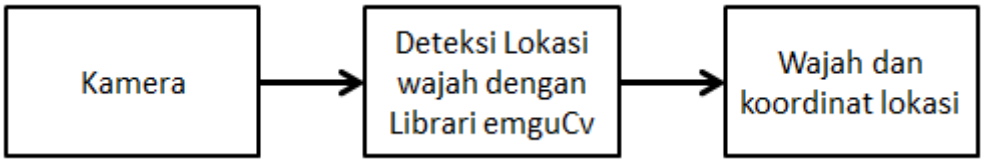
# BAB III
# ANALISIS DAN PERANCANGAN

## III.1 Gambaran Umum Sistem

Secara umum sistem pengenalan wajah ini mempunyai beberapa tahapan yaitu pendeteksian wajah, ekstraksi wajah, dan pengenalan wajah. Pendeteksian wajah adalah tahap untuk menentukan dan memisahkan bagian dari suatu citra diam (*still image*) yang terdefinisi sebagai wajah, dan ekstraksi wajah merupakan pengambilan bagian dari fitur-fitur wajah. Data *learning* (data yang digunakan untuk 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).

Untuk pendeteksian wajah ini menggunakan fungsi dari *haarClasified* yang menjadi library dari emguCv. Hasil dari deteksi wajah berupa informasi koordinat lokasi yang dapat digunakan pada proses selanjutnya yaitu untuk *resize* dan menyimpan ke dalam file gambar baru yang akan digunakan untuk mendeteksi fitur-fitur pada wajah, seperti pada Gambar III.1.




Gambar III.1 Diagram deteksi wajah

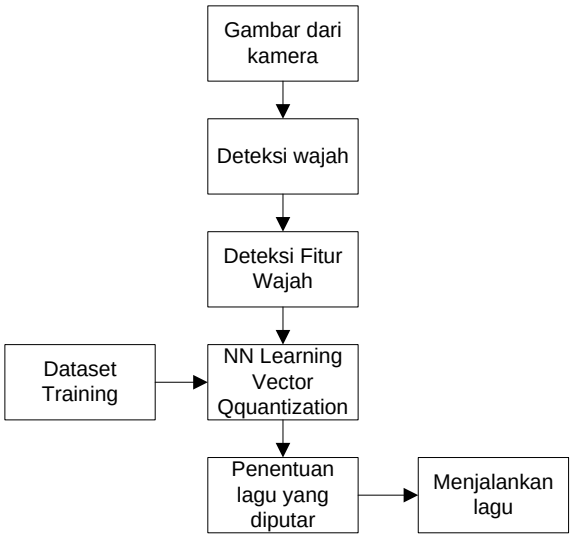
Pada tahap pertama akan dilakukan pengambilan gambar atau *capture* gambar dari kamera dan akan mendeteksi wajah dari gambar tersebut untuk selanjutnya dilakukan *cropping* area wajah saja. Karena untuk mempermudah pada saat proses deteksi fitur wajah.

Selanjutnya proses deteksi fitur wajah, yaitu proses mendeteksi fitur-fitur wajah seperti alismata, mata, dan mulut. Yang kemudian ditentukan koordinat

dari masing-masing fitur tersebut. Koordinat yang didapat kemudian akan dilakukan perhitungan untuk mendapatkan nilai fitur yang dibutuhkan, seperti nilai fitur jarak titik pusat kedua alismata dengan titik pusat kedua mata (F1) , lebar mata (F2) , lebar mulut (F3) , jarak kedua alismata (F4) , jarak kedua mata (F5) , dan jarak titik pusat kedua alismata dengan titik pusat mulut (F6). Yang nantinya itu kan dijadikan sebagai inputan dalam proses *learning vector quantization.*

Pada proses *learning vector quantization* yaitu terdapat dua fungsional *training* dan *testing*. Pada proses *training* data masukan akan dilakukan proses pembelajaran untuk mendapatkan nilai bobot (w) akhir yang bertujuan sebagai acuan pada proses *testing*.

Setelah proses sebelumnya telah berhasil maka proses *testing* akan dilakukan untuk mengenali ekspresi wajah. Jika ekspresi wajah berhasil dikenali maka secara otomatis sistema akan menjalankan musik. Untuk gambaran sistemnya dapat dilihat pada Gambar III.2.




Gambar III.2 Diagram Sistem

## III.2 Analisis

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.

### III.2.1 Analisis Ekspresi

Ekspresi wajah atau mimik adalah hasil dari satu atau lebih gerakan atau posisi otot pada wajah. Ekspresi wajah merupakan salah satu bentuk komunikasi, dan dapat menyampaikan keadaan emosi dari seseorang kepada orang yang mengamatinya.

Ada beberapa ciri khusus dari macam-macam ekspresi wajah. Berdasarkan jurnal yang berjudul pemilihan playlist dengan mengenali ekspresi wajah pengguna menggunakan neural network backpropagation menjelaskan bahwa ada beberapa ciri dari ekspresi wajah. Seperti pada ekspresi marah posisi alismata dan mata posisinya berdekatan, kedua mata melebar, dan jarak alismata mendekat. Sedangkan pada ekspresi senang tinggi mulut semakin melebar.

### III.2.2 Analisis Data Inputan

Data inputan yang akan diambil dalam sistem ini adalah sebuah citra yang diambil melalui *webcam* secara *real time*. Namun hanya fitur wajah saja yang digunakan sebagai inputan untuk menjalankan sistem ini. Dengan posisi wajah menghadap lurus ke *webcam*.

Citra wajah diambil dengan menggunakan *webcam* sebagai citra *real time*. Dari citra wajah nanti akan dideteksi fitur-fiturnya seperti mulut, mata, dan alismata menggunakan fungsi *haarClasified* yang merupakan *librery* dari emguCv. Dari deteksi tersebut nanti akan didapat koordinat-koordinat dari masing-masing fitur tersebut.

### III.2.3 Analisis Metode

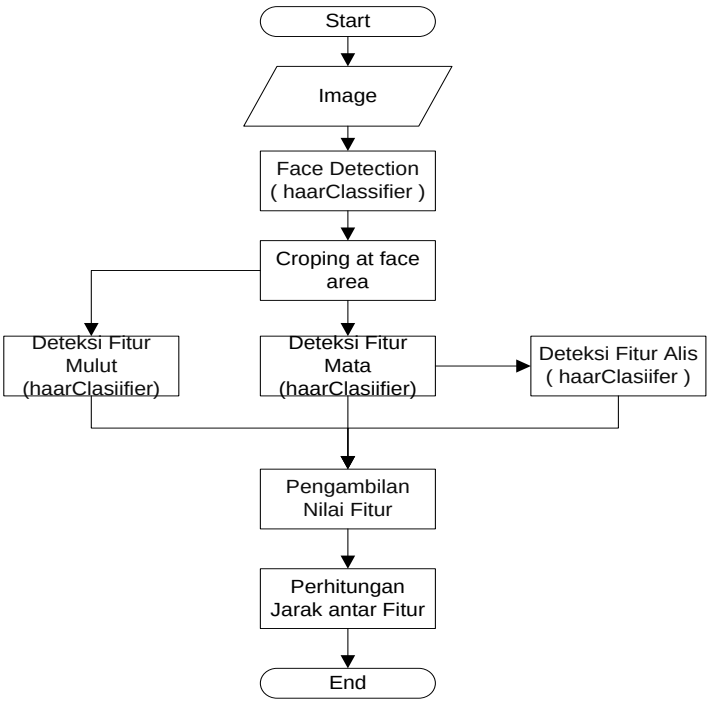
Analisis metode menjelaskan tentang cara pengerjaan atau cara perhitungan dari metode yang digunakan dalam sistem ini.

#### III.2.3.1 Menghitung Nilai Fitur

Mendeteksi ekspresi wajah dengan menghitung perbedaan nilai fitur dan jarak fitur, maka sistem ini dibangun agar mampu mengenali dan mengambil nilai-nilai fitur yang digunakan dalam deteksi wajah (menentukan fitur-fitur wajah), ekstraksi nilai-nilai fitur wajah, ada fitur utama dari wajah yang dianggap mempengaruhi perubahan ekspresi wajah, yaitu:

1. Fitur mulut
2. Fitur mata
3. Fitur alismata

Secara umum kerja sistem untuk mendapatkan nilai fitur dapat dilihat pada Gambar III.3.

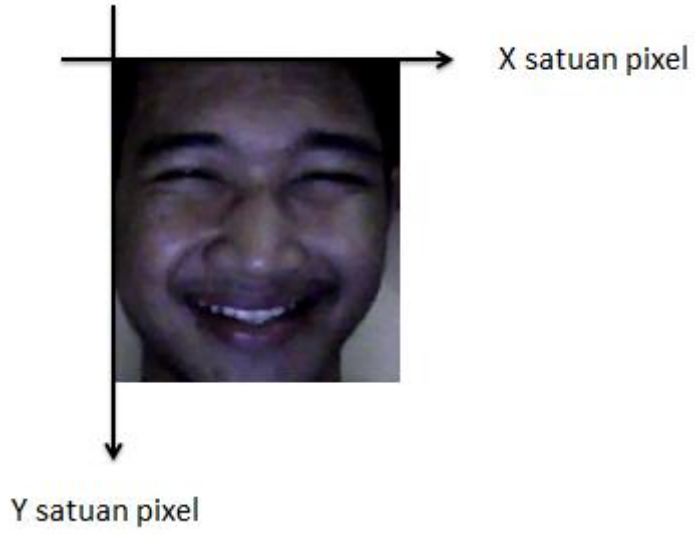



Gambar III.3 Diagram Proses Pengambilan Nilai

Berdasarkan deteksi fitur tersebut, dapat diambil 6 nilai fitur yang berpengaruh dari ekspresi wajah:

1. Jarak alis dengan mata = F1
2. Lebar mata = F2
3. Lebar mulut = F3
4. Jarak kedua alis mata = F4
5. Jarak kedua mata = F5
6. Jarak alis mata dengan mulut = F6

Setelah pengenalan lokasi wajah dilakukan pencarian nilai fitur dari wajah dengan memanfaatkan fungsi *haarClasified* yang telah dibuat untuk deteksi fitur mulut, mata, dan alismata. Dari fitur yang didaptkan berupa rectangular posisi dari fitur tersebut, dapat diambil nilai koordinat dari titik tepinya dan dihitung nilai fiturnya berdasarkan titik koordinat tersebut. Di mana posisi koordinat terhitung dimulai dari sisi pojok kiri atas ( perhatikan Gambar III.4 ).

