tulisan huruf t sehingga didapatkan area yang berfokus pada karakter tulisan huruf t agar dapat diproses dengan baik pada tahap selanjutnya.

4.1.3.3 Konversi RGB ke Grayscale

Setelah proses memasukkan citra tulisan huruf t dengan cara menginputkan file citra tulisan pada aplikasi, maka citra tersebut secara otomatis memiliki ciri warna RGB. Oleh karena itu gambar tulisan huruf t tersebut harus dikonversi ke grayscale terlebih dahulu. Gambar 4.2 berikut merupakan gambar yang dikonversi RGB ke grayscale.



Gambar 4.2 Gambar RGB

Gambar 4.2 adalah 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 dibawah ini:

Tabel 4.1 Nilai RGB

Piksel(1,1)	Piksel(1,2)
R= 179	R= 155
G= 187	G= 180
B= 168	B= 124
Piksel(2,1)	Piksel(2,2)
R= 120	R= 79
G= 100	G= 66
B= 102	B= 60

Dengan menggunakan rumus 2.15 konversi RGB ke Grayscale), maka akan dilakukan perhitungan sebagai berikut :

$$\begin{aligned} \text{Grayscale Piksel (1,1)} &= (0.2989 * 179) + (0.5870 * 187) + (0.1141 * 168) \\ &= 182.4409 \end{aligned}$$

$$\text{Grayscale Piksel (1,2)} = (0.2989 * 155) + (0.5870 * 180) + (0.1141 * 124)$$

$$= 166.138$$

$$\text{Grayscale Piksel (2,1)} = (0.2989 * 120) + (0.5870 * 100) + (0.1141 * 102)$$

$$= 106.2062$$

$$\text{Grayscale Piksel (2,2)} = (0.2989 * 79) + (0.5870 * 66) + (0.1141 * 60)$$

$$= 69.2011$$

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.

Tabel 4.2 Nilai Hasil Konversi RGB ke *Grayscale*

Piksel (1,1) 182.4409	Piksel (1,2) 166.138
Piksel (2,1) 106.2062	Piksel (2,2) 69.2011

Berikut adalah *pseudo code* untuk konversi pola tulisan huruf dari ciri warna RGB menjadi *grayscale*:

Program Convert RGB to Gray

Deklarasi

Img {input gambar 3 dimensi}

R {representasi nilai merah (red)}

G {representasi nilai hijau (green)}

B {representasi nilai biru (blue)}

Gray {representasi nilai grayscale yang akan dicari}

Algoritma

1) $r(i,:) = (:,:,1);$

2) $g(i,:) = (:,:,2);$

3) $b(i,:) = (:,:,3);$

4) $gray(i,:) = (0.2989*r) + (0.587*g) + (0.1140*b)$

Algoritma 4.1 Pseudocode Konversi RGB ke Grayscale (Putra, 2010)

Keterangan Algoritma 4.1 adalah sebagai berikut:

- 1) Langkah untuk mendapatkan nilai R (*Red*) dari satu piksel citra
- 2) Langkah untuk mendapatkan nilai G (*Green*) dari satu piksel citra
- 3) Langkah untuk mendapatkan nilai B (*Blue*) dari satu piksel citra
- 4) Rumus untuk mendapatkan nilai *gray* dari satu piksel citra berdasarkan nilai R, G, dan B yang telah didapatkan dengan langkah sebelumnya.

4.1.3.4 Konversi Grayscale ke Biner

Setelah didapatkan nilai grayscale dari masing-masing piksel citra, maka akan dilanjutkan dengan mengubah nilai masing-masing piksel dari format grayscale menjadi biner (Shatil, 2006). Dengan menggunakan rumus 2.16 (rumus konversi grayscale ke biner), dan menggunakan nilai ambang (T) = 128 , maka nilai biner yang diperoleh masing-masing piksel adalah sebagai berikut.

$$\text{Piksel (1,1)} = 182.4409 > T$$

$$= 1$$

$$\text{Piksel (1,2)} = 166.138 > T$$

$$= 1$$

$$\text{Piksel (2,1)} = 106.2062 < T$$

$$= 0$$

$$\text{Piksel (2,2)} = 69.2011 < T$$

$$= 0$$

Dari perhitungan konversi *grayscale* ke *biner* yang telah dilakukan, maka nilai *biner* dari masing-masing piksel dapat dilihat pada Tabel 4.3 berikut.

Tabel 4.3 Nilai Hasil Konversi Grayscale ke Biner

Piksel (1,1) 1	Piksel (1,2) 1
Piksel (2,1) 0	Piksel (2,2) 0

Pseudo code untuk konversi citra dari format *grayscale* ke *biner* adalah sebagai berikut:

Program Convert Gray To Bin

Deklarasi

img_gray; {matriks nilai grayscale}

T; {nilai ambang}

Algoritma

1. T=128;
2. If(img_gray(i,:) >=T);
3. img_bin=1;
4. else
5. img_bin=0;

Algoritma 4.2 Pseudocode Konversi Citra Grayscale ke Biner (Putra, 2010)

Keterangan Algoritma 4.2 adalah sebagai berikut

- 1) Untuk menetapkan nilai ambang (T)
- 2) Kondisi jika nilai *grayscale* dari piksel sebuah citra lebih besar atau sama dengan nilai ambang
- 3) Nilai *grayscale* piksel sebuah citra diubah menjadi bernilai 1
- 4) Jika hasil tidak sesuai kondisi awal
- 5) Nilai *grayscale* pixel sebuah citra diubah menjadi bernilai 0
- 6) Akhir dari sebuah kondisi.

4.1.3.5 Resize

Pola citra karakter tulisan huruf *t* yang telah diinputkan dilakukan proses *cropping* dan konversi seperti tahap sebelumnya maka akan dilakukan proses *resize* untuk mengubah ukuran semula menjadi matriks berukuran $m \times n$ piksel. Pada penelitian ini, aplikasi pengenalan karakter berdasarkan pola tulisan huruf *t* dengan menggunakan matriks 20 x 20 piksel.

4.1.4 Analisis Processing

Analisis processing adalah tahapan yang terdiri dari proses ekstraksi ciri citra menggunakan metode PCA (*Principal Component Analysis*) yang nantinya akan

diperoleh nilai *PC* untuk digunakan pada tahapan klasifikasi metode LVQ 2.1 (*learning vector quantization 2.1*). Berikut Merupakan tahapan ekstraksi ciri citra menggunakan metode PCA (*Principal Component Analysis*).

4.1.4.1 PCA (*Principal Component Analysis*)

Setelah proses pre-processing dilakukan yaitu proses cropping dan grayscale selanjutnya akan dilakukan proses pencarian nilai ciri citra huruf t dengan metode principal component analisis (PCA). Hasil dari proses ekstraksi ciri dengan metode PCA adalah nilai bobot citra tulisan huruf atau *principal component* (PC). Nilai PC inilah yang menjadi ciri dari suatu citra huruf t Berikut ini adalah algoritma yang digunakan pada metode PCA (*Principal Component Analysis*):

1. Buat matriks data set (data latih) yang berisi data citra huruf t.
2. Buat matriks nilai rata-rata (mean) citra huruf t dari matriks data set.
3. Buat matriks selisih dengan cara mengurangkan setiap elemen matriks data set atau data latih dengan matriks rata-rata citra huruf t.
4. Membuat matriks kovarian dari matriks selisih citra huruf t data latih.
5. Cari nilai eigenvector dan eigenvalue dari matriks kovarian yang telah dicari.
6. Cari nilai principal component (PC) dari eigenvector yang telah dicari berdasarkan eigenvalue tertinggi dengan mengurutkannya terlebih dahulu.

Berikut ini adalah *flowchart* metode PCA (*Principal Component Analysis*):

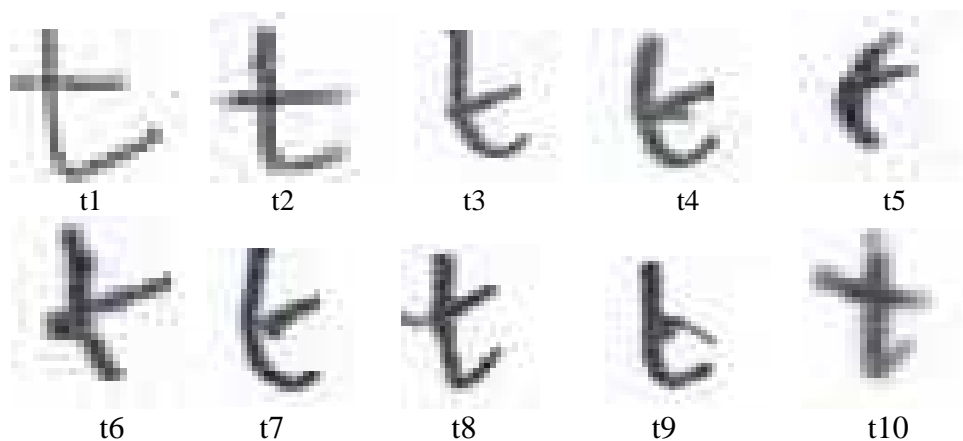


Gambar 4.3 Flowchart PCA sistem klasifikasi tulisan huruf t

Perhitungan metode PCA (*Principal Component Analysis*)

4.1.4.1.1 Membuat data set

Terdapat beberapa citra yang akan dijadikan data set atau data latih. Berikut ini beberapa citra huruf t yang akan dijadikan data set atau data latih.



