

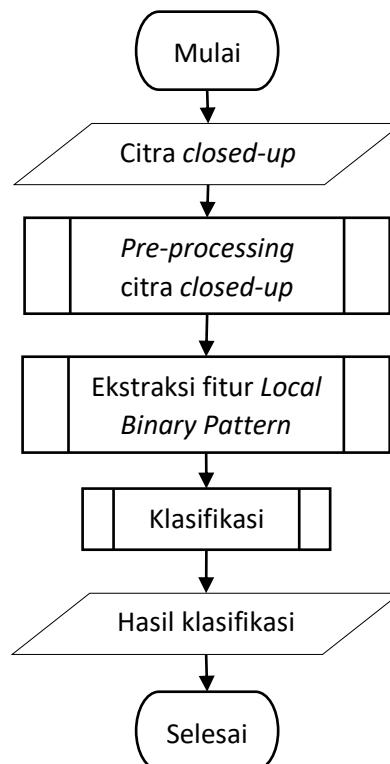
## BAB 4 PERANCANGAN

Bab ini menjelaskan penerapan tahapan perancangan dari sistem pengenalan emosi berdasarkan ekspresi mikro menggunakan metode *Local Binary Pattern* hingga menampilkan hasil klasifikasi.

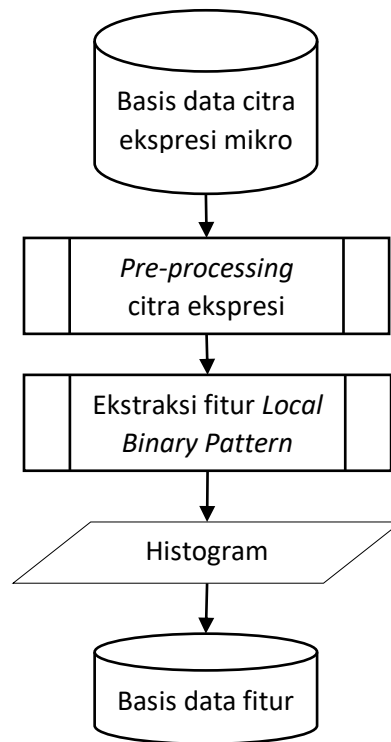
### 4.1 Perancangan Algoritme Proses Implementasi

Secara garis besar terdapat tiga proses utama yang diimplementasikan ke dalam program untuk penelitian ini, yaitu *pre-processing*, ekstraksi fitur *Local Binary Pattern*, klasifikasi KNN. Pada tahap *pre-processing*, citra masukan akan diolah hingga menghasilkan citra yang nantinya dapat diproses pada tahapan ekstraksi fitur *Local Binary Pattern*. Kemudian, diambil fitur dari hasil ekstraksi fitur *Local Binary Pattern* berupa nilai histogram 0-255 dari citra *Local Binary Pattern* yang digunakan untuk klasifikasi. Alur dari proses implementasi ditunjukkan pada Gambar 4.1.

Selain proses utama terdapat proses untuk pelatihan data untuk proses klasifikasi. Alur algoritme pelatihan data hampir sama dengan alur proses utama, perbedaannya setelah melakukan ekstraksi fitur LBP dan didapatkan nilai histogram kemudian disimpan dalam basis data berupa dokumen *excel*. Alur dari proses implementasi ditunjukkan pada Gambar 4.2.