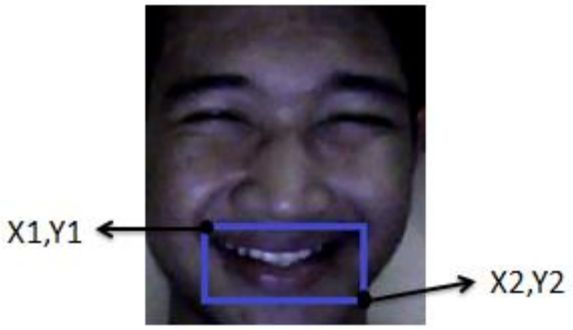



Gambar III.4 Sistem koordinat

Di mana nilai koordinat yang diperoleh dari masing – masing fitur adalah sebagai berikut :

1. Fitur Mulut

Seperti yang terlihat pada Gambar III.5 deteksi mulut menghasilkan nilai koordinat x1,y1,x2,dan y2 sehingga dapat dihitung nilai fitur lebar mulut (F3) dengan perbedaan nilai koordinat pada sumbu y.




Gambar III.5 Fitur Mulut

Misalkan x1= 3; y1= 15; x2= 10; y2= 18

Maka, untuk menghitung nilai fitur lebar mulut (F3) adalah sebagai berikut:

F3 = Y2 – Y1

F3=18-15

F3=3

Jadi, untuk nilai fitur lebar mulut (F3) adalah 3

Kemudian menghitung koordinat titik tengah mulut, untuk menghitung nilai fitur yang lainnya.

Menghitung koordinat titik tengah X dari mulut:

titikTengahMulutX = ( (X2 - X1) / 2 ) + X1

titikTengahMulutX= ( (10 - 3) / 2 + 3)

titikTengahMulutX= 3.5+3

titikTengahMulutX= 6.5

Menghitung koordinat titik tengah Y dari mulut:

titikTengahMulut Y = ((Y2 - Y1) / 2 ) + Y1

titikTengahMulutY = ((18-15) / 2 ) + 15

titikTengahMulutY = 1.5 + 15

titikTengahMulutY = 16.5

Jadi, koordinat titik tengah mulut adalah (titikTengahMulutX, titikTengahMulutY) atau (6.5 , 16.5).

2. Fitur mata

Pada Gambar III.6 fitur mata dikenali 2 buah rectangle yang menyatakan posisi dari mata kiri dan mata kanan, sehingga nilai koordinat nya disimpan dalam array (mataX1[],mataY1[],mataX2[], mataY2[]).




Gambar III.6 Fitur Mata

Misalkan:

mataX1[1] = 1 mataY1[1] = 5

mataX1[2] = 4 mataY1[2] = 7

mataX2[1] = 8 mataY2[1] = 6

mataX2[2] = 12 mataY2[2] = 7

maka, untuk menghitung nilai fitur lebar mata (F2) adalah sebagai berikut:

$$F2 = \frac{(mataY1[2] - mataY1[1]) + (mataY2[2] - mataY2[1])}{2}$$

$$F2 = \frac{(7 - 5) + (7 - 6)}{2}$$

$$F2 = \frac{2 + 1}{2}$$

$$F2 = 1.5$$

Jadi, untuk nilai fitur lebar mata (F2) adalah 1,5.

Selanjutnya mencari titik tengah mata1 dan mata 2 untuk menghitung jarak antara kedua mata (F5).

Mencari koordinat titik tengah mata pertama

a. Sumbu x

centermataX1 = ((mataX1[2]-mataX1[1])/2)+mataX1[1]

centermataX1 = ((4 – 1)/2)+1

centermataX1 = 1.5+1

centermataX1 = 2.5

b. Sumbu y

centermataY1 = ((mataY1[2]-mataY1[1])/2)+mataY1[1]

centermataY1 = ((7 – 5) / 2) + 5

centermataY1 = 1 + 5

centermataY1 = 6

Sehingga koordinat titik tengah mata pertama adalah (2.5 , 6).

Selanjutnya mencari koordinat titik tengah mata kedua

c. Sumbu x

centermataX2 = ((mataX2[2]-mataX2[1])/2)+mataX2[1]

centermataX2 = ((12 – 8) / 2) + 8

centermataX2 = 2 + 8

centermataX2 = 10

d. Sumbu y

centermataY2 = ((mataY2[2]-mataY2[1])/2)+mataY2[1]

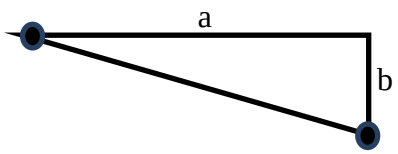
centermataY2 = ((7 – 6) / 2) + 6

centermataY2 = 0.5 + 6

centermataY2 = 6.5

Sehingga titik tengah mata kedua adalah (10 , 6.5).

Karena posisi titik tengah belum tentu dalam kesejajaran , maka perhitungan jarak mata menggunakan *pytagoras*. Sehingga untuk mencari nilai fitur jarak antara kedua mata (F5) adalah sebagai berikut:




a. Mencari nilai a

$$a = |centermataX2 - centermataX1|$$

$$a = |10 - 2.5|$$

$$a = |7.5|$$

b. Mencari nilai b

$$b = |centermataY2 - centermataY1|$$

$$b = |6.5 - 6|$$

$$b = |0.5|$$

c. Mencari nilai fitur f5

$$F5 = \sqrt{a^2 + b^2}$$

$$F5 = \sqrt{7.5^2 + 0.5^2}$$

$$F5 = \sqrt{56.25 + 0.25}$$

$$F5 = \sqrt{56.5}$$

$$F5 = 7.52$$

Jadi, untuk nilai fitur jarak antara kedua mata (F5) adalah 7,52.

Kemudian mencari koordinat titik tengah kedua mata untuk menghitung jarak alismata dengan mata.

a. Sumbu x

centermataX= ((centermataX2 – centermataX1)/2)+ centermataX1

centermataX = ((10 – 2.5) / 2) +2.5

centermataX = 3.75 + 2.5

centermataX = 6.25

b. Sumbu y

centermataY = ((centermataY2 – centermataY1)/2)+ centermataY1

centermataY = ((6.5 – 6) / 2) + 6

centermataY = 0.25 + 6

centermataY = 6.25

Jadi, untuk koordinat titik tengah kedua mata adalah (6.25 , 6.25).

3. Fitur alismata

Penentuan fitur alis berdasarkan posisi dari fitur posisi mata , sehingga nilai fitur alis bergantung dari deteksi fitur mata. Berdasarkan Gambar III.7 koordinat pixel lokasi alis yang didapat akan dicari nilai dari kordinat x terendah dan x tertinggi (menyatakan titik ujung alis ), yaitu alisX1[], alisY1[],alisX2[],alisY2[].



Gambar III.7 fitur alismata

Misalkan:

alisX1[1] = 1 alisY1[1] = 1

alisX1[2] = 5 alisY1[2] = 2

alisX2[1] = 7 alisY2[1] = 1

alisX2[2] = 13 alisY2[2] = 2

Menentukan nilai fitur jarak antara alismata dengan mata (F1), jarak antara kedua alismata (F4), dan jarak antara alismata dengan mulut (F6) dengan menentukan nilai centeralisX dan centeralisY terlebih dahulu. Maka untuk menentukan nilai koordinatnya adalah sebagai berikut:

1. Menentukan koordinat titik tengah darai masing-masing alismata terlebih dahulu.
   a. Titik tengah alismata pertama

      Sumbu X

      centeralisX1 = ((alisX1[2]-alisX1[1])/2) + alisX1[1]

      centeralisX1 = ((5 – 1)/2) +1

      centeralisX1 = 2 +1

      centeralisX1 = 3

      Sumbu Y

      centeralisY1 = ((alisY1[2]-alisY1[1])/2) + alisY1[1]

      centeralisY1 = ((2 – 1)/2)+1

      centeralisY1 = 0.5 +1

centeralisY1 = 1.5

Sehingga untuk koordinat titik tengah alis pertama adalah (3 , 1.5).

b. Titik tengah alismata kedua

Sumbu X

centeralisX2 = ((alisX2[2]-alisX2[1])/2) + alisX2[1]

centeralisX2 = ((13 – 7) / 2) +7

centeralisX2 = 3 + 7

centeralisX2 = 10

Sumbu Y

centeralisY2 = ((alisY2[2]-alisY2[1])/2) + alisY2[1]

centeralisY2 = ((2 – 1) / 2) +1

centeralisY2 = 0.5 +1

centeralisY2 = 1.5

Sehingga untuk koordinat titik tengah alismata kedua adalah (10 , 1.5).

2. Menentukan titik tengah antara dua alismata tersebut

Sumbu X

centerduaalisX = ((centeralisx2 – centeralisx1)/2) + centeralisx1

centerduaalisX = ((10 – 3)/2) +3

centerduaalisX = 3.5 + 3

centerduaalisX = 6.5

Sumbu Y

centerduaalisY = ((centeralisy2 – centeralisy1)/2) + centeralisy1

centerduaalisY = ((1.5 – 1.5)/2) + 1.5

centerduaalisY = 0 + 1.5

centerduaalisY = 1.5

Sehingga untuk koordinat dua alismata tersebut adalah (6.5 , 1.5).

