BAB III

ANALISIS DAN PERANCANGAN

Analisis merupakan proses penguraian konsep ke dalam bagian-bagian yang lebih sederhana, sehingga struktur logisnya menjadi jelas. Metode untuk menguji, menilai, dan memahami sistem pemikiran yang kompleks dengan memecahnya ke dalam unsur-unsur yang lebih sederhana sehingga hubungan antar unsur-unsur itu menjadi jelas.

3.1 Analisis Sistem

Wajah merupakan salah satu bagian dari manusia yang memiliki ciri berbeda untuk setiap manusia. Wajah dapat digunakan untuk mengenali seseorang, salah satunya dengan sistem pengenalan wajah misalnya untuk absensi, pendataan penduduk dan sistem pengamanan.

Artificial Intelligence merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik manusia. Contoh sistem berbasis *artificial intelligence* adalah sistem pengenalan gambar menggunakan jaringan syaraf tiruan, yaitu suatu sistem dimana sistem tersebut dapat mengenali gambar dengan menggunakan data yang sudah diperoleh dari proses pembelajaran gambar yang telah dilakukan sebelumnya.

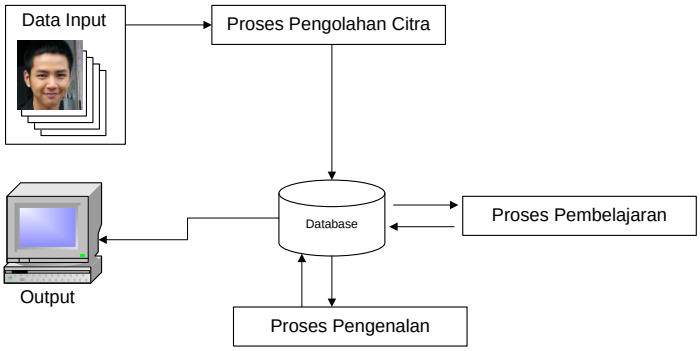
Terdapat 2 jenis metode pembelajaran pada jaringan syaraf tiruan, yaitu pembelajaran terawasi (*supervised learning*) dan pembelajaran tak terawasi (*unsupervised learning*). Pembelajaran terawasi adalah pembelajaran yang

memiliki target keluaran yang diharapkan, sedangkan pembelajaran tak terawasi tidak memiliki target keluaran. Untuk melakukan pengenalan wajah, pembelajaran terawasi lebih cocok karena menggunakan target keluaran, diantaranya yang termasuk metode pembelajaran terawasi adalah *backpropagation* dan *learning vector quantization*. *Backpropagation* merupakan metode pembelajaran yang menggunakan nilai keluaran eror untuk mengubah nilai bobotnya dalam arah mundur. *Learning vector quantization* merupakan metode pembelajaran pada lapisan kompetitif yang akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor masukan .

Metode *backpropagation* dan *learning vector quantization* memiliki kelebihan dan kekurangan masing-masing dalam proses pembelajaran untuk pengenalan gambar. *Backpropagation* menggunakan 2 alur dalam perhitungan bobot, yaitu propagasi maju (*forward*) dan propagasi mundur (*backward*). Sedangkan *learning vector quantization* hanya menggunakan 1 alur dalam perhitungannya, yaitu alur maju. Keduanya memiliki perbedaan dalam penentuan hasil pengenalan.

Sistem perbandingan metode jaringan syaraf tiruan *backpropagation* dan *learning vector quantization* dibutuhkan untuk dapat mengetahui dan merekomendasikan metode yang lebih baik dari segi akurasi dan waktu diantara kedua metode tersebut dalam pengenalan wajah. Metode *backpropagation* dan *learning vector quantization* memiliki langkah-langkah yang sama dalam sistem pengenalan wajah. Data *learning* (data yang digunakan untu proses pembelajaran) terlebih dahulu harus dipelajari untuk mendapatkan bobot yang kemudian

disimpan kedalam *database*, dan dapat digunakan pada proses pengenalan data *test* (data yang akan dikenali). Gambaran sistem secara umum dapat dilihat pada Gambar III.1.



Gambar III.1 Analisis Sistem

Gambar III.1 menggambarkan bahwa data masukan berupa 5 buah gambar wajah. Terdapat 3 buah proses utama yaitu proses pengolahan citra, proses pembelajaran dan proses pengenalan. Semua hasil proses disimpan ke dalam *database* dan hasil dari setiap proses ditampilkan pada layar komputer.

3.1.1 Analisis Metode Pembelajaran dan Pengenalan

Metode pembelajaran dan pengenalan yang dibandingkan pada sistem ini adalah metode *backpropagation* dan *learning vector quantization*. Tabel III.1 sampai Tabel III.3 merupakan perbandingan dasar dari kedua metode tersebut:

Tabel III.1 Perbandingan Metode *Backpropagation* dan LVQ

Parameter	*Backpropagation*	LVQ
Alur Pembelajaran	Menggunakan 2 alur, yaitu alur maju (*forward propagation*) dan alur mundur (*backpropagation*)	Menggunakan 1 alur yaitu alur maju.

Tabel III.2 Perbandingan Metode *Backpropagation* dan LVQ (Lanjutan)

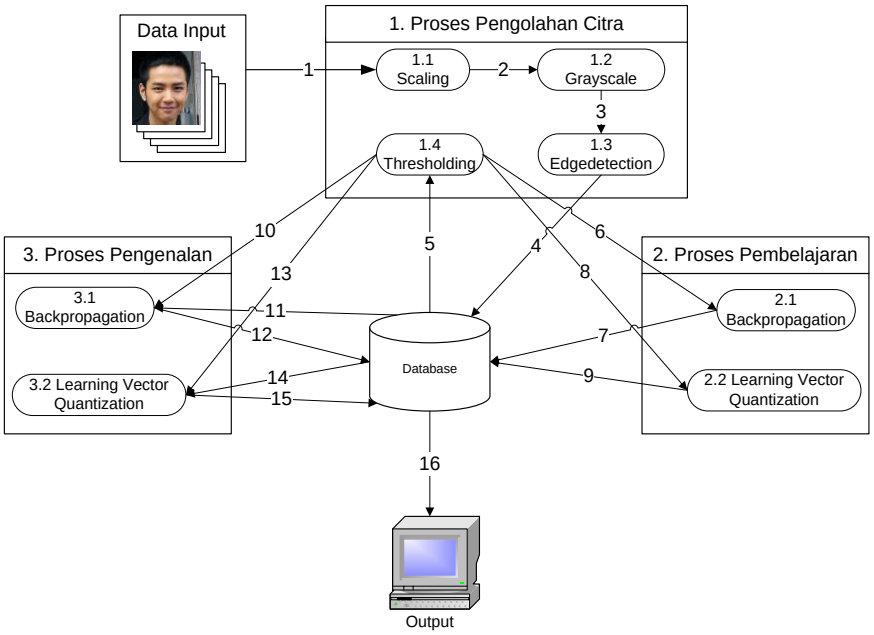
Parameter	*Backpropagation*	LVQ
Parameter Pembelajaran	Parameter pembelajaran yang digunakan adalah maksimal perulangan, rasio pembelajaran dan minimal eror.	Parameter pembelajaran yang digunakan adalah maksimal perulangan, rasio pembelajaran, pengurang rasio dan minimal eror.
Bobot Pembelajaran	Memiliki 2 nilai bobot yaitu bobot pada *hidden layer* (v) dan bobot pada *output layer* (w).	Memiliki 1 nilai bobot yaitu bobot pada keluaran (w).
	Nilai awal setiap bobot diambil dari nilai random yang cukup kecil.	Nilai awal bobot diambil dari salah satu data masukan untuk setiap kelas.
	Perubahan bobot dihitung dengan alur mundur (*backpropagation*), yaitu dengan mengubah nilai bobot pada *layer output* (w), kemudian mengubah nilai bobot pada *hidden layer* (v).	Perubahan bobot dihitung berdasarkan nilai kelas hasil perhitungan. Jika kelas hasil perhitungan sama dengan kelas target maka menggunakan rumus tambah, dan sebaliknya menggunakan rumus kurang.

Tabel III.3 Perbandingan Metode *Backpropagation* dan LVQ (Lanjutan)

Parameter	*Backpropagation*	LVQ
Perulangan Pembelajaran	Perulangan dilakukan selama perulangan belum mencapai maksimal perulangan dan nilai Mean Square Error (MSE) masih lebih kecil dari minimal eror.	Perulangan dilakukan selama perulangan belum mencapai maksimal perulangan atau nilai perubahan rasio pembelajaran masih lebih kecil dari minimal eror.
Kompleksitas Algoritma	Perhitungan lebih rumit karena menggunakan 11 rumus dalam perhitungan pembelajaran, dan 4 rumus dalam perhitungan pengenalan.	Perhitungan sederhana karena hanya menggunakan 3 rumus dalam perhitungan pembelajaran dan 1 rumus dalam perhitungan pengenalan.
Hasil Pembelajaran	Berupa nilai bobot pada *hidden layer* (v) dan bobot pada *output layer* (w).	Berupa nilai bobot (w).
Hasil Pengenalan	Berupa angka biner dari nilai kelas yang dikenali.	Berupa nilai kelas yang dikenali.

3.1.2 Analisis Proses

Pada penjelasan analisis sistem telah dijelaskan bahwa sistem ini memiliki 3 proses utama, setiap proses memiliki sub proses yang memiliki peranan dalam pengenalan wajah. Gambaran analisis proses digambarkan pada Gambar III.2.



Gambar III.2 Analisis Proses

Pada Gambar III.2 terdapat 3 proses utama, 8 sub proses serta 14 alur proses. Alur proses dari sistem ini adalah:

1. Gambar masukan sebanyak 5 buah melalui proses *scaling* untuk diubah ukurannya, sehingga ukuran setiap gambar menjadi sama.
2. Setelah melalui pross *scaling*, gambar masukan melalui proses *grayscale* untuk mengubah warna gambar menjadi keabu-abuan.
3. Setelah gambar masukan berwarna keabu-abuan, kemudian melalui proses *edgedetection* (deteksi tepi), untuk mendapatkan tepi dari setiap gambar.
4. Kemudian gambar masukan disimpan kedalam *database*, yang disimpan berupa keterangan gambar, gambar asli hasil proses *scaling* dan gambar hasil proses *edge detection*.
5. Proses ke-5 yaitu *thresholding* dilakukan ketika gambar masukan yang telah melalui proses *edgedetection* akan digunakan untuk proses

pembelajaran(proses 6 dan 8) dan pengenalan(proses 10 dan 12). Proses ini mengubah gambar hasil *edgedetection* menjadi matriks representasi gambar tersebut dengan isi nilai biner 1 atau 0.

6. Proses ini merupakan proses pembelajaran menggunakan metode *backpropagation*. Setiap matriks gambar *learning* akan dipelajari dan menghasilkan 3 jenis nilai bobot berupa matriks dan array. Kemudian dilanjutkan pada proses 7.
7. Setelah 3 nilai bobot *backpropagation* didapatkan, bobot-bobot tersebut disimpan ke dalam *database*.
8. Proses 8 merupakan proses pembelajaran menggunakan metode *learning vector quantization*. Setiap matriks gambar *learning* akan dipelajari dan menghasilkan 1 matriks nilai bobot. Kemudian dilanjutkan pada proses 9.
9. Matriks bobot hasil pembelajaran *learning vector quantization* disimpan ke dalam *database*.
10. Proses ini merupakan proses pengenalan gambar *test* menggunakan metode *backpropagation*. Gambar *test* hasil proses *thresholding* menjadi data masukan ke dalam proses ini.
11. Mengambil nilai bobot hasil pembelajaran *backpropagation* yang telah disimpan di dalam *database* untuk proses pengenalan *backpropagation*. Hasil pengenalan berupa nilai kelas gambar yang cocok. Kemudian dilanjutkan pada proses 12.
12. Data pengenalan dan hasilnya disimpan ke dalam *database*.

13. Proses 13 merupakan proses pengenalan gambar *test* menggunakan metode *learning vector quantization*. Gambar *test* hasil proses *thresholding* menjadi data masukan ke dalam proses ini.
14. Mengambil nilai bobot hasil pembelajaran *learning vector quantization* yang telah disimpan di dalam *database* untuk melakukan pengenalan *learning vector quantization*. Hasil pengenalan berupa nilai kelas gambar yang cocok. Kemudian dilanjutkan pada proses 15.
15. Data pengenalan dan hasilnya disimpan ke dalam *database*.
16. Data-data masukan, hasil pembelajaran dan hasil pengenalan ditampilkan melalui layar komputer.

Untuk lebih jelas mengenai proses-proses yang dilalui pada sistem ini, berikut ini penjelasannya:

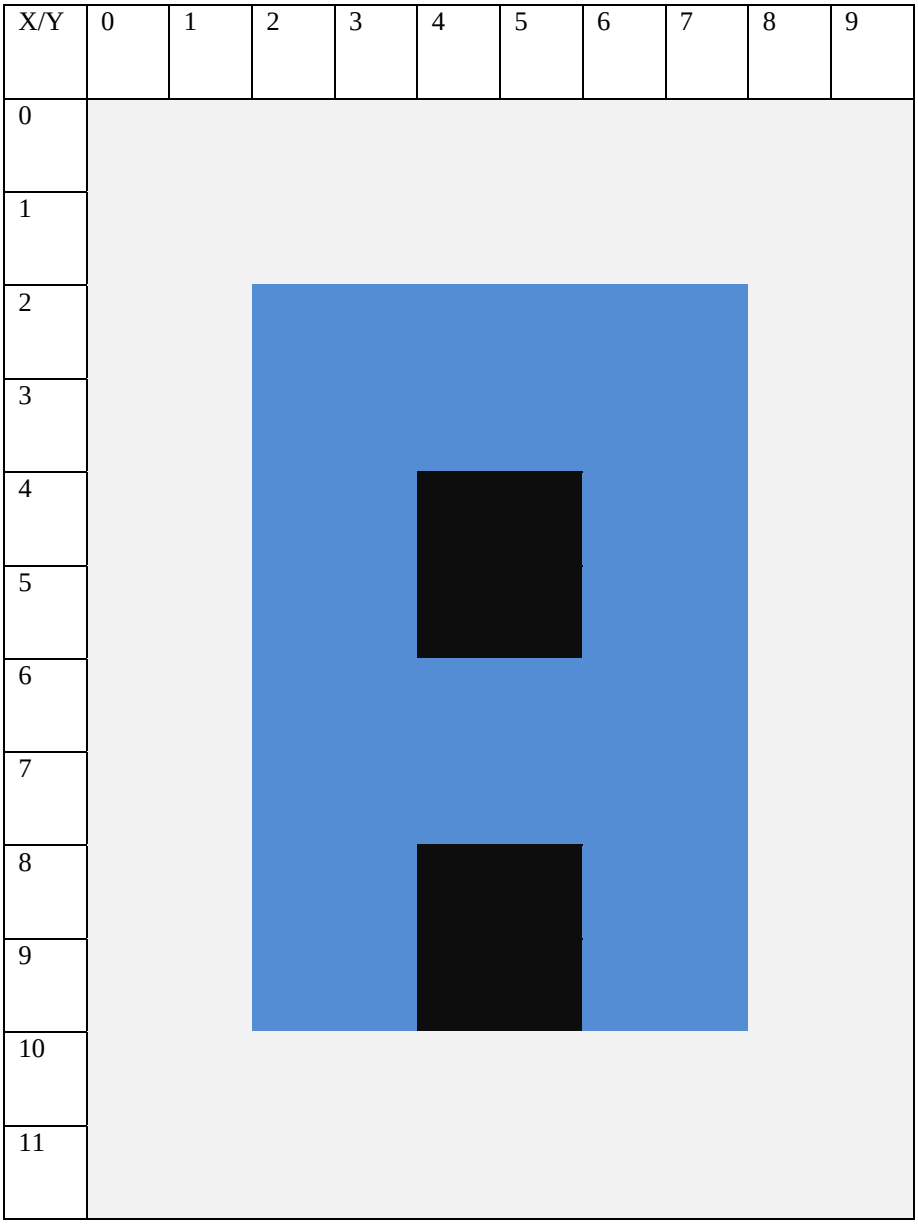
1. Proses Pengolahan Citra

   Proses pengolahan citra ini merupakan proses awal yang dijalankan ketika melakukan penambahan data. Data masukan akan melalui proses *scaling*, *grayscale* dan *edgedetection* terlebih dahulu sebelum dimasukkan kedalam *database*. Sedangkan proses *thresholding* dilakukan ketika data masukan, baik data tes maupun data *training* akan digunakan untuk proses pembelajaran atau pengenalan.

   Setiap proses pada proses pengolahan citra menggunakan nilai pixel dari citra untuk perhitungannya. Setiap pixel pada citra terdiri dari 3 unsur warna dasar yaitu *Red* (R), *Green* (G) dan *Blue* (B), yang sering disingkat RGB, dan setiap unsur memiliki nilai antara 0-255 yang menjadi kode

kedalaman warna dari masing-masing unsur (penjelasan lebih lengkap pada subbab 2.3.3 Pengenalan Citra).

Sebagai contoh perhitungan pada proses-proses pengolahan citra, digunakan gambar huruf “A” dengan ukuran lebar 10 pixel dan panjang 12 pixel seperti terlihat pada Gambar III.3.



Gambar III.3 Gambar huruf A dengan ukuran lebar 10 pixel dan panjang 12 pixel

1.1. Scaling

Scaling merupakan proses mengubah ukuran gambar digital, hal ini perlu dilakukan agar semua gambar digital data masukan memiliki ukuran yang sama. *Pseudocode* untuk proses *scaling* ditunjukkan pada Gambar III.5. Langkah-langkah proses *scaling* adalah:

a) Tentukan ukuran tujuan, dalam contoh ini lebar 5 pixel dan panjang 6 pixel.

b) Hitung skala perubahan panjang dan lebar dengan cara membagi panjang atau lebar awal dengan panjang atau lebar tujuan.

$$nwidthfactor = \frac{width}{nwidth} = \frac{10}{5} = 2$$

$$nheightfactor = \frac{height}{nheight} = \frac{12}{6} = 2$$

c) Untuk setiap matriks gambar hasil, dilakukan perhitungan d) sampai g)

d) Tentukan kordinat pixel disekitar matriks yang sedang diulang. X dan y menunjukkan kordinat yang sedang dihitung. Berikut ini perhitungan untuk kordinat 0,0.

$$fr_x = x * nwidthfactor = 0 * 2 = 0$$

$$fr_y = y * nheightfactor = 0 * 2 = 0$$

$$C_x = fr_x + 1 = 0 + 1 = 1$$

$$C_y = fr_y + 1 = 0 + 1 = 1$$

$$f_x = x * nwidthfactor - fr_x = 0 * 0 - 0 = 0$$

$$f_y = y * nheightfactor - fr_y = 0 * 0 - 0 = 0$$

$n_x = 1{,}0 - f_x = 1{,}0 - 0 = 1{,}0$

$n_y = 1{,}0 - f_y = 1{,}0 - 0 = 1{,}0$

e) Ambil nilai pixel pada 4 kordinat, yaitu:

C1 = pixel(fr_x, fr_y) = pixel(0,0)

C2 = pixel(C_x, fr_y) = pixel(1,0)

C3 = pixel(fr_x, C_y) = pixel(0,1)

C4 = pixel(C_x, C_y) = pixel(1,1)

Nilai RGB untuk pixel pada C1, C2, C3 dan C4 adalah sebagai berikut:

	Red (R)	*Green* (G)	*Blue* (B)
C1 :	242	242	242
C2 :	242	242	242
C3 :	242	242	242
C4 :	242	242	242

f) Hitung nilai RGB baru dengan menggunakan nilai RGB dari keempat kordinat yang sudah diambil.

R:

$b1 = (n_x * C1.R + f_x * C2.R) = (1{,}0 * 242 + 0 * 242) = 242$

$b2 = (n_x * C3.R + f_x * C4.R) = (1{,}0 * 242 + 0 * 242) = 242$

$R = (n_y * b1 + f_y * b2) = (1{,}0 * 242 + 0 * 242) = 242$

G:

$b1 = (n_x * C1.G + f_x * C2.G) = (1{,}0 * 242 + 0 * 242) = 242$

$b2 = (n_x * C3.G + f_x * C4.G) = (1{,}0 * 242 + 0 * 242) = 242$

$$G = (n_y * b1 + f_y * b2) = (1{,}0 * 242 + 0 * 242) = 242$$

B:

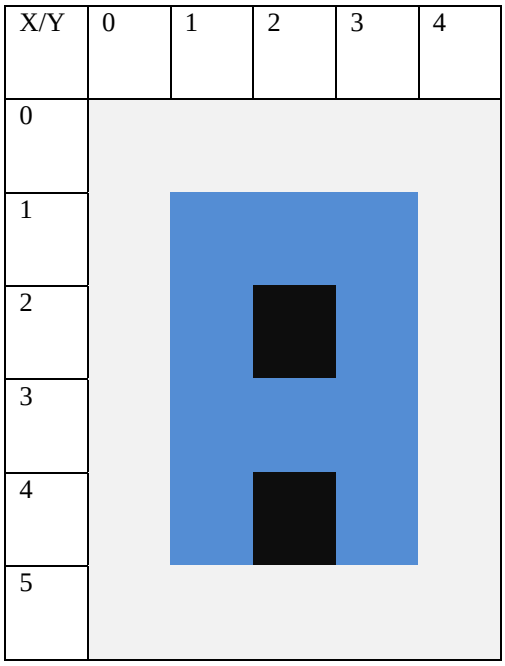
$$b1 = (n_x * C1.B + f_x * C2.B) = (1{,}0 * 242 + 0 * 242) = 242$$

$$b2 = (n_x * C3.B + f_x * C4.B) = (1{,}0 * 242 + 0 * 242) = 242$$

$$B = (n_y * b1 + f_y * b2) = (1{,}0 * 242 + 0 * 242) = 242$$

g) Set nilai pixel pada kordinat yang sedang diulang dengan nilai RGB hasil dari perhitungan. Sehingga pixel(0,0) memiliki nilai R=242, G=242 dan B=242.

Hasil akhir dari perhitungan *scaling* untuk seluruh pixel pada gambar huruf "A" dengan ukuran lebar 10 pixel dan panjang 12 pixel adalah gambar huruf "A" dengan ukuran lebar 5 pixel dan panjang 6 pixel seperti terlihat pada Gambar III.4.