Gambar 4.4 Citra Huruf t

Kemudian dari citra huruf t yang berukuran 20x 20 piksel di atas dikonversikan ke dalam bentuk matriks sebagai berikut :

$$\begin{array}{l}
 \text{Matriks t1} = \begin{bmatrix} 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \\
 \text{Matriks t2} = \begin{bmatrix} 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} \\
 \text{Matriks t3} = \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \\
 \text{Matriks t4} = \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \\
 \text{Matriks t5} = \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \\
 \text{Matriks t6} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \\
 \text{Matriks t7} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} \\
 \text{Matriks t8} = \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \\
 \text{Matriks t9} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 0 & \dots & 0 \end{bmatrix} \\
 \text{Matriks t10} = \begin{bmatrix} 1 & 1 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix}
 \end{array}$$

4.1.4.1.2 Matriks nilai rata-rata.

Membuat matriks nilai rata-rata, sesuai dengan persamaan 2.3. Diketahui:

$$M = 10$$

$$\psi = \frac{1}{1} \sum_{n=1}^1 r_n = \frac{1}{1}$$

$$\begin{aligned} & \begin{bmatrix} 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} + \\ & \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} \\ & + \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} + \\ & \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 0 & \dots & 0 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} \\ & \psi = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0,9 & 0,6 & \dots & 1 \\ 1 & 1 & 0,5 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0,7 & 1 & \dots & 1 \end{bmatrix} \end{aligned}$$

4.1.4.1.3 Matrik selisih

Membuat matrik selisih, sesuai dengan persamaan 2.4

$$\varphi = \begin{matrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \\ \vdots \\ \mathbf{I} \end{matrix} \begin{bmatrix} 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0,9 & 0,6 & \dots & 1 \\ 1 & 1 & 0,5 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0,7 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0,1 & -0,3 & 0 \\ 0 & 0 & -0,6 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0,3 & 0 & 0 \end{bmatrix}$$

$$\varphi = \begin{matrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \\ \vdots \\ \mathbf{I} \end{matrix} \begin{bmatrix} 1 & 0 & 1 & \dots & 1 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0,9 & 0,6 & \dots & 1 \\ 1 & 1 & 0,5 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0,7 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0,1 & 0,4 & 0 \\ 0 & 0 & 0,5 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0,3 & 0 & 0 \end{bmatrix}$$

$$\varphi = \begin{matrix} \mathbf{I} \\ \mathbf{I} \\ \mathbf{I} \\ \vdots \\ \mathbf{I} \end{matrix} \begin{bmatrix} 1 & 1 & 1 & \dots & 0 \\ 1 & 0 & 0 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 0,9 & 0,6 & \dots & 1 \\ 1 & 1 & 0,5 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0,7 & 1 & \dots & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & -0,9 & -0,6 & 0 \\ 0 & 0 & 0,5 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & -0,7 & 0 & 0 \end{bmatrix}$$

4.1.4.1.4 Matriks Kovarian

Membuat matriks kovarian, berdasarkan persamaan 2.5

C=

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0,1 & -0,3 & 0 & \dots & 0 & 0,1 & 0,4 & 0 & \dots & 0 & -0,9 & -0,6 & 0 \\ 0 & 0 & -0,6 & 0 & \dots & 0 & 0 & 0,5 & 0 & \dots & 0 & 0 & 0,5 & 0 \\ 0 & 0,3 & 0 & 0 & \dots & 0 & 0,3 & 0 & 0 & \dots & 0 & -0,7 & 0 & 0 \end{bmatrix} \mathbf{X} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0,3 \\ 0 & -0,3 & -0,6 & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0,3 \\ 0 & 0,4 & 0,5 & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \\ 0 & -0,9 & 0 & -0,7 \\ 0 & -0,6 & 0,5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0,7 & 1,1 & \dots & 0 \\ 0 & 0 & 22,8 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 22,9 & 0 & \dots & 0 \end{bmatrix}$$

4.1.4.1.5 Mencari Nilai *Eigenvector* dan *Eigenvalue*

Mencari nilai eigenvalue, sesuai dengan persamaan 2.7

$$\det(C - I) = 0$$

$$= \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0,7 & 1,1 & \dots & 0 \\ 0 & 0 & 22,8 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 22,9 & 0 & \dots & 0 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}$$

$$= \det \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 3,8477 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 5,5702 & \dots & 0 \end{bmatrix}$$

4.1.4.1.6 Mencari Nilai *Eigenvector*, sesuai dengan persamaan 2.8

Untuk mencari nilai *eigenvector* digunakan persamaan diatas tapi dengan mensubstitusikann nilai *eigenvalue*.

Maka didapatkan nilai eigen vectornya adalah :

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & -0.085959 & -0.11063 & \dots & 0.31913 \\ 0 & 0 & -0.030594 & \dots & 0.20889 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0.057852 & 0 & \dots & 0 \end{bmatrix}$$

- Mencari nilai principal component (PC)

$$U1 = V. 1 =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & -0.085959 & -0.11063 & \dots & 0.31913 \\ 0 & 0 & -0.030594 & \dots & 0.20889 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0.057852 & 0 & \dots & 0 \end{bmatrix}$$

$$X \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0,1 & -0,3 & \dots & 0 \\ 0 & 0 & -0,6 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0,3 & 0 & \dots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & -2.3086 & \dots & 0 \\ 0 & 0 & -2.7851 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1.1543 & 0 & \dots & 0 \end{bmatrix}$$

$$U2 = V. 2 =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & -0.085959 & -0.11063 & \dots & 0.31913 \\ 0 & 0 & -0.030594 & \dots & 0.20889 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0.057852 & 0 & \dots & 0 \end{bmatrix}$$

$$X \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0,1 & 0,4 & \dots & 0 \\ 0 & 0 & 0,5 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0,3 & 0 & \dots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1.5391 & \dots & 0 \\ 0 & 0 & 2.7851 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1.1543 & 0 & \dots & 0 \end{bmatrix}$$

$$U10 = V. 3 =$$

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & -0.085959 & -0.11063 & \dots & 0.31913 \\ 0 & 0 & -0.030594 & \dots & 0.20889 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0.057852 & 0 & \dots & 0 \end{bmatrix}$$

$$X \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & -0,9 & -0,6 & \dots & 0 \\ 0 & 0 & 0,5 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -0,7 & 0 & \dots & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & -0,9 & -0,6 & \dots & 0 \\ 0 & 0 & 0,5 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -0,7 & 0 & \dots & 0 \end{bmatrix}$$

4.1.5 Analisis Classification dengan Learning Vector Quantization

Setelah melalui beberapa tahapan yaitu data *collection – preprocessing – processing*, maka tahapan terakhir untuk mendapatkan hasil dari pengenalan karakter huruf t adalah classification. Tahap klasifikasi ini mengklasifikasikan vektor input yang telah diperoleh sebelumnya untuk dijadikan data latih (*training*) dan diuji dengan data uji (*testing*) dengan menerapkan jaringan syaraf tiruan yaitu dengan menggunakan metode *learning vector quantization 2.1*.

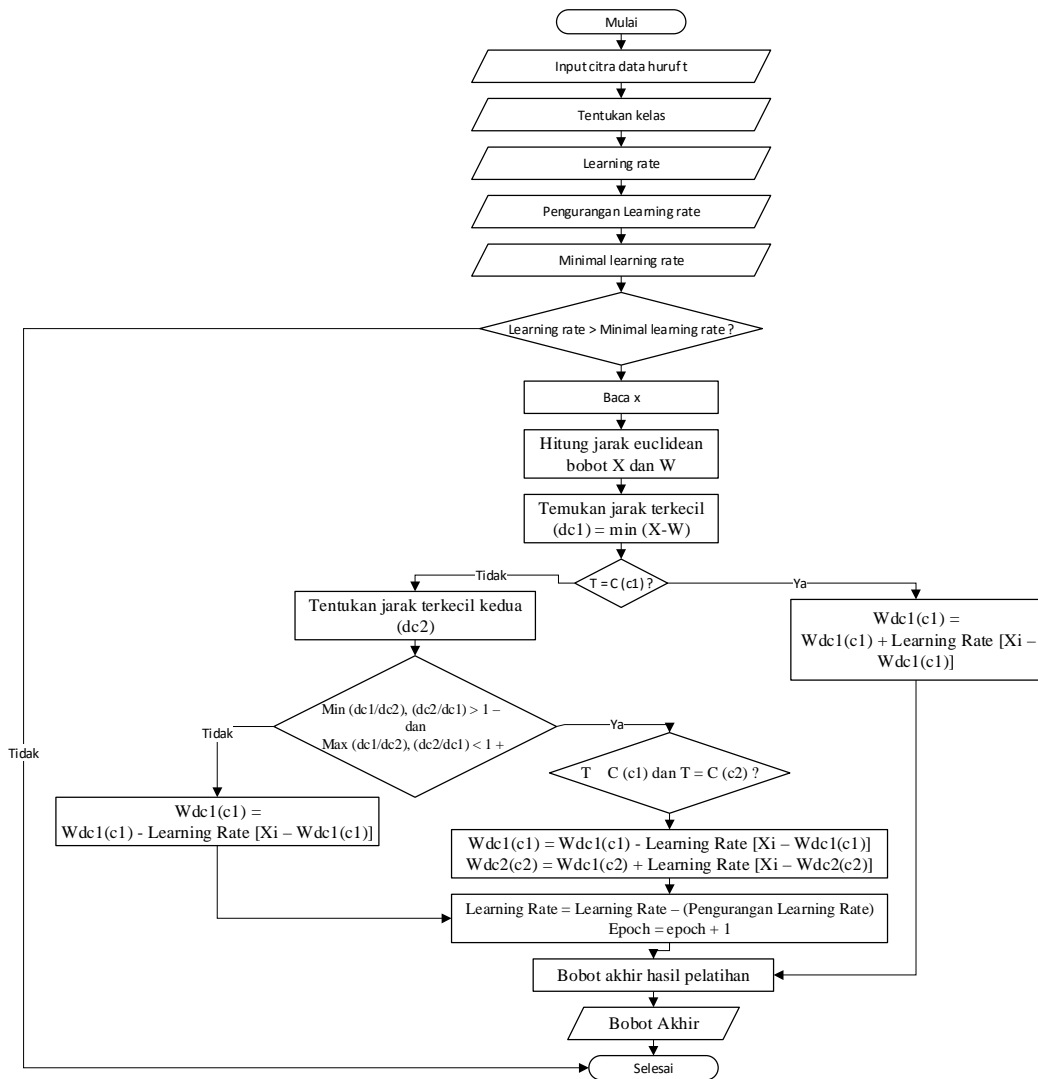
4.1.5.1 Flowchart

4.1.5.2 Learning Vector Quantization

Flowchart digunakan untuk menggambarkan aliran proses metode LVQ 2.1 agar lebih mudah dipahami dalam penerapannya pada penelitian ini. *Flowchart* LVQ 2.1 pada rancangan sistem pengenalan karakter seseorang berdasarkan citra pola tulisan huruf t ini terdiri atas dua bagian yaitu *flowchart* proses pembelajaran (*training*) LVQ2.1 dan *flowchart* proses pengujian (*testing*). Setiap bagiannya akan digambarkan seperti berikut.