3. Menghitung nilai fitur lebar alis dengan mata (F1)

$$F1 = \sqrt{(\text{centeralisx} - \text{centermatax})^2 + (\text{centeralisy} - \text{centermatay})^2}$$

$$F1 = \sqrt{(6.5 - 6.25)^2 + (1.5 - 6.25)^2}$$

$$F1 = \sqrt{(0.25)^2 + (-4.75)^2}$$

$$F1 = \sqrt{0.06 + 22.56}$$

$$F1 = \sqrt{22.62}$$

$$F1 = 4.76$$

Jadi, untuk nilai fitur lebar alismata dengan mata (F1) adalah 4.76.

4. Menghitung nilai fitur jarak antara kedua alismata (F4)

$$F4 = \sqrt{(\text{centeralisx2} - \text{centeralisx1})^2 + (\text{centeralisy2} - \text{centeralisy1})^2}$$

$$F4 = \sqrt{(10 - 3)^2 + (1.5 - 1.5)^2}$$

$$F4 = \sqrt{(7)^2 + (0)^2}$$

$$F4 = \sqrt{49 + 0}$$

$$F4 = 7$$

Jadi, untuk nilai fitur jarak antara kedua alismata (f4) adalah 7

5. Menghitung nilai fitur jarak antara alismata dengan mulut (F6)

$$F6 = \sqrt{(\text{titikTengahMulutX} - \text{centeralisX})^2 + (\text{titikTengahMulutY} - \text{centeralisY})^2}$$

$$F6 = \sqrt{(6.5 - 6.5)^2 + (16.5 - 1.5)^2}$$

$$F6 = \sqrt{(0)^2 + (15)^2}$$

$$F6 = \sqrt{0 + 225}$$

$$F6 = 15$$

Jadi, untuk nilai fitur jarak antara alismata dengan mulut (F6) adalah 15.

Dari proses perhitungan nilai fitur di atas didapatkan 6 layer input yaitu jarak alismata dengan titik pusat mata (F1 = 4.76) , lebar mata (F2 = 1.5) , lebar mulut (F3 = 3) , jarak kedua alismata (F4 = 7) , jarak kedua mata (F5 = 7.52) , dan jarak titik pusat kedua alismata dengan titik pusat mulut (F6 = 15). Nilai-nilai fitur tersebut yang nantinya akan menjadi nilai input dalam perhitungan *learning vector quantization.*

### III.2.3.2 *Training Learning Vector Quantization*

Proses pembelajaran dengan metode *learning vector quantization* menggunakan 1 alur perhitungan yaitu alur maju.

Berikut tahapan pada proses pembelajaran LVQ terhadap data *training* tersebut:

1. Pada contoh perhitungan ini terdapat 3 buah kelas, yaitu kelas 1 untuk ekspresi "Senyum", kelas 2 untuk ekspresi "Normal", dan kelas 3 ekspresi "Marah". Setiap kelas (kelas 1, kelas 2, dan kelas 2) diambil salah satu array vektornya dan dijadikan sebagai inisialisasi bobot (W). Sedangkan lainnya menjadi data masukan (X). Hasilnya terlihat seperti pada Tabel III-1.

Tabel III-1 Inisialisasi dan pembelajaran LVQ

| Inisialisasi nilai bobot (W) | | | |
|---|---|---|---|
| Ekspresi | Kelas | Nilai fitur (W1,W2,W3,W4,W5,W6) | Inisialisasi Nilai W |
| Senyum | 1 | (4.76, 1.5, 3, 7, 7.52, 15) | Bobot (W) |
| Normal | 2 | (2.51, 3.9, 4, 11.5, 10.01, 16.34) | Bobot (W) |
| Marah | 3 | (3.86, 3.7, 5, 16.02, 10.5, 18.37) | Bobot (W) |
| Data training | | | |
| Huruf | Kelas | Nilai fitur (f1,f2,f3,f4,f5,f6) | Data masukan (X) |
| Senyum | 1 | (5.27, 4.8, 5, 11.09, 10.8, 29.91) | X1 |
| Normal | 2 | (2.36, 3, 5, 13.6, 9.8, 17.2) | X2 |
| Marah | 3 | (1.73, 4, 4, 11.4, 10.2, 16.51) | X3 |

2. Tentukan nilai maksimum perulangan, *error minimum*, rasio pembelajaran dan pengurangan rasio. Pada contoh ini digunakan batasan-batasan yang diambil dari penelitian yang telah dilakukan oleh Wuryandari [1], dengan batasan sebagai berikut:

Maksimum perulangan = 10

Eror minimum = 0,1

Rasio pembelajaran = 0,1

Pengurangan rasio = 0,1

3. Perhitungan 4) sampai 5) dilakukan selama perulangan masih lebih kecil dari maksimum perulangan atau rasio pembelajaran masih lebih besar dari eror minimum.
4. Untuk setiap data masukan (x) dihitung terhadang setiap data bobot (w) menggunakan rumus $Cj = \sqrt{\sum_{i=1}^{n}(F_i - w_{ij})^2}$. Perhitungan jarak (C) untuk data masukan X1 terhadap masing-masing data bobot (W) adalah sebagai berikut:
   a. Jarak X1 terhadap Bobot(W) kelas-1

$$C_1 = \sqrt{\begin{matrix}(5.27-4.76)^2+(4.8-1.5)^2+(5-3)^2+(12.09-7)^2 \\ +(10.8-7.52)^2+(29.91-15)^2\end{matrix}}$$

$$C_1 = \sqrt{(0.51)^2+(3.3)^2+(2)^2+(4.09)^2+(3.28)^2+(14.91)^2}$$

$$C_1 = \sqrt{0.26+10.89+4+16.73+10.76+222.31}$$

$$C_1 = \sqrt{264.94}$$

$$C_1 = 16.28$$

   b. Jarak X1 terhadap Bobot(W) kelas-2

$$C_2 = \sqrt{\begin{matrix}(5.27-2.51)^2+(4.8-3.9)^2+(5-4)^2+(11.09-11.5)^2 \\ +(10.8-10.01)^2+(29.91-16.34)^2\end{matrix}}$$

$$C_2 = \sqrt{(2.76)^2+(0.9)^2+(1)^2+(-0.41)^2+(0.79)^2+(13.57)^2}$$

$$C_2 = \sqrt{7.62+0.81+1+0.17+0.62+184.14}$$

$$C_2 = \sqrt{194.36}$$

$$C_2 = 13.94$$

   c. Jarak X1 terhadap Bobot(W) kelas-3

$$C_3 = \sqrt{\begin{matrix}(5.27-3.86)^2+(4.8-3.7)^2+(5-5)^2+(11.09-16.02)^2 \\ +(10.8-10.5)^2+(29.91-18.37)^2\end{matrix}}$$

$$C_3 = \sqrt{(1.41)^2+(1.1)^2+(0)^2+(-4.93)^2+(0.3)^2+(11.54)^2}$$

$C_3 = \sqrt{1.99 + 1.21 + 0 + 24.3 + 0.09 + 133.17}$

$C_3 = \sqrt{160.76}$

$C_3 = 12.68$

Dari hasil perhitungan di atas didapat tiga nilai jarak X1 terhadap masing-masing nilai bobot yaitu, C1 = 16.28 , C2 = 13.94 , dan C3 = 12.68 .

5. Dilihat jarak (C) terkecil yang diperoleh pada perhitungan terhadap X1. Kemudian diperiksa apakah target x1 sesuai dengan kelas bobot dengan jarak terkecil. Untuk hasil dari perhitungan di atas menunjukan bahwa jarak terkecil ada pada bobot ke-3. Target x1 adalah kelas 1 dan kelas bobot ke-3 adalah kelas 3, hal ini berarti target tidak sesuai dengan hasil perhitungan.
6. Dilakukan perhitungan perubahan bobot terhadap bobot dengan jarak terkecil yaitu bobot ke-1 dengan rumus:

   Jika kelas target sesuai kelas bobot:

   W(baru) = W(lama) + α [x - W(lama)]

   Jika kelas target tidak sesuai kelas bobot:

   W(baru) = W(lama) - α [x - W(lama)]

   α = rasio pembelajaran

   Dalam contoh perhitungan di atas, hasil dari perhitungan menunjukan bahwa kelas target tidak sesuai dengan kelas bobot dengan jarak terkecil, sehingga perhitungan untuk perubahan nilai bobot (W) adalah sebagai berikut:

   $W_{1,1} = 4.76 - 0.1 * (5.27 - 4.76) = 4.7$

   $W_{1,2} = 1.5 - 0.1 * (4.8 - 1.5) = 1.17$

   $W_{1,3} = 3 - 0.1 * (5 - 3) = 2.8$

   $W_{1,4} = 7 - 0.1 * (11.09 - 7) = 6.59$

   $W_{1,5} = 7.52 - 0.1 * (10.8 - 7.52) = 7.19$

   $W_{1,6} = 15 - 0.1 * (29.91 - 15) = 13.51$
7. Langkah 4) sampai 6) dikerjakan terhadap semua masukan (x), setelah selesai dilakukan perhitungan perubahan nilai rasio dengan rumus:

   $\text{rasio} = \text{rasio} - (\text{pengurangan rasio} * \text{rasio})$

   $\text{rasio} = 0.1 - (0.1 * 0.1) = 0.09$

8. Perhitungan diulang sampai perulangan berhenti atau rasio pembelajaran kurang dari sama dengan error minimum.

Sehingga nanti akan mendapatkan nilai bobot (W) akhir. Yang nantinya akan digunakan untuk proses testing. Pada Tabel III-2 menunjukan nilai bobot (W) terakhir.