Gambar III.4 Gambar huruf A dengan ukuran lebar 5 pixel dan panjang 6 pixel

Keterangan rumus:

nwidthfactor	:	Faktor rasio skala lebar	x	:	Koordinat x pixel yang sedang dibaca
nheightfactor	:	Faktor rasio skala tinggi	y	:	Koordinat y pixel yang sedang dibaca
width	:	Lebar lama	fr	:	Kordinat pixel sekitar
height	:	Tinggi lama	c	:	Nilai pixel sekitar
nwidth	:	Lebar baru	f	:	Kordinat pixel sekitar
nheight	:	Tinggi baru	n	:	Kordinat sekitar f

```
nWidthFactor ← temp.Width / newWidth
nHeightFactor ← temp.Height / newHeight
for x ← 0 to bmap.Width do
   for y ← 0 to bmap.Height do
        fr_x ← Floor(x * nWidthFactor)
      fr_y ← Floor(y * nHeightFactor)
      cx ← fr_x + 1
      if (cx >= temp.Width)then
               cx ← fr_x
       endif
      cy ← fr_y + 1
      if (cy >= temp.Height) then
               cy ← fr_y
       endif
      fx ← x * nWidthFactor - fr_x
      fy ← y * nHeightFactor - fr_y
      nx ← 1.0 - fx
       ny ← 1.0 - fy
       color1 ← temp.Pixel(fr_x, fr_y)
      color2 ← temp.Pixel(cx, fr_y)
      color3 ← temp.Pixel(fr_x, cy)
      color4 ← temp.Pixel(cx, cy)
      bp1 ← (byte)(nx * color1.B + fx * color2.B)
       bp2 ← (byte)(nx * color3.B + fx * color4.B)
       nBlue ← (byte)(ny * (double)(bp1) + fy * (double)(bp2))
      bp1 ← (byte)(nx * color1.G + fx * color2.G)
      bp2 ← (byte)(nx * color3.G + fx * color4.G)
      nGreen ← (byte)(ny * (double)(bp1) + fy * (double)(bp2))
      bp1 ← (byte)(nx * color1.R + fx * color2.R)
      bp2 ← (byte)(nx * color3.R + fx * color4.R)
      nRed ← (byte)(ny * (double)(bp1) + fy * (double)(bp2))
       bmap.SetPixel(x y,FromArgb(255, nRed, nGreen, nBlue))
   endfor
endfor
```

Gambar III.5 *Pseudocode* Proses *Scaling*

1.2. *Grayscale*

Grayscale merupakan proses untuk mengubah warna gambar menjadi keabu-abuan. Gambar digital yang sudah melalui proses *scaling* (Gambar III.4) menjadi masukan dalam proses ini. Proses *grayscale* dilakukan dengan mengubah nilai RGB setiap pixel gambar menjadi 1 nilai yang sama. Sehingga setiap pixel memiliki nilai yang sama untuk ketiga unsur warnanya. *Pseudocode* untuk proses *grayscale* ditunjukkan pada Gambar III.8. Berikut ini langkah-langkah proses *grayscale* dengan contoh gambar huruf "A" dengan ukuran lebar 5 pixel dan panjang 6 pixel yang ditunjukan pada Gambar III.4.

Nilai RGB untuk setiap pixel pada gambar huruf "A" yang akan diubah menjadi keabu-abuan ditunjukkan pada Gambar III.6.

X/Y	0	1	2	3	4
0	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)
1	(242, 242, 242)	(84, 141, 212)	(84, 141, 212)	(84, 141, 212)	(242, 242, 242)
2	(242, 242, 242)	(84, 141, 212)	(13, 13, 13)	(84, 141, 212)	(242, 242, 242)
3	(242, 242, 242)	(84, 141, 212)	(84, 141, 212)	(84, 141, 212	(242, 242, 242)
4	(242, 242, 242)	(84, 141, 212	(13, 13, 13)	(84, 141, 212	(242, 242, 242)
5	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)

Gambar III.6 Nilai Pixel Gambar Huruf "A"

a) Setiap pixel dihitung nilai *Grayscale* nya dengan rumus sebagai berikut:

$$Grayscale = \frac{R + G + B}{3}$$

$$Grayscale(0,0) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(0,1) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(0,2) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(0,3) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(0,4) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(1,0) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(1,1) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(1,2) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(1,3) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(1,4) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(2,0) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(2,1) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(2,2) = \frac{(13 + 13 + 13)}{3} = 13$$

$$Grayscale(2,3) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(2,4) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(3,0) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(3,1) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(3,2) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(3,3) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(3,4) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(4,0) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(4,1) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(4,2) = \frac{(13 + 13 + 13)}{3} = 13$$

$$Grayscale(4,3) = \frac{(84 + 141 + 212)}{3} = 145{,}67 = 146$$

$$Grayscale(4,4) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(5,0) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(5,1) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(5,2) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(5,3) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

$$Grayscale(5,4) = \frac{(242 + 242 + 242)}{3} = 242 = 242$$

b) Nilai RGB setiap pixel diset ulang dengan nilai *grayscale* yang dihasilkan pada perhitungan langkah a). Hasilnya ditunjukkan pada Gambar III.7.

X/Y	0	1	2	3	4
0	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)
1	(242, 242, 242)	(146, 146, 146)	(146, 146, 146)	(146, 146, 146)	(242, 242, 242)
2	(242, 242, 242)	(146, 146, 146)	(13, 13, 13)	(146, 146, 146)	(242, 242, 242)
3	(242, 242, 242)	(146, 146, 146)	(146, 146, 146)	(146, 146, 146)	(242, 242, 242)
4	(242, 242, 242)	(146, 146, 146)	(13, 13, 13)	(146, 146, 146)	(242, 242, 242)
5	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)

Gambar III.7 Nilai Pixel Hasil Proses *Grayscale*

```
for i ← 0 to width do
    for j ← 0 to height do
        pixelgray ← (gambar[i][j].R + gambar[i][j].G + gambar[i][j].B) / 3
          gambar[i][j].R ← pixelgray
          gambar[i][j].G ← pixelgray
          gambar[i][j].B ← pixelgray
    endfor
endfor
```

Gambar III.8 *Pseudocode* Proses *Grayscale*

1.3. *Edgedetection*

Gambar digital yang sudah berwarna keabu-abuan (Gambar III.7) menjadi data masukan pada proses ini. Proses *edgedetection* ini dijalankan dengan menggunakan metode sobel. Metode sobel menggunakan 2 matriks berukuran 3x3 yaitu matriks Gx dan Gy. Kedua matriks tersebut digunakan untuk menghitung perbedaan warna pada pixel yang sedang dihitung dengan pixel disekitarnya secara horizontal dan vertikal. Matriks Gx digunakan untuk menghitung secara horizontal dan matriks Gy secara vertikal. Matriks Gx dan Gy pada metode sobel ditunjukkan pada Gambar III.9.

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Gambar III.9 Matriks Metode Sobel

Pada perhitungan deteksi tepi ini, semua pixel dihitung kecuali pixel yang berada pada tepi gambar, yaitu pada contoh ini yang memiliki nilai x=0, x=5, y=0 dan y=4. Hal ini dilakukan karena di dalam perhitungan nilai tepi sebuah pixel, harus menggunakan nilai pixel dari pixel-pixel disekitar pixel yang sedang dihitung, sedangkan pixel pada tepi gambar tidak memiliki beberapa pixel disekitarnya yang dibutuhkan dalam perhitungan. *Pseudocode* untuk proses deteksi tepi dengan metode sobel ditunjukkan pada Gambar III.11.

Berikut langkah perhitungan untuk setiap pixelnya:

a) Perhitungan pada pixel (1,1); nilai kordinat i=1, j=1. Dilakukan perulangan perhitungan terhadap sisi x dan y. Masing-masing sisi x dan y dimulai dari -1 hingga 1. Perulangan pada sisi x diwakili oleh variabel wi dan pada sumbu y diwakili oleh variabel hw. Pada setiap perulangan dilakukan perhitungan sebagai berikut:

$C = (\,p(j+wi, i+hw).R \;+\; p(j+wi, i+hw).G \;+$

$p(j+wi, i+hw).B)/3$

$Xbaru \mathrel{+}= gx(hw+1, wi+1) * C$

$Ybaru \mathrel{+}= gy(hw+1, wi+1) * C$

Berikut ini hasil perhitungan pada pixel(1,1) :

1) Perulangan 1

Hw= -1 ; wi= -1;

$$C = \frac{p(0,0).R \;+\; p(0,0).G \;+\; p(0,0).B}{3} = 242$$

$Xbaru \mathrel{+}= gx(0,0) * C = -242$

$Ybaru \mathrel{+}= gy(0,0) * C = 242$

2) Perulangan 2

Hw= -1 ; wi= 0;

$$C = \frac{p(1,0).R \;+\; p(1,0).G \;+\; p(1,0).B}{3} = 242$$

$Xbaru \mathrel{+}= gx(0,1) * C = -242$

$Ybaru \mathrel{+}= gy(0,1) * C = 726$

3) Perulangan 3

Hw= -1 ; wi= 1;

$$C = \frac{p(2,0).R \ + \ p(0,0).G \ + \ p(0,0).B}{3} = 242$$

$$Xbaru \mathrel{+}= gx(0,2) * C = 0$$

$$Ybaru \mathrel{+}= gy(0,2) * C = 968$$

4) Perulangan 4

Hw= 0; wi= -1;

$$C = \frac{p(0,1).R \ + \ p(0,1).G \ + \ p(0,1).B}{3} = 242$$

$$Xbaru \mathrel{+}= gx(1,0) * C = -484$$

$$Ybaru \mathrel{+}= gy(1,0) * C = 968$$

5) Perulangan 5

Hw= 0 ; wi= 0;

$$C = \frac{p(1,1).R \ + \ p(1,1).G \ + \ p(1,1).B}{3} = 146$$

$$Xbaru \mathrel{+}= gx(1,1) * C = -484$$

$$Ybaru \mathrel{+}= gy(1,1) * C = 968$$

6) Perulangan 6

Hw= 0 ; wi= 1;

$$C = \frac{p(2,1).R \ + \ p(2,1).G \ + \ p(2,1).B}{3} = 146$$

$$Xbaru \mathrel{+}= gx(1,2) * C = -192$$

$$Ybaru \mathrel{+}= gy(1,2) * C = 968$$

7) Perulangan 7

Hw= 1 ; wi= -1;

$$C = \frac{p(0,2).R \ + \ p(0,2).G \ + \ p(0,2).B}{3} = 242$$

$$Xbaru \mathrel{+}= gx(2,0) * C = -434$$

$$Ybaru \mathrel{+}= gy(2,0) * C = 726$$

8) Perulangan 8

Hw= 1 ; wi= 0;

$$C = \frac{p(1,2).R \ + \ p(1,2).G \ + \ p(1,2).B}{3} = 146$$

$$Xbaru \mathrel{+}= gx(2,1) * C = -434$$

$$Ybaru \mathrel{+}= gy(2,1) * C = 434$$

9) Perulangan 9

Hw= 1 ; wi= 1;

$$C = \frac{p(2,2).R \ + \ p(2,2).G \ + \ p(2,2).B}{3} = 13$$

$$Xbaru \mathrel{+}= gx(2,2) * C = -421$$

$$Ybaru \mathrel{+}= gy(2,2) * C = 421$$

Hasil akhir perhitungan Xbaru = -421 dan Ybaru = 421.

b) Jika $Xbaru^2 + Ybaru^2$ lebih besar dari 128^2, maka nilai RGB pixel pada titik (1,1) diubah menjadi 0, sedangkan jika hasilnya lebih kecil dari 128^2 maka nilai RGB pixel(1,1) diubah menjadi 255.

c) Langkah a) dan b) diulang terhadap semua pixel kecuali pixel tepi gambar. Gambar III.10 merupakan hasil pada perhitungan seluruh pixel.

X/Y	0	1	2	3	4
0	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)
1	(242, 242, 242)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(242, 242, 242)
2	(242, 242, 242)	(0, 0, 0)	(255, 255, 255)	(0, 0, 0)	(242, 242, 242)
3	(242, 242, 242)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(242, 242, 242)
4	(242, 242, 242)	(0, 0, 0)	(255, 255, 255)	(0, 0, 0)	(242, 242, 242)
5	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)	(242, 242, 242)

Gambar III.10 Gambar huruf "A" hasil proses *Edgedetection*

```
gx ←{ { -1, 0, 1 }, { -2, 0, 2 }, { -1, 0, 1 } }
gy ←{ { 1, 2, 1 }, { 0, 0, 0 }, { -1, -2, -1 } }
for i ←1 to Height - 1 do
     for j ←1 to Width - 1 do
         new_x ←0
         new_y ←0
         for hw ←-1 to 1 do
             for wi ←-1 to 1  do
                c ←(b[j + wi][i + hw].B + b[j + wi][i + hw].R + b[j +
wi][i + hw].G) / 3
                new_x ←new_x  + gx[hw + 1, wi + 1] * c
                new_y ←new_y  + gy[hw + 1, wi + 1] * c
             endfor
         endfor
          if (new_x * new_x + new_y * new_y > 128 * 128) then
                b1[j][i] ←0
          else
                 b1[j][i] ←255
          endif
    endfor
endfor
```

Gambar III.11 *Pseudocode* Proses Deteksi Tepi Metode Sobel

Setelah proses *edgedetection* selesai terhadap semua gambar masukan, gambar-gambar tersebut disimpan ke direktori dan keterangannya disimpan ke dalam *database*.

1.4. *Thresholding*

Thresholding merupakan proses untuk merepresentasikan gambar digital hasil *edgedetection* kedalam bentuk matriks (array 2 dimensi) dengan isi 1 atau 0. Data masukan dalam proses ini merupakan gambar digital hasil proses *edgedetection* yang diambil dari direktori. Gambar III.14 menunjukkan *pseudocode* proses *thresholding*.

Berikut ini langkah-langkah proses *thresholding*:

a) Tentukan ambang batas nilai RGB. Dalam hal ini diambil batas 128.
b) Baca setiap pixel gambar, jika nilai rata-rata RGBnya lebih besar dari 128, maka matriks pada kordinat yang sama dengan pixel yang sedang dibaca diisi dengan nilai 0, dan sebaliknya, jika nilainya lebih kecil atau sama dengan 128, matriksnya diisi dengan 0. Gambar III.12 merupakan matriks hasil proses *thresholding* dari gambar huruf A yang telah melalui proses *edgedetection* (Gambar II.10).

X/Y	0	1	2	3	4
0	0	0	0	0	0
1	0	1	1	1	0
2	0	1	0	1	0
3	0	1	1	1	0
4	0	1	0	1	0
5	0	0	0	0	0

Gambar III.12 Matriks Gambar Huruf A Setelah Proses *Thresholding*

c) Kemudian matriks (array 2 dimensi, 5 kolom dan 6 baris) hasil langkah b) diubah menjadi array 1 dimensi dengan 1 kolom dan 30 baris. Hasilnya ditunjukkan pada Gambar III.13.

X/Y	0
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1
8	1
9	0
10	0
11	1
12	0
13	1
14	0
15	0
16	1
17	1
18	1
19	0
20	0
21	1
22	0
23	1
24	0
25	0
26	0
27	0
28	0
29	0

Gambar III.13 Array Gambar Hasil *Thresholding*

```
for i ← 0 to width do
     for j ← 0 to height do
           c ← (b[i][j].R + b[i][j].G + b[i][j].B) / 3
           if (c > 128) then
                hasil[z] ← 0
                z←z+1
          else
                 hasil[z] ← 1
                z←z+1
           endif
     endfor
endfor
```

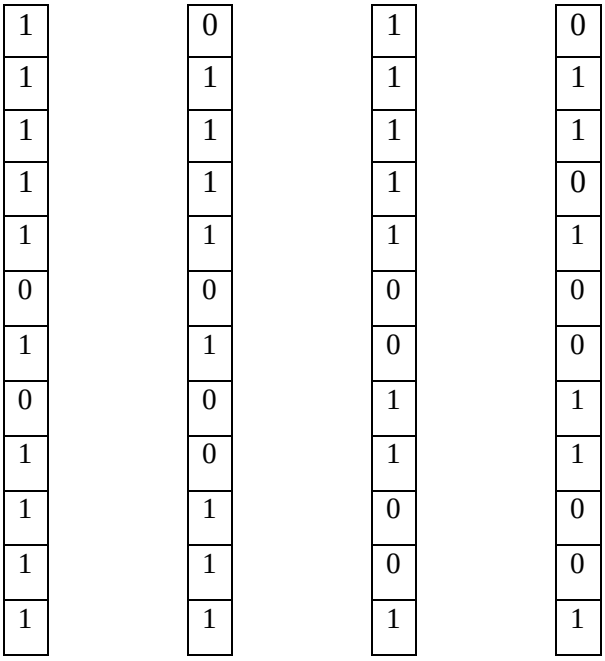
Gambar III.14 *Pseudocode* Proses *Thresholding*

Array 1 dimensi hasil proses *thresholding* digunakan oleh proses pembelajaraan dan pengenalan.

2. Proses Pembelajaran

Proses pembelajaran merupakan proses dimana komputer belajar dari sejumlah data yang ada dan menghasilkan nilai bobot. Bobot merupakan nilai yang menujukkan seberapa besar koneksi antara neuron. Nilai bobot akan bertambah jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan, dan sebaliknya. Proses pembelajaran pada sistem ini dilakukan dengan menggunakan 2 buah metode, yaitu *backpropagation* dan *learning vector quantization*.

Data *training* yang digunakan sebagai contoh pada proses pembelajaran adalah array 1 dimensi dari gambar huruf “A” hasil proses *thresholding* sebelumnya (a) ditambah dengan 1 buah array 1 dimensi dari gambar huruf “A” (b) serta 2 buah array 1 dimensi dari gambar huruf “C” ((c) dan (d)) dengan struktur yang berbeda yang telah melalui proses pengolahan citra seperti data (a). Data *training* ditunjukkan pada Gambar III.15.



Gambar III.15 Empat Buah Array 1 Dimensi Data *Training*

2.1. Pembelajaran *Backpropagation*

Pembelajaran dengan metode *backpropagation* menggunakan 2 alur perhitungan, yaitu alur maju dan alur mundur. Gambar III.16 merupakan *Pseudocode* proses pembelajaran *backpropagation*.

Berikut perhitungan pembelajaran LVQ terhadap data *training* tersebut:

a) Keempat array dijadikan data masukan dengan target sesuai kelas dari huruf tersebut. Hasilnya seperti terlihat pada Tabel III.4 dan Tabel III.5.

Tabel III.4 Data Masukan Proses Pembelajaran *Backpropagation*

Huruf	Target	Vektor	
A	1	1 1 1 1 0 1 1 1 1 1 0 1	Masukan (x1)
C	2	1 1 1 1 0 0 1 0 0 1 1 1	Masukan (x2)

Tabel III.5 Data Masukan Proses Pembelajaran *Backpropagation* (Lanjutan)

Huruf	Target	Vektor	
A	1	0 1 0 1 0 1 1 1 1 1 0 1	Masukan (x3)
C	2	0 1 1 1 0 0 1 0 0 0 1 1	Masukan (x4)

b) Ubah nilai target kedalam bentuk angka biner seperti pada Tabel III.6.

Tabel III.6 Biner Target

Huruf	Target	Biner
A	1	0 1
C	2	1 0
A	1	0 1
C	2	1 0

c) Tentukan nilai maksimum perulangan, eror minimum, dan rasio pembelajaran. Pada contoh perhitungan ini digunakan batasan sebagai berikut:

Maksimum perulangan = 100

Eror minimum = 0,01

Rasio pembelajaran = 0,05

d) Inisialisasikan bobot awal dengan nilai acak antara 0 dan 1.

i. Bobot awal masukan ke *hidden* (v)

v={{0.4102,0.9779,0.5471,0.8970,0.5512,0.3478,0.5682,0.
9118,0.8610,0.5925,0.6030,0.6290},
{0.8696,0.2377,0.2268,0.3614,0.0049,0.3730,0.5506,0.
1408,0.5958,0.2685,0.4945,0.5134},
{0.5791,0.8777,0.4060,0.3135,0.3189,0.0049,0.6812,0.
4258,0.4435,0.4208,0.5435,0.0731},
{0.1048,0.7064,0.6371,0.6615,0.5699,0.1068,0.5909,0.
7129,0.5468,0.1753,0.2096,0.9170}}

ii. Bobot awal bias ke *hidden* (v0)

v0={0,0,0,0}

Semua nilai v0 dianggap 0;

iii. Bobot awal *hidden* ke keluaran (w)

w={{0.9413,0.2376},{0.2289,0.8795},{0.8252,0.1634},{0.
2756,0.7456}}

iv. Bobot awal bias ke keluaran (w0)

w0=0

Nilai w0 dianggap 0.

e) Perulangan dilakukan selama nilai perulangan lebih kecil dari maksimal perulangan dan nilai kuadrat eror lebih besar dari nilai eror minimum. Setiap data dilakukan perulangan dilakukan perhitungan seperti pada langkah e) sampai g)

f) Alur maju:

Data = x1

Perulangan = 1

Operasi pada *hidden layer* :

$$z_in_j = v0_i + \sum_{i=1}^{n} x_i * v_{i,j}$$

$$z_j = \frac{1}{1 + e^{-z_in_j}}$$

$$z_in_1 = 0 + (1 * 0.4102) + (1 * 0.9779) + (1 * 0.5471) + (1 * 0.8970) + (0 * 0.5512) + (1 * 0.3478) + (1 * 0.5682) + (1 * 0.9118) + (1 * 0.8610) + (1 * 0.5925) + (0 * 0.6030) + (1 * 0.6290) = 6.8971$$

$$z_in_2 = 4.6297$$

$$z_in_3 = 4.6738$$

$$z_in_4 = 5.9928$$

$$z_1 = \frac{1}{1 + e^{-6.8971}} = 0.9990$$

$$z_2 = \frac{1}{1 + e^{-4.6297}} = 0.9903$$

$$z_3 = \frac{1}{1 + e^{-4.6738}} = 0.9907$$

$$z_4 = \frac{1}{1 + e^{-5.9928}} = 0.9975$$

Operasi pada *output layer* :

$$y_in_k = w0 + \sum_{j=1}^{n} w_{jk} * z_j$$

$$y_k = \frac{1}{1 + e^{-y_in_k}}$$

$$y_in_k = 0 + (0.9413 * 0.9990) + (0.2289 * 0.9903) +$$

$$(0.8252 * 0.9907) + (0.2756 * 0.9975) = 3.1166$$

$$y_k = \frac{1}{1 + e^{-3.1166}} = 0.9576$$

Hitung nilai eror :

$$eror = T_{ik} - y_k = 1 - 0.9576 = 0.0424$$

g) Alur mundur

Hitung faktor unit kesalahan (δ) :

$$\delta_k = (T_{ik} - y_k) * \left(\frac{1}{1 + e^{-y_in}}\right) * \left[1 - \left(\frac{1}{1 + e^{-y_in}}\right)\right]$$

$$\delta_k = (1 - 0.9576) * \left(\frac{1}{1+e^{-3.1166}}\right) * \left[1 - \left(\frac{1}{1+e^{-3.1166}}\right)\right] =$$

$$0.0017$$

Hitung suku perubahan bobot w (Δw_i) :

$$\Delta w_{jk} = \alpha * \delta_k * z_j$$

α = rasio

$$\delta = faktor\ unit\ kesalahan$$

$$\Delta w_{10} = 0.05 * 0.0017 * 0.9990 = 0.0001$$

$$\Delta w_{20} = 0.05 * 0.0017 * 0.9903 = 0.0001$$

$$\Delta w_{30} = 0.05 * 0.0017 * 0.9907 = 0.0001$$

$$\Delta w_{40} = 0.05 * 0.0017 * 0.9975 = 0.0001$$

Hitung faktor unit kesalahan pada *layer* tersembunyi :

$$\delta_in_j = \delta * w_{jk}$$

$$\delta_in_1 = 0.0017 * 0.9413 = 0.0016$$

$$\delta_in_2 = 0.0017 * 0.2289 = 0.0004$$

$\delta_in_3 = 0.0017 * 0.8252$=0.0014

$\delta_in_4 = 0.0017 * 0.2756$ =0.0005

Hitung faktor kesalahan unit tersembunyi :

$$\delta_i = \delta_in_i * \left(\frac{1}{1+e^{-z_j}}\right) * \left[1 - \left(\frac{1}{1+e^{-z_j}}\right)\right]$$

$$\delta_1 = 0.0016 * \left(\frac{1}{1+e^{-0.9990}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.9990}}\right)\right] = 0.0003$$

$$\delta_2 = 0.0004 * \left(\frac{1}{1+e^{-0.9903}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.9903}}\right)\right] = 0.0001$$

$$\delta_3 = 0.0014 * \left(\frac{1}{1+e^{-0.9907}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.9907}}\right)\right] = 0.0003$$

$$\delta_4 = 0.0005 * \left(\frac{1}{1+e^{-0.9975}}\right) * \left[1 - \left(\frac{1}{1+e^{-0.9975}}\right)\right] = 0.0001$$