1. *Flowchart* proses pembelajaran (*training*)

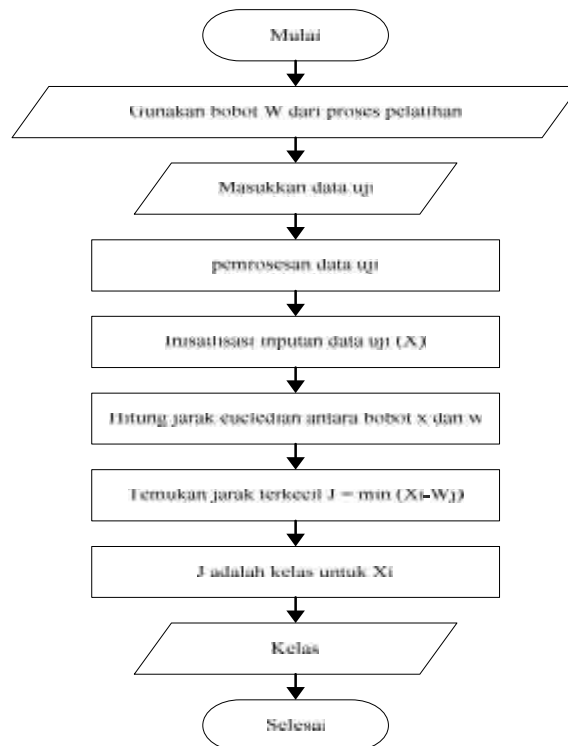
Flowchart proses pembelajaran untuk rancangan sistem pengenalan karakter seseorang berdasarkan citra pola tulisan huruf t dapat dilihat pada Gambar 4.5 berikut.



Gambar 4.5 Flowchart Proses Pembelajaran (training) LVQ 2.1

2. *Flowchart* proses pengujian (*testing*)

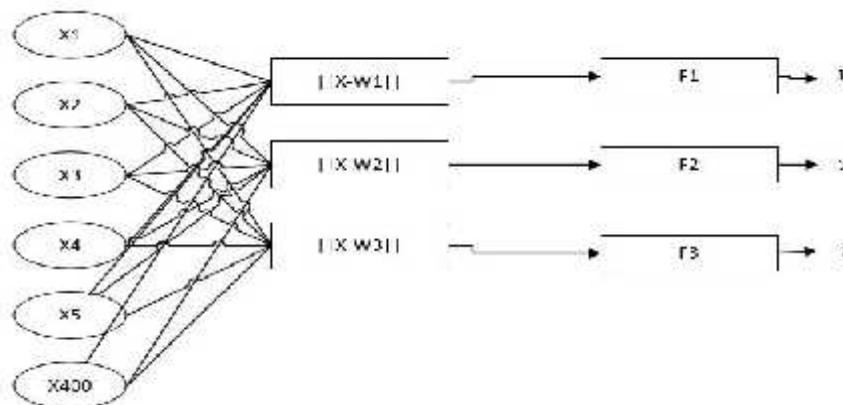
Flowchart proses pengujian untuk rancangan sistem pengenalan karakter seseorang berdasarkan citra pola tulisan huruf t dapat dilihat seperti Gambar 4.6 sebagai berikut.



Gambar 4.6 *Flowchart* Proses Pengujian (*testing*) LVQ 2.1

4.1.5.3 *Arsitektur Learning Vector Quantization*

Arsitektur jaringan syaraf tiruan dengan metode *learning vector quantization 2.1* pada pengenalan karakter huruf *t* dapat dilihat pada Gambar 4.7 berikut:



Gambar 4.7 *Arsitektur* LVQ 2.1

Keterangan Gambar 4.7 :

- x_1 sampai dengan x_m = nilai input
- $\|x - w_1\|$ sampai dengan $\|x - w_n\|$ = jarak bobot
- H_1 sampai dengan H_n = lapisan output
- D_1 sampai dengan D_n = nilai output
- m : jumlah elemen input
- n = jumlah kelas.

Pelatihan LVQ dikatakan berhasil jika pelatihan konvergen, dan gagal jika pelatihan divergen. Suatu pelatihan dikatakan konvergen jika kesalahan pada setiap iterasi pelatihan selalu mengecil, sampai pada titik dimana nilai bobot pada setiap neuron telah mencapai nilai yang paling baik untuk data pelatihan yang diberikan. Sebaliknya, pelatihan dikatakan divergen jika kesalahan pada pelatihan tidak cenderung mengecil menuju sebuah titik tertentu.

Adapun algoritma LVQ sebagai berikut [7]:

1. Tetapkan bobot (w), maksimum iterasi/epoch MaxEpoch), Error minimum (Eps) dan learning rate .
2. Masukkan :Input : $x(m,n)$; dimana
 - a. m = jumlah input dan
 - b. n = jumlah data Target : $T(1,n)$
3. Tetapkan kondisi awal : Epoch=0
4. Kerjakan jika: (epoch < MaxEpoch) atau ($>Eps$)
 - a. Epoch=epoch + 1
 - b. Kerjakan untuk $i = 1$ sampai n
 - i. Tentukan J sehingga $\|x-w_j\|$ adalah minimum
 - ii. Perbaiki w_j dengan ketentuan:
Jika $T = J$, maka: $w_j(\text{baru}) = w_j(\text{lama}) + (x - w_j(\text{lama}))$
Jika $T \neq J$, maka: $w_j(\text{baru}) = w_j(\text{lama}) - (x - w_j(\text{lama}))$
 - c. Kurangi nilai

4.1.6 Normalisasi Data

Pada perhitungan jarak *euclidean*, atribut berskala panjang dapat mempunyai pengaruh yang lebih besar dari pada atribut berskala pendek. Oleh sebab itu untuk mencegah hal tersebut maka harus dilakukan normalisasi data terhadap nilai atribut menjadi kisaran 0 sampai 1. Proses normalisasi tersebut menggunakan persamaan 2.7. Sedangkan data masukan ordinal yakni memiliki peringkat (rank), akan dinormalisasikan menggunakan persamaan 2.8.

Tabel 4.3 merupakan data citra huruf tulisan huruf t yang akan digunakan untuk mengklasifikasikan karakter seseorang dengan 3 kelas yang sudah ditentukan.

Tabel 4.4 Data Citra huruf t

No	Variabel	Data Citra Huruf t									
		t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
1	X ₁	0	0	0	0	0	0	0	0	0	0
2	X ₂	0	0	0	0	0	0	0	0	0	0
3	X ₃	0	0	0	0	0	0	0	0	0	0
4	X ₄	0	0	0	0	0	0	0	0	0	0
5	X ₅	0	0	0	0	0	0	0	0	0	0
6	X ₆	0	0	0	0	0	0	0	0	0	0
7	X ₇	0	0	0	0	0	0	0	0	0	0
8	X ₈	0	0	0	0	0	0	0	0	0	0
9	X ₉	0	0	0	0	0	0	0	0	0	0
10	X ₁₀	0	0	0	0	0	0	0	0	0	0
11	X ₈₆	0.39581	-0.92355	-0.92355	0.39581	0.39581	0.39581	-0.92355	0.39581	0.39581	0.39581
12	X ₈₇	0.26387	-1.0555	0.26387	0.26387	0.26387	-1.0555	0.26387	0.26387	0.26387	0.26387
13	X ₂₀₈	-2.0368	4.7508	-2.0368	-2.0368	-2.0368	-2.0368	4.7508	-2.0368	-2.0368	4.7508
14	X ₂₀₉	0	0	0	0	0	0	1.3574	1.3574	1.3574	-5.4294
15	X ₃₆₀	0	0	0	0	0	0	0	0	0	0
16	X ₂₁₁	0	0	0	0	0	0	0	0	0	0
17	X ₂₁₂	0	0	0	0	0	0	0	0	0	0

No	Variabel	Data Citra Huruf t									
		t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
18	X ₂₁₃	0	0	0	0	0	0	0.67868	0.67868	-6.1081	0.67868
19	X ₂₂₇	2.9797	-4.4695	2.9797	-4.4695	2.9797d	-4.4695	-4.4695	2.9797	2.9797	2.9797
20	X ₂₂₈	-3.7246	3.7246	3.7246	-3.7246	-3.7246	-3.7246	3.7246	-3.7246	3.7246	3.7246
21	X ₂₂₉	0	0	0	0	0	0	0	0	0	0
22	X ₂₃₀	0	0	0	0	0	0	0	0	0	0
23	X ₂₃₁	0	0	0	0	0	0	0	0	0	0
24	X ₂₃₂	0	0	0	0	0	0	0	0	0	0
25	X ₂₃₃	0	0	0	0	0	0	0	0	0	0
26	X ₂₆₇	1.8404	-7.3616	1.8404	1.8404	1.8404	1.8404	-7.3616	1.8404	1.8404	1.8404
27	X ₃₀₇	2.8282	-11.313	2.8282	2.8282	2.8282	2.8282	-11.313	2.8282	2.8282	2.8282
28	X ₃₁₀	-9.8986	4.2423	4.2423	-9.8986	4.2423	-9.8986	4.2423	4.2423	4.2423	4.2423
29	X ₃₆₉	2.4249	-21.824	2.4249	2.4249	2.4249	2.4249	2.4249	2.4249	2.4249	2.4249
30	X _n	0	0	0	0	0	0	0	0	0	0
Kelas		1	1	2	2	2	2	2	2	3	3

Keterangan Tabel diatas.

- 1 X_n adalah keseluruhan data, n = 400.
- 2 X₁, X₂, X₃, ..., X_n adalah variabel *input* yang mempengaruhi kriteria citra huruf t pada sistem klasifikasi penentuan karakter seseorang berdasarkan citra huruf t.
- 3 t₁ – t₁₀ merupakan citra data latih huruf t dari yang pertama sampai kesepuluh
- 4 Kelas adalah target yang akan dicapai oleh variabel citra huruf t tersebut.

- 5 Proses normalisasi untuk Tabel 4.3 diatas adalah sebagai berikut, dimana nilai “ya” dan “tidak” akan diubah kedalam angka 0 dan 1 dan nilai selain “ya” dan “tidak” akan dikonversi nilainya antara 0 sampai 1 karena variabelnya mengandung nilai range sesuai dengan data yang ada di Basis Data sistem pengenalan karakter berdasarkan tulisan huruf t.

Tabel 4.5 Normalisasi untuk Keterangan Ya dan Tidak

Keterangan	Normalisasi
Ya	1
Tidak	0

Berdasarkan Tabel 4.3, maka dihasilkan nilai-nilai baru berupa numerik yang dapat dilihat pada Tabel 4.5. Kemudian data tersebut akan dijadikan untuk inialisasi bobot dan data latih.