Tabel III-2 Nilai Bobot Hasil Training

| Ekspresi | Kelas | Nilai Bobot akhir (W) |
|---|---|---|
| Senyum | 1 | (4.7, 1.17, 2.8, 6.59, 7.19, 13.51) |
| Normal | 2 | (2.51, 3.9, 4.69, 11.48, 10.01, 16.24) |
| Marah | 3 | (3.86, 3.7, 5, 16.07, 10.51, 18.36) |

### III.2.3.3 *Testing Learning Vector Quantization*

Proses pengenalan dengan metode *learning vector quantization* menggunakan rumus yang sama dengan perhitungan bobot pada proses pembelajaran *learning vector quantization*.

Langkah langkahnya adalah sebagai berikut:

1. Data *test* dihitung terhadap setiap data bobot (w) dengan data test, seperti pada Tabel III-3:

Tabel III-3 Data Test

| Data Test (X) (F1,F2,F3,F4,F5,F6) | Kelas | Ekspresi |
|---|---|---|
| (2.18, 3.7, 5, 11.6, 11.2, 17.91) | ? | ? |

Perhitungan jarak (C) untuk data test terhadap masing-masing bobot (W) menggunakan rumus $Cj = \sqrt{\sum_{i=1}^{n}(F_i - w_{ij})^2}$ adalah sebagai berikut:

a. Jarak x terhadap Bobot(W) kelas-1

$$C_1 = \sqrt{\begin{matrix}(2.18-4.7)^2+(3.7-1.17)^2+(5-2.8)^2+(11.6-6.59)^2 \\ +(11.2-7.19)^2+(17.91-13.51)^2\end{matrix}}$$

$$C_1 = \sqrt{(-2.52)^2+(2.53)^2+(2.2)^2+(5.01)^2+(4.01)^2+(4.4)^2}$$

$$C_1 = \sqrt{6.35+6.4+4.84+25.1+16.1+19.336}$$

$$C_1 = \sqrt{78.13}$$

$$C_1 = 8.84$$

b. Jarak x terhadap Bobot(W) kelas-2

$$C_2 = \sqrt{\begin{matrix}(2.18-2.51)^2+(3.7-3.9)^2+(5-4.69)^2+(11.6-11.48)^2+ \\ (11.2-10.01)^2+(17.91-16.24)^2\end{matrix}}$$

$$C_2 = \sqrt{(-0.33)^2+(-0.2)^2+(0.31)^2+(0.12)^2+(1.19)^2+(1.67)^2}$$

$$C_2 = \sqrt{0.11+0.04+0.09+0.01+1.42+2.79}$$

$$C_2 = \sqrt{4.46}$$

$$C_2 = 2.11$$

c. Jarak x terhadap Bobot(W) kelas-3

$$C_3 = \sqrt{\begin{matrix}(2.18-3.86)^2+(3.7-3.7)^2+(5-5)^2+(11.6-16.07)^2 \\ +(11.2-10.51)^2+(17.91-18.36)^2\end{matrix}}$$

$$C_3 = \sqrt{(-1.68)^2+(0)^2+(0)^2+(-4.47)^2+(0.69)^2+(-0.45)^2}$$

$$C_3 = \sqrt{2.82+0+0+19.98+0.48+0.2}$$

$$C_3 = \sqrt{23.48}$$

$$C_3 = 4.84$$

2. Dilihat dari 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 didapat nilai terkecil dari ketiga jarak (C) adalah C2 dimana jarak ini adalah jarak terhadap kelas 2 yaitu ekspresi normal.

### III.2.4 Analisis Kebutuhan Perangkat Lunak

#### III.2.4.4 Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menjelaskan kebutuhan sistem yang akan dijalankan. Berikut kebutuhan-kebutuhannya.

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 sistem pemilihan lagu ini, perangkat keras yang digunakan berupa sebuah komputer yang berbasis desktop. Dan spesifikasi yang digunakan dalam menjalankan sistem ini yaitu:

a. Processor 2,27 GHz
b. Hard Disk 320 GB
c. Memory 2 GB
d. Monitor
e. Webcame, tipe VCR (Video Camera Recorder)
f. Mouse
g. Keyboard

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 sistem ini adalah sebagai berikut :

1. Sistem Operasi Windows 8.1
2. Microsoft Visual Studio 2010
3. XAMPP MySql

4. Emgu CV 2.2.1, digunakan sebagai *library* tambahan untuk pengkodean sistem.

3. Kebutuhan Pengguna

Untuk menjalankan sistem dibutuhkan spesifikasi pengguna yang dapat menjalankan fungsi-fungsi dari sistem ini. Adapun spesifikasi pengguna yang dibutuhkan adalah sebagai berikut:

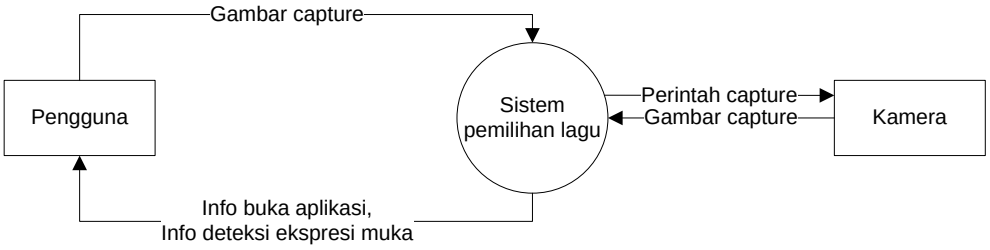
1. Dapat mengoperasikan komputer.
2. Mengerti bahasa Inggris dasar dan bahasa Indonesia.

### III.2.4.5 Kebutuhan Fungsional

Analisis kebutuhan fungsional menjelaskan proses-proses yang terjadi di dalam sistem ini. Perangkat lunak ini dimodelkan menggunakan *DFD* (*Data Flow Diagram*). *Tools* yang digunakan adalah Diagram Konteks, *Data Flow Diagram* dan Spesifikasi Proses yang dibuat menggunakan Microsoft Visio 2007 sebagai perangkat lunak yang digunakan.

1. Diagram Konteks

Diagram konteks adalah diagram yang menggambarkan masukan, proses, dan keluaran secara umum yang terjadi pada sistem. Berikut ini adalah diagram konteks perangkat lunak pemilihan lagu pada Gambar III.8.



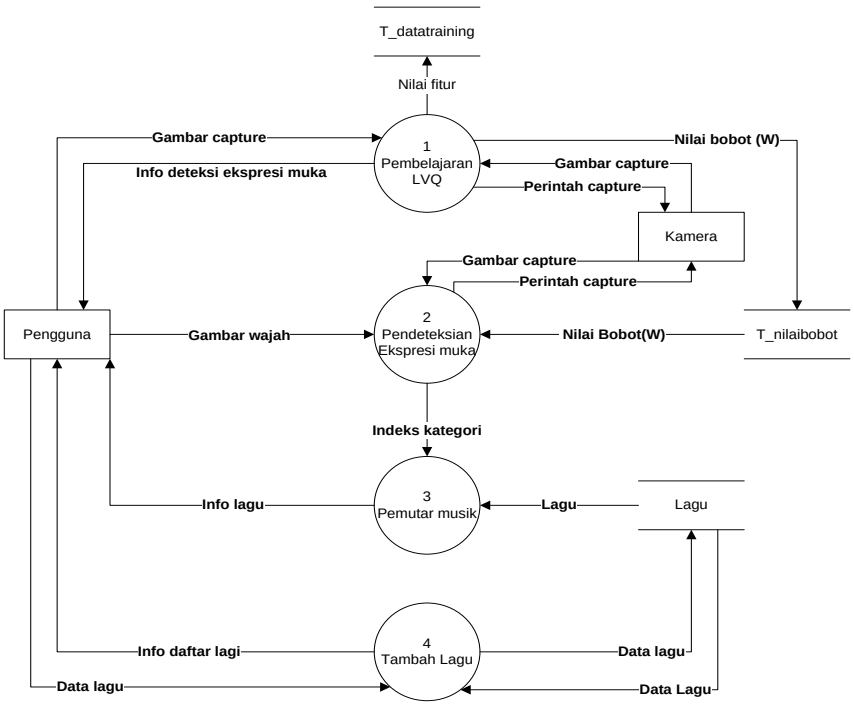

Gambar III.8 Diagram konteks sistem pemilihan lagu

*2. Data Flow Diagram ( DFD )*

*Data flow diagram* merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau lingkungan fisik dimana data tersebut akan disimpan. Salah satu keuntungan DFD adalah memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan.

a. DFD level 1

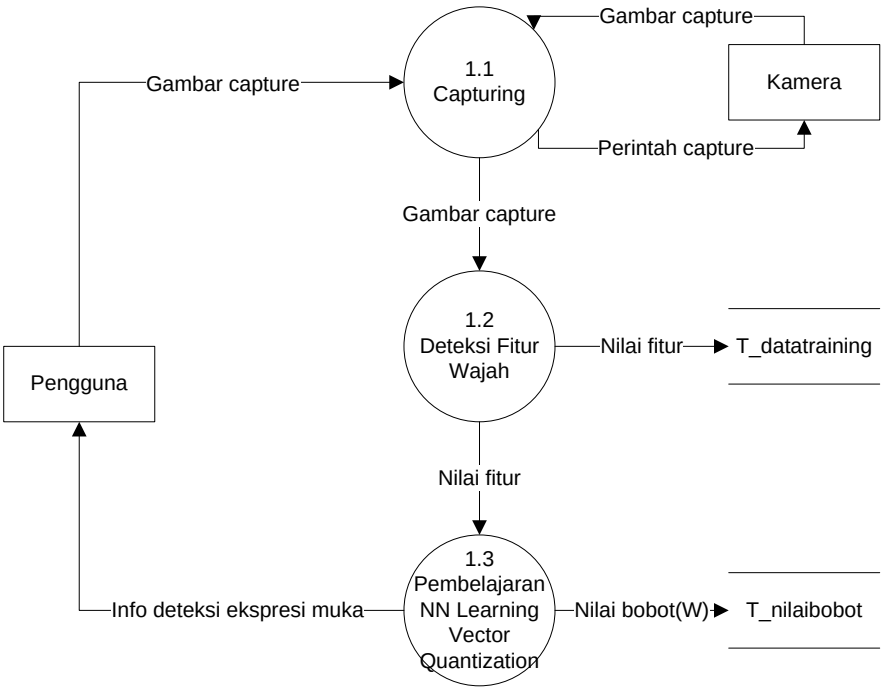
DFD level 1 dibuat untuk menjelaskan lebih detail lagi jika pada diagram kontek ada proses yang harus dijelaskan lebih terperinci lagi. DFD level 1 terdapat 4 proses, yaitu pembelajaran LVQ, pendeteksian ekspresi wajah, pemutar musik, dan tambah lagu. Untuk gambaran DFD level 1 dapat dilihat pada Gambar III.9.