Hitung suku perubahan bobot v :

$$\Delta v_{i,j} = \alpha * \delta_j * x_{i,j}$$

$$\Delta v_{1,1} = 0.05 * 0.0017 * 1 = 0.0000$$

Δv= {{0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000},
{0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000},
{0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000},
{0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000,
0.0000, 0.0000, 0.0000, 0.0000, 0.0000}}

h) Perubahan bobot

Hitung bobot w baru :

$w_{ik} = w_{ik} + \Delta w_{ik}$

$w_{10} = 0.9413 + 0.0001 = 0.9414$

$w_{20} = 0.2289 + 0.0001$=0.2290

$w_{30} = 0.8252 + 0.0001$=0.8253

$w_{40} = 0.2756 + 0.0001 = 0.2757$

Hitung bobot v baru :

$v_{i,j} = v_{i,j} + \Delta v_{i,j}$

v={{0.4102,0.9779,0.5471,0.8970,0.5512,0.3478,0.5682,0.911
8,0.8610,0.5925,0.6030,0.6290},
{0.8696,0.2377,0.2268,0.3614,0.0049,0.3730,0.5506,0.14
08,0.5958,0.2685,0.4945,0.5134},
{0.5791,0.8777,0.4060,0.3135,0.3189,0.0049,0.6812,0.42
58,0.4435,0.4208,0.5435,0.0731},
{0.1048,0.7064,0.6371,0.6615,0.5699,0.1068,0.5909,0.71
29,0.5468,0.1753,0.2096,0.9170}}

i) Perhitungan dilakukan pada setiap perulangan hingga kondisi berhenti bernilai benar, kemudian berlanjut pada data ke-2 (x). Berikut ini merupakan bobot akhir untuk data 1 setelah perulangan selesai:

v={{0.4122,0.9819,0.5511,0.9010,0.5512,0.3480,0.5722,0.9120,0.
8612,0.5946,0.6069,0.6330},
{0.8701,0.2387,0.2278,0.3624,0.0049,0.3730,0.5516,0.1408,
0.5958,0.2690,0.4955,0.5144},

{0.5809,0.8813,0.4095,0.3171,0.3189,0.0050,0.6848,0.4259,
0.4436,0.4226,0.5469,0.0767},
{0.1054,0.7076,0.6383,0.6627,0.5699,0.1068,0.5921,0.7129,
0.5468,0.1759,0.2108,0.9182}}

w={0.9627,0.2497,0.8463,0.2970}

```
while ((epoch <= ME) and (MSE > eps)) do
        epoch←epoch+1
        for baris ←0 to jmlGambar do
               for j ←0 to 4 do
                    Z_net[j] ←0
                    Z[j] ←0
                    for i ←0 to maxI do
                           Z_net[j] ←Z_net[j] + (P[i][baris] * V[i][j])
                    endfor
                    Z[j] ←1 / (1 + Pow(e, -Z_net[j]))
               endfor
               MS[baris]←0

         for k ←0 to maxK do
                     Y_net[k] ←0
                     for j ←0 to 4 do
                           Y_net[k] ←Y_net[k] +(Z[j] * W[k][j])
                     endfor
                     Y[k] ←1 / (1 + Pow(e, -Y_net[k]))
                     error[k] ←Pow(T[k,baris] - Y[k],2)
                     MS[baris] ←MS[baris] + error[k]
               endfor
               MS[baris] ←MS[baris] / 7
               MSE +←MS[baris]
              for  k = 0 to maxK do
                    DK[k] = (T[k][baris] - Y[k]) * Y[k] * (1 - Y[k])
              endfor
         for k ←0 to maxK do
                 for j ←0 to 4 do
                          dW[k][j] ←rate * DK[k] * Z[j]
                 endfor
         endfor
         for j ←0 to 4 do
                 DK_net[j] ←0
                 for k ←0 to maxK do
                       DK_net[j] ←DK_net[j] +(DK[k]*W[k][j])
                 endfor
```

```
          endfor
         for j ←0 to 4 do
                 DS[j] ←DK_net[j] * Z[j] * (1 - Z[j])
         endfor
         for j ←0 to 4 do
                for i ←0 to maxI do
                            dV[i,j] ←rate * DS[j] * P[i,baris]
                endfor
          endfor
           for j ←0 to 4 do
                  for k ←0 to maxK do
                            W[k,j] ←W[k,j] +dW[k,j]
                  endfor
           endfor

               for j←0 to 4 do
                  for i ←0 tomaxI do
                            V[i,j] ←V[i,j] +dV[i,j]
                   endfor
           endfor
                MSE ←MSE / jmlGambar
endwhile
```

Gambar III.16 *Pseudocode* Proses Pembelajaran *Backpropagation*

2.2. Pembelajaran *Learning vector quantization*

Proses pembelajaran dengan metode *learning vector quantization* menggunakan 1 alur perhitungan yaitu alur maju. Pada contoh perhitungan ini terdapat 2 buah kelas, yaitu kelas 1 untuk huruf "A" dan kelas 2 untuk huruf "C". *Pseudocode* proses pembelajaran lvq ditunjukkan pada Gambar III.17.

Berikut perhitungan pembelajaran LVQ terhadap data *training* tersebut:

a) Setiap kelas (kelas 1 dan kelas 2) diambil salah satu array vektornya dan dijadikan sebagai inisialisasi bobot (W). Sedangkan array lainnya menjadi data masukan (X) Hasilnya terlihat pada Tabel III.7.

Tabel III.7 Inisialisasi Bobot dan Data Masukan Proses Pembelajaran LVQ

Huruf	Kelas	Vektor	
A	1	1 1 1 1 0 1 1 1 1 1 0 1	Bobot (w1)
C	2	1 1 1 1 0 0 1 0 0 1 1 1	Bobot (w2)
A	1	0 1 0 1 0 1 1 1 1 1 0 1	Masukan (x1)
C	2	0 1 1 1 0 0 1 0 0 0 1 1	Masukan (x1)

b) Tentukan nilai maksimum perulangan, eror minimum, rasio pembelajaran dan pengurangan rasio. Pada contoh ini digunakan batasan sebagai berikut:

Maksimum perulangan = 5

Eror minimum = 0,01

Rasio pembelajaran = 0,05

Pengurangan rasio = 0,1

c) Perhitungan d) sampai f) dilakukan selama perulangan masih lebih kecil dari maksimum perulangan atau rasio pembelajaran masih lebih besar dari eror minimum.

d) Untuk setiap data masukan (x) dihitung terhadang setiap data bobot (w) dengan rumus:

$$C_j = \sqrt{\sum_{i=1}^{n} \left(x_i - w_{ji}\right)^2}$$

Perulangan = 1

Data ke-1 (x1)

Jarak pada:

1) Bobot ke-1 (w1)

$C_1 = \sqrt{((0-1)^2 + (1-1)^2 + (0-1)^2 + (1-1)^2 + (0-0)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (0-0)^2 + (1-1)^2)} = 1{,}41$

2) Bobot ke-2 (w2)

$C_2 = \sqrt{((0-1)^2 + (1-1)^2 + (0-1)^2 + (1-1)^2 + (0-0)^2 + (1-0)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + (1-1)^2 + (0-1)^2 + (1-1)^2)} = 2{,}45$

e) Dilihat jarak (C) terkecil yang diperoleh pada perhitungan terhadap x1. Kemudian diperiksa apakah target x1 sesuai dengan kelas bobot dengan jarak terkecil. Hasil dari perhitungan di atas menunjukan bahwa jarak terkecil ada pada bobot ke-1. Target x1 adalah kelas 1 dan kelas bobot ke-1 juga kelas 1, hal ini berarti target sesuai dengan hasil perhitungan.

f) Dilakukan perhitungan perubahan bobot terhadap bobot dengan jarak terecil yaitu bobot ke-1 dengan rumus:

Jika kelas target sesuai kelas bobot:

$$Wj(baru) = Wj(lama) + \alpha [x - Wj(lama)]$$

Jika kelas target tidak sesuai kelas bobot:

$$Wj(baru) = Wj(lama) - \alpha [x - Wj(lama)]$$

α = rasio pembelajaran

Dalam contoh perhitungan ini, hasil dari perhitungan menunjukan bahwa kelas target sesuai dengan kelas bobot dengan jarak terkecil, sehingga perhitungannya adalah sebagai berikut:

$W_{1,1} = 1 + 0{,}05 * (0 - 1) = 0{,}95$

$W_{1,2} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,3} = 1 + 0{,}05 * (0 - 1) = 0{,}95$

$W_{1,4} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,5} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,6} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,7} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,8} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,9} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,10} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,11} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,12} = 1 + 0{,}05 * (1 - 1) = 1$

g) Langkah d) sampai f) dikerjakan terhadap semua masukan (x), setelah selesai dilakukan perhitungan perubahan nilai rasio dengan rumus:

$rasio = rasio - pengurangan\, rasio * rasio$

$rasio = 0{,}05 - 0{,}1 * 0{,}05 = 0{,}045$

h) Perhitungan diulang hingga perulangan berhenti dan mendapatkan matriks nilai bobot terakhir. Pada contoh ini perulangan berhenti

pada perulangan ke-15 dengan hasil akhir bobot ditunjukkan pada Tabel III.8.

Tabel III.8 Bobot Hasil Pembelajaran

Huruf	Kelas	Vektor	
A	1	0,66 | 1 | 0,66 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1	Bobot (w1)
C	2	0,66 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0,66 | 1 | 1	Bobot (w2)

```
while ((E < ME) or (rate > eps)) do
      E←E+1
      for i ← 0 to  jmlGbr do
            for k ← 0 to  jmlKls do
                  hasil ← 0
                  for j ← 0 to lebar do
                        hasil ← hasil + Pow((P[j][i] - W[j][k]), 2)
                  endfor
                 H[k] ←  Sqrt(hasil)
            endfor
            C ← 0
           for k ← 1 to jmlKls do
                 if (H[k] < H[C]) then
                        C ← k
                endif
           endfor
           if (TK[C] == TP[i]) then
                 for j ← 0 to lebar do
                      W[j][C] ← W[j][C] + rate * (P[j][C] - W[j][C])
                 endfor
           else
                 for j ← 0 to lebar do
                      W[j][C] ← W[j][C] - rate * (P[j][C] - W[j][C])
                 endfor
           endif
      endfor
      rate ← rate - (0.1 * rate)
endfor
```

Gambar III.17 *Pseudcode* Proses Pembelajaran LVQ

3. Proses Pengenalan

Proses pengenalan merupakan proses dimana komputer mengenali data *test* dengan menggunakan nilai bobot yang telah diperoleh dari proses pembelajaran. Sebagai contoh, data yang akan dikenali adalah array 1 dimensi representasi dari gambar huruf A yang telah melalui proses pengolahan citra seperti terlihat pada Gambar III.18.

1
1
1
1
1
0
1
0
0
1
1
1

Gambar III.18 Array 1 Dimensi Huruf "A"

3.1. Pengenalan *Backpropagation*

Proses pengenalan dengan metode *backpropagation* hanya menggunakan 1 alur, yaitu alur maju (langkah f) pada proses pembelajaran). *Pseudocode* proses pengenalan *backpropagation* ditunjukkan pada Gambar III.19.

Berikut ini langkah-langkah pengenalannya:

a) Hitung data *test* menggunakan sesuai perhitungan pada alur maju. Perhitungan ini dilakukan satu persatu terhadap setiap hasil w dan v dari setiap data. Jika data hasil sudah sesuai dengan dengan tujuan maka perhitungan dihentikan.

$$z_in_i = v0_i + \sum_{i=1}^{n} x_i * v_{i,j}$$

$$z_i = \frac{1}{1 + e^{-z_in_i}}$$

$$z_in_1 = 0.1586 + (1 * 0.4122) + (1 * 0.9819) + (0 * 0.5511) + (1 * 0.9010) + (0 * 0.5512) + (1 * 0.3480) + (1 * 0.5722) + (1 * 0.9120) + (1 * 0.8612) + (1 * 0.5946) + (0 * 0.6069) + (1 * 0.6330) = 5.9625$$

$$z_in_2 = 3.5381$$

$$z_in_3 = 3.7088$$

$$z_in_4 = 5.2577$$

$$z_1 = \frac{1}{1 + e^{-5.9625}} = 0.9974$$

$$z_2 = \frac{1}{1 + e^{-3.5381}} = 0.9717$$

$$z_3 = \frac{1}{1 + e^{-3.7088}} = 0.9761$$

$$z_4 = \frac{1}{1 + e^{-5.2577}} = 0.9948$$

Operasi pada *output layer* :

$$y_in_k = w0 + \sum_{i=1}^{n} w_{ik} * z_i$$

$$y_k = \frac{1}{1 + e^{-y_in_k}}$$

$$y_in = 0 + (0.9267 * 0.9974) + (0.2497 * 0.9717) +$$

$$(0.8463 * 0.9761) + (0.2970 * 0.9948) = 3.2029$$

$$y = \frac{1}{1 + e^{-3.2029}} = 0.9609$$

b) Nilai y hasil perhitungan pada langkah a) menjadi hasil pengenalan. Kemudian ditentukan batas ambang (*threshold*) dari nilai hasil. Diambil *threshold* = 0.5, artinya jika nilai y memiliki angka dibelakang koma lebih besar atau sama dengan 5, maka nilai tersebut dibulatkan keatas, dan sebaliknya. Pada perhitungan di atas didapatkan nilai y={0.3675,0.9609} , maka hasil y = {0,1}, artinya data *test* termasuk kelas 1, yang memiliki nilai biner {0,1}, yaitu huruf A.

```
for j ← 0 to 4 do
     Z_net[j] ← 0
     Z[j] ← 0
     for i ← 0 to maxI do
         Z_net[j] ← Z_net[j] + (P[i][baris] * V[i][j])
     endfor
     Z[j] ← 1 / (1 + Pow(e, -Z_net[j]))
endfor
for k ← 0 to maxK do
     Y_net[k] ← 0
     for j ← 0 to 4 do
         Y_net[k] ← Y_net[k] +(Z[j] * W[k][j])
     endfor
     Y[k] ← 1 / (1 + Pow(e, -Y_net[k]))
endfor
```

Gambar III.19 *Pseudocode* Proses Pengenalan *Backpropagation*

3.2. Pengenalan *Learning vector quantization*

Proses pengenalan dengan metode *learning vector quantization* menggunakan rumus yang sama dengan perhitungan bobot pada proses pembelajaran *learning vector quantization*. Gambar III.20 merupakan *pseudocode* proses pengenalan *learning vector quantization.*

Berikut ini langkah-langkahnya:

a) Data *test* dihitung terhadang setiap data bobot (w) dengan rumus:

$$C_j = \sqrt{\sum_{i=1}^{n} \left(x_i - w_{ji}\right)^2}$$

Jarak pada:

1) Bobot ke-1 (w1)

$C_1 = \sqrt{((1-0{,}66)\text{^}2 + (1-1)\text{^}2 + (0-0{,}66)\text{^}2 + (1-1)\text{^}2 + (0-0)\text{^}2 + (1-1)\text{^}2 + (1-1)\text{^}2 + (1-1)\text{^}2 + (1-1)\text{^}2 + (0-0)\text{^}2 + (1-1)\text{^}2)} = 0{,}74$

2) Bobot ke-2 (w2)

$C_2 = \sqrt{((1-0{,}66)\text{^}2 + (1-1)\text{^}2 + (0-1)\text{^}2 + (1-1)\text{^}2 + (0-0)\text{^}2 + (1-0)\text{^}2 + (1-1)\text{^}2 + (1-0)\text{^}2 + (1-0)\text{^}2 + (1-0{,}66)\text{^}2 + (0-1)\text{^}2 + (1-1)\text{^}2)} = 2{,}29$

b) Dilihat jarak (C) terkecil yang diperoleh pada perhitungan terhadap data *test*. Kelas bobot yang memiliki jarak terkecil menjadi hasil dari proses pengenalan ini. Dalam perhitungan di atas menghasilkan jarak terkecil ada pada bobot ke-1 (w1) dan w1

merupakan kelas 1, sehingga data *test* termasuk ke dalam kelas 1 yaitu huruf A.

```
for k ←0 to  jmlKls do
     hasil ←0
     for j ←0 to lebar do
         hasil ←hasil + Pow((P[j][i] - W[j][k]), 2)
     endfor
     H[k] ←  Sqrt(hasil)
endfor
C ←0
for k ←1 to jmlKls do
    if (H[k] < H[C]) then
        C ←k
    endif
endfor
```

Gambar III.20 *Pseudocode* Proses Pengenalan *Learning Vector Quantization*.

3.1.3 Analisis Kebutuhan Data Non Fungsional

Analisis kebutuhan non fungsional menjelaskan pendukung sistem yang akan dijalankan. Adapun kebutuhan-kebutuhannya yaitu:

3.1.3.1 Kebutuhan Perangkat Keras

Komputer terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya.

Pada aplikasi perbandingan algoritma ini, perangkat keras yang digunakan berupa sebuah komputer yang berbasis desktop. Adapun spesifikasi yang digunakan dalam penelitian ini yaitu:

1. Processor : Intel Pentium Core 2 Duo @ 2.2 GHz
2. Memory : 1 GB
3. VGA : 512 MB
4. Hardisk : 50 MB
5. Monitor : 17 inch
6. Mouse dan Keyboard

3.1.3.2 Kebutuhan Perangkat Lunak

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi ini adalah sebagai berikut :

1. Sistem Operasi Windows 7
2. Microsoft Visual Studio 2010
3. Microsoft SQL Server 2008

3.1.3.3 Kebutuhan User

Untuk menjalankan aplikasi dibutuhkan spesifikasi user yang dapat menjalankan fungsi-fungsi dari aplikasi ini. Hanya terdapat satu jenis user dalam aplikasi ini, yaitu Penguji. Adapun spesifikasi user yang dibutuhkan adalah sebagai berikut:

a. Dapat mengoperasikan komputer.
b. Mengerti bahasa Inggris dasar dan bahasa Indonesia.
c. Mengerti maksud penggunaan algoritma *backpropagation* dan *learning vector quantization*.

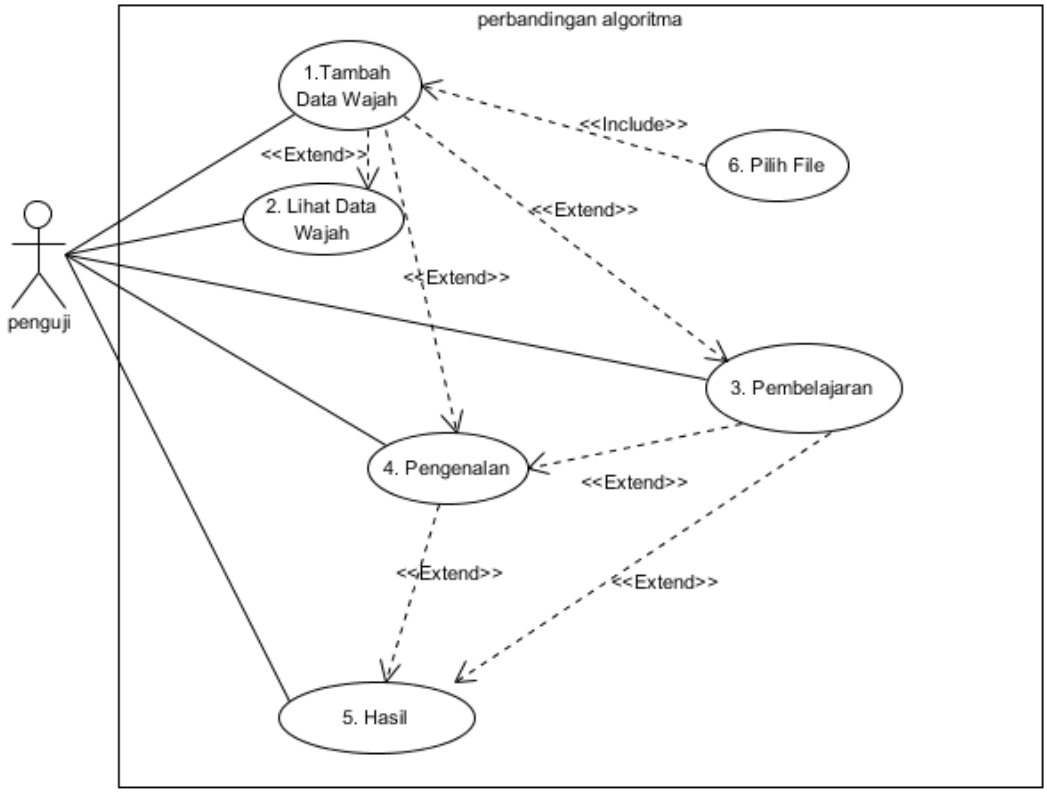
d. Mengerti istilah-istilah dalam algoritma *backpropagation* dan *learning vector quantization*.

3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menjelaskan proses-proses yang terjadi di dalam aplikasi ini. Proses-proses tersebut dijelaskan dalam usecase diagram, skenario diagram, activity diagram, sequence diagram dan class diagram.

3.1.4.1 Use Case Diagram

Use case diagram menggambarkan proses yang ada di dalam aplikasi ini. Use case diagram aplikasi ini digambarkan pada Gambar III.21.