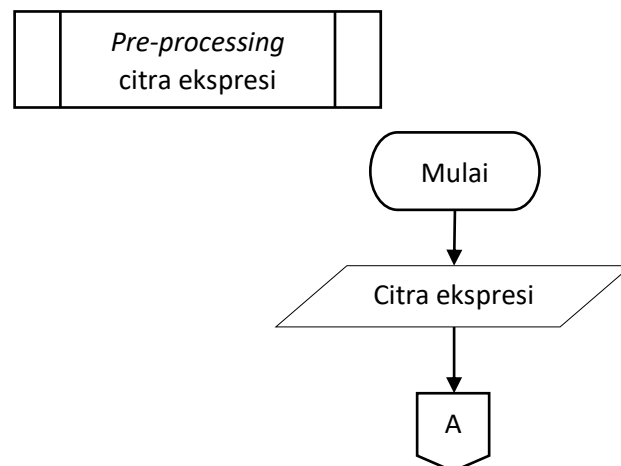
Gambar 4.1 Alur perancangan sitem

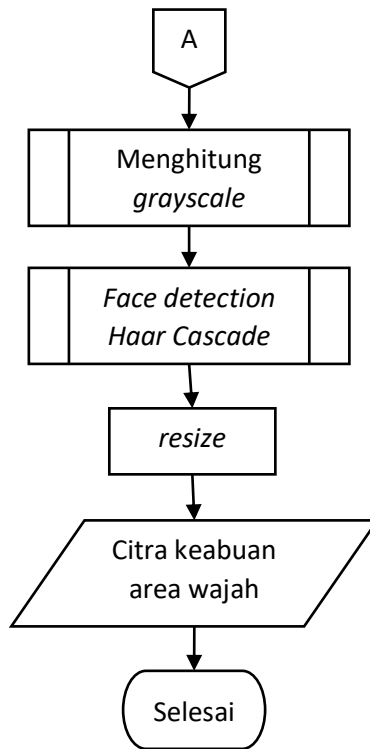


Gambar 4.2 Alur perancangan data latih

#### 4.1.1 Pre-processing Implementasi

*Pre-processing* adalah proses pengolahan citra paling awal yang berfungsi untuk memperbaiki kualitas citra sebelum dilakukan tahapan selanjutnya. Pada penelitian ini terdapat dua tahapan *pre-processing*, yaitu tahapan *pre-processing* pada data latih dan *pre-processing* pada tahapan proses implementasi. Kedua tahapan *pre-processing* ini memiliki alur yang sama. Alur dari kedua *pre-processing* ditunjukkan pada Gambar 4.3.