Tabel 4.6 Hasil Normalisasi Data Citra Huruf t yang akan dijadikan sebagai Inialisasi Bobot (Vektor W) dan Data Latih (Vektor X) dalam 3 Kelas

No	Variabel	Data Citra huruf t					
		Inialisasi Bobot (Vektor W)			Data Latih (Vektor X)		
		t1	t3	t9	t2	t4	t10
1	X ₁	0	0	0	0	0	0
2	X ₂	0	0	0	0	0	0
3	X ₃	0.83613	0	0	0	0	0.70978
4	X ₄	0.83613	0	0.73954	0.88076	0	0.73954
5	X ₅	0.78411	0	0	0.78411	0	0.50861
6	X ₆	0.53505	0.94998	0.40856	0.53505	0	0.26901
7	X ₇	0.33236	0.70994	0.19548	0.85441	0.52218	0
8	X ₈	0.46113	0.26468	0.44856	0.78842	0.68758	0.2139
9	X ₉	0.49752	0	0.22418	0.69366	0.70208	0.79898
10	X ₁₀	0.80323	0	0.57684	0.61794	0.42316	0.57684
11	X ₈₆	0.61109	0.80276	0.49261	0.35763	0.14212	0.3729
12	X ₈₇	0.37544	0.72957	0.24016	0.7747	0.553	0.055543
13	X ₂₀₈	0.26716	0	0.25006	0.97275	0.52258	0.87414
14	X ₂₀₉	0.56829	0.14083	0.33344	0.7368	0.80095	0.34889
15	X ₃₆₀	0.80323	0	0.57684	0.61794	0.42316	0.57684
16	X ₂₁₁	0.73644	0.46055	0	0.73644	0.47625	0.73644
17	X ₂₁₂	0.67211	0.30238	0.5239	0.73644	0.61409	0.73644
18	X ₂₁₃	0.76806	0.42893	0	0.76806	0.67995	0.76806
19	X ₂₂₇	0.81877	0.9316	0.70008	0.51684	0	0.6272
20	X ₂₂₈	0.10629	0.74888	0.81167	0.93293	0.38574	0.73153
21	X ₂₂₉	0.49752	0	0.22418	0.69366	0.70208	0.79898
22	X ₂₃₀	0.80323	0	0.57684	0.61794	0.42316	0.57684

No	Variabel	Data Citra huruf t					
		Inisialisasi Bobot (Vektor W)			Data Latih (Vektor X)		
		t1	t3	t9	t2	t4	t10
23	X ₂₃₁	0.73644	0.46055	0	0.73644	0.47625	0.73644
24	X ₂₃₂	0.67211	0.30238	0.5239	0.73644	0.61409	0.73644
25	X ₂₃₃	0.73644	0.36534	0.47417	0.73644	0.64431	0.73644
26	X ₂₆₇	0.63279	1.8404	0.50714	-7.3616	0.73719	0.38739
27	X ₃₀₇	0.79403	2.8282	0.67442	-11.313	0.85259	0.59531
28	X ₃₁₀	0	4.2423	1	4.2423	0	1
29	X ₃₆₉	0.62394	2.4249	0.41938	-21.824	0.87871	1
30	X _n	0	0	0	0	0	0

Sedangkan untuk data uji yang akan digunakan juga dinormalisasi seperti data latih pada Tabel 4.5 diatas. Tabel 4.6 merupakan data Citra huruf t untuk data uji yang sudah dinormalisasi.

Tabel 4.7 Data Citra huruf t yang sudah dinormalisasi, yang akan dijadikan sebagai Data Uji.

No	Variabel	Data Citra Huruf t
		t5
1	X ₁	0
2	X ₂	0
3	X ₃	0
4	X ₄	0
5	X ₅	0
6	X ₆	0
7	X ₇	0.26021
8	X ₈	0.68758
9	X ₉	0.70208
10	X ₁₀	0.57684
11	X ₈₆	0.14212
12	X ₈₇	0.3013
13	X ₂₀₈	0.52258
14	X ₂₀₉	0.80095
15	X ₃₆₀	0.57684
16	X ₂₁₁	0.46055
17	X ₂₁₂	0.63065
18	X ₂₁₃	0.60055
19	X ₂₂₇	0.72421
20	X ₂₂₈	0.38574
21	X ₂₂₉	0.70208
22	X ₂₃₀	0.57684
23	X ₂₃₁	0.46055
24	X ₂₃₂	0.63065
25	X ₂₃₃	0.55607
26	X ₂₆₇	0.5468
27	X ₃₀₇	0.70062

No	Variabel	Data Citra Huruf t
		t5
28	X ₃₁₀	1
29	X ₃₆₉	0.87871
30	X _n	0

Proses normalisasi data adalah sebagai berikut :

Data masukan yang dinormalisasikan dengan persamaan 2.7 adalah nilai data citra huruf t1

- a. Normalisasi untuk nilai data citra huruf t1 :

Nilai X untuk data pertama = 0

Nilai min (X) Nilai dari seluruh data citra huruf t1 = -15.805

Nilai max (X) Nilai data citra huruf t1 dari seluruh data = 9.6381

$$\begin{aligned} \text{Maka nilai } X^* \text{ (normalisasi)} &= X - \min(X) / \max(X) - \min(X) \\ &= 0 - (-15.805) / 9.6381 - (-15.805) \\ &= 0.62119003 \end{aligned}$$

- b. Normalisasi untuk nilai data citra huruf t2 :

Nilai X untuk data pertama = 0

Nilai min (X) Nilai data citra huruf t2 dari seluruh data = -21.824

Nilai max (X) Nilai luas lantai rumah dari seluruh data = 13.493

$$\begin{aligned} \text{Maka nilai } X^* \text{ (normalisasi)} &= X - \min(X) / \max(X) - \min(X) \\ &= 0 - (-21.824) / 13.493 - (-21.824) \\ &= 0.61794603 \end{aligned}$$

- c. Normalisasi data untuk citra huruf t3 :

Nilai X untuk data pertama = 0

Nilai min (X) Nilai data citra huruf t3 dari seluruh data = -9.5431

Nilai max (X) Nilai data citra huruf t3 dari seluruh data = 13.493

$$\begin{aligned} \text{Maka nilai } X^* \text{ (normalisasi)} &= X - \min(X) / \max(X) - \min(X) \\ &= 0 - (-9.5431) / 13.493 - (-9.5431) \\ &= 0.41426717 \end{aligned}$$

4.1.7 Perhitungan Manual Menggunakan Metode *Learning Vector Quantization* (LVQ) 2.1

Pada proses klasifikasi menggunakan *Learning Vector Quantization 2.1* (LVQ) 2.1, nilai dari hasil normalisasi yang berbentuk numerik akan menjadi acuan dalam menetapkan klasifikasi penentuan karakter seseorang dan sebagai tahapan proses LVQ 2.1 dalam menentukan kelas dari data uji. Berikut adalah tahapan.

4.1.7.1 Menetapkan parameter awal metode

Parameter yang digunakan yaitu :

- a. *Learning Rate* () = 0.01
- b. Pengurangan *Learning Rate* = 0.005*
- c. Minimal *Learning Rate* (min) = 0.01
- d. Nilai window () = 0.5

4.1.7.2 Menentukan input pertama pada data citra huruf t yang akan dijadikan sebagai inialisasi bobot (vektor W)

Tabel 4.8 Data Input (Vektor W) Pada Data Citra Huruf t

No	Variabel	Data Citra huruf t		
		Inialisasi Bobot (Vektor W)		
		t1	t3	t9
1	X ₁	0	0	0
2	X ₂	0	0	0
3	X ₃	0.83613	0	0
4	X ₄	0.83613	0	0.73954
5	X ₅	0.78411	0	0
6	X ₆	0.53505	0.94998	0.40856
7	X ₇	0.33236	0.70994	0.19548
8	X ₈	0.46113	0.26468	0.44856
9	X ₉	0.49752	0	0.22418
10	X ₁₀	0.80323	0	0.57684
11	X ₈₆	0.61109	0.80276	0.49261
12	X ₈₇	0.37544	0.72957	0.24016
13	X ₂₀₈	0.26716	0	0.25006
14	X ₂₀₉	0.56829	0.14083	0.33344
15	X ₃₆₀	0.80323	0	0.57684
16	X ₂₁₁	0.73644	0.46055	0
17	X ₂₁₂	0.67211	0.30238	0.5239
18	X ₂₁₃	0.76806	0.42893	0
19	X ₂₂₇	0.81877	0.9316	0.70008
20	X ₂₂₈	0.10629	0.74888	0.81167
21	X ₂₂₉	0.49752	0	0.22418
22	X ₂₃₀	0.80323	0	0.57684
23	X ₂₃₁	0.73644	0.46055	0

No	Variabel	Data Citra huruf t		
		Inisialisasi Bobot (Vektor W)		
		t1	t3	t9
24	X ₂₃₂	0.67211	0.30238	0.5239
25	X ₂₃₃	0.73644	0.36534	0.47417
26	X ₂₆₇	0.63279	1.8404	0.50714
27	X ₃₀₇	0.79403	2.8282	0.67442
28	X ₃₁₀	0	4.2423	1
29	X ₃₆₉	0.62394	2.4249	0.41938
30	X _n	0	0	0

4.1.7.3 Menentukan data citra huruf t untuk data latih (vektor X)

Pada perhitungan ini digunakan 2 data citra huruf t dapat dilihat pada Tabel 4.22 berikut.