Gambar III.9 DFD Level 1 Sistem Pemilihan Lagu

b. DFD Level 2 Bagian 1 Pembelajaran LVQ

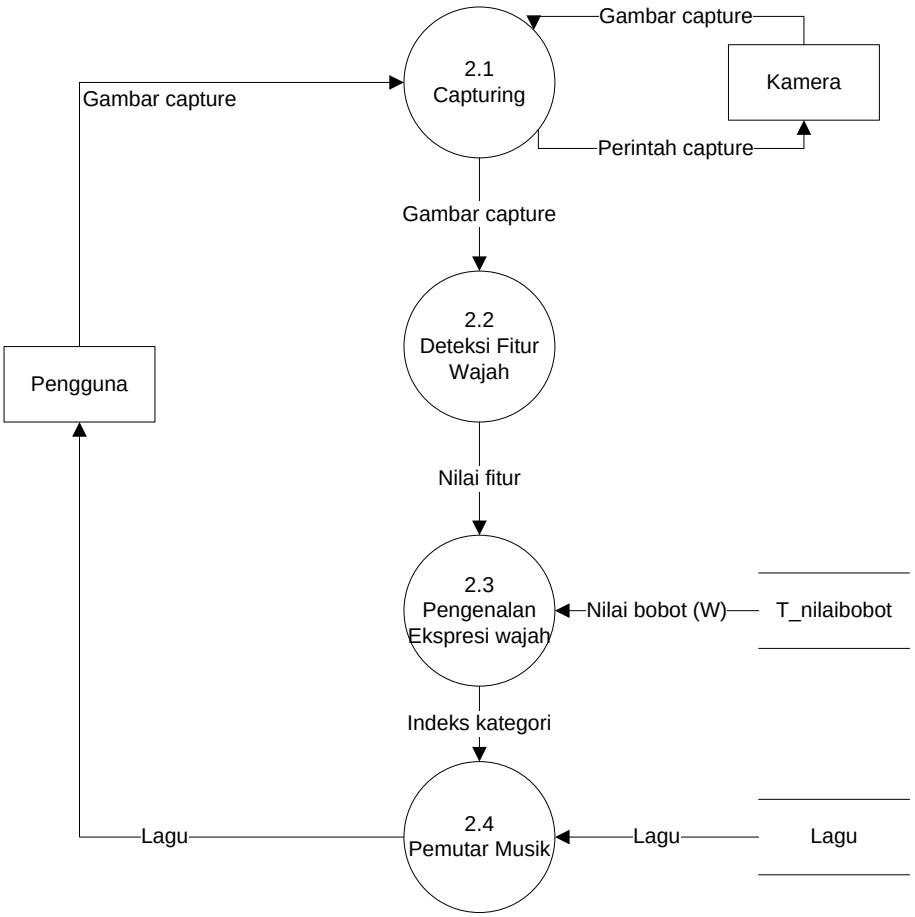
DFD level 2 bagian 1 pembelajaran LVQ terdiri dari 3 proses yaitu *capturing*, deteksi fitur wajah, dan NN *lerning vector quantization*. Untuk gambaran DFD level 2 bagian 2 dapat dilihat padaGambar III.10.




Gambar III.10 DFD Level 2 Bagian 1 Pembelajaran LVQ

c. DFD level 2 bagian 2 Pengenalan Ekspresi Wajah

DFD level 2 bagian 2 pengenalan ekspresi wajah terdiri dari 4 proses yaitu capturing, deteksi fitur wajah, *learning vector quantization*, dan pemutar musik. Untuk gambaran DFD level 2 bagian 2 dapat dilihat pada Gambar III.11.




Gambar III.11 DFD level 2 bagian 2 Pengenalan Ekspresi Wajah

3. Spesifikasi Proses

Spesifikasi proses digunakan untuk menggambarkan semua proses model aliran data yang tergambar pada DFD yang tercantum pada Tabel III-4 Spesifikasi Proses Sistem pemilihan lagu

Tabel III-4 Spesifikasi Proses Sistem pemilihan lagu

| NO | PROSES | KETERANGAN |
|---|---|---|
| 1 | Nomor Proses | 1 |
| | Nama Proses | Pembelajaran LVQ |
| | Deskripsi | Mendeteksi ekspresi wajah oleh *webcam* |

| | | |
|---|---|---|
| | Input | Citra wajah dari *webcam* |
| | Output | Nilai bobot(W) |
| | Proses | Sistem mendeteksi ekspresi wajah senang, marah, atau normal |
| | Logika Proses | 1. Citra wajah dideteksi fitur-fiturnya menggunakan fungsi haar<br>2. Menghitung nilai-nilai fitur<br>3. Mengklasifikasikan ekspresi muka berdasarkan nilai-nilai fitur yang didapat menggunakan LVQ<br>4. Mendapatkan nilai bobot(W)<br>5. Menyimpan nilai bobot(W) ke data training |
| 2 | Nomor Proses | 2 |
| | Nama Proses | Pengenalan ekspresi wajah |
| | Deskripsi | Mendeteksi ekspresi wajah oleh *webcam* |
| | Input | Citra wajah dari *webcam* |
| | Output | Hasil ekspresi wajah |
| | Proses | Sistem mendeteksi ekspresi wajah senang, marah, atau normal |
| | Logika Proses | 1. Citra wajah dideteksi fitur-fiturnya menggunakan fungsi haar<br>2. Menghitung nilai-nilai fitur<br>3. Mengklasifikasikan ekspresi muka berdasarkan nilai-nilai fitur yang didapat menggunakan LVQ<br>4. Mendapatkan hasil ekspresi wajah |
| 3 | Nomor Proses | 3 |
| | Nama Proses | Pemutar musik |
| | Deskripsi | Memutarkan musik yang sesuai dengan ekspresi muka dari pengguna |
| | Input | Variabel indek ekspresi |
| | Output | Lagu |
| | Proses | Menjalankan musik |
| | Logika Proses | 1. Mendapatkan inputan hasil ekspresi wajah berupa indeks 1, 2, atau 3<br>2. Menjalankan lagu yang sesuai dengan ekspresi wajah tersebut |
| 4 | Nomor Proses | 4 |
| | Nama Proses | Tambah Lagu |
| | Deskripsi | Menambahkan lagu kedalam playlist |
| | Input | Lagu dan kategori ekspresi |

| | | |
|---|---|---|
| | Output | Daftar Lagu dalam playlist |
| | Proses | Menambahkan data lagu kedalam playlist |
| | | 1.pilih lagu yang mau ditambahkan.<br>2.pilih kategori lagu<br>3.simpan data lagu |
| 5 | Nomor Proses | 1.1 |
| | Nama Proses | *Capturing* |
| | Deskripsi | Mengidentifikasi wajah pengguna |
| | Input | Citra dari *webcam* |
| | Output | Gambar *capture* |
| | Proses | Mengcapturing dari hasil citra *rea ltime* |
| | Logika Proses | 1. Citra *real time* yang merupakan objek di-*capturing*<br>2. Objek di-*cropping* pada fitur wajahnya |
| 6 | Nomor Proses | 1.2 |
| | Nama Proses | Deteksi fitur wajah |
| | Deskripsi | Mengidentifikasi fitur-fitur wajah |
| | Input | Citra wajah dari *webcam* |
| | Output | Nilai fitur |
| | Proses | Menghitung nilai dari fitur-fitur |
| | Logika Proses | 1. Menentukan koordinat fitur-fitur seperti mulut, mata, dan alismata<br>2. Menentukan nilai-nilai fitur |
| 7 | Nomor Proses | 1.3 |
| | Nama Proses | Pembelajaran LVQ |
| | Deskripsi | Mengklasifikasikan ekspresi wajah pengguna |
| | Input | Data fitur wajah |
| | Output | Nilai bobot(W) |
| | Proses | Menentukan nilai bobot(W) dari hasil pembelajaran LVQ |
| | Logika Proses | 1. Mengisi parameter yang akan digunakan dalam pembelajaran.<br>2. Melakukan validasi apakah semua parameter pembelajaran sudah diisi<br>3. Jika parameter belum diisi semua, akan menampilkan pesan" parameter belum lengkap" kemudian kembali ke langkah 1.<br>4. Jika parameter sudah lengkap, sistem mengambil semua data training dari database.<br>5. Melakukan pembelajaran dengan metode learning |