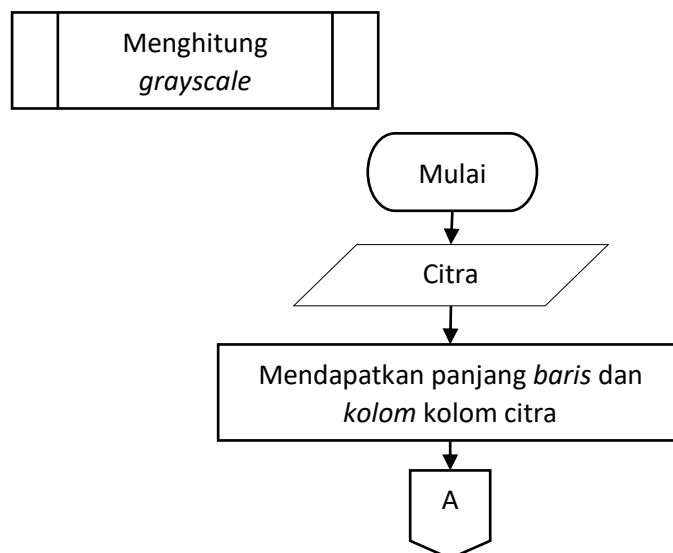


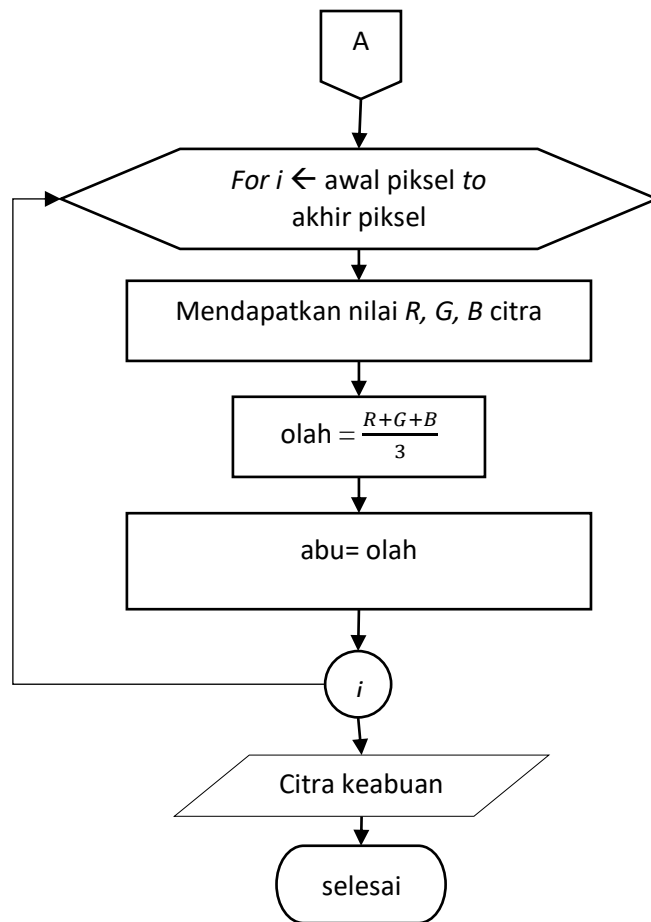


Gambar 4.3 Diagram alir *pre-processing* citra ekspresi mikro

#### 4.1.2 Menghitung *Grayscale*

*Grayscale* adalah warna piksel yang berada dalam rentang warna hitam dan putih atau dapat dikatakan keabuan. *Grayscale* dilakukan dengan merubah gambar yang memiliki intensitas gambar *red* (R), *green* (G), *blue* (B) menjadi abu-abu (*gray*). Dengan cara mengambil nilai piksel dari R, G, dan B lalu ketiga nilai tersebut ditambah dan dibagi dengan tiga sehingga didapatkan nilai piksel keabuan.