Tabel 4.9 Data Latih (Vektor X) Citra Huruf t

No	Variabel	Data Citra huruf t		
		Inisialisasi Data Latih (Vektor X)		
		t2	t4	t6
1	X ₁	0	0	0
2	X ₂	0	0	0
3	X ₃	0	0	0
4	X ₄	0.88076	0	0
5	X ₅	0.78411	0	0
6	X ₆	0.53505	0	0
7	X ₇	0.85441	0.52218	0.26021
8	X ₈	0.78842	0.68758	0.68758
9	X ₉	0.69366	0.70208	0.70208
10	X ₁₀	0.61794	0.42316	0.57684
11	X ₈₆	0.35763	0.14212	0.14212
12	X ₈₇	0.7747	0.553	0.3013
13	X ₂₀₈	0.97275	0.52258	0.52258
14	X ₂₀₉	0.7368	0.80095	0.80095
15	X ₃₆₀	0.61794	0.42316	0.57684
16	X ₂₁₁	0.73644	0.47625	0.46055
17	X ₂₁₂	0.73644	0.61409	0.63065
18	X ₂₁₃	0.76806	0.67995	0.60055
19	X ₂₂₇	0.51684	0	0.72421
20	X ₂₂₈	0.93293	0.38574	0.38574
21	X ₂₂₉	0.69366	0.70208	0.70208
22	X ₂₃₀	0.61794	0.42316	0.57684
23	X ₂₃₁	0.73644	0.47625	0.46055
24	X ₂₃₂	0.73644	0.61409	0.63065
25	X ₂₃₃	0.73644	0.64431	0.55607
26	X ₂₆₇	0.29841	0.73719	0.5468
27	X ₃₀₇	0	0.85259	0.70062
28	X ₃₁₀	0.73806	0	1

No	Variabel	Data Citra huruf t		
		Inisialisasi Data Latih (Vektor X)		
		t2	t4	t6
29	X ₃₆₉	0	0.87871	0.87871
30	X _n	0	0	0

4.1.7.4 Melakukan perhitungan data latih pada *Epoch 1* atau iterasi 1.

Proses perhitungan data latih (proses pembelajaran) ini bertujuan untuk mencari jarak terdekat dari masing-masing kelas. Proses perhitungan dapat dilihat sebagai berikut, dimulai dari data latih ke-1 dengan menggunakan rumus 2.1 (perhitungan jarak *euclidean*).

0,0,0, 0.88076, 0.78411, 0.53505, 0.85441, 0.78842, 0.69366, 0.61794, 0.35763, 0.7747, 0.97275, 0.7368, 0.61794, 0.73644, 0.73644, 0.76806, 0.51684, 0.93293, 0.69366, 0.61794, 0.73644, 0.73644, 0.73644, 0.29841,0, 0.73806,0,0 → Data latih ke-1

0,0, 0.83613, 0.83613, 0.78411, 0.53505, 0.33236, 0.46113, 0.49752, 0.80323, 0.61109, 0.37544, 0.26716, 0.56829, 0.80323, 0.73644, 0.67211, 0.76806, 0.81877, 0.10629, 0.49752, 0.80323, 0.73644, 0.67211, 0.73644, 0.63279, 0.79403,0, 0.62394,0 → Bobot

T = 1

Bobot ke-1

=

$$\sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 0.83613)^2 + (0.88076 - 0.83613)^2 + (0.78411 - 0.78411)^2 + (0.53505 - 0.53505)^2 + (0.85441 - 0.33236)^2 + (0.14212 - 0.46113)^2 + (0.553 - 0.49752)^2 + (0.52258 - 0.80323)^2 + (0.80095 - 0.61109)^2 + (0.42316 - 0.37544)^2 + (0.97275 - 0.26716)^2 + (0.7368 - 0.56829)^2 + (0.61794 - 0.80323)^2 + (0.73644 - 0.73644)^2 + (0.73644 - 0.67211)^2 + (0.76806 - 0.76806)^2 + (0.51684 - 0.81877)^2 + (0.93293 - 0.10629)^2 + (0.69366 - 0.49752)^2 + (0.61794 - 0.80323)^2 + (0.73644 - 0.73644)^2 + (0.73644 - 0.67211)^2 + (0.73644 - 0.73644)^2 + (0.29841 - 0.63279)^2 + (0 - 0.79403)^2 + (0.73806 - 0)^2 + (0 - 0.62394)^2 + (0 - 0)^2}$$

=

$$\sqrt{0 + 0 + 0.69911 + 0.01991 + 0 + 0 + 0.17350 + 0.10176 + 0.00307 + 0.07876 + 0.03604 + 0.04772 + 0.49785 + 0.02839 + 0.03433 + 0 + 0.00413 + 0 + 0.81877 + 0.82664 + 0.19614 + 0.03433 + 0 + 0.06433 + 0 + 0.11180 + 0.63048 + 0.54473 + 0.38930 + 0}$$

$$= \sqrt{5.34109}$$

$$= 2.31107$$

0,0,0, 0.88076, 0.78411, 0.53505, 0.85441, 0.78842, 0.69366, 0.61794, 0.35763, 0.7747,
 0.97275, 0.7368, 0.61794, 0.73644, 0.73644, 0.76806, 0.51684, 0.93293, 0.69366,
 0.61794, 0.73644, 0.73644, 0.73644, 0.29841,0, 0.73806,0,0 → Data latih ke-1

0, 0, 0, 0, 0, 0.94998, 0.70994, 0.26468, 0, 0, 0.80276, 0.72957, 0, 0.14083, 0, 0.46055,
 0.30238, 0.42893, 0.9316, 0.74888, 0, 0, 0.46055, 0.30238, 0.36534, 0.84685, 0.92033,
 0.3144, 0.25159, 0 → Bobot

$$T = 1$$

Bobot ke-2

=

$$\sqrt{\begin{aligned} &(0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0.88076 - 0)^2 + (0.78411 - 0)^2 + \\ &(0.53505 - 0.94998)^2 + (0.85441 - 0.70994)^2 + (0.14212 - 0.26468)^2 + \\ &(0.553 - 0)^2 + (0.52258 - 0)^2 + (0.80095 - 0.80276)^2 + \\ &(0.42316 - 0.72957)^2 + (0.97275 - 0)^2 + (0.7368 - 0.14083)^2 + \\ &(0.61794 - 0)^2 + (0.73644 - 0.46055)^2 + (0.73644 - 0.30238)^2 + \\ &(0.76806 - 0.42893)^2 + (0.51684 - 0.9316)^2 + (0.93293 - 0.74888)^2 + \\ &(0.69366 - 0)^2 + (0.61794 - 0)^2 + (0.73644 - 0.46055)^2 + \\ &(0.73644 - 0.30238)^2 + (0.73644 - 0.36534)^2 + (0.29841 - 0.84685)^2 + \\ &(0 - 0.92033)^2 + (0.73806 - 0.3144)^2 + (0 - 0.25159)^2 + (0 - 0)^2 \end{aligned}}$$

=

$$\sqrt{\begin{aligned} &0 + 0 + 0 + 0.77573 + 0.61482 + 0.17216 + 0.02087 + 0.01502 + 0.30580 + 0.27308 \\ &+ 0 + 0.09388 + 0.94624 + 0.35518 + 0.38184 + 0.07611 + 0.18840 + 0.11500 + \\ &0.17202 + 0.03387 + 0.38184 + 0.54234 + 0.07611 + 0.18840 + 0.13771 + 0.30078 \\ &+ 0.84700 + 0.42366 + 0.06329 + 0 \end{aligned}}$$

$$= \sqrt{7.85633}$$

$$= 2.80291$$

0,0,0, 0.88076, 0.78411, 0.53505, 0.85441, 0.78842, 0.69366, 0.61794, 0.35763, 0.7747,
 0.97275, 0.7368, 0.61794, 0.73644, 0.73644, 0.76806, 0.51684, 0.93293, 0.69366,
 0.61794, 0.73644, 0.73644, 0.73644, 0.29841,0, 0.73806,0,0 → Data latih ke-1

0, 0, 0, 0.73954, 0, 0.40856, 0.19548, 0.44856, 0.22418, 0.57684, 0.49261, 0.24016,
 0.25006, 0.33344, 0.57684, 0, 0.5239, 0, 0.70008, 0.81167, 0.22418, 0.57684, 0, 0.5239,
 0.47417, 0.50714, 0.67442, 1, 0.41938, 0 → Bobot

T = 1

Bobot ke-3

=

$$\sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0.88076 - 0.73954)^2 + (0.78411 - 0)^2 + (0.53505 - 0.40856)^2 + (0.85441 - 0.19548)^2 + (0.14212 - 0.44856)^2 + (0.553 - 0.22418)^2 + (0.52258 - 0.57684)^2 + (0.80095 - 0.49261)^2 + (0.42316 - 0.24016)^2 + (0.97275 - 0.25006)^2 + (0.7368 - 0.33344)^2 + (0.61794 - 0.57684)^2 + (0.73644 - 0)^2 + (0.73644 - 0.5239)^2 + (0.76806 - 0)^2 + (0.51684 - 0.70008)^2 + (0.93293 - 0.81167)^2 + (0.69366 - 0.22418)^2 + (0.61794 - 0.57684)^2 + (0.73644 - 0)^2 + (0.73644 - 0.5239)^2 + (0.73644 - 0.47417)^2 + (0.29841 - 0.50714)^2 + (0 - 0.67442)^2 + (0.73806 - 1)^2 + (0 - 0.41938)^2 + (0 - 0)^2}$$

=

$$\sqrt{0 + 0 + 0 + 0.01994 + 0.61482 + 0.01599 + 0.43418 + 0.09390 + 0.10812 + 0.00294 + 0.55528 + 0.03348 + 0.52228 + 0.02646 + 0.00168 + 0.54234 + 0.04517 + 0.58991 + 0.03357 + 0.01470 + 0.22041 + 0.00168 + 0.54234 + 0.04517 + 0.06878 + 0.04356 + 0.45484 + 0.06861 + 0.17587 + 0}$$

$$= \sqrt{5.49240}$$

$$= 2.34358$$

Jarak terkecil adalah bobot ke-1. (C = 1) = 1. Target data latih adalah 1, Sehingga C=T. Maka bobot ke-1 diperbarui dengan rumus 2.2.

W₁(baru) =

(0,0, 0.83613, 0.83613, 0.78411, 0.53505, 0.33236, 0.46113, 0.49752, 0.80323, 0.61109, 0.37544, 0.26716, 0.56829, 0.80323, 0.73644, 0.67211, 0.76806, 0.81877, 0.10629, 0.49752, 0.80323, 0.73644, 0.67211, 0.73644, 0.63279, 0.79403,0, 0.62394,0) + 0.05 { (0,0,0, 0.88076, 0.78411, 0.53505, 0.85441, 0.78842, 0.69366, 0.61794, 0.35763, 0.7747, 0.97275, 0.7368, 0.61794, 0.73644, 0.73644, 0.76806, 0.51684, 0.93293, 0.69366, 0.61794, 0.73644, 0.73644, 0.73644, 0.29841,0, 0.73806,0,0) –

(0,0, 0.83613, 0.83613, 0.78411, 0.53505, 0.33236, 0.46113, 0.49752, 0.80323, 0.61109, 0.37544, 0.26716, 0.56829, 0.80323, 0.73644, 0.67211, 0.76806, 0.81877, 0.10629, 0.49752, 0.80323, 0.73644, 0.67211, 0.73644, 0.63279, 0.79403,0, 0.62394,0)}

= 0, 0, 0.87794, 0.8339, 0.78411, 0.53505, 0.30626, 0.44476, 0.48772, 0.8125, 0.62377, 0.35547, 0.23188, 0.55986, 0.8125, 0.73644, 0.66889, 0.76806, 0.83386, 0.064957, 0.48772, 0.8125, 0.73644, 0.66889, 0.73644, 0.64951, 0.83374, 0.036903, 0.65514, 0

$W_1(\text{baru}) =$

0, 0, 0.87794, 0.8339, 0.78411, 0.53505, 0.30626, 0.44476, 0.48772, 0.8125, 0.62377, 0.35547, 0.23188, 0.55986, 0.8125, 0.73644, 0.66889, 0.76806, 0.83386, 0.064957, 0.48772, 0.8125, 0.73644, 0.66889, 0.73644, 0.64951, 0.83374, 0.036903, 0.65514, 0

Kemudian lanjut kepada data latih ke-2 dengan rumus 2.1 (perhitungan jarak *euclidean*).