Gambar III.21 Use Case Diagram Aplikasi Perbandingan Algoritma

3.1.4.2 Skenario Diagram

Skenario diagram menjelaskan skenario dari setiap proses yang terdapat pada use case diagram. Skenario diagram pada aplikasi ini dijelaskan pada Tabel III.9 sampai III-14 berikut ini:

Tabel III.9 Skenario diagram proses Tambah data Wajah

Identifikasi	
Nomor	1
Nama	Tambah Data Wajah
Tujuan	Menambah data gambar wajah
Deskripsi	Proses menambah data informasi dan gambar wajah
Aktor	Penguji
Skenario Utama	
Kondisi Awal	Data dan gambar wajah belum ada

Aksi Aktor	Reaksi Sistem
1. Penguji berada pada tampilan sistem perbandingan algoritma	
2. Penguji memilih tombol tambah data wajah	3. Menampilkan form tambah data wajah
4. Penguji mengisi form tambah data berupa gambar dan informasinya	
5. Memilih tombol Simpan	6. Melakukan validasi semua data sudah terisi atau belum.
	7. Jika data belum lengkap, menampilkan pesan “data belum diisi semua”, kemudian kembali ke tahap 4.
	8. Jika data sudah lengkap, melakukan proses pengolahan

	citra (*scaling, grayscale, edgedetection*).
	9. Simpan gambar citra asli dan hasil *edge detection* kedalam direktori aplikasi, kemudian simpan data ke dalam database.
	10. Menampilkan pesan "tambah data berhasil".
Kondisi Akhir	Data gambar dan informasi wajah baru telah ditambahkan

Tabel III.10 Skenario Diagram Proses Lihat Data Wajah

Identifikasi	
Nomor	2
Nama	Lihat Data Wajah
Tujuan	Menampilkan data gambar wajah manusia
Deskripsi	Menampilkan data informasi dan gambar wajah manusia
Aktor	Penguji
Skenario Utama	
Kondisi Awal	Data gambar wajah sudah ada
Aksi Aktor	Reaksi Sistem
1. Penguji berada pada tampilan sistem perbandingan algoritma	
2. Penguji memilih tombol lihat data wajah	3. Menampilkan tampilan lihat data wajah
4. Penguji memilih data	5. Mengambil detail data dari database
	6. Menampilkan detail data
Kondisi Akhir	Menampilkan detail data yang ada di dalam *database*

Tabel III.11 Skenario Diagram Proses Pembelajaran

Identifikasi	
Nomor	3
Nama	Pembelajaran
Tujuan	Melakukan pembelajaran sesuai algoritma yang digunakan
Deskripsi	Proses melakukan pembelajaran gambar wajah manusia
Aktor	Penguji
Skenario Utama	
Kondisi Awal	Data pembelajaran belum ada

Aksi Aktor	Reaksi Sistem
1. Penguji berada pada tampilan sistem perbandingan algoritma	
2. Penguji memilih tombol pembelajaran	3. Menampilkan form pembelajaran
4. Penguji mengisi parameter yang akan digunakan dalam pembelajaran	
5. Memilih tombol pelajari	6. Melakukan validasi apakah semua parameter pembelajaran sudah diisi.
	7. Jika parameter belum diisi semua, akan menampilkan pesan “parameter pembelajaran belum lengkap”, kemudian kembali ke langkah 4.
	8. Jika parameter sudah lengkap, sistem mengambil semua data dan data *training* dari dalam

	database.
	9. Melakukan proses thresholding dan menghasilkan matriks data masukan proses pembelajaran.
	10. Melakukan proses pembelajaran dengan metode *learning vector quantization.*
	11. Menyimpan bobot W hasil pembelajaran LVQ
	12. Melakukan pembelajaran dengan metode *backpropagation.*
	13. Menyimpan bobot W dan V hasil pembelajaran BP
	14. Menampilkan seluruh hasil pembelajaran.
Kondisi Akhir	Data *training* telah dipelajari. Dan bobot hasil pembelajaran disimpan ke dalam *database*, kemudian ditampilkan.

Tabel III.12 Skenario Diagram Proses Pengenalan

Identifikasi	
Nomor	4
Nama	Pengenalan
Tujuan	Melakukan pengenalan gambar sesuai algoritma yang digunakan
Deskripsi	Proses melakukan pengenalan gambar wajah manusia
Aktor	Penguji
Skenario Utama	

Kondisi Awal	Gambar *test* belum dikenali
Aksi Aktor	Reaksi Sistem
1. Penguji berada pada tampilan sistem perbandingan algoritma	
2. Penguji memilih tombol pengenalan	3. Menampilkan form pengenalan
4. Penguji memilih gambar yang akan dikenali	
5. Memilih tombol kenali	6. Melakukan validasi apakah data tes yang akan dikenali telah dipilih.
	7. Jika belum ada data tes yang dipilih maka akan menampilkan pesan “data tes pengenalan belum dipilih”, kemudian kembali ke langkah 4.
	8. Jika data tes sudah dipilih, sistem melakukan proses *thresholding* terhadap data tes yang dipilih.
	9. Mengambil data bobot LVQ dan BP dari dalam database.
	10. Melakukan proses pengenalan dengan metode LVQ dan BP
	11. Menghitung waktu dan akurasi pengenalan.
	12. Menyimpan hasil pengenalan ke dalam database.
	13. Menampilkan hasil pengenalan dan grafiknya.

Kondisi Akhir	Data *test* telah dikenali dan hasil pengenalan disimpan ke dalam *database.* Kemudian hasil pengenalan ditampilkan.

Tabel III.13 Skenario Diagram Proses Hasil

Identifikasi		
Nomor	5	
Nama	Hasil	
Tujuan	Menampilkan hasil perhitungan pengenalan gambar dan grafik	
Deskripsi	Proses menampilkan hasil pengenalan gambar	
Aktor	Penguji	
Skenario Utama		
Kondisi Awal	Hasil pengenalan atau pembelajaran belum ditampilkan	
Aksi Aktor		Reaksi Sistem
1. Penguji berada pada tampilan sistem perbandingan algoritma		
2. Penguji memilih tombol hasil		3. Menampilkan form hasil
4. Penguji memilih tombol hasil pembelajaran		5. Menampilkan tampilan hasil pembelajaran
6. Penguji memilih data hasil pembelajaran yang akan ditampilkan		7. Sistem menampilkan detail hasil pembelajaran yang dipilih
8. Penguji memilih tombol hasil pengenalan		9. Menampilkan tampilan hasil pengenalan
10. Penguji memilih data hasil pengenalan yang akan ditampilkan		11. Sistem akan menampilkan detail hasil pengenalan yang dipilih
12. Penguji memilih tombol hasil		13. Menampilkan grafik seluruh

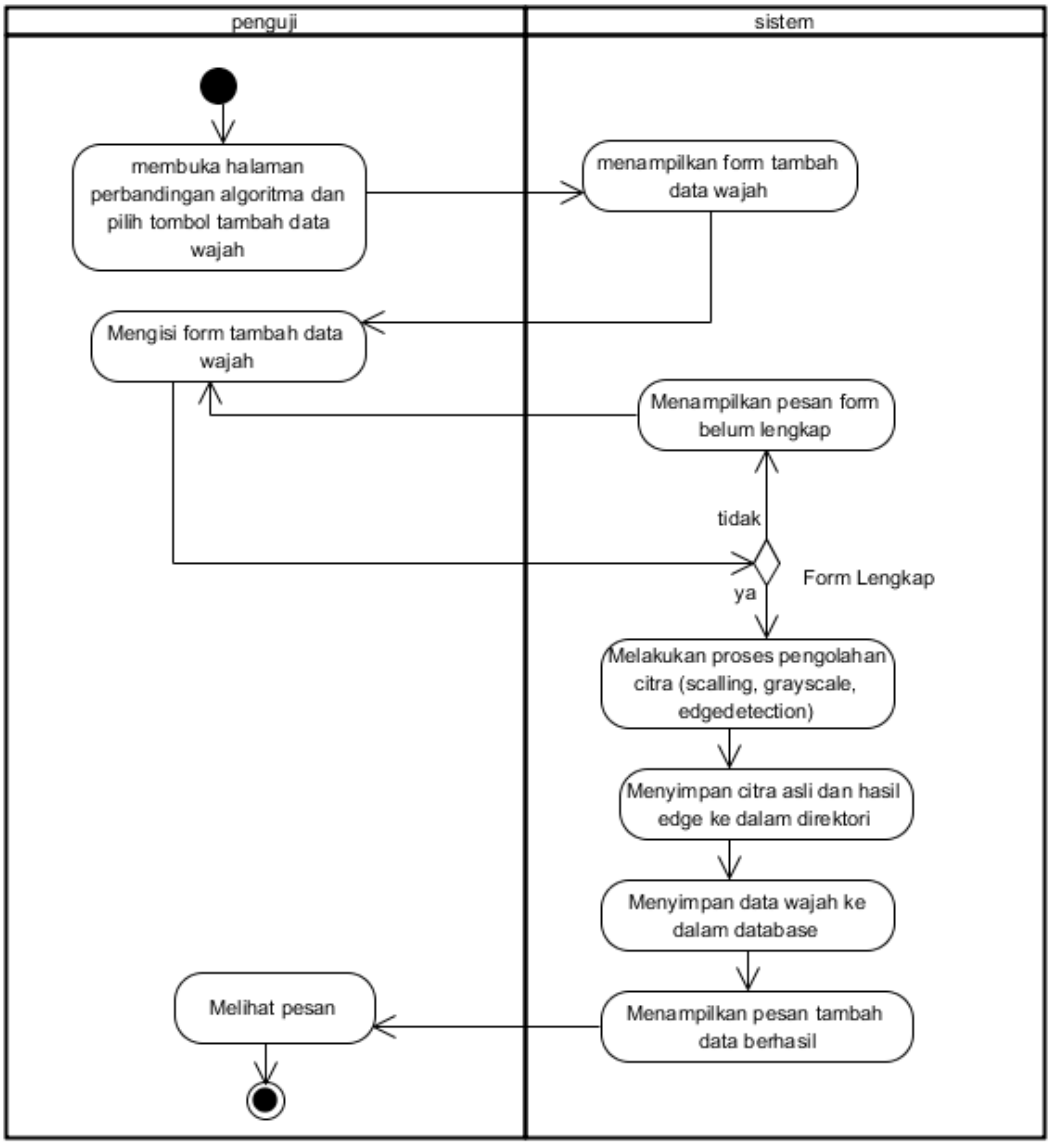
seluruh pengenalan	pengenalan.
Kondisi Akhir	Hasil pengenalan atau pembelajaran ditampilkan

Tabel III.14 Skenario Diagram Proses Pilih File

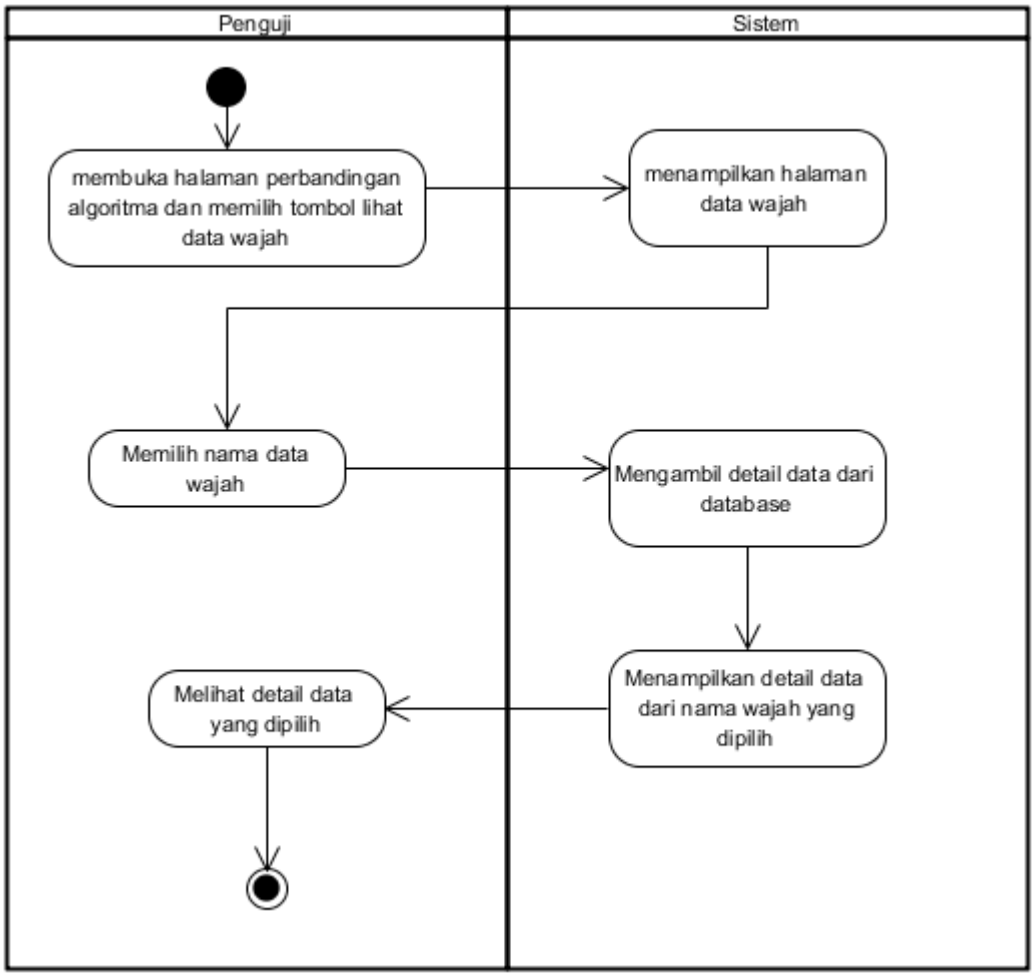
Identifikasi		
Nomor	6	
Nama	Pilih File	
Tujuan	Mengambil file gambar dari direktori di luar sistem	
Deskripsi	Proses untuk mengambil file gambar dengan format .jpg dari luar direktori di luar sistem untuk dimasukkan ke dalam sistem.	
Aktor	Penguji	
Skenario Utama		
Kondisi Awal	File gambar pada tampilan tambah data wajah belum dipilih	
Aksi Aktor		Reaksi Sistem
1. Penguji berada pada tampilan form tambah data wajah		
2. Penguji memilih tombol Pilih file		3. Menampilkan form direktori file
4. Penguji memilih file yang akan dimasukkan ke dalam sistem		5. Menutup form direktori file
		6. Menampilkan file yang dipilih dan alamat direktorinya
Kondisi Akhir	File gambar sudah dipilih dan ditampilkan pada form tambah data wajah	

3.1.4.3 Activity Diagram

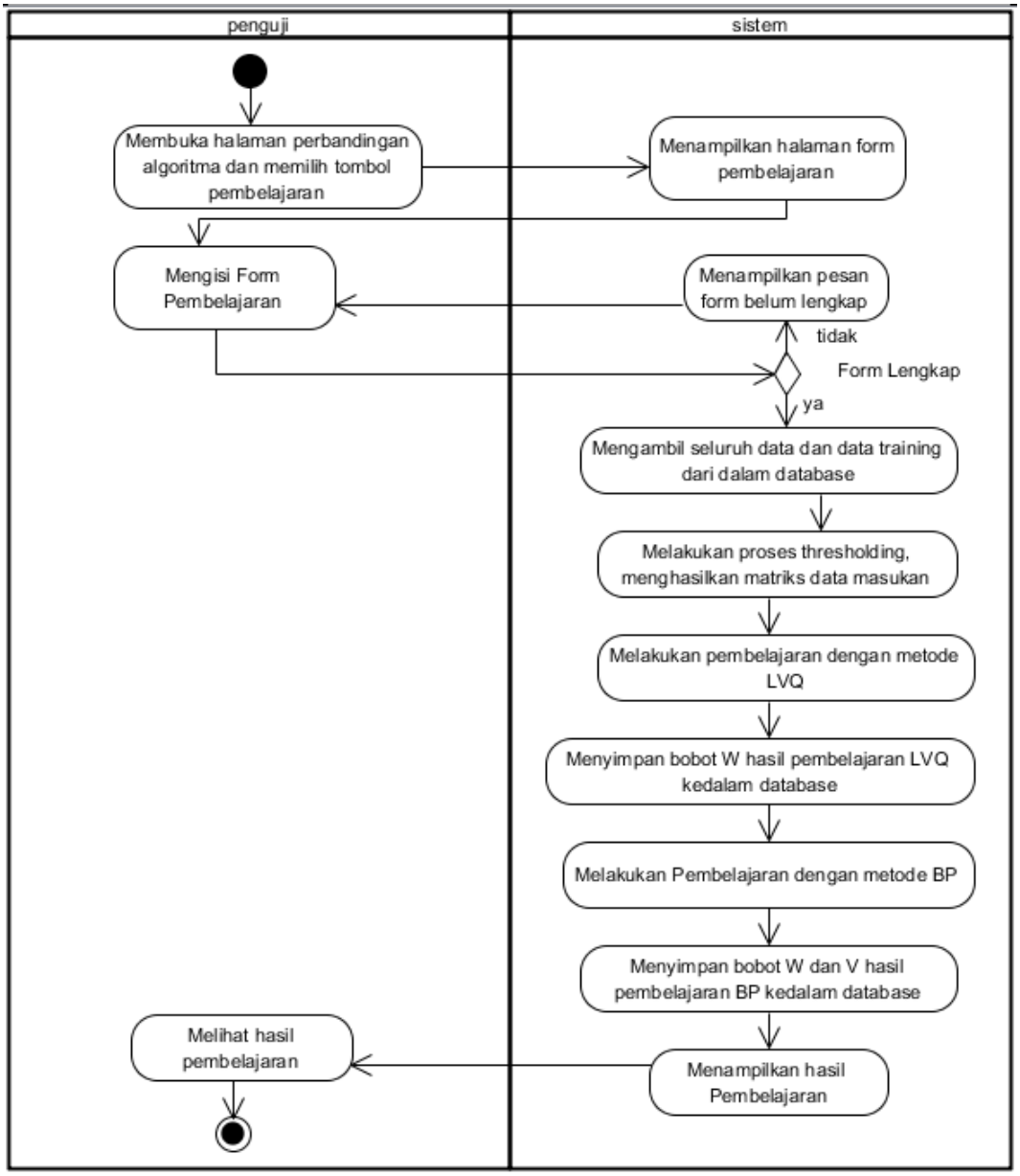
Activity diagram menjelaskan alur aktivitas dari setiap usecase. Activity diagram sistem ini ditungjukkan pada Gambar III.22 sampai III.27.



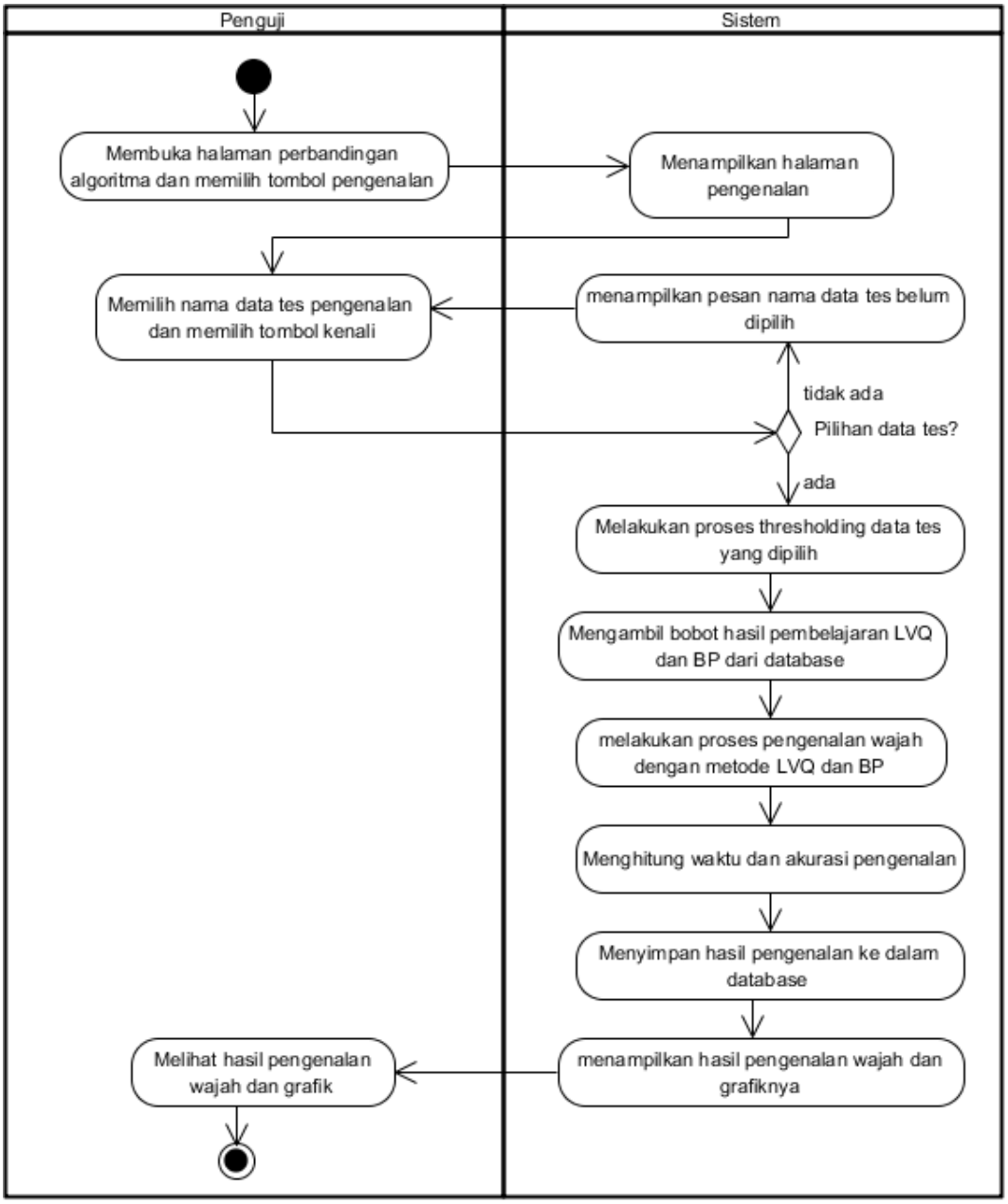
Gambar III.22 Activity Diagram Tambah Data Wajah