| | | |
|---|---|---|
| | | vector quantization.<br>6. Menyimpan bobot (w) hasil pembelajaran. |
| 8 | Nomor Proses | 2.1 |
| | Nama Proses | Capturing |
| | Deskripsi | Mengidentifikasi wajah pengguna |
| | Input | Citra dari *webcam* |
| | Output | Gambar *capture* |
| | Proses | Mengcapturing dari hasil citra *real time* |
| | Logika Proses | a. Citra *real time* yang merupakan objek di-*capturing*<br>b. Objek di-*cropping* pada fitur wajahnya |
| 9 | Nomor Proses | 2.2 |
| | Nama Proses | Deteksi fitur wajah |
| | Deskripsi | Mengidentifikasi fitur-fitur wajah |
| | Input | Citra wajah dari *webcam* |
| | Output | Nilai fitur |
| | Proses | Menghitung nilai dari fitur-fitur |
| | Logika Proses | 1. Menentukan koordinat fitur-fitur seperti mulut, mata, dan alismata<br>2. Menentukan nilai-nilai fitur |
| 10 | Nomor Proses | 2.3 |
| | Nama Proses | *NN Learning Vector Quantization* |
| | Deskripsi | Mengklasifikasikan ekspresi wajah pengguna |
| | Input | Data fitur wajah |
| | Output | Hasil ekspresi wajah |
| | Proses | Menentukan variabel indeks ekspresi wajah dari pengguna |
| | Logika Proses | 1. Data fitur dihitung terhadap setiap data bobot (w)<br>2. Dilihat jarak (C) terkecil yang diperoleh pada perhitungan terhadap data test. Kelas bobot yang memiliki jarak terkecil menjadi hasil dari proses pengenalan ini.<br>3. Memberikan info hasil ekspresi wajah berupa indeks kategori yaitu 1,2,3. |

| | | |
|---|---|---|
| 11 | Nomor Proses | 2.4 |
| | Nama Proses | Pemutar musik |
| | Deskripsi | Memutarkan musik yang sesuai dengan ekspresi muka dari pengguna |
| | Input | Variabel indek ekspresi |
| | Output | Lagu |
| | Proses | Menjalankan musik |
| | Logika Proses | 1. Mendapatkan inputan hasil ekspresi wajah<br>2. Menjalankan lagu yang sesuai dengan ekspresi wajah tersebut |

4. Kamus Data

Kamus data merupakan suatu tempat penyimpanan dari data dan informasi yang dibutuhkan oleh suatu sistem, dapat dilihat pada Tabel III-5.

Tabel III-5 Kamus Data

| No | Kamus | Keterangan |
|---|---|---|
| 1 | Nama | Gambar capture |
| | Digunakan pada | Pengguna – level 2 bagian 1<br>Pengguna – level 2bagian 2<br>Kamera – level 2 bagian 1<br>Kamera – level 2 bagian 2 |
| | Deskripsi | Citra real dari pengguna |
| | Struktur data | citraRGB |
| | CitraRGB | [0,255], [0,255], [0,255], [0,255],....n |
| 2 | Nama | Nilai fitur |
| | Digunakan pada | Proses deteksi fitur wajah – level 2 bagian 1<br>Proses deteksi fitur wajah – level 2 bagian 2 |
| | Deskripsi | Nilai-nilai dari fitur wajah |
| | Struktur data | (F1 + F2 + F3 + F4 + F5 + F6), kelas |
| | F1, F2, F3, F4, F5, F6<br>kelas | {0-9}<br>[1\|2\|3] |
| 3 | Nama | Nilai bobot |

| | | |
|---|---|---|
| | Digunakan pada | Proses Pembelajaran LVQ – level 2bagian 1<br>Proses pengenalan ekspresi wajah –level 2 bagian 2 |
| | Deskripsi | Nilai dari masing-masing nilai bobot |
| | Struktur data | (W1+W2+W3+W4+W5+W6), kelas |
| | W1,W2,W3,W4,W5,W6<br>kelas | {0-9}<br>[1\|2\|3] |
| 4 | Nama | Data lagu |
| | Digunakan pada | Pengguna – level 1 |
| | deskripsi | Data lagu yang akan ditambahkan ke dalam playlist |
| | Struktur data | Url_lagu,kelas |
| | Url_lagu<br><br><br>Kelas | {0-9}<br>{‘A-Z’\|’a-z’\|0-9\|`~!@#$%^&*()_+}<br>{“:’?><}<br>[1\|2\|3] |

### III.3 Perancangan

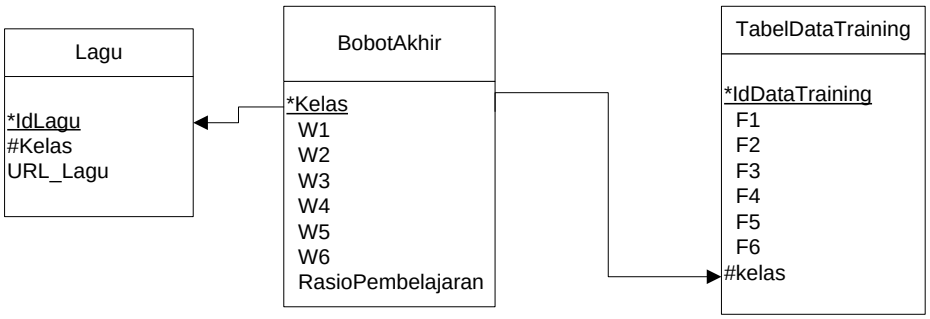
Perancangan sistem ini menjelaskan perancangan yang akan dibangun pada sistem pemilihan lagu ini. Sebuah penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi.

#### III.3.1 Perancangan Database

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

##### III.3.1.1 Diagram Relasi

Diagram relasi menggambarkan relasi setiap tabel di dalam database, seperti terlihat pada Gambar III.12.




Gambar III.12 Diagram Relasi

### III.3.1.2 Struktur Table

Strukture table menjelaskan tentang deskripsi setiap atribut yang dimiliki oleh setiap tabel yang digunakan pada sistem ini. Pada sistem ini terdapat dua buah tabel yaitu tabel data nilai bobot(W) dan tabel data *training*. Untuk struktur tabel data nilai bobot dapat dilihat pada Tabel III-6 dan untuk tabel data *training* dapat dilihat pada Tabel III-7 sedangkan untuk tabel data lagu dapat dilihat pada Tabel III-8.

Tabel III-6 Struktur Tabel Data Nilai Bobot(W)

| Nama | Tipe | Range | Keterangan |
|---|---|---|---|
| Kelas | int | 1,2,3 | PK |
| W1 | Double | 0..9 | - |
| W2 | Double | 0..9 | - |
| W3 | Double | 0..9 | - |
| W4 | Double | 0..9 | - |
| W5 | Double | 0..9 | - |
| W6 | Double | 0..9 | - |
| Rasio Pembelajaran | Float | 0..9 | - |

Tabel III-7 Struktur Tabel Data *Training*

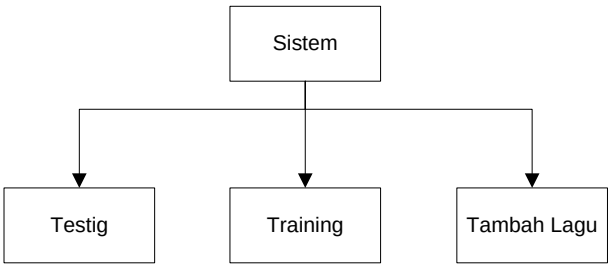
| Nama | Tipe | Range | Ketetangan |
|---|---|---|---|
| Id_data | Int | 0...9 | Primery Key, Auto Increment |
| F1 | Float | 0...9 | - |
| F2 | Float | 0...9 | - |
| F3 | Float | 0...9 | - |
| F4 | Float | 0...9 | - |
| F5 | Float | 0...9 | - |
| F6 | Float | 0...9 | - |
| Kelas | Int | 1,2,3 | Forign Key |

Tabel III-8 Struktur Tabel Data Lagu

| Nama | Type | Range | Keterangan |
|---|---|---|---|
| Id_lagu | Int | 0..9 | Primery Key |
| Url_lagu | String | 100 | |
| Kelas | Int | 1,2,3 | Forign Key |

### III.3.2 Struktur Menu

Perancangan ini menjelaskan struktur menu yang akan dibuat pada sistem ini. Gambar III.13 menunjukan struktur menu dari sistem ini. Di mana pada saat sistem dijalankan, sistem akan menampilkan tiga buah piliahan menu, yaitu menu *testing*, menu *training*, dan tambah lagu.



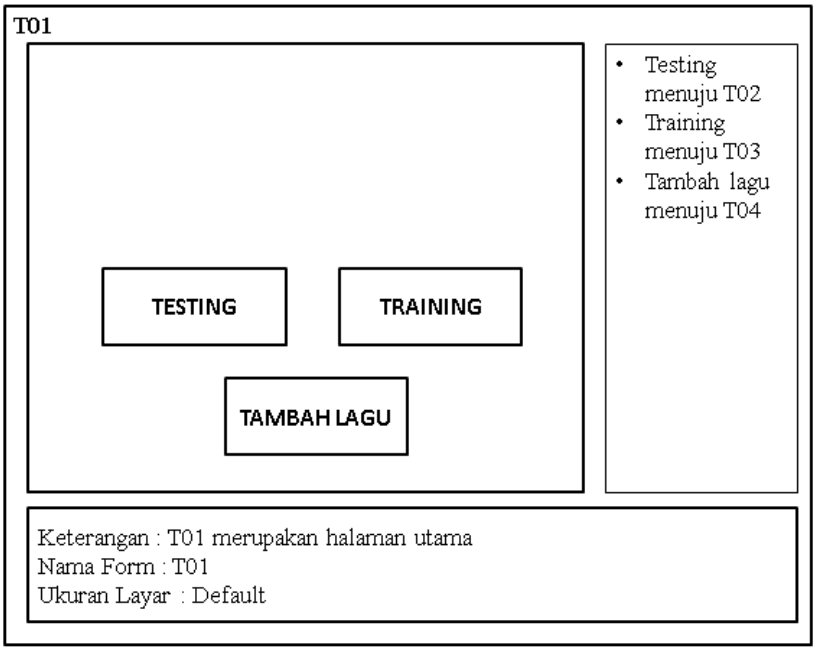

Gambar III.13 Struktur Menu Sistem Pemilihan Lagu

### III.3.3 Perancangan Antar Muka

Perancanga antar muka ini dibagi menjadi 3 rancangan antarmuka. Rancangan form tersebut adalah *form* menu utama, *form testing*, dan *form training.*