0, 0 ,0, 0, 0, 0, 0.52218, 0.68758, 0.70208, 0.42316, 0.14212, 0.553, 0.52258, 0.80095, 0.42316, 0.47625, 0.61409, 0.67995, 0, 0.38574, 0.70208, 0.42316, 0.47625, 0.61409, 0.64431, 0.73719, 0.85259, 0, 0.87871, 0 → Data latih ke-2

0, 0, 0.87794, 0.8339, 0.78411, 0.53505, 0.30626, 0.44476, 0.48772, 0.8125, 0.62377, 0.35547, 0.23188, 0.55986, 0.8125, 0.73644, 0.66889, 0.76806, 0.83386, 0.064957, 0.48772, 0.8125, 0.73644, 0.66889, 0.73644, 0.64951, 0.83374, 0.036903, 0.65514, 0 → Bobot

$T = 2$

Bobot ke-1

=

$$\begin{aligned}
& \sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 0.87794)^2 + (0 - 0.8339)^2 + (0 - 0.78411)^2 +} \\
& \quad (0 - 0.53505)^2 + (0.52218 - 0.30626)^2 + (0.68758 - 0.44476)^2 + \\
& \quad (0.70208 - 0.48772)^2 + (0.42316 - 0.8125)^2 + (0.14212 - 0.62377)^2 + \\
& \quad (0.553 - 0.35547)^2 + (0.52258 - 0.23188)^2 + (0.80095 - 0.55986)^2 + \\
& \quad (0.42316 - 0.8125)^2 + (0.47625 - 0.73644)^2 + (0.61409 - 0.66889)^2 + \\
& \quad (0.67995 - 0.76806)^2 + (0 - 0.83386)^2 + (0.38574 - 0.064957)^2 + \\
& \quad (0.70208 - 0.48772)^2 + (0.42316 - 0.8125)^2 + (0.47625 - 0.73644)^2 + \\
& \quad (0.61409 - 0.66889)^2 + (0.64431 - 0.73644)^2 + (0.73719 - 0.64951)^2 + \\
& \quad (0.85259 - 0.83374)^2 + (0 - 0.036903)^2 + (0.87871 - 0.65514)^2 + (0 - 0)^2} \\
& = \sqrt{8.8357} \\
& = 2.97249
\end{aligned}$$

0, 0, 0, 0, 0, 0, 0.52218, 0.68758, 0.70208, 0.42316, 0.14212, 0.553, 0.52258, 0.80095, 0.42316, 0.47625, 0.61409, 0.67995, 0, 0.38574, 0.70208, 0.42316, 0.47625, 0.61409, 0.64431, 0.73719, 0.85259, 0, 0.87871, 0 → Data latih ke-2

0, 0, 0, 0, 0, 0.94998, 0.70994, 0.26468, 0, 0, 0.80276, 0.72957, 0, 0.14083, 0, 0.46055, 0.30238, 0.42893, 0.9316, 0.74888, 0, 0, 0.46055, 0.30238, 0.36534, 0.84685, 0.92033, 0.3144, 0.25159, 0 → Bobot

T = 1

Bobot ke-2

$$\begin{aligned}
& \sqrt{(0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 +} \\
& \quad (0 - 0.94998)^2 + (0.52218 - 0.70994)^2 + (0.68758 - 0.26468)^2 + \\
& \quad (0.70208 - 0)^2 + (0.42316 - 0)^2 + (0.14212 - 0.80276)^2 + \\
& \quad (0.553 - 0.72957)^2 + (0.52258 - 0)^2 + (0.80095 - 0.14083)^2 + \\
& \quad (0.42316 - 0)^2 + (0.47625 - 0.46055)^2 + (0.61409 - 0.30238)^2 + \\
& \quad (0.67995 - 0.42893)^2 + (0 - 0.9316)^2 + (0.38574 - 0.74888)^2 + \\
& \quad (0.70208 - 0)^2 + (0.42316 - 0)^2 + (0.47625 - 0.46055)^2 + \\
& \quad (0.61409 - 0.30238)^2 + (0.64431 - 0.36534)^2 + (0.73719 - 0.84685)^2 + \\
& \quad (0.85259 - 0.92033)^2 + (0 - 0.3144)^2 + (0.87871 - 0.25159)^2 + (0 - 0)^2} \\
& \sqrt{0 + 0 + 0 + 0 + 0 + 0.90246 + 0.03525 + 0.17884 + 0.49291 + 0.17906} \\
& \quad + 0.43644 + 0.03117 + 0.27308 + 0.43575 + 0.17906 + 0.00024 + 0.09716 + 0.06301 + \\
& \quad 0.86787 + 0.13187 + 0.49291 + 0.17906 + 0.00024 + 0.09716 + 0.07782 + 0.01202 \\
& \quad + 0.00458 + 0.09884 + 0.39327 + 0} \\
& = \sqrt{6.5689} \\
& = 2.56298
\end{aligned}$$

0, 0, 0, 0, 0, 0, 0.52218, 0.68758, 0.70208, 0.42316, 0.14212, 0.553, 0.52258, 0.80095, 0.42316, 0.47625, 0.61409, 0.67995, 0, 0.38574, 0.70208, 0.42316, 0.47625, 0.61409, 0.64431, 0.73719, 0.85259, 0, 0.87871, 0 → Data latih ke-2

0, 0, 0, 0.73954, 0, 0.40856, 0.19548, 0.44856, 0.22418, 0.57684, 0.49261, 0.24016, 0.25006, 0.33344, 0.57684, 0, 0.5239, 0, 0.70008, 0.81167, 0.22418, 0.57684, 0, 0.5239, 0.47417, 0.50714, 0.67442, 1, 0.41938, 0 → Bobot

T = 1

Bobot ke-3

=

$$\sqrt{\begin{aligned} &(0 - 0)^2 + (0 - 0)^2 + (0 - 0)^2 + (0 - 0.79433)^2 + (0 - 0)^2 + \\ &(0 - 0.40856)^2 + (0.52218 - 0.19548)^2 + (0.68758 - 0.44856)^2 + \\ &(0.70208 - 0.22418)^2 + (0.42316 - 0.57684)^2 + (0.14212 - 0.49261)^2 + \\ &(0.553 - 0.24016)^2 + (0.52258 - 0.25006)^2 + (0.80095 - 0.33344)^2 + \\ &(0.42316 - 0.57684)^2 + (0.47625 - 0)^2 + (0.61409 - 0.5239)^2 + \\ &(0.67995 - 0)^2 + (0 - 0.70008)^2 + (0.38574 - 0.81167)^2 + \\ &(0.70208 - 0.22418)^2 + (0.42316 - 0.57684)^2 + (0.47625 - 0)^2 + \\ &(0.61409 - 0.5239)^2 + (0.64431 - 0.47417)^2 + (0.73719 - 0.50714)^2 + \\ &(0.85259 - 0.67442)^2 + (0 - 1)^2 + (0.87871 - 0.41938)^2 + (0 - 0)^2 \end{aligned}}$$

$$= \sqrt{6.4686}$$

$$= 2.54334$$

Jarak terkecil adalah bobot ke-3. (C = 3) = 2.54334. Target data latih adalah 1, Sehingga C T. Maka periksa apakah jarak *runner-up* masih masuk dalam window () dengan rumus 2.4.

D₁ (Jarak terdekat pertama) = 2.54334 dan D₂ (Jarak terdekat kedua) = 2.56298

$$= \text{Min} \left[\frac{2.5}{2.5}, \frac{2.5}{2.5} \right] > 1 - 0.5$$

dan

$$\text{Max} \left[\frac{2.5}{2.5}, \frac{2.5}{2.5} \right] < 1 + 0.5$$

$$= \text{Min} [0.992 , 1.007] > 0.5 \text{ dan } \text{Max} [0.996 , 1.003] < 1.5$$

$$= (T) \text{ dan } (T) = T$$

Nilainya adalah *True*. Kedua bobot (W) dengan jarak terkecil (W_3 dan W_2) diperbarui dengan ketentuan :

Vektor W yang tidak termasuk kedalam kelas yang sama dengan vektor X akan diperbaharui dengan dengan rumus 2.6.