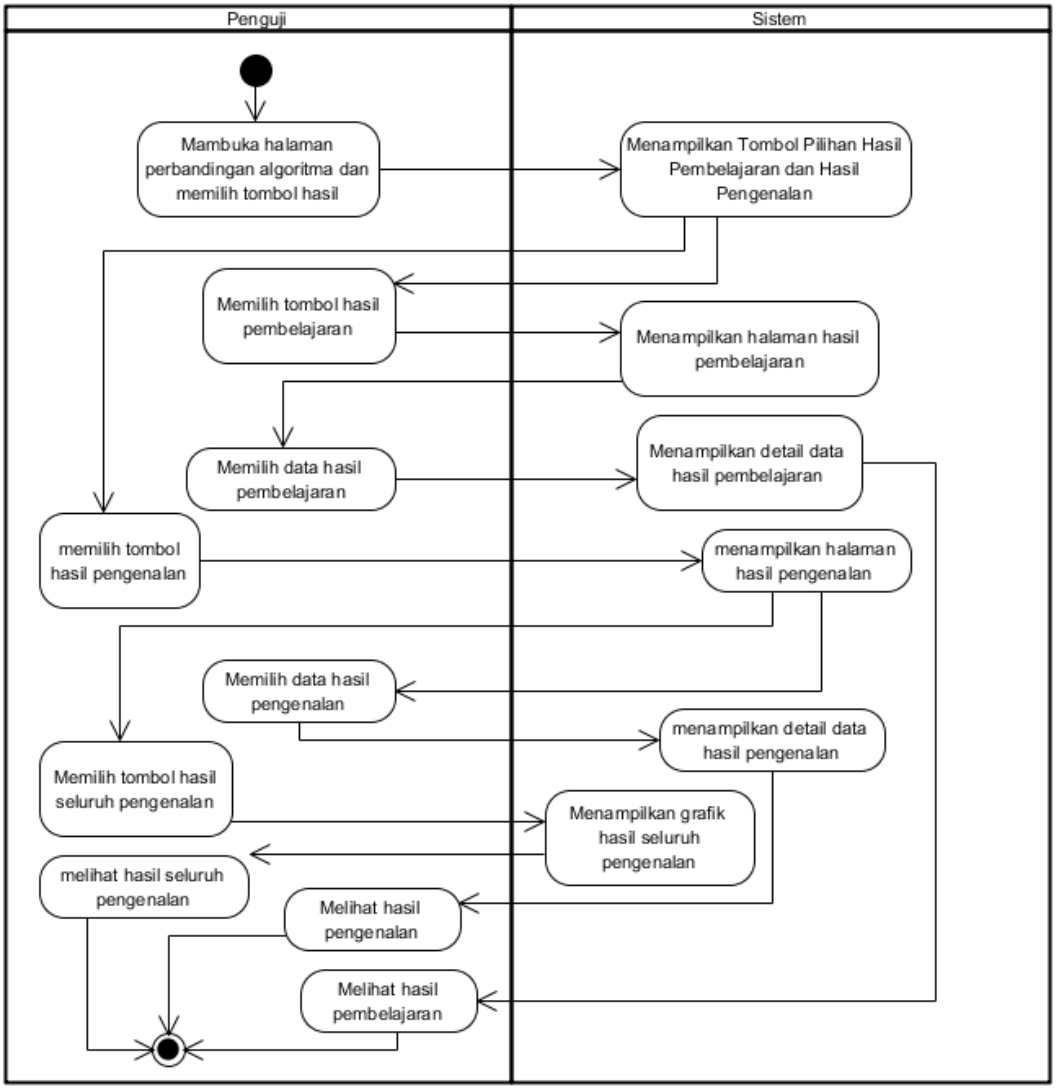
Gambar III.23 Activity Diagram Lihat Data Wajah



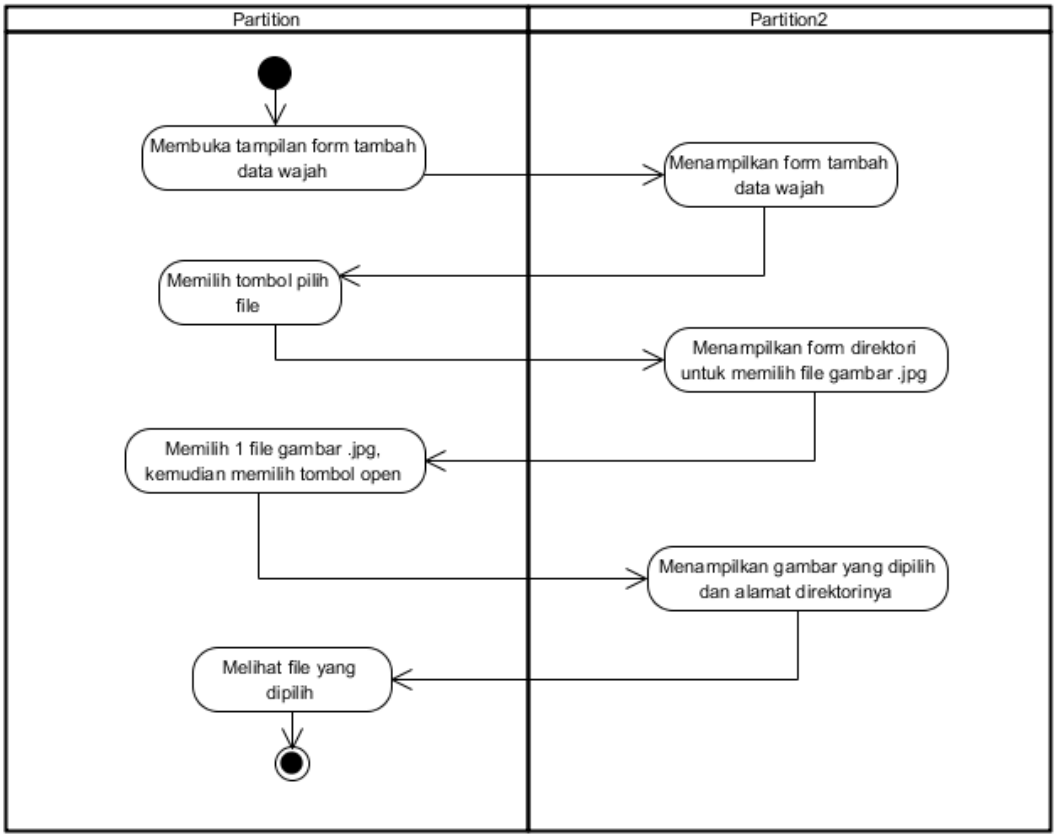
Gambar III.24 Activity Diagram Pembelajaran



Gambar III.25 Activity Diagram Pengenalan



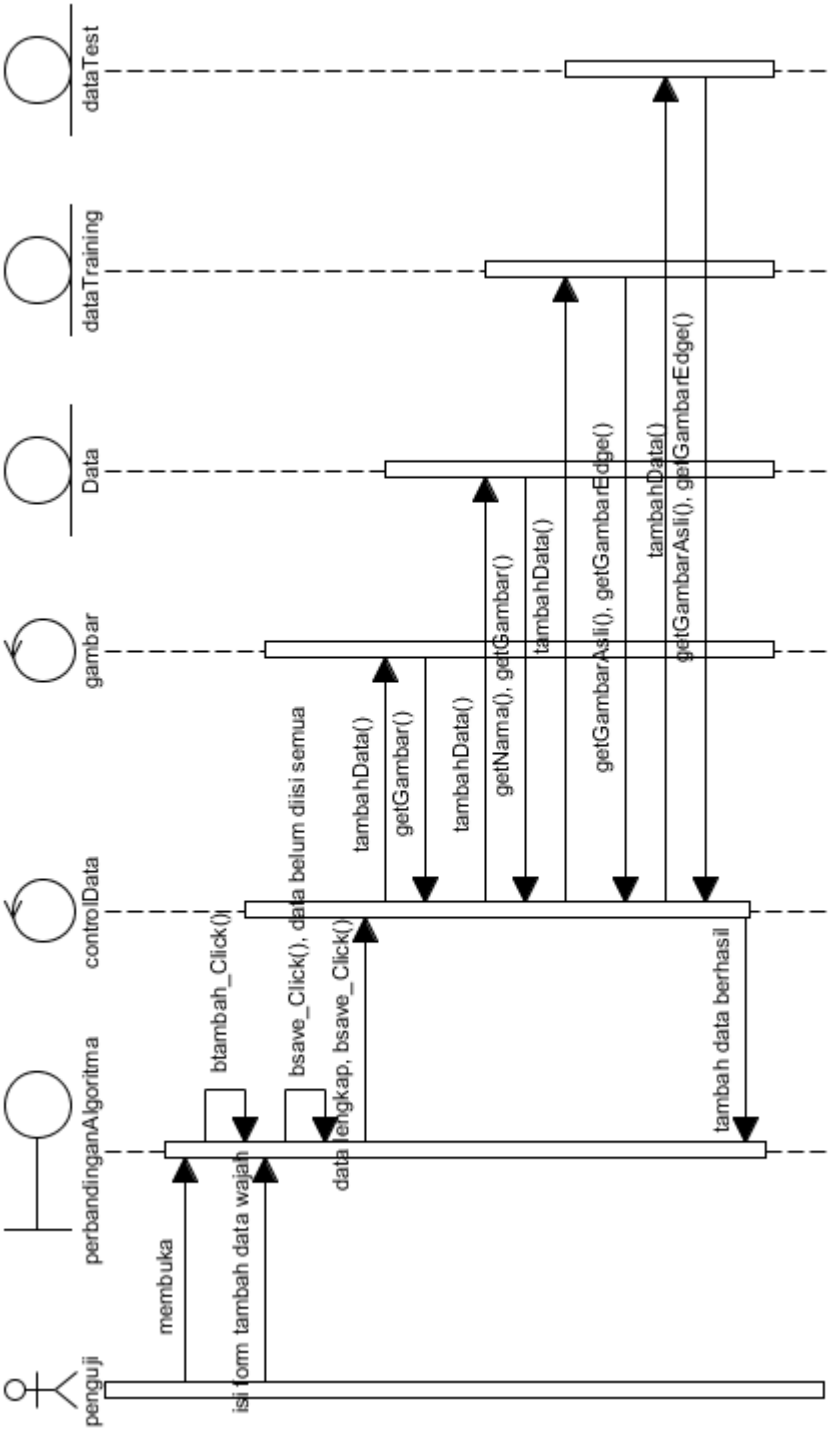
Gambar III.26 Activity Diagram Hasil



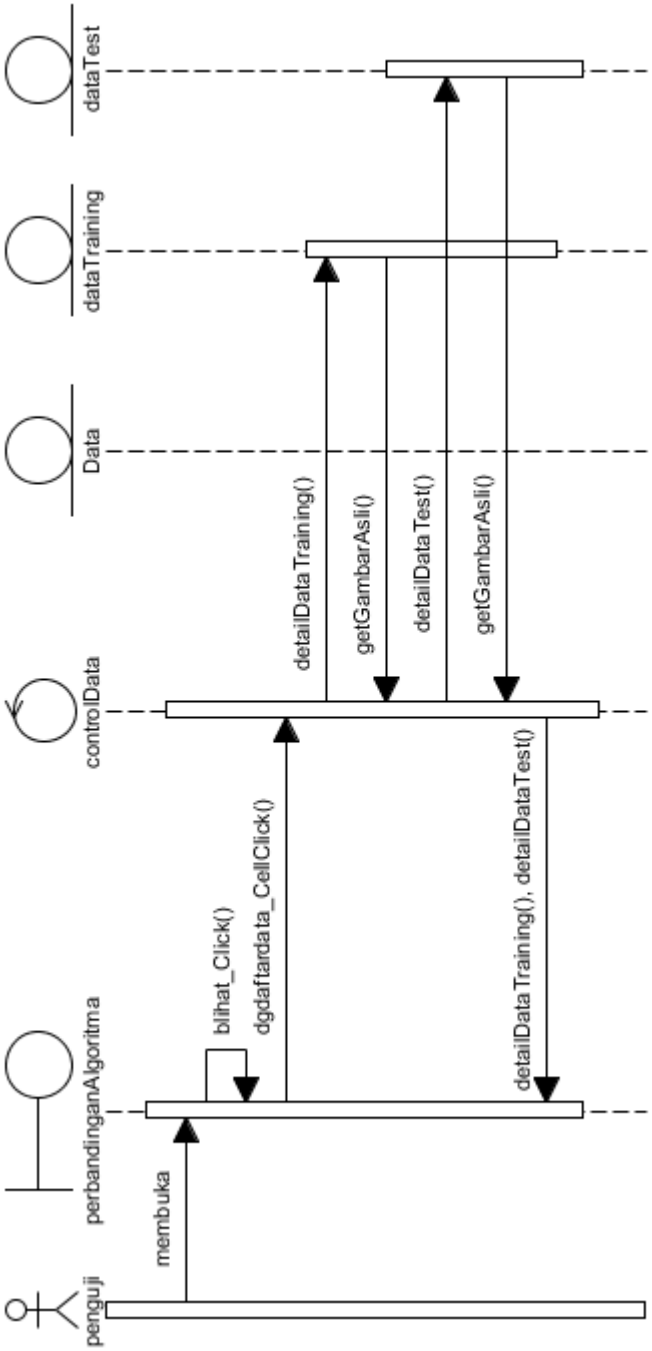
Gambar III.27 Activity Diagram Pilih File

3.1.4.4 Sequence Diagram

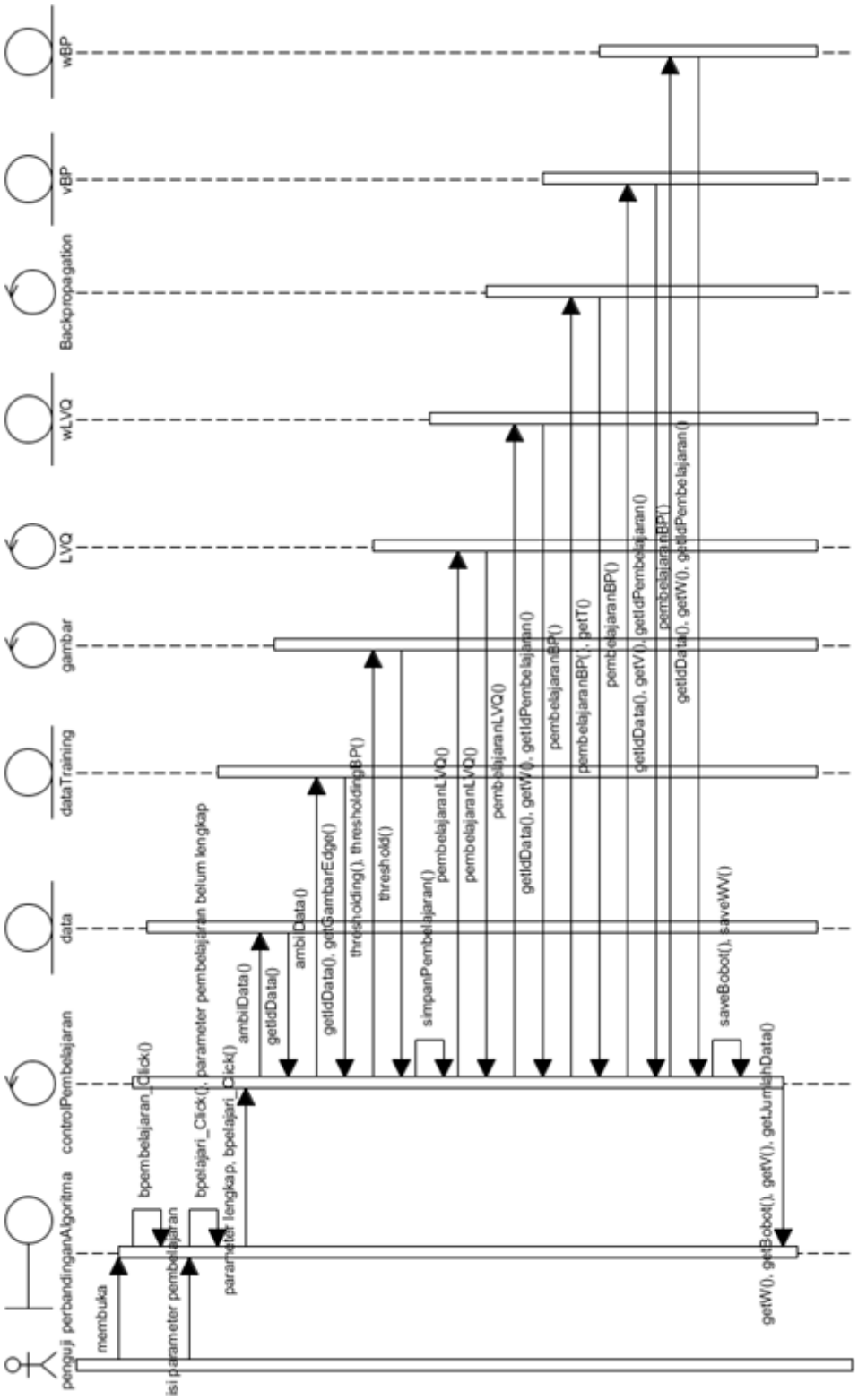
Sequence diagram menjelaskan alur yang akan berjalan dalam sistem ini. Sequence diagram sistem ini ditunjukkan pada Gambar III.28 sampai III.33.



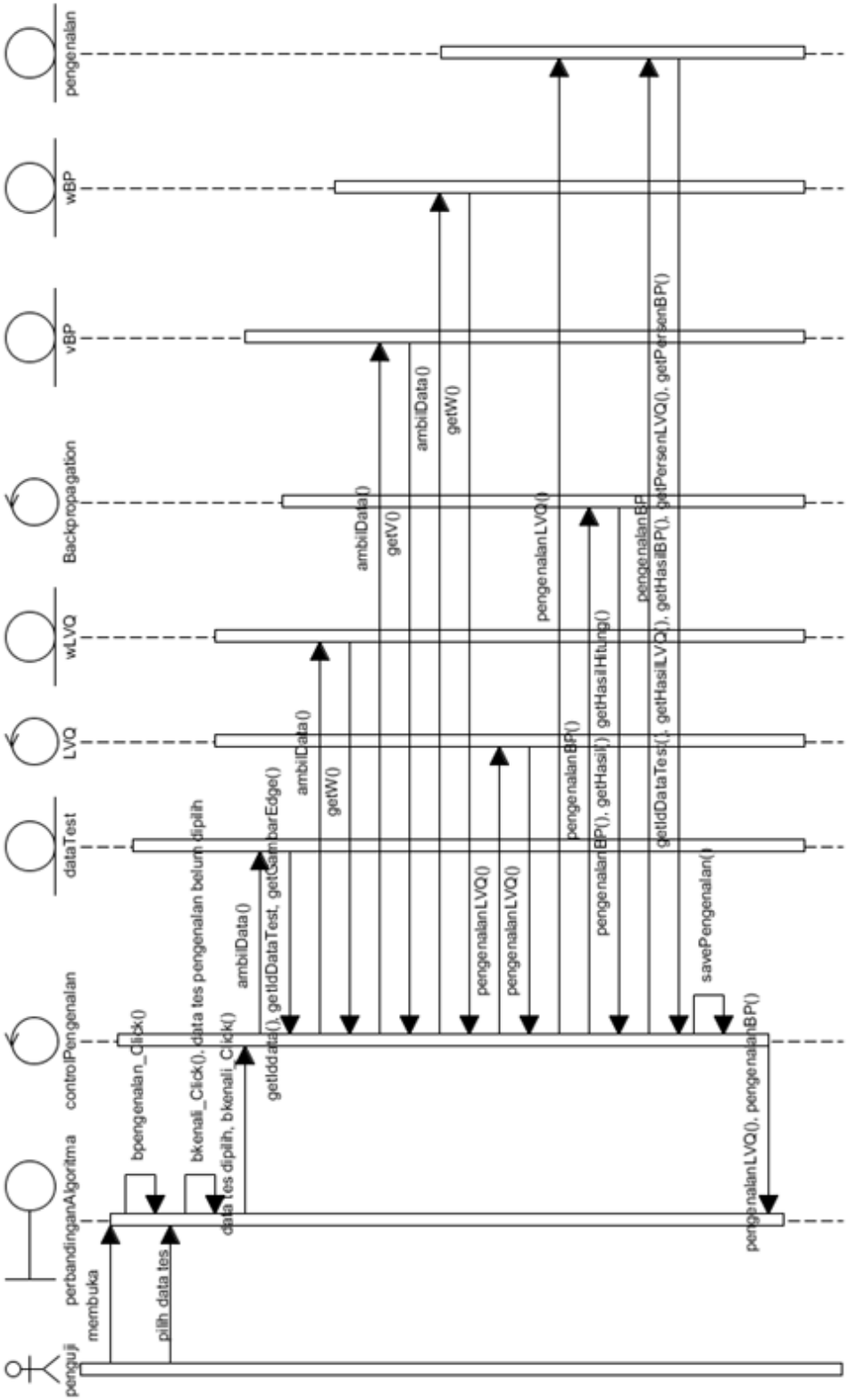
Gambar III.28 Sequence Diagram Proses Tambah Data Wajah



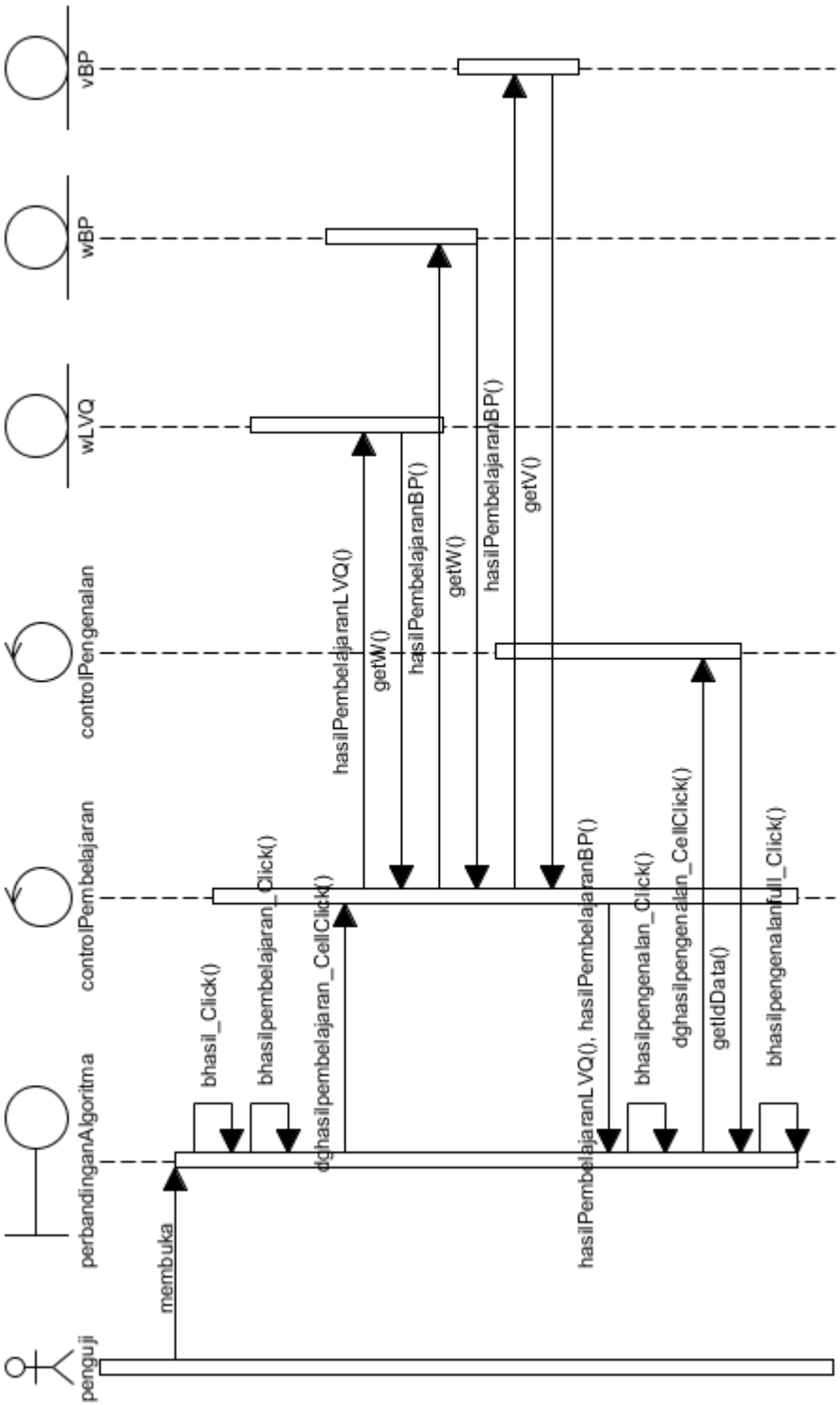
Gambar III.29 Sequence Diagram Proses Lihat Data Wajah