#### III.3.3.1 Desain Form Menu Utama

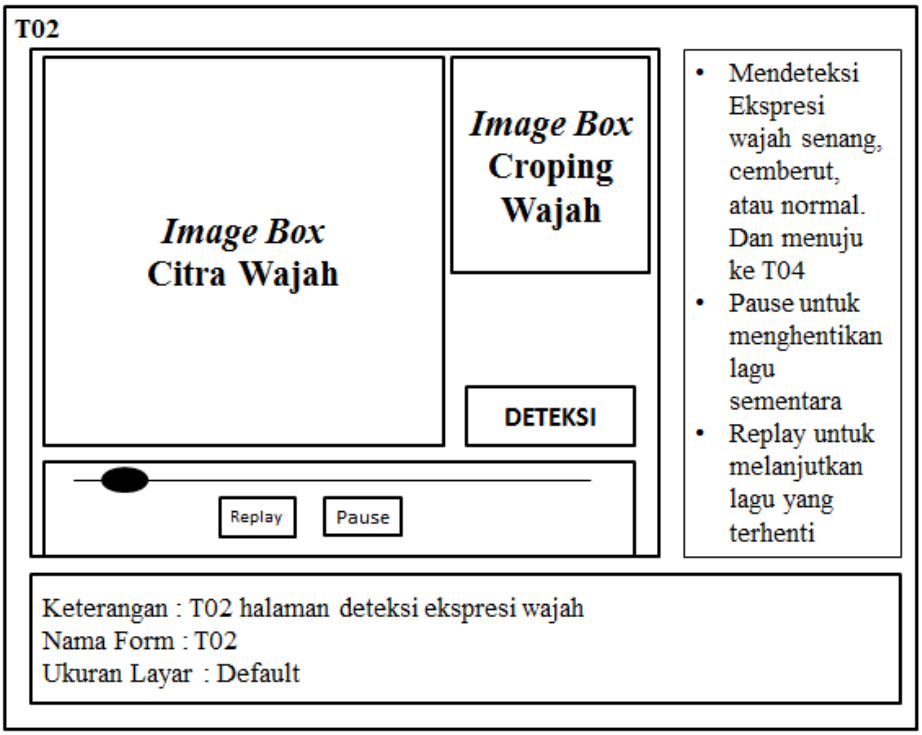
*Form* menu utama merupakan *form* yang digunakan sebagai tampilan pada saat pengguna membuka sistem. Berikut ini adalah desain tampilan form sistem pada Gambar III.14:




Gambar III.14 Desain Form Menu Utama

### III.3.3.2 *Desain Form Testing*

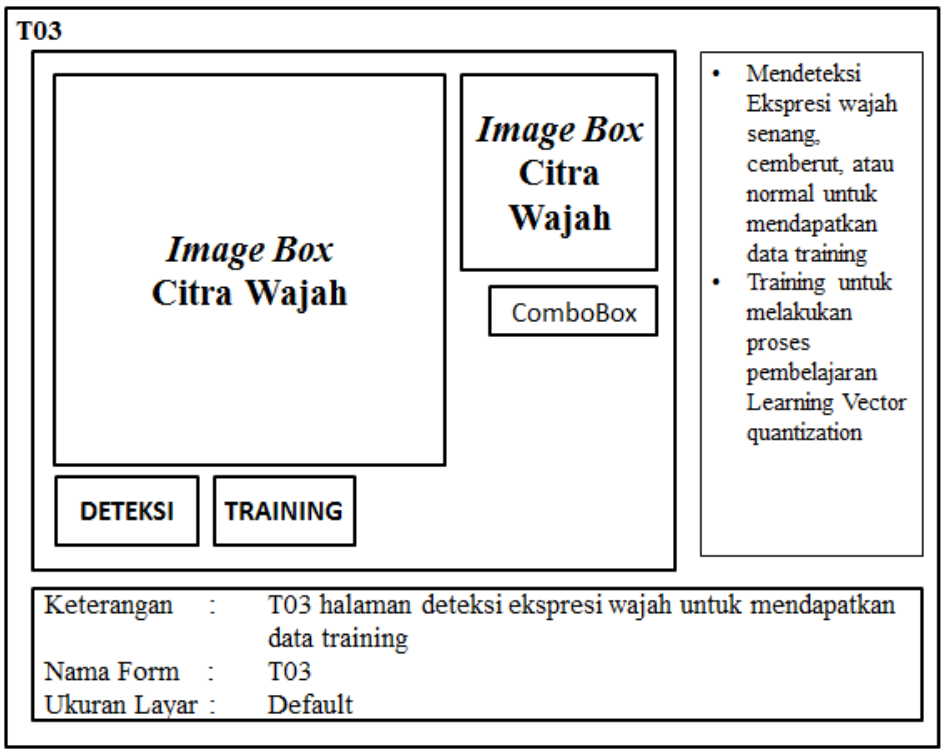
*Form testing* merupakan *form* yang digunakan sebagai tampilan pada saat pengguna melakukan deteksi ekspresi muka untuk menjalankan lagu. Berikut ini adalah desain tampilan form sistem pada Gambar III.15:




Gambar III.15 *Desain Form Testing*

### III.3.3.3 *Desain Form Training*

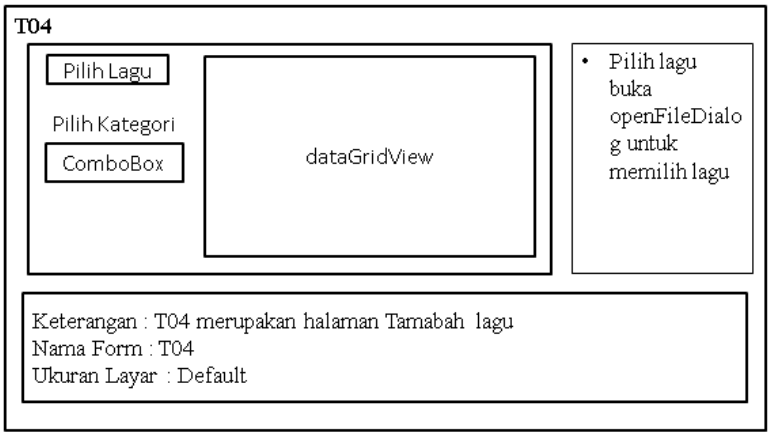
*Form training* merupakan *form* yang digunakan sebagai tampilan pada saat pengguna melakukan deteksi ekspresi muka untuk mendapatkan data *training*. Berikut ini adalah desain tampilan *form training* pada Gambar III.16:




Gambar III.16 Desain *Form Training*

### III.3.3.4 Desain *Form* Tambah Lagu

*Form* tambah lagu merupakan *form* yang digunakan untuk menambahkan lagu kedalam playlist. Berikut ini adalah desain tampilan *form* tambah lagu pada Gambar III.17.



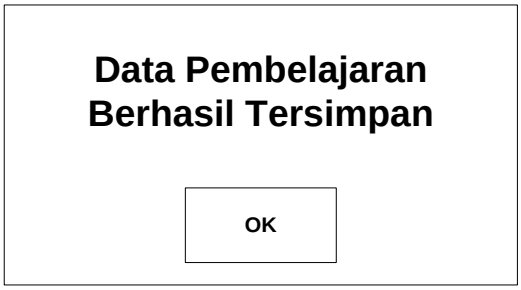

Gambar III.17 Desain *Form* Tambah Lagu

### III.3.4 Perancangan Pesan

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

a. Pesan Pembelajaran Berhasil Tersimpan

Pesan pemberitahuan berhasil tersimpan digunakan untuk memberi informasi bahwa proses pembelajaran *learning vector quantization* telah berhasil. Perancangan pesan pembelajaran berhasil tersimpan ditunjukan pada Gambar III.18.




Gambar III.18 Pesan Pembelajaran Berhasil Tersimpan

b. Pesan Pembelajaran Gagal

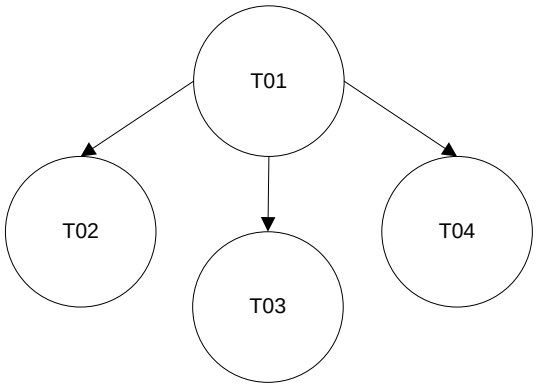
Pesan pembelajaran gagal digunakan untuk memberikan informasi bahwa proses pembelajaran gagal. Perancangan pesan pembelajaran gagal ditunjukkan pada Gambar III.19.




Gambar III.19 Pesan Pembelajaran Gagal

### III.3.5 Perancangan Jaringan Semantik

Pada Gambar III.20. menjelaskan tentang gambaran hubungan antar modul sistem pemutar lagu berdasarkan ekspresi wajah.