$$W_3(\text{baru}) = (0, 0, 0, 0.73954, 0, 0.40856, 0.19548, 0.44856, 0.22418, 0.57684, 0.49261, 0.24016, 0.25006, 0.33344, 0.57684, 0, 0.5239, 0, 0.70008, 0.81167, 0.22418, 0.57684, 0, 0.5239, 0.47417, 0.50714, 0.67442, 1, 0.41938, 0) - 0.05$$

$$\{(0, 0, 0, 0, 0, 0, 0.52218, 0.68758, 0.70208, 0.42316, 0.14212, 0.553, 0.52258, 0.80095, 0.42316, 0.47625, 0.61409, 0.67995, 0, 0.38574, 0.70208, 0.42316, 0.47625, 0.61409, 0.64431, 0.73719, 0.85259, 0, 0.87871, 0) - (0, 0, 0, 0.73954, 0, 0.40856, 0.19548, 0.44856, 0.22418, 0.57684, 0.49261, 0.24016, 0.25006, 0.33344, 0.57684, 0, 0.5239, 0, 0.70008, 0.81167, 0.22418, 0.57684, 0, 0.5239, 0.47417, 0.50714, 0.67442, 1, 0.41938, 0)\}$$

$$= 0, 0, 0, 0.77652, 0, 0.42898, 0.17914, 0.43661, 0.36029, 0.58452, 0.51013, 0.22452, 0.23644, 0.31007, 0.58452, -0.023813, 0.51939, 0.033997, 0.73508, 0.83297, 0.36029, 0.58452, 0.023813, 0.51939, 0.46567, 0.66551, 1, 0.39641, 0$$

$$W_3(\text{baru}) = (0, 0, 0, 0.77652, 0, 0.42898, 0.17914, 0.43661, 0.36029, 0.58452, 0.51013, 0.22452, 0.23644, 0.31007, 0.58452, -0.023813, 0.51939, 0.033997, 0.73508, 0.83297, 0.36029, 0.58452, 0.023813, 0.51939, 0.46567, 0.66551, 1, 0.39641, 0$$

$$W_1(\text{baru}) = (0, 0, 0.87794, 0.8339, 0.78411, 0.53505, 0.30626, 0.44476, 0.48772, 0.8125, 0.62377, 0.35547, 0.23188, 0.55986, 0.8125, 0.73644, 0.66889, 0.76806, 0.83386, 0.064957, 0.48772, 0.8125, 0.73644, 0.66889, 0.73644, 0.64951, 0.83374, 0.036903, 0.65514, 0) - 0.05 \{$$

$$\begin{aligned}
& (0, 0, 0, 0, 0, 0, 0.52218, 0.68758, 0.70208, 0.42316, 0.14212, 0.553, 0.52258, \\
& 0.80095, 0.42316, 0.47625, 0.61409, 0.67995, 0, 0.38574, 0.70208, 0.42316, \\
& 0.47625, 0.61409, 0.64431, 0.73719, 0.85259, 0, 0.87871, 0) - \\
& (0, 0, 0.87794, 0.8339, 0.78411, 0.53505, 0.30626, 0.44476, 0.48772, \\
& 0.8125, 0.62377, 0.35547, 0.23188, 0.55986, 0.8125, 0.73644, 0.66889, \\
& 0.76806, 0.83386, 0.064957, 0.48772, 0.8125, 0.73644, 0.66889, \\
& 0.73644, 0.64951, 0.83374, 0.036903, 0.65514, 0) \} \\
& = 0, 0, 0.043897, 0.041695, 0.039205, 0.026753, 0.53297, 0.69972, \\
& 0.71279, 0.4037, 0.11803, 0.56288, 0.53711, 0.813, 0.4037, 0.46324, \\
& 0.61135, 0.67554, 0.041693, 0.40178, 0.71279, 0.4037, 0.46324, \\
& 0.61135, 0.6397, 0.74157, 0.85353, 0.00184, 0.88989, 0
\end{aligned}$$

$$\begin{aligned}
W_1(\text{baru}) = & 0, 0, 0.043897, 0.041695, 0.039205, 0.026753, 0.53297, 0.69972, \\
& 0.71279, 0.4037, 0.11803, 0.56288, 0.53711, 0.813, 0.4037, 0.46324, \\
& 0.61135, 0.67554, 0.041693, 0.40178, 0.71279, 0.4037, 0.46324, \\
& 0.61135, 0.6397, 0.74157, 0.85353, 0.00184, 0.88989, 0
\end{aligned}$$

Sedangkan vektor W yang termasuk kedalam kelas yang sama dengan vektor X akan diperbaharui dengan dengan rumus 2.5.

$$\begin{aligned}
W_2(\text{baru}) = & (0, 0, 0, 0, 0, 0.94998, 0.70994, 0.26468, 0, 0, 0.80276, 0.72957, 0, 0.14083, \\
& 0, 0.46055, 0.30238, 0.42893, 0.9316, 0.74888, 0, 0, 0.46055, 0.30238, \\
& 0.36534, 0.84685, 0.92033, 0.3144, 0.25159, 0) + 0.05 \\
& \{ (0, 0, 0, 0, 0, 0, 0.52218, 0.68758, 0.70208, 0.42316, 0.14212, 0.553, \\
& 0.52258, 0.80095, 0.42316, 0.47625, 0.61409, 0.67995, 0, 0.38574, 0.70208, \\
& 0.42316, 0.47625, 0.61409, 0.64431, 0.73719, 0.85259, 0, 0.87871, 0) - \\
& (0, 0, 0, 0, 0, 0.94998, 0.70994, 0.26468, 0, 0, 0.80276, 0.72957, 0, 0.14083, \\
& 0, 0.46055, 0.30238, 0.42893, 0.9316, 0.74888, 0, 0, 0.46055, 0.30238, \\
& 0.36534, 0.84685, 0.92033, 0.3144, 0.25159, 0) \} \\
& = 0, 0, 0, 0, 0, 0.90248, 0.70055, 0.28583, 0.035104, 0.021158, 0.76973, \\
& 0.72074, 0.026129, 0.17384, 0.021158, 0.46134, 0.31797, 0.44148, \\
& 0.88502, 0.73072, 0.035104, 0.021158, 0.46134, 0.31797, 0.37929, \\
& 0.84136, 0.91694, 0.29868, 0.28295, 0
\end{aligned}$$

W₂(baru) = 0, 0, 0, 0, 0, 0.90248, 0.70055, 0.28583, 0.035104, 0.021158, 0.76973,
 0.72074, 0.026129, 0.17384, 0.021158, 0.46134, 0.31797, 0.44148,
 0.88502, 0.73072, 0.035104, 0.021158, 0.46134, 0.31797, 0.37929,
 0.84136, 0.91694, 0.29868, 0.28295, 0

Maka dihasilkan :

W₃(baru)= 0, 0, 0, 0.77652, 0, 0.42898, 0.17914, 0.43661, 0.36029,
 0.58452, 0.51013, 0.22452, 0.23644, 0.31007, 0.58452, -0.023813,
 0.51939, 0.033997, 0.73508, 0.83297, 0.36029, 0.58452, 0.023813,
 0.51939, 0.46567, 0.66551, 1, 0.39641, 0

W₁(baru)= 0, 0, 0.043897, 0.041695, 0.039205, 0.026753, 0.53297, 0.69972,
 0.71279, 0.4037, 0.11803, 0.56288, 0.53711, 0.813, 0.4037, 0.46324,
 0.61135, 0.67554, 0.041693, 0.40178, 0.71279, 0.4037, 0.46324,
 0.61135, 0.6397, 0.74157, 0.85353, 0.00184, 0.88989, 0

W₂(baru)= 0, 0, 0, 0, 0, 0.90248, 0.70055, 0.28583, 0.035104, 0.021158, 0.76973,
 0.72074, 0.026129, 0.17384, 0.021158, 0.46134, 0.31797, 0.44148,
 0.88502, 0.73072, 0.035104, 0.021158, 0.46134, 0.31797, 0.37929,
 0.84136, 0.91694, 0.29868, 0.28295, 0

Selanjutnya dilakukan hingga kondisi minimal *learning rate* (min) terpenuhi. Setelah diperoleh bobot akhir pada proses pembelajaran maka bobot tersebut digunakan dalam proses pengujian (*testing*). Algoritma proses pengujian ini dimulai dengan memasukkan data yang akan diuji dan bobot akhir yang didapat, kemudian menghitung jarak terdekatnya untuk selanjutnya dicari nilai atau bobot yang paling kecil. Jika bobot terkecil tersebut sesuai dengan target maka proses pengujian benar, jika tidak maka pengujian tersebut tidak akurat. Berikut adalah perhitungan pengujian :

Data Uji : 0, 0, 0, 0, 0., 0, 0.26021, 0.68758, 0.70208, 0.57684, 0.14212, 0.3013, 0.52258,
 0.80095, 0.57684, 0.46055, 0.63065, 0.60055, 0.72421, 0.38574, 0.70208,
 0.57684, 0.46055, 0.63065, 0.55607, 0.5468, 0.70062, 1, 0.87871, 0

Tentukan target = 2. Target digunakan untuk melihat kesesuaian antara target dan hasil perhitungan oleh sistem. Jika target dan perhitungan oleh sistem

hasilnya sama, maka perhitungan sistem akurat. Kemudian hitung jarak *euclidean* antara vektor input (data uji) dan bobot (W) dengan rumus 2.1.

Bobot ke-1

$$\sqrt{\begin{aligned} &(0 - 9.10087)^2 + (0 - 2.70122)^2 + (0 - 5.28259)^2 + (0 - 5.35285)^2 + (0 - -2.11360)^2 + \\ &(0 - 8.39824)^2 + (0.26021 - 1.48878)^2 + (0.68758 - 1.80992)^2 + \\ &(0.70208 - -1.59616)^2 + (0.57684 - -2.77626)^2 + (0.14212 - -3.69831)^2 + \\ &(0.3013 - -3.33609)^2 + (0.52258 - 2.67586)^2 + (0.80095 - 4.58776)^2 + \\ &(0.57684 - 2.04288)^2 + (0.46055 - -9.56021)^2 + (0.63065 - -1.89982)^2 + \\ &(0.60055 - -5.16676)^2 + (0.72421 - -1.98458)^2 + (0.38574 - 1.00550)^2 + \\ &(0.70208 - 6.94791)^2 + (0.57684 - 0.57684)^2 + (0.46055 - 8.19360)^2 + \\ &(0.63065 - 3.52562)^2 + (0.55607 - 6.35290)^2 + (0.5468 - -2.30065)^2 + \\ &(0.70062 - -0.00267)^2 + (1 - 0.05417)^2 + (0.87871 - 11.59972)^2 + (0 - 0)^2 \end{aligned}}$$

=

$$\sqrt{\begin{aligned} &0 + 0 + 0 + 0 + 0 + 0 + 1.50938 + 1.25964 + 5.28190 + 11.24327 \\ &+14.74890 + 13.12170 + 4.63661 + 14.33992 + 2.14927 + 100.41563 + 5.70126 + \\ &33.26186 + 7.33775 + 0.38410 + 39.010 + 0 + 59.77686 \\ &+8.38085 + 33.60323 + 8.10797 + 0.49461 \\ &+0.89459 + 0.09884 + 114.94005 + 0 \end{aligned}}$$

= 2.54334

4.1.8 Parameter *Learning Vector Quantization 2.1* yang digunakan dalam Matlab

Parameter *learning vector quantization 2.1* yang digunakan dalam Matlab dapat dilihat pada Algoritma 4.3 berikut:

```
function [waktu Wlvq WlvqC ep ] = lvq(data, kelas, lr, win, epoch)
    plr=0.1;
    mlr=0;
```

Algoritma 4.3 parameter learning vector quantization 2.1 yang digunakan dalam matlab

Keterangan Algoritma 4.3 adalah sebagai berikut:

1. Lr = *learning rate*
2. Win = *window*

3. Plr = Penurunan *learning rate*

4. Mlr = *Minimum learning rate*

4.2 Perancangan

Tahap perancangan adalah tahap yang dilakukan setelah tahap analisa, Perancangan dilakukan untuk dijadikan pedoman dalam membuat sebuah aplikasi pengenalan karakter huruf t. Perancangan yang dilakukan adalah *flowchart*, dan perancangan tampilan (*interface*) untuk aplikasi yang akan dibuat.