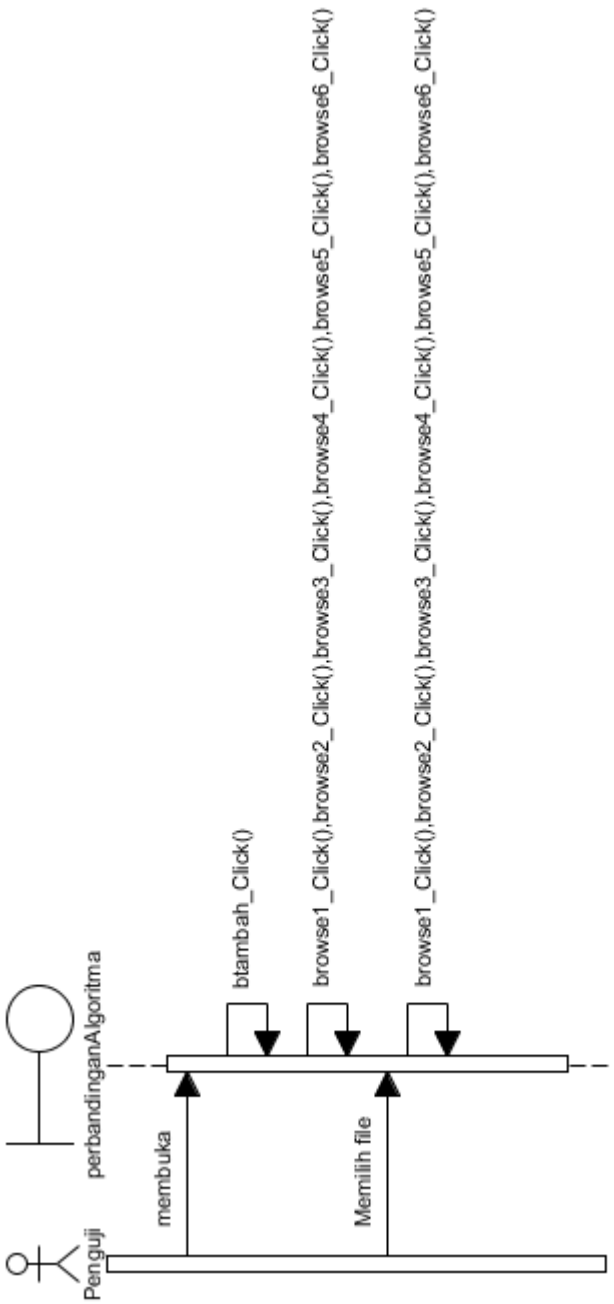
Gambar III.30 Sequence Diagram Proses Pembelajaran



Gambar III.31 Sequence Diagram Proses Pengenalan



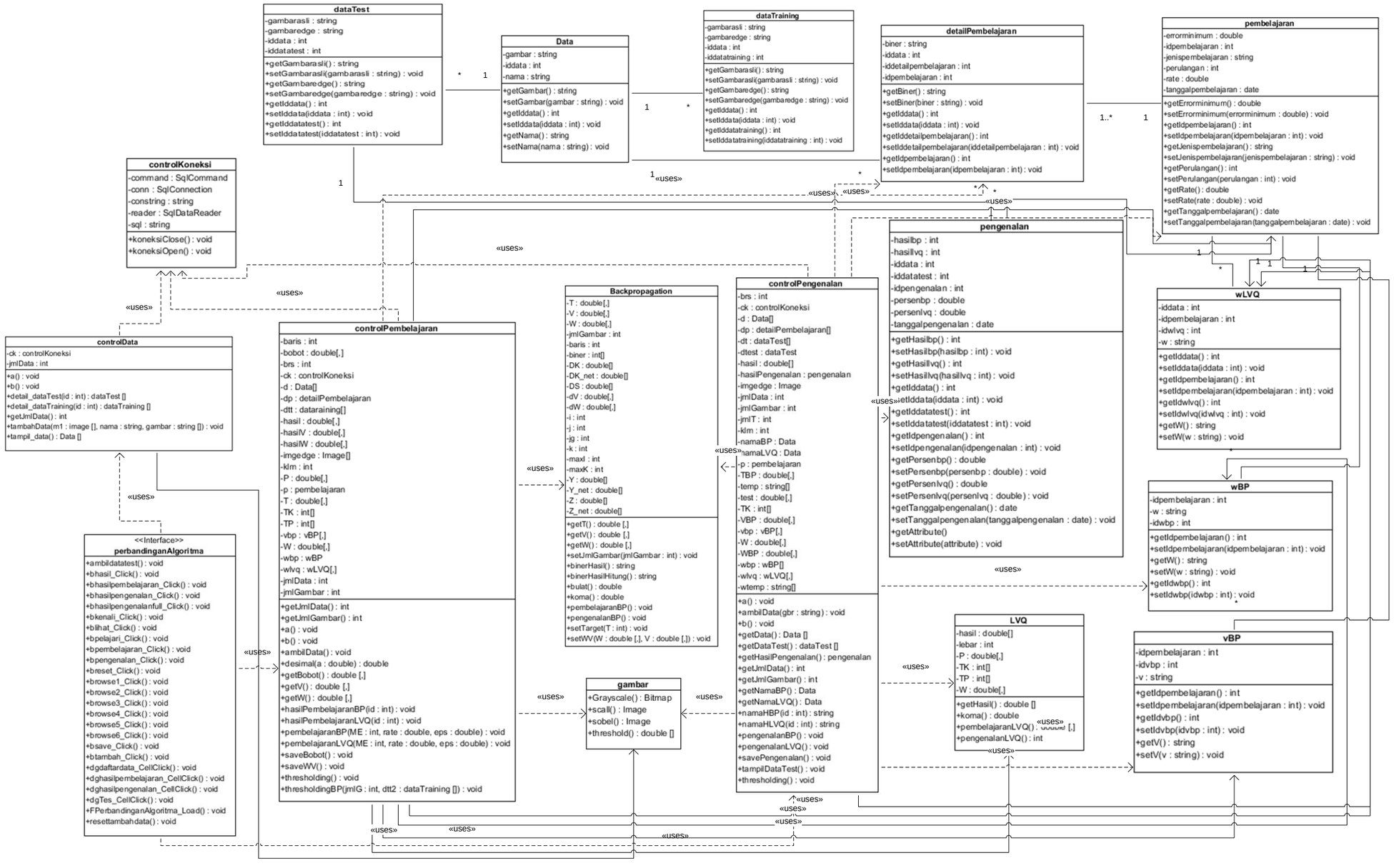
Gambar III.32 Sequence Diagram Proses Hasil



Gambar III.33 Sequence Diagram Pilih File

3.1.4.5 Class Diagram

Class diagram menjelaskan analisis class yang akan digunakan, terutama pada class *Entity*. Class diagram sistem ini digambarkan pada Gambar III.34.



Gambar III.34 Class Diagram Aplikasi Perbandingan Algoritma

3.2 Perancangan Sistem

Perancangan sistem menjelaskan perancangan class dan *database* yang akan dibangun pada sistem ini.

3.2.1 Perancangan Class

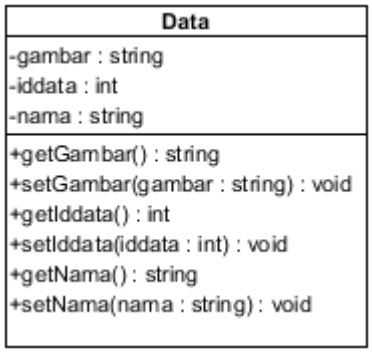
Perancangan class menjelaskan class-class yang akan dibuat. Terdapat 3 jenis class yaitu class *Entity*, class boundary dan class control.

3.2.1.1 Perancangan Class Entity (Data)

Class entity merupakan class yang berisi data, mewakili setiap table yang ada di dalam database dan memiliki method getter dan setter untuk setiap variablenya. Terdapat 9 class entity, yaitu Data, dataTest, dataTraining, pembelajaran, detailPembelajaran, pengenalan, wBP, vBP, wLVQ.

1. **Data**

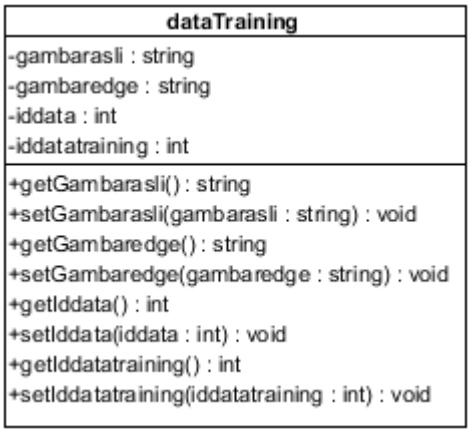
   Class Data merupakan class entity yang mewakili table data. Gambar III.35 menggambarkan perancangan class Data.



Gambar III.35 Perancangan class data

2. dataTraining

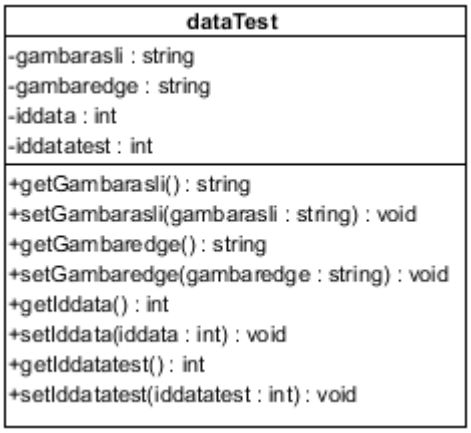
Class dataTraining merupakan class entity yang mewakili table dataTraining. Gambar III.36 menggambarkan perancangan class dataTraining.



Gambar III.36 Perancangan class dataTraining

3. dataTest

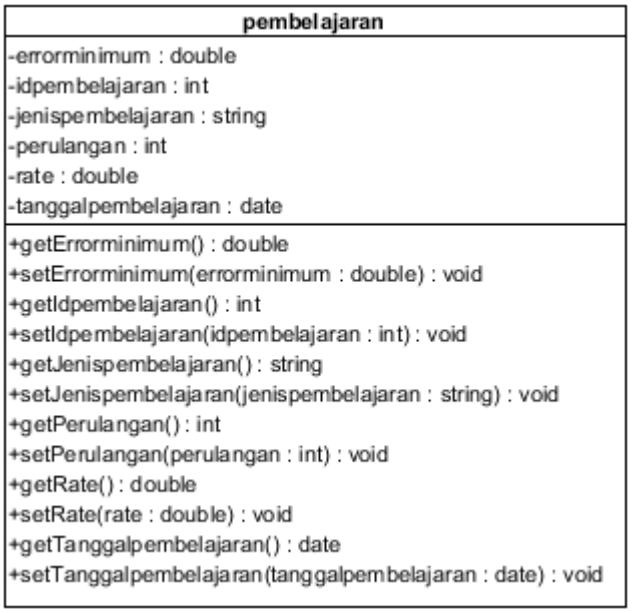
Class dataTest merupakan class entity yang mewakili table dataTest. Gambar III.37 menggambarkan perancangan class dataTest.



Gambar III.37 Perancangan class dataTest

4. Pembelajaran

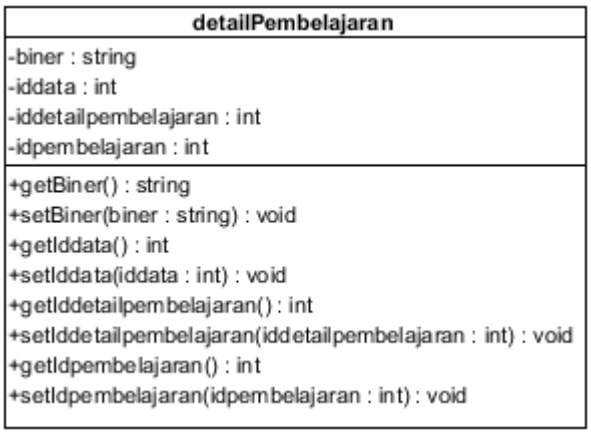
Class pembelajaran merupakan class entity yang mewakili table pembelajaran. Gambar III.38 menggambarkan perancangan class pembelajaran.



Gambar III.38 Perancangan class pembelajaran

5. detailP**embelajaran**

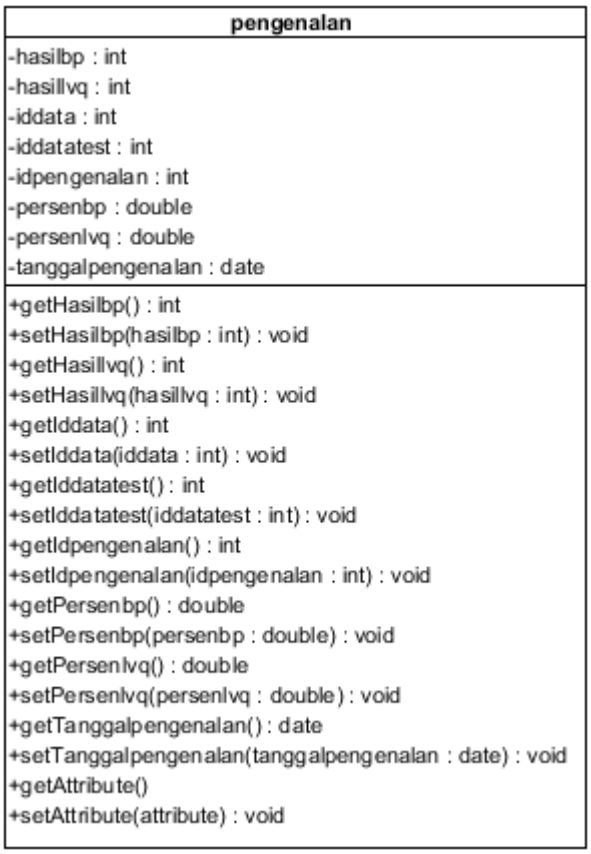
Class detailPembelajaran merupakan class entity yang mewakili table detailPembelajaran. Gambar III.39 menggambarkan perancangan class detailPembelajaran.



Gambar III.39 Perancangan class detailPembelajaran

6. Pengenalan

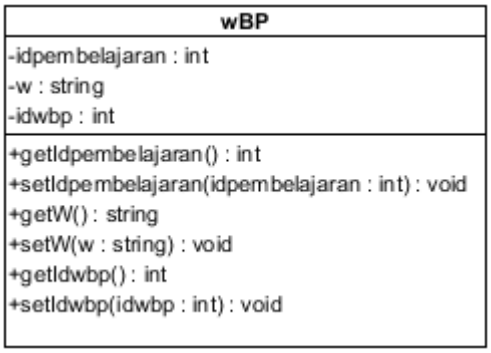
Class pengenalan merupakan class entity yang mewakili table pengenalan. Gambar III.40 menggambarkan perancangan class pengenalan.



Gambar III.40 Perancangan class pengenalan

7. wBP

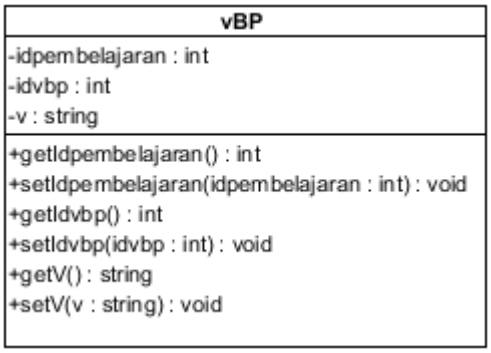
Class wBP merupakan class entity yang mewakili table wBP. Gambar III.41 menggambarkan perancangan class wBP.



Gambar III.41 Perancangan class wBP

8. vBP

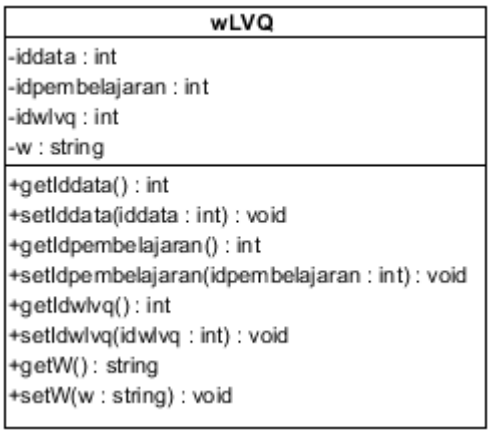
Class vBP merupakan class entity yang mewakili table vBP. Gambar III.42 menggambarkan perancangan class vBP.



Gambar III.42 Perancangan class vBP

9. wLVQ

Class wLVQ merupakan class entity yang mewakili table wLVQ. Gambar III.43 menggambarkan perancangan class wLVQ.



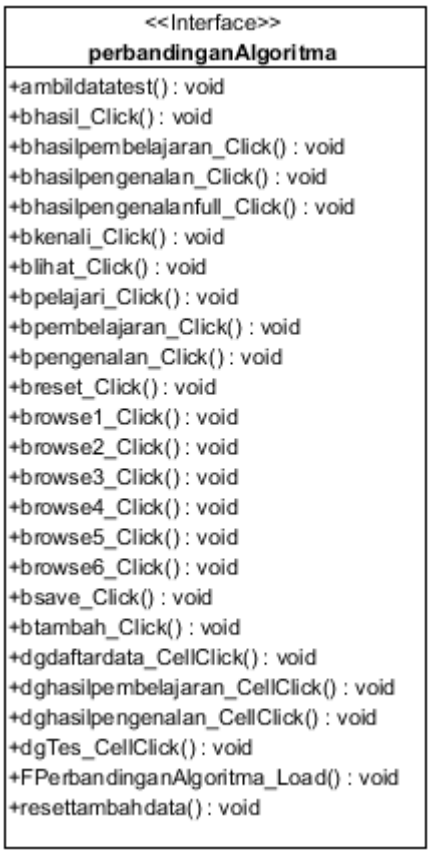
Gambar III.43 Perancangan class wLVQ

3.2.1.2 Perancangan Class Boundary (Interface)

Class boundary merupakan class yang berisi interface. Pada sistem ini terdapat 1 buah class boundary, yaitu FperbandinganAlgoritma.

1. **FPerbandinganAlgoritma**

   Class FperbandinganAlgoritma merupakan class interface yang menampilkan semua proses yang dijalankan dalam sistem ini. Gambar III.44 menggambarkan perancangan class FPerbandinganAlgoritma.



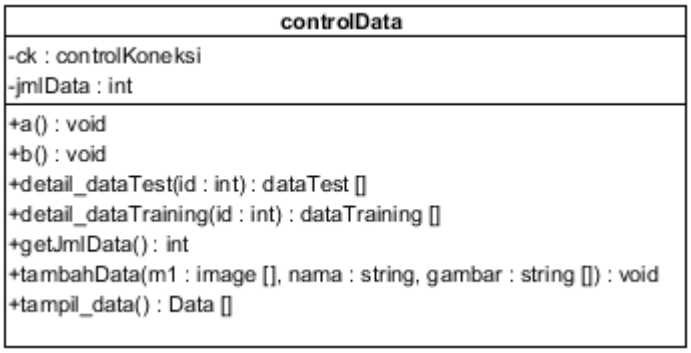
Gambar III.44 Class FPerbandinganAlgoritma

3.2.1.3 Perancangan Class Control

Class control merupakan class yang berisi fungsi-fungsi perhitungan dan penghubung ke database. Pada sistem ini terdapat 7 buah class control, yaitu controlData, controlPembelajaran, controlPengenalan, controlKoneksi, Backpropagation, LVQ, gambar.

1. **Class controlData**

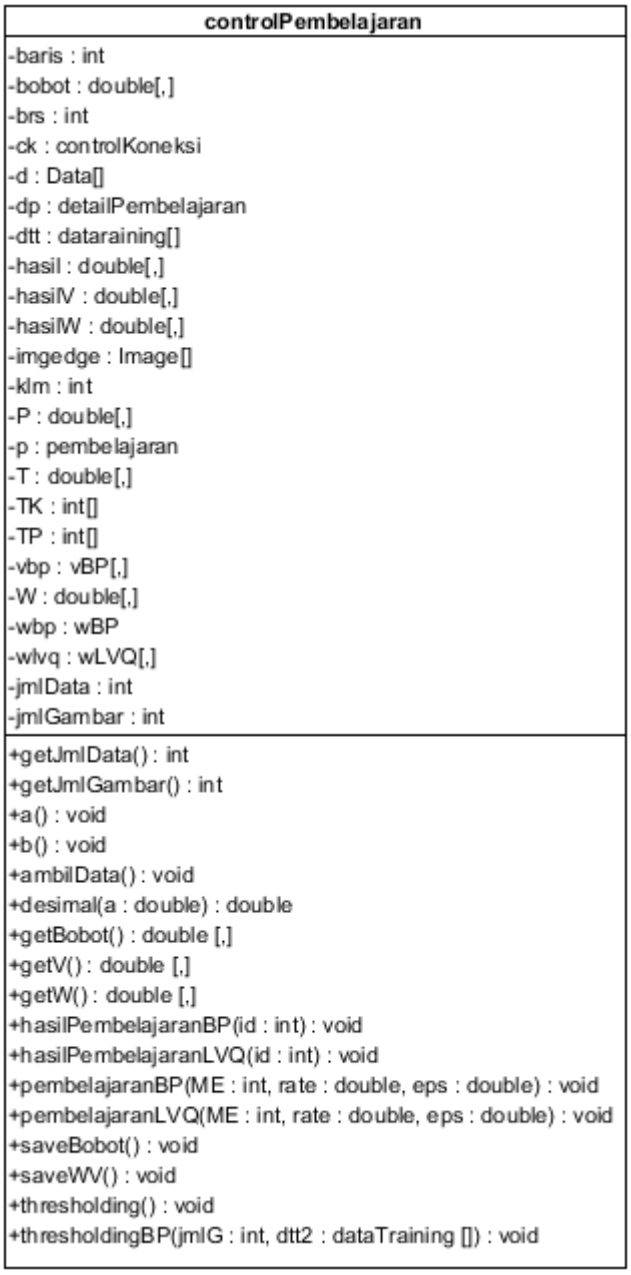
   Perancangan class controlData dijelaskan pada Gambar III.45.



Gambar III.45 Class controlData

2. **Class controlPembelajaran**

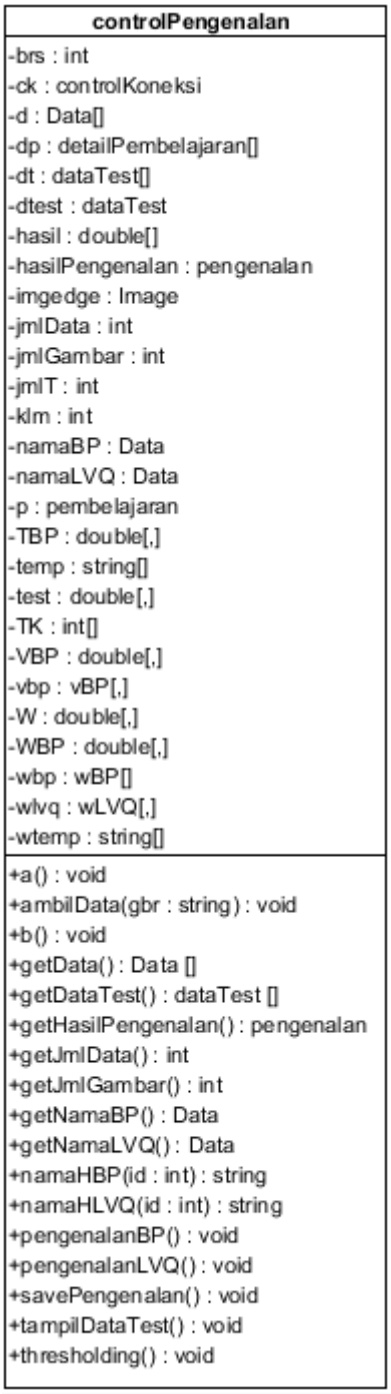
   Gambar III.46 Menjelaskan perancangan class controlPembelajaran.