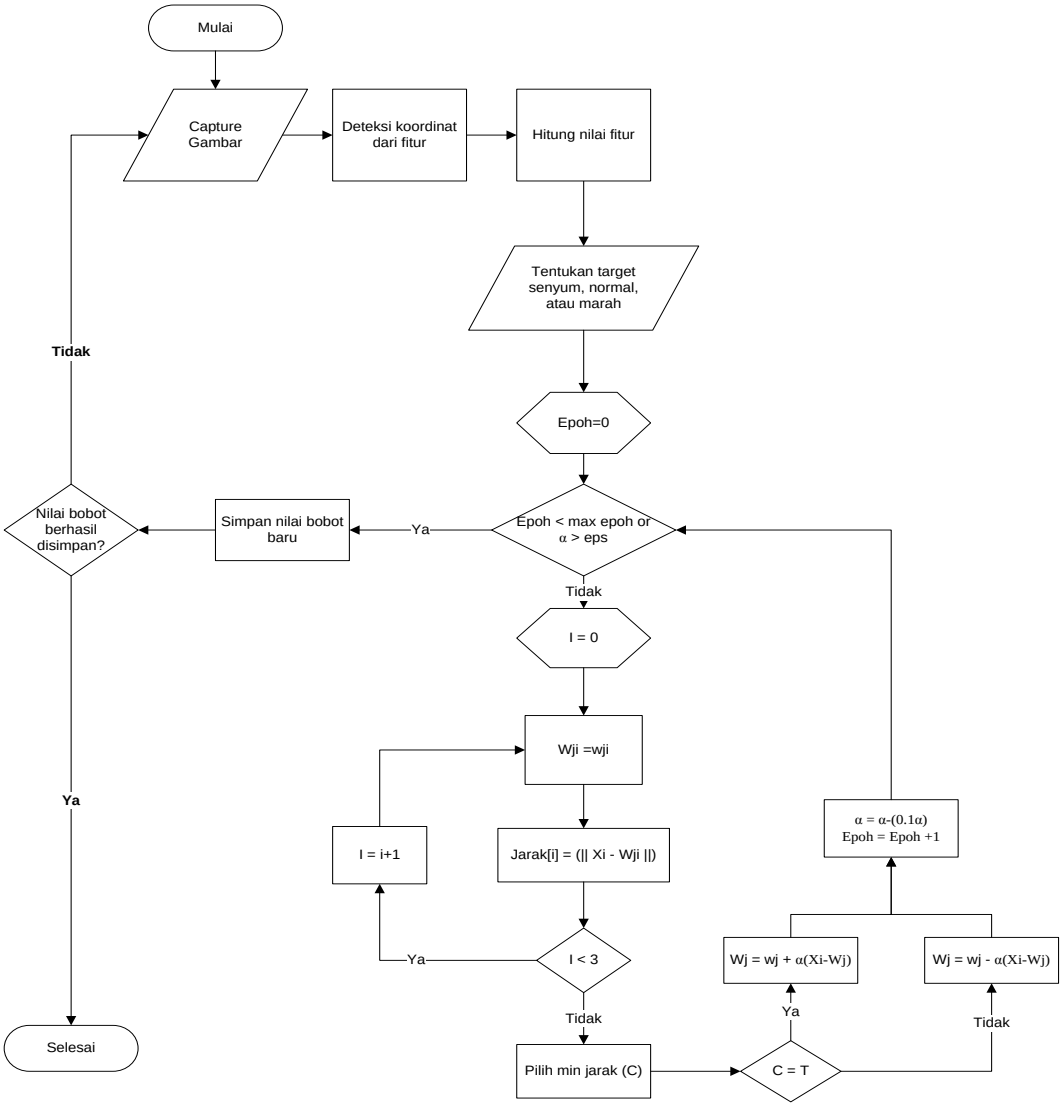

Gambar III.20 Jaringan Semantik

### III.3.6 Perancangan Prosedural

Dalam sistem ini terdapat dua fungsional utama yaitu proses training dan proses testing. Maka, untuk prosedural sistem ini akan dibagi menjadi dua yaitu prosedural training dan testing.

#### III.3.6.1 *Training*

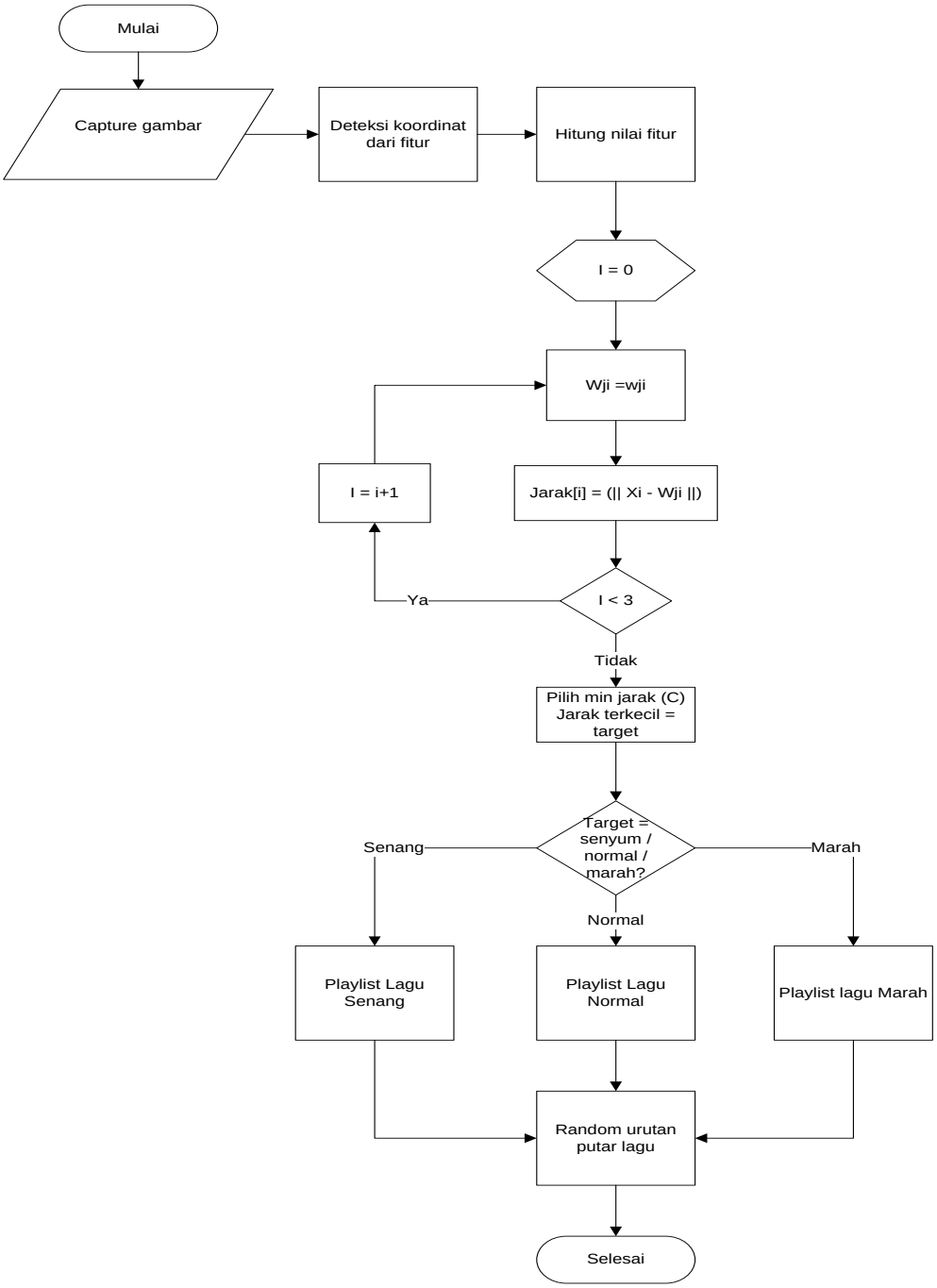
Pada Gambar III.21 ini menjelaskan tentang *prosedural training*.




Gambar III.21 *Prosedural training*

### III.3.6.2 *Testing*

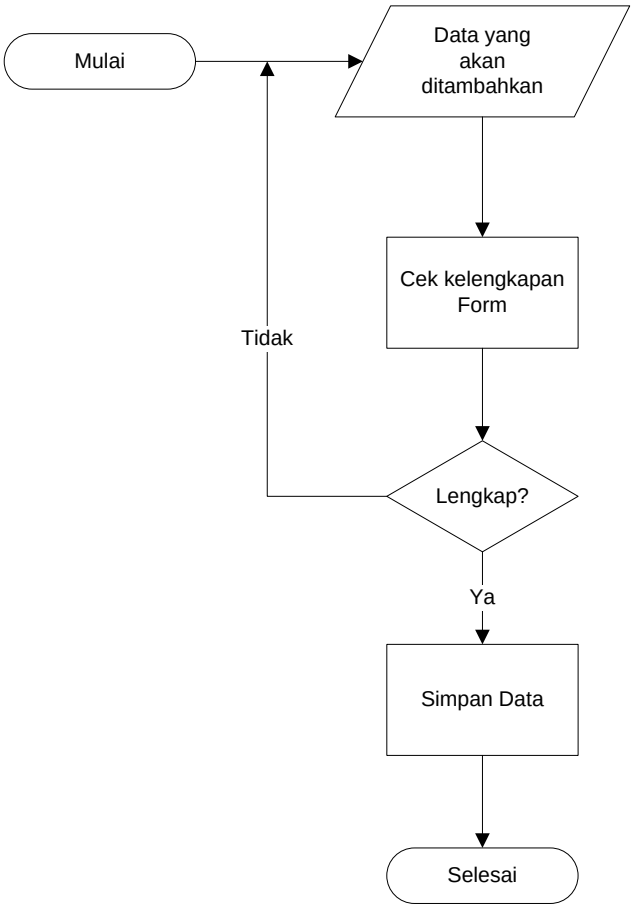
Pada Gambar III.22 ini menjelaskan tentang *prosedural testing*.




Gambar III.22 *Prosedural Testing*

### III.3.6.3 Tambah Lagu

Pada Gambar III.23 menjelaskan tentang *prosedural* tambah lagu pada *playlist.*




Gambar III.23 *Prosedural* Tambah Lagu