4.2.1 Flowchart

Alur proses aplikasi pengenalan karakter huruf t menggunakan metode *Learning Vector Quatization 2.1* dapat dilihat pada Gambar 4.8 *flowchart* di bawah ini.



Gambar 4.8 *Flowchart* Aplikasi Pengenalan Karakter Huruf t

Alur proses aplikasi yang dibuat dimulai dengan menginputkan citra data latih huruf t pada aplikasi. Setelah itu pilih kelas citra data latih huruf t, kemudian gambar rgb akan dikonversikan ke grayscale dan gambar grayscale akan di konversikan ke biner, kemudian dilakukan *cropping* untuk meminimalkan *background* pada objek huruf. Setelah itu gambar akan dilakukan *resize* untuk mengubah ukuran matriks menjadi matriks berukuran 20x20 piksel. Kemudian akan dilakukan proses klasifikasi menggunakan metode jaringan syaraf tiruan *learning vector quantization 2.1* sehingga karakter huruf yang di-input-kan dapat dikenali.

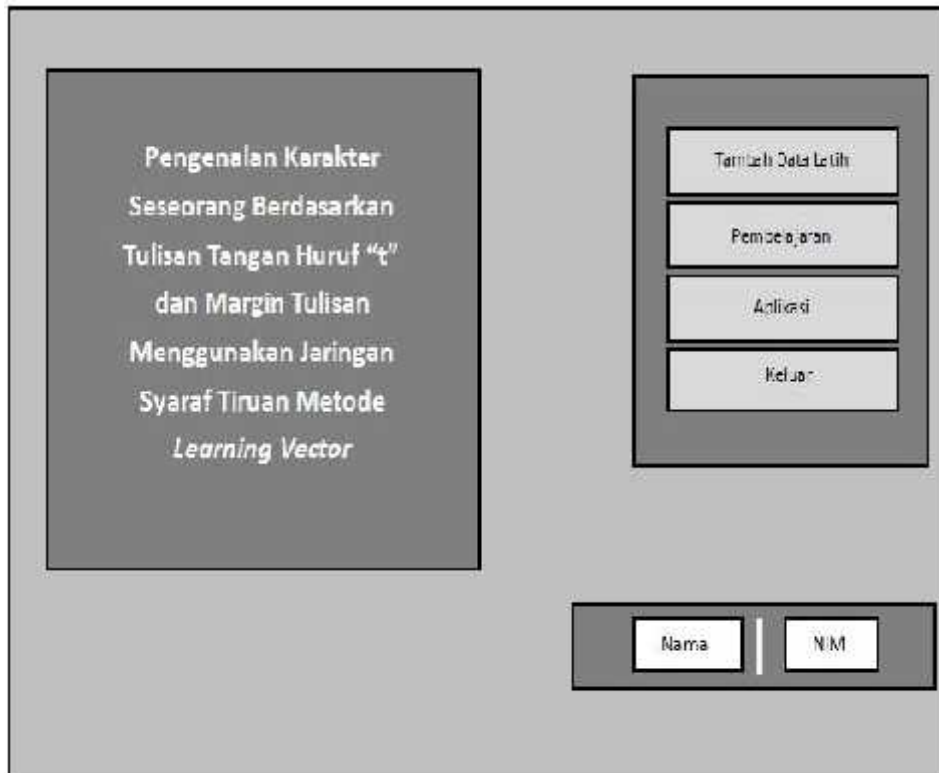
4.2.2 Perancangan Tampilan Aplikasi (*Interface*)

Perancangan tampilan (*interface*) penting dilakukan dalam membuat sebuah sistem/aplikasi. *Interface* menjadi sebuah sarana pengembangan yang digunakan untuk membuat komunikasi yang baik antara sistem/aplikasi dan pemakainya. Sebuah perancangan *interface* yang baik meliputi tampilan yang baik, konsisten dalam sisi bahasa, dan mudah dipahami.

Perancangan tampilan pada aplikasi pengenalan karakter tulisan tangan huruf t dan margin tulisan yang akan dibuat meliputi perancangan tampilan halaman utama, perancangan tampilan tambah data latih, perancangan tampilan Pembelajaran, dan perancangan tampilan pengujian aplikasi.

4.2.2.1 Perancangan Tampilan Halaman Utama

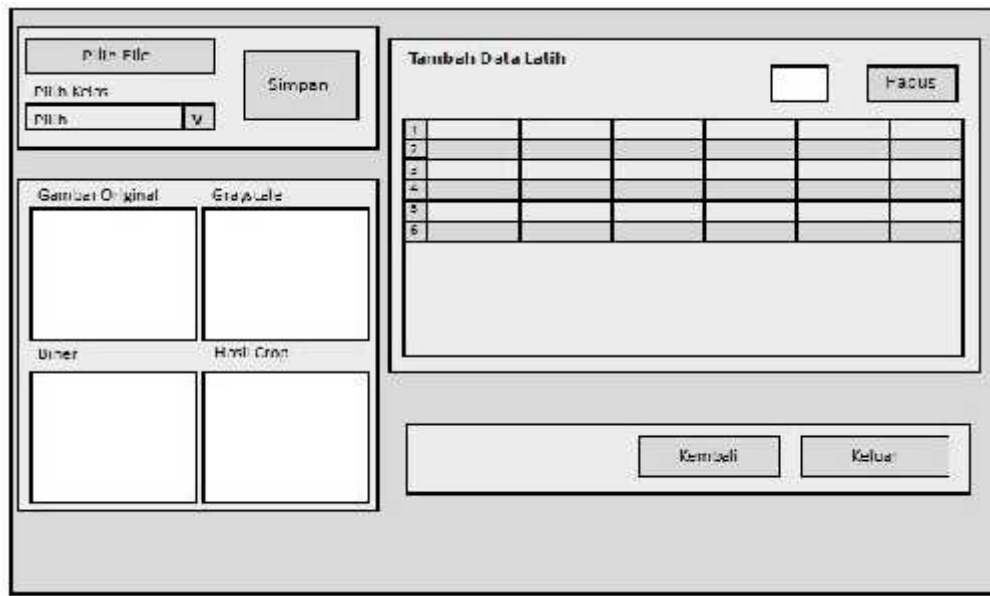
Halaman utama adalah halaman yang pertama muncul saat aplikasi dijalankan. Perancangan tampilan halaman utama meliputi *button* tambah data latih untuk menambah data latih karakter tulisan tangan huruf, *button* pembelajaran untuk melakukan pembelajaran terhadap data latih yang telah di masukkan kedalam aplikasi dan *button* Pengujian aplikasi melakukan pengujian terhadap data uji tulisan tangan huruf t dan margin tulisan dan *button* keluar untuk keluar dari aplikasi. Tampilan halaman utama dapat dilihat pada Gambar 4.9 berikut.



Gambar 4.9 Perancangan Tampilan Halaman Utama

4.2.2.2 Perancangan Tampilan Tambah Data Latih

Tampilan tambah data latih merupakan tampilan untuk menambahkan/menginputkan data latih citra tulisan tangan huruf t yang berukuran 20x20 piksel yang akan dikenali oleh aplikasi. Pada tampilan ini terdapat *button* pilih file untuk menginputkan data latih, *button* pilih kelas untuk menentukan kelas dari data latih citra tulisan huruf t yang diinputkan, *button* simpan untuk dapat menyimpan data latih citra tulisan tangan huruf t sehingga dapat dilanjutkan ke tahap berikutnya, *button* hapus untuk menghapus data latih yang sudah diinputkan, *button* kembali untuk kembali kehalaman utama dan *button* keluar untuk keluar dari aplikasi.

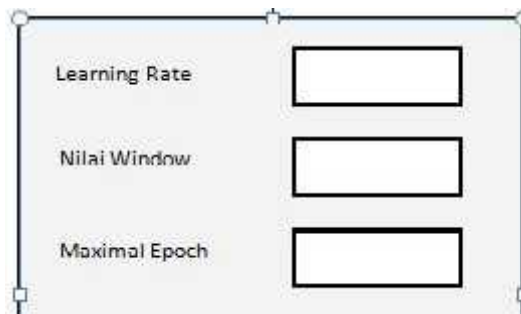


Gambar 4.10 Perancangan Tampilan Tambah Data Latih

4.2.2.3 Perancangan Tampilan Pembelajaran.

Pada perancangan tampilan pembelajaran berisikan *learnig rate* merupakan parameter yang digunakan dalam jaringan syaraf tiruan dan mempengaruhi kecepatan dimana jaringan syaraf tiruan tiba di solusi minimum, nilai windows yaitu nilai yang digunakan sebagai daerah yang harus dipenuhi untuk memperbaharui vektor referensi pemenang dan runner up jika berada dikelas yang berbeda, maximal epoch adalah jumlah epoch atau iterasi maksimum yang boleh dilakukan selama pelatihan. Iterasi akan berhenti jika nilai epoch mencapai epoch maksimum.

Perancangan tampilan pembelajaran dapat dilihat pada Gambar 4.11 berikut:



Gambar 4.11 Perancangan Tampilan Pembelajaran

4.2.2.4 Perancangan Tampilan Pengujian Citra Tulisan Tangan Huruf t dan Margin Tulisan

Pada perancangan tampilan pengujian margin tulisan dan citra tulisan tangan huruf t ini menggunakan data uji citra tulisan tangan huruf t yang telah di crop dan menggunakan angka untuk pengujian margin tulisan. Perancangan tampilan Pengujian citra tulisan tangan huruf t dan margin dapat dilihat pada Gambar 4.12 berikut:

The image shows a software interface for handwriting margin testing. It features a grid of 20 selection buttons labeled 'Pilih Huruf T1' through 'Pilih Huruf T20'. Below the grid is a 'Proses' button. At the bottom, there is a 'KARAKTER' label above a large empty text input area, and two buttons labeled 'KEMBAL' and 'MASUK'.

Gambar 4.12 Perancangan Tampilan Pengujian Margin Tulisan dan Citra Tulisan Tangan Huruf t