Gambar III.46 Class controlPembelajaran

3. Class controlPengenalan

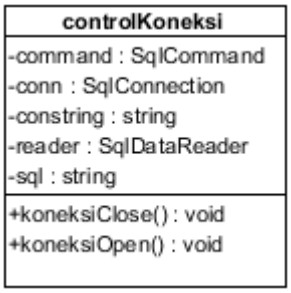
Perancangan class controlPengenalan digambarkan pada Gambar III.47.



Gambar III.47 Class controlPengenalan

4. Class controlKoneksi

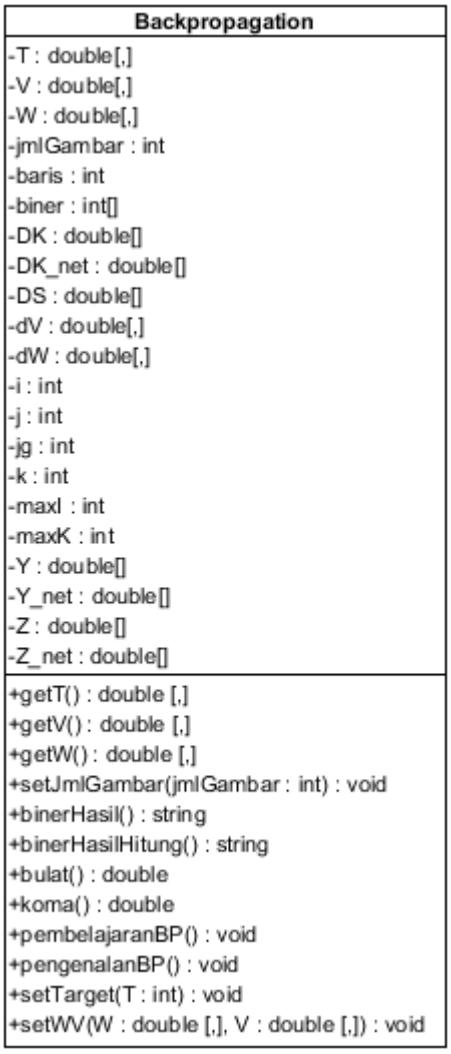
Perancangan class controlKoneksi digambarkan pada Gambar III.48



Gambar III.48 Class controlKoneksi

5. Class Backpropagation

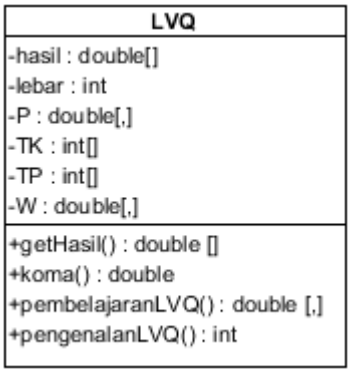
Perancangan class backpropagation dijelaskan pada Gambar III.49.



Gambar III.49 Class backpropagation

6. Class LVQ

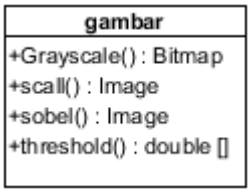
Perancangan class LVQ dijelaskan pada Gambar III.50.



Gambar III.50 Class LVQ

7. Class gambar

Perancangan class gambar dijelaskan pada Gambar III.51.



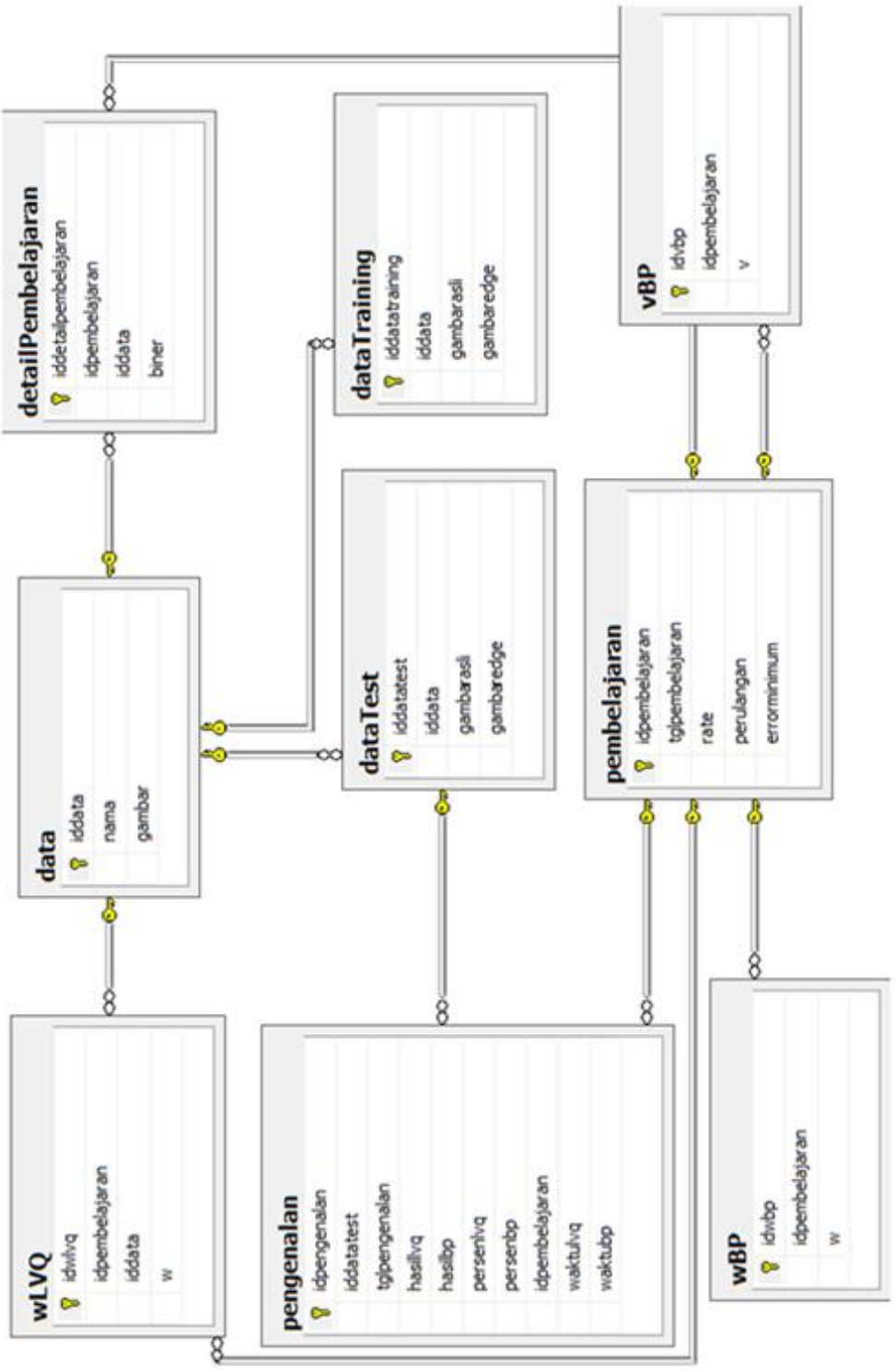
Gambar III.51 Class Gambar

3.2.2 Perancangan *Database*

Perancangan *database* menjelasankan basis data yang akan dibuat pada sistem ini.

3.2.3 Diagram Relasi

Diagram relasi menggambarkan relasi setiap tabel di dalam *database*, seperti terlihat pada Gambar III.52.



Gambar III.52 Diagram Relasi

3.2.4 Struktur File

Struktur file menjelaskan deskripsi setiap atribut yang dimiliki oleh setiap tabel yang digunakan pada sistem ini.

1. **Tabel data**

   Tabel data menyimpan iddata, nama dan gambar wajah dari setiap orang yang datanya digunakan dalam sistem ini. Struktur file tabel data dapat dilihat pada Tabel III.15.

Tabel III.15 Struktur File Tabel Data

Nama	Tipe	Panjang	Range	Keterangan
iddata	int	-	0..9	PK
nama	varchar	50	A..Z	
gambar	varchar	100	A..Z, 0..9	

2. **Tabel data*Test***

   Tabel data*Test* menyimpan iddata*test*, iddata, gambarasli dan gambaredge wajah untuk pengenalan dari setiap orang yang datanya digunakan dalam sistem ini. Struktur file tabel data*Test* dapat dilihat pada Tabel III.16.

Tabel III.16 Struktur File Tabel data*Test*

Nama	Tipe	Panjang	Range	Keterangan
iddatatest	int	-	0..9	PK
iddata	int	-	0..9	FK Ref data.iddata
gambarasli	varchar	100	A..Z,0..9	
gambaredge	varchar	100	A..Z, 0..9	

3. Tabel data*Training*

Tabel data*Training* menyimpan iddata*training*, iddata, gambarasli dan gambaredge wajah untuk pembelajaran dari setiap orang yang datanya digunakan dalam sistem ini. Struktur file tabel data*Training* dapat dilihat pada Tabel III.17.

Tabel III.17 Struktur File Tabel data*Training*

Nama	Tipe	Panjang	Range	Keterangan
iddata*training*	int	-	0..9	PK
iddata	int	-	0..9	FK Ref data.iddata
gambarasli	varchar	100	A..Z,0..9	
gambaredge	varchar	100	A..Z, 0..9	

4. Tabel pembelajaran

Tabel pembelajaran menyimpan idpembelajaran, tglpembelajaran dan jenispembelajaran dari setiap pembelajaran yang dilakukan. Struktur file tabel pembelajaran dapat dilihat pada Tabel III.18.

Tabel III.18 Struktur File Tabel pembelajaran

Nama	Tipe	Panjang	Range	Keterangan
idpembelajaran	int	-	0..9	PK
tglpembelajaran	datetime	-	A..Z,0..9	
rate	float	-	0..9	
perulangan	float	-	0..9	
errorminimum	float	-	0..9	

5. Tabel pengenalan

Tabel pengenalan menyimpan idpengenalan, iddata*test*, iddata, hasilLVQ, hasilBP, persenLVQ dan persenBP dari setiap pengenalan yang dilakukan. Struktur file tabel pengenalan dapat dilihat pada Tabel III.19.

Tabel III.19 Struktur File Tabel pengenalan

Nama	Tipe	Panjang	Range	Keterangan
idpengenalan	int	-	0..9	PK
iddata*test*	int	-	0..9	FK Ref data*Test*.iddata *test*
idPembelajaran	int	-	0..9	FK Ref pembelajaran.idPembelajaran
hasilLVQ	int	-	0..9	
hasilBP	int	-	0..9	
persenLVQ	float	-	0..9	
persenBP	float	-	0..9	
waktuLVQ	float	-	0..9	
waktuBP	float	-	0..9	

6. Tabel detailPembelajaran

Tabel detaiPembelajaran menyimpan iddetailpembelajaran, idpembelajaran, iddata dan biner setiap pembelajaran dengan jenis *backpropagation*. Struktur file tabel detailPembelajaran dapat dilihat pada Tabel III.20.

Tabel III.20 Struktur File Tabel detailPembelajaran

Nama	Tipe	Panjang	Range	Keterangan
iddetailpembelajaran	int	-	0..9	PK
idpembelajaran	int	-	0..9	FK ref pembelajaran.idpembelajaran
iddata	int	-	0..9	FK Ref data.iddata
biner	varchar	15	A..Z, 0..9	

7. Tabel wLVQ

Tabel wLVQ menyimpan idwlvq, idpembelajaran, iddata, dan wyang merupakan data bobot hasil pembelajaran LVQ. Struktur file tabel wLVQ dapat dilihat pada Tabel III.21.

Tabel III.21 Struktur File Tabel wLVQ

Nama	Tipe	Panjang	Range	Keterangan
idwlvq	int	-	0..9	PK
idpembelajaran	int	-	0..9	FK ref pembelajaran.idpembelajaran
iddata	int	-	0..9	FK Ref data.iddata
w	varchar	8000	0..9, #	

8. Tabel wBP

Tabel wBP menyimpan idwbp, idpembelajaran, dan w yang merupakan data bobot w hasil pembelajaran *Backpropagation*. Struktur file tabel wBP dapat dilihat pada Tabel III.22.

Tabel III.22 Struktur File Tabel wBP

Nama	Tipe	Panjang	Range	Keterangan
idwbp	int	-	0..9	PK
idpembelajaran	int	-	0..9	FK ref pembelajaran.idpembelajaran
w	varchar	8000	0..9, #	

9. Tabel vBP

Tabel vBP menyimpan idvbp, idpembelajaran, dan v yang merupakan data bobot v hasil pembelajaran *Backpropagation*. Struktur file tabel vBP dapat dilihat pada Tabel III.23.

Tabel III.23 Struktur File Tabel vBP

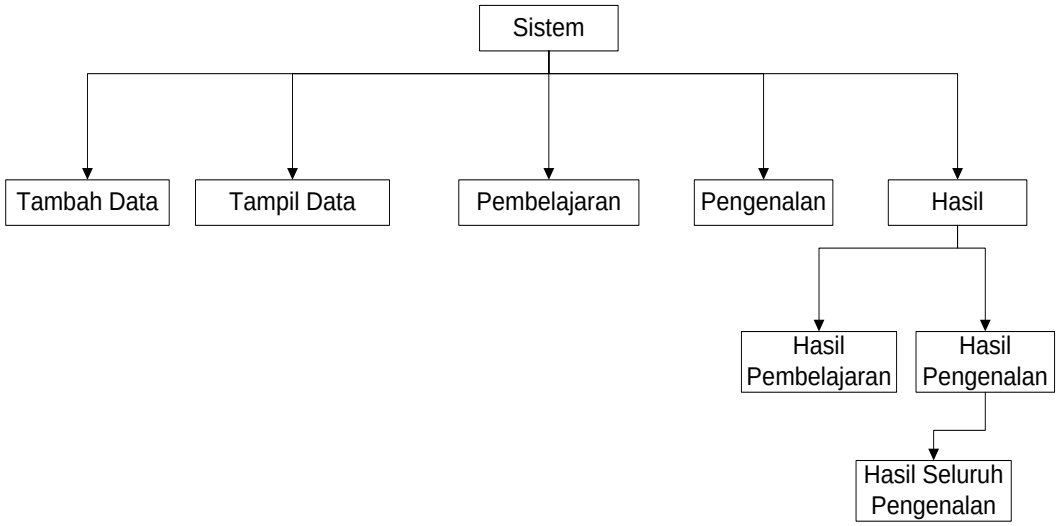
Nama	Tipe	Panjang	Range	Keterangan
idvbp	int	-	0..9	PK
idpembelajaran	int	-	0..9	FK ref pembelajaran.idpembelajaran
v	varchar	8000	0..9, #	

3.3 Perancangan Antarmuka

Perancangan Antarmuka menjelaskan antarmuka yang akan dibuat pada sistem ini. Perancangan antarmuka terdiri dari 3 sub judul yaitu perancangan menu, perancangan form dan perancangan pesan.

3.3.1 Perancangan Menu

Perancangan menu menjelaskan struktur menu yang akan dibuat pada suatu sistem. Gambar III.53 menunjukkan perancangan menu pada sistem ini.



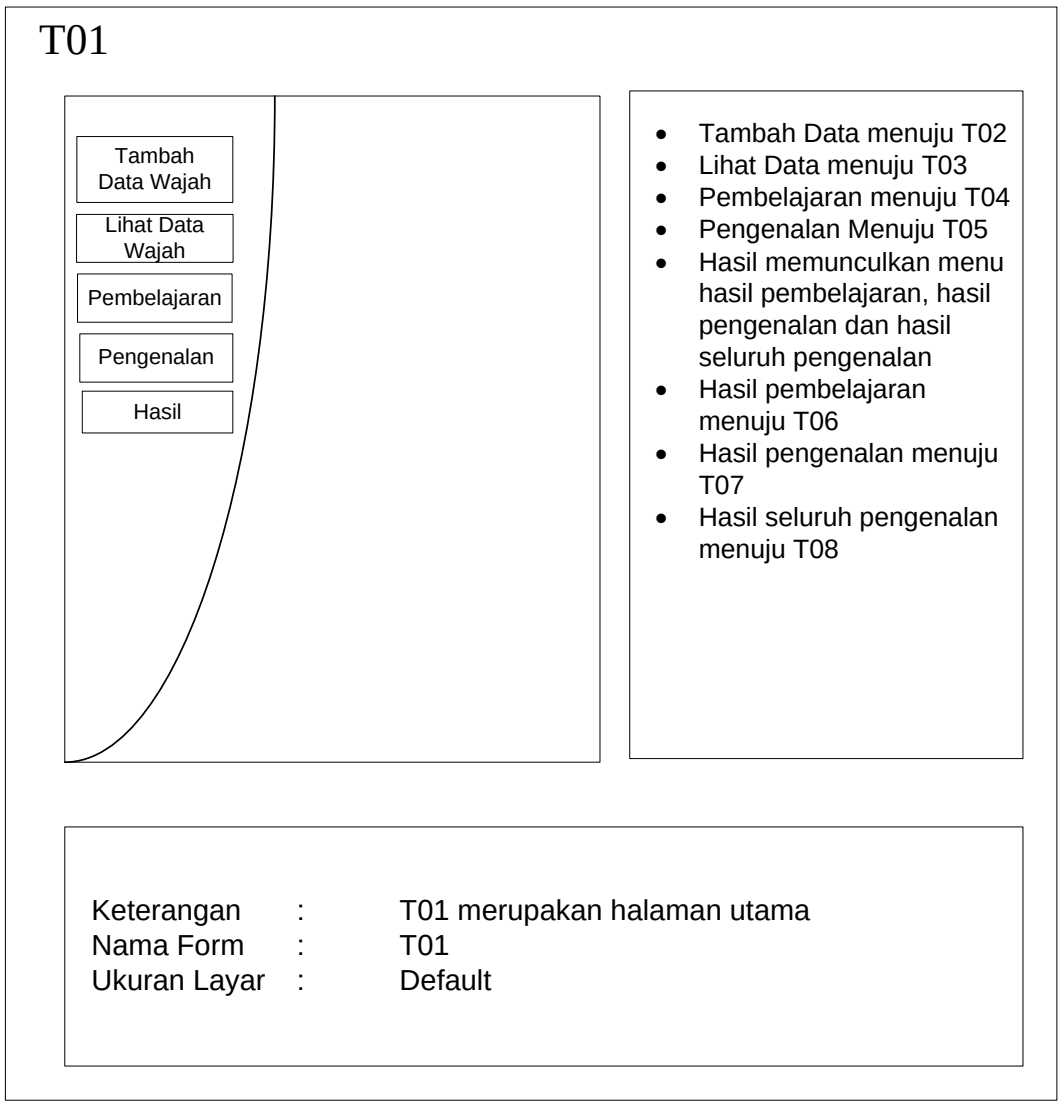
Gambar III.53 Struktur Menu Aplikasi Perbandingan Algoritma

3.3.2 Perancangan Form

Perancangan form dibagi menjadi lima rancangan antarmuka. Rancangan form tersebut yaitu rancangan form tampilan utama, rancangan form tambah data, rancangan form lihat data dan rancangan form layar ambil foto.

3.3.2.1 Rancangan Form Tampilan Utama

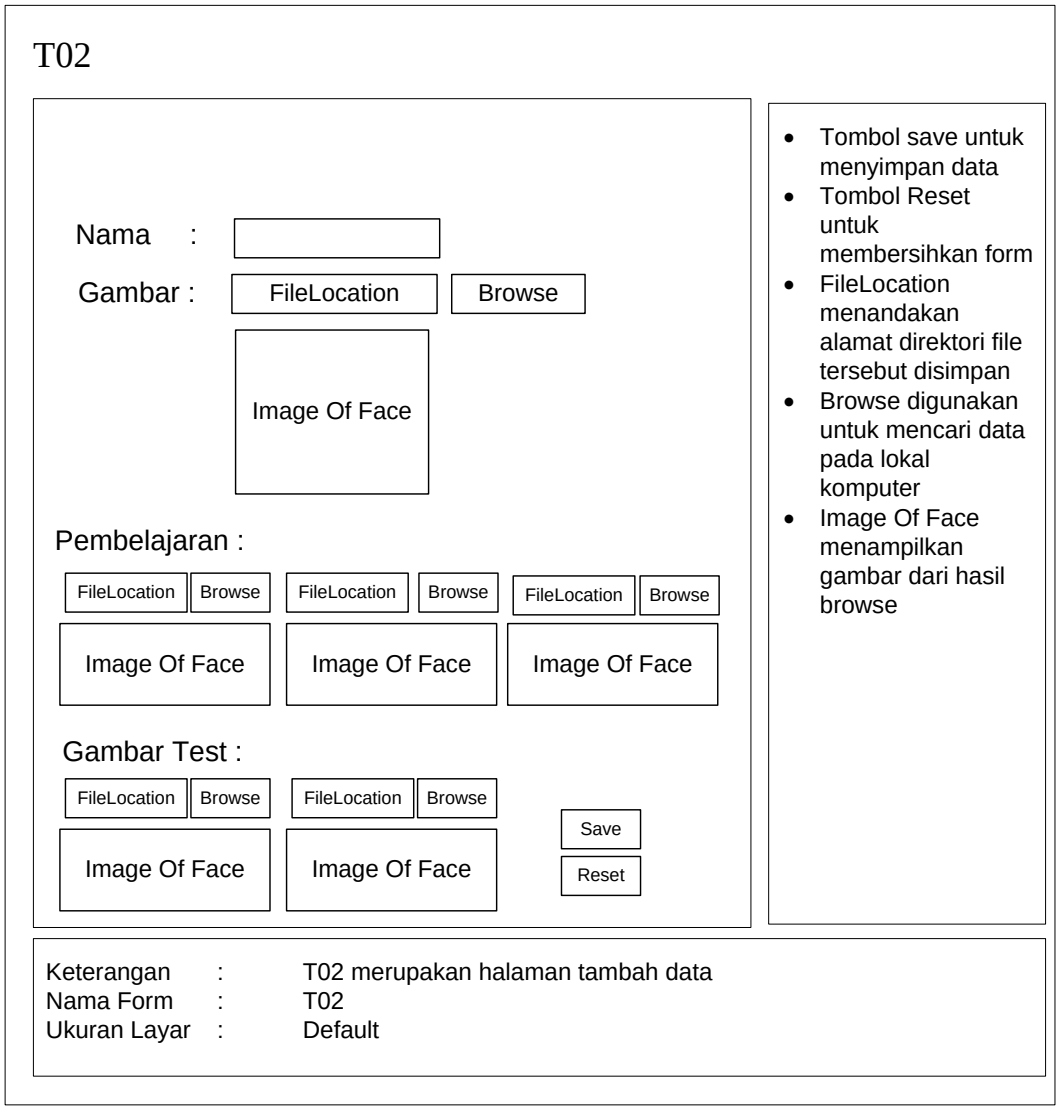
Rancangan form tampilan utama menampilkan tampilan awal program aplikasi. Tampilan ini menunjukan menu-menu yang dapat diakses oleh user. Rancangan form tampilan utama dapat dilihat pada Gambar III.54.



Gambar III.54 Rancangan Form Tampilan Utama

3.3.2.2 Rancangan Form Tambah Data

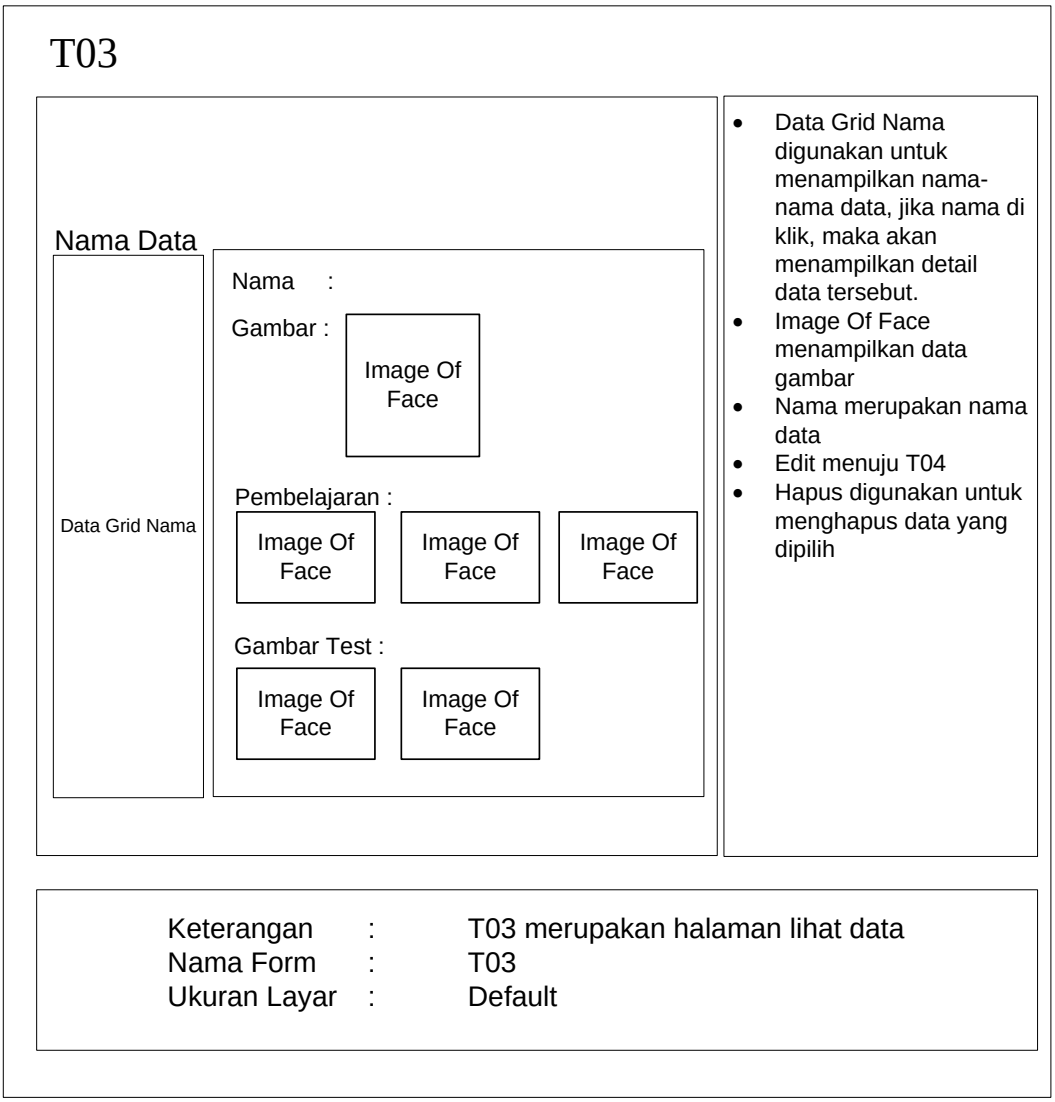
Rancangan form tambah data digunakan untuk menambah objek gambar wajah. Rancangan form tambah data dapat dilihat pada Gambar III.55.



Gambar III.55 Rancangan Form Tambah Data

3.3.2.3 Rancangan Form Lihat Data

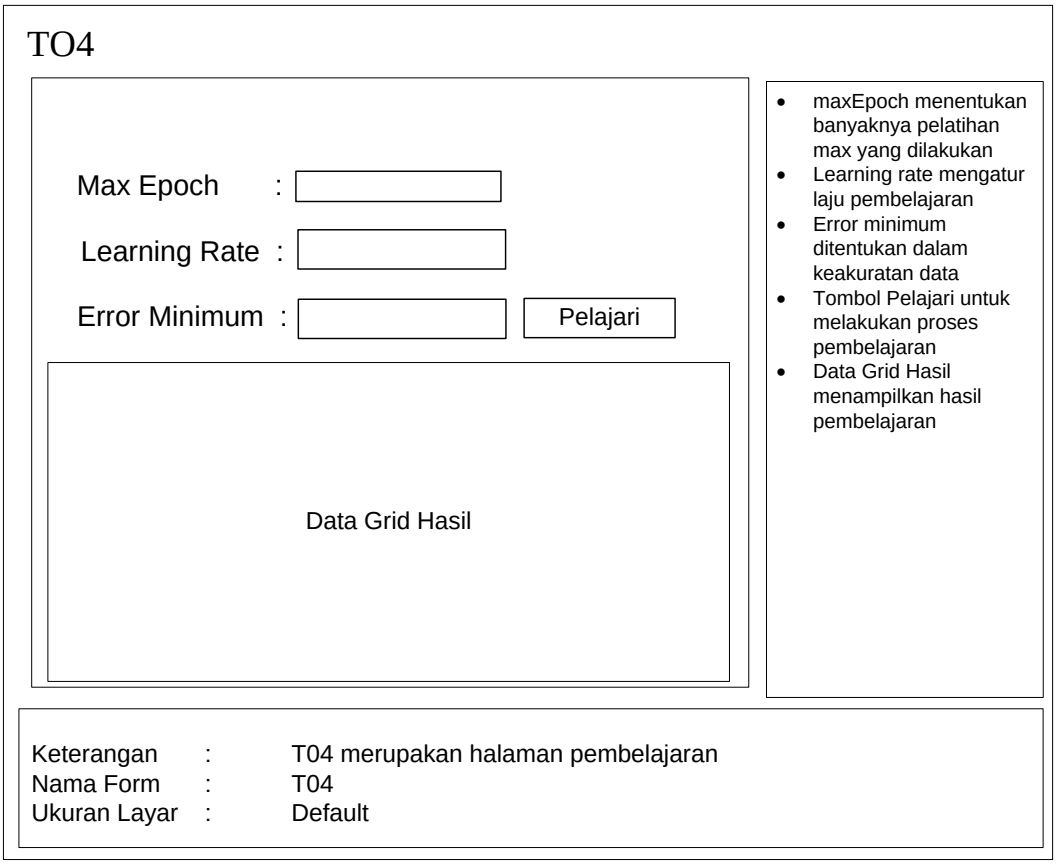
Rancangan form lihat data digunakan untuk menampilkan objek gambar yang sebelumnya telah dimasukkan. Rancangan form lihat data dapat dilihat pada Gambar III.56.



Gambar III.56 Rancangan Form Lihat Data

3.3.2.4 Rancangan Form Pembelajaran

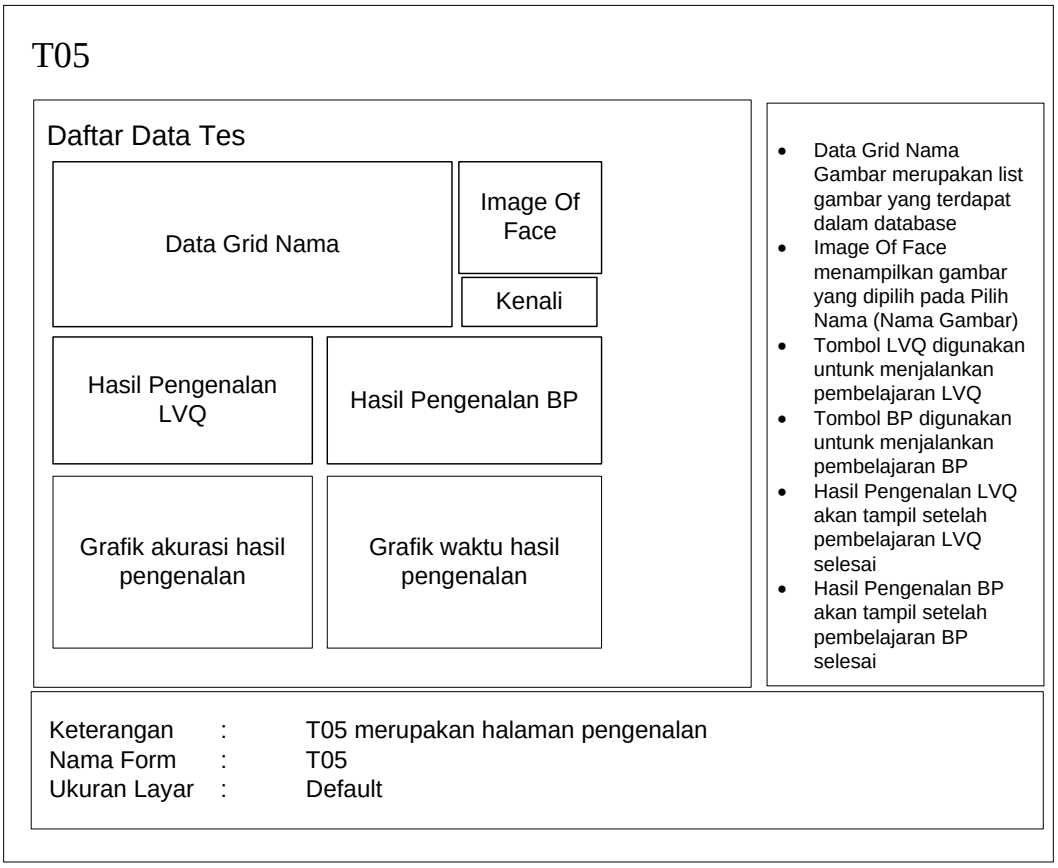
Rancangan form pembelajaran digunakan untuk memasukan parameter pembelajaran yang akan dilakukan. Pada rancangan form pembelajaran terdapat tombol “pelajari” untuk memulai pembelajaran. Rancangan form pembelajaran dapat dilihat pada Gambar III.57.



Gambar III.57 Rancangan Form Pembelajaran

3.3.2.5 Rancangan Form Pengenalan

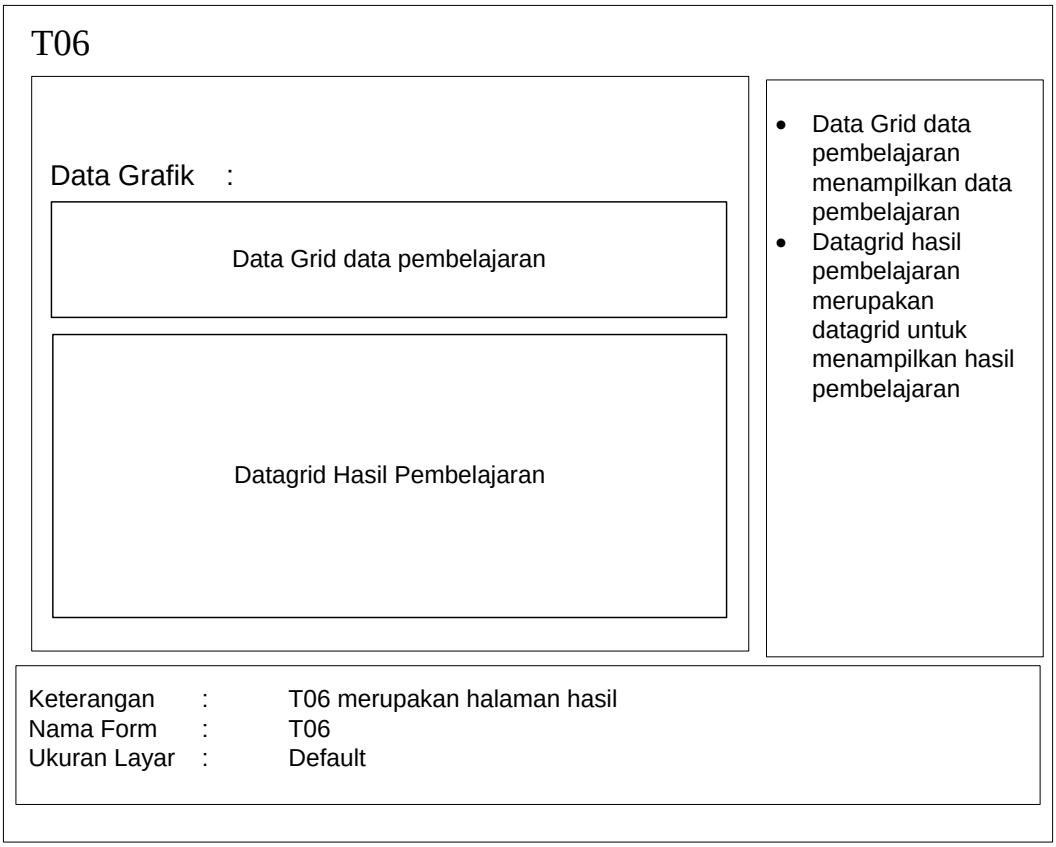
Rancangan form pengenalan digunakan untuk mengenali gambar sesuai pembelajaran yang dipiih. Pada rancangan form pengenalan terdapat tombol "Kenali" untuk melakukan pengenalan. Rancangan form dapat dilihat pada Gambar III.58.



Gambar III.58 Rancangan Form Pengenalan

3.3.2.6 Rancangan Form Hasil Pembelajaran

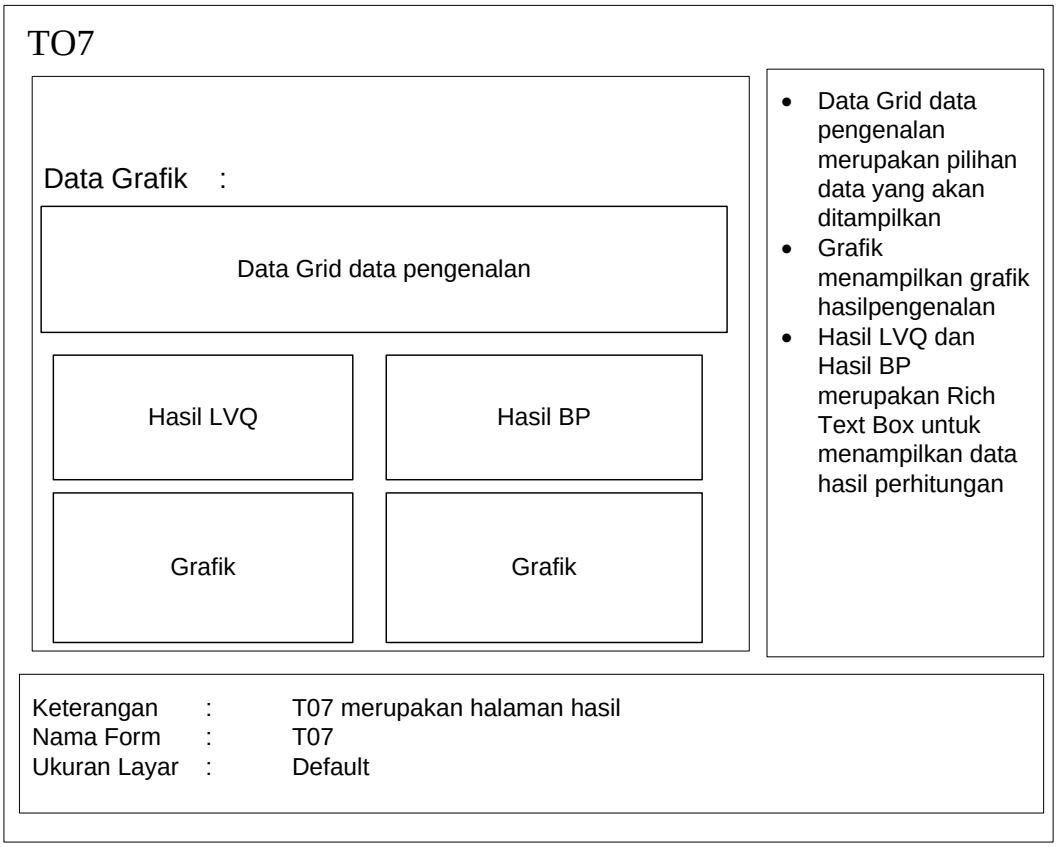
Rancangan form hasil digunakan untuk menampilkan hasil pembelajaran yang telah dilakukan. Pada rancangan form hasil pembelajaran terdapat datagrid yang menampilkan data hasil pembelajaran, klik salah satu data untuk melihat detail hasil. Rancangan form dapat dilihat pada Gambar III.59.



Gambar III.59 Rancangan Form Hasil Pembelajaran

3.3.2.7 Rancangan Form Hasil Pengenalan

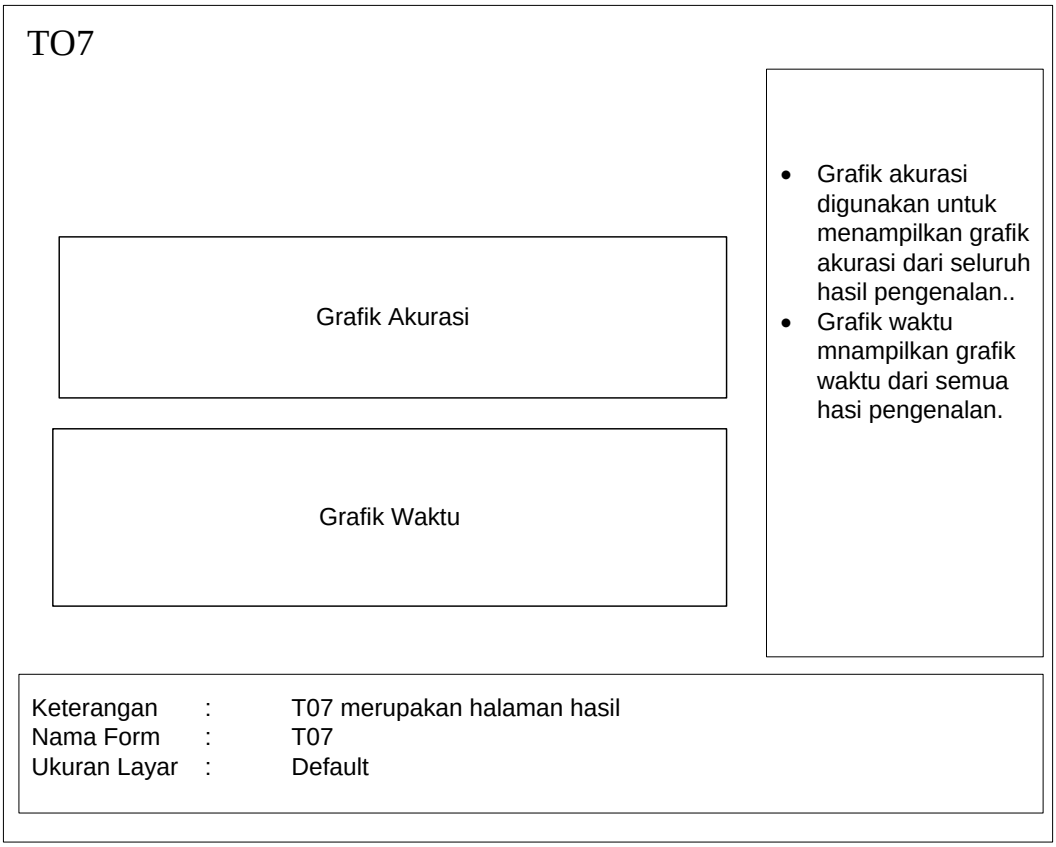
Rancangan form hasil digunakan untuk menampilkan hasil pengenalan yang telah dilakukan. Pada rancangan form hasil pengenalan terdapat datagrid yang menampilkan data pengenalang, klik salah satu data untuk melihat detail hasil pengenalan. Rancangan form dapat dilihat pada Gambar III.60.



Gambar III.60 Rancangan Form Hasil Pengenalan

3.3.2.8 Rancangan Form Hasil SeluruhPengenalan

Rancangan form hasilseluruh pengenalan digunakan untuk menampilkan grafik persentase akurasi dan waktu hasil pengenalan yang telah dilakukan. Rancangan form dapat dilihat pada Gambar III.61.



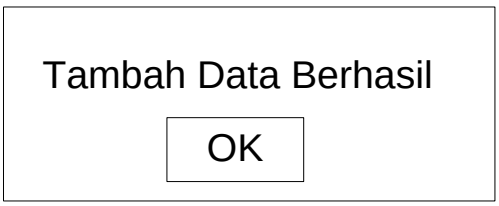
Gambar III.61 Rancangan Form Hasil Sekuruh Pengenalan

3.3.3 Perancangan Pesan

Pesan digunakan untuk memberitahu bahwa sebuah proses telah selesai dilakukan atau terdapat kesalahan dalam masukan proses.

1. **Pesan Tambah Data Berhasil**

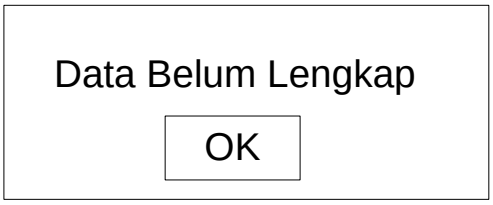
   Pesan tambah data berhasil digunakan untuk memberikan informasi bahwa proses tambah data berhasil. Perancangan pesan tambah data berhasil ditunjukkan pada Gambar III.62



Gambar III.62 Pesan Tambah Data Berhasil

2. Pesan Tambah Data Gagal

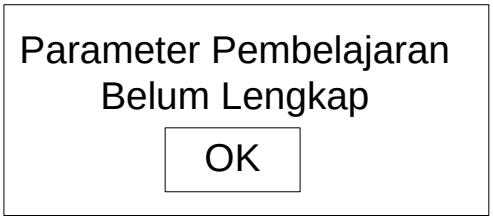
Pesan tambah data gagal digunakan untuk memberikan informasi bahwa proses tambah data gagal. Perancangan pesan tambah data gagal ditunjukkan pada Gambar III.63.



Gambar III.63 Pesan Tambah Data Gagal

3. Pesan Pembelajaran Gagal

Pesan tambah data gagal digunakan untuk memberikan informasi bahwa proses tambah data gagal. Perancangan pesan tambah data gagal ditunjukkan pada Gambar III.64.



Gambar III.64 Pesan Pembelajaran Gagal