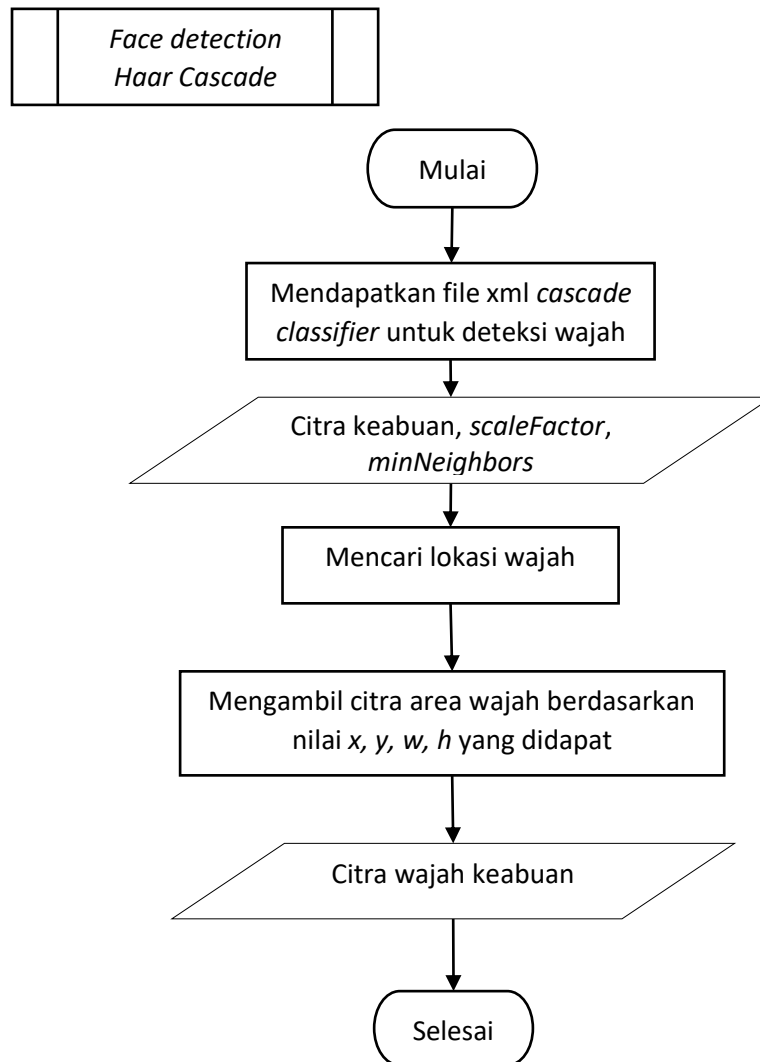




**Gambar 4.4** Diagram alir menghitung *grayscale*

### 4.1.3 Face detection Haar Cascade

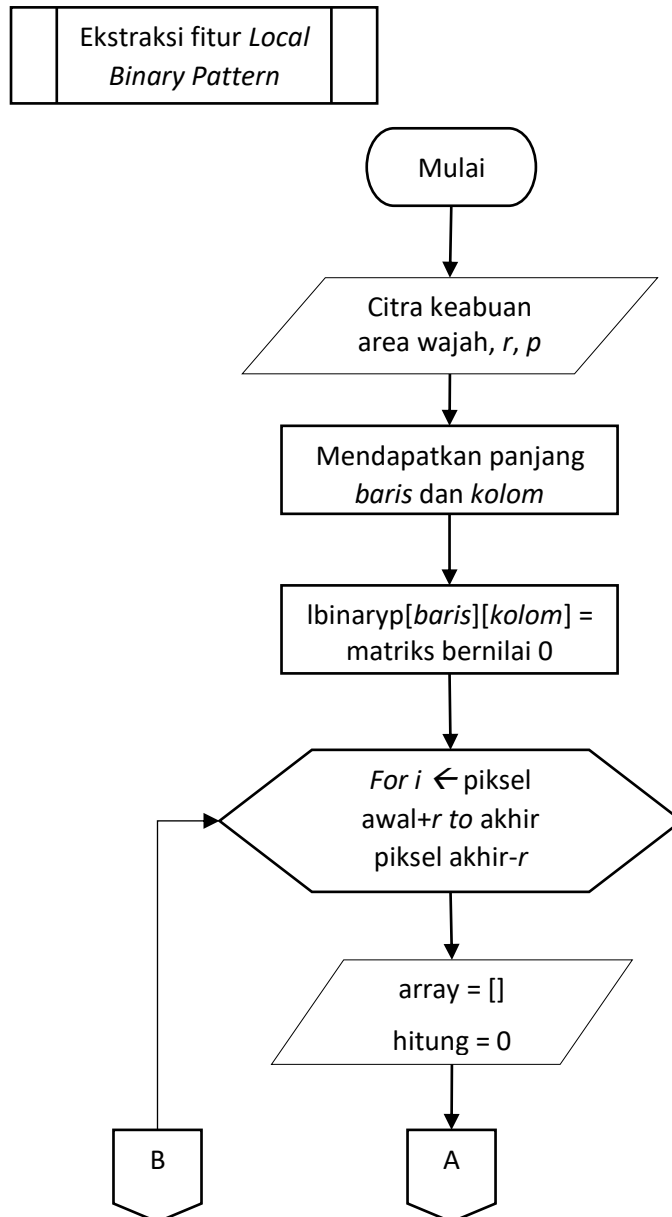
Proses mendeteksi wajah membutuhkan citra RGB yang telah diproses menjadi citra keabuan. *Face detection Haar Cascade* digunakan untuk mendapatkan area wajah agar lebih mudah dalam mengenali ekspresi mikro. Pencarian wajah dilakukan dengan menggunakan *library* pendeteksi wajah yang sudah disediakan oleh *OpenCV*. Alur dari *face detection Haar Cascade* menggunakan *library OpenCV* ditunjukkan pada Gambar 4.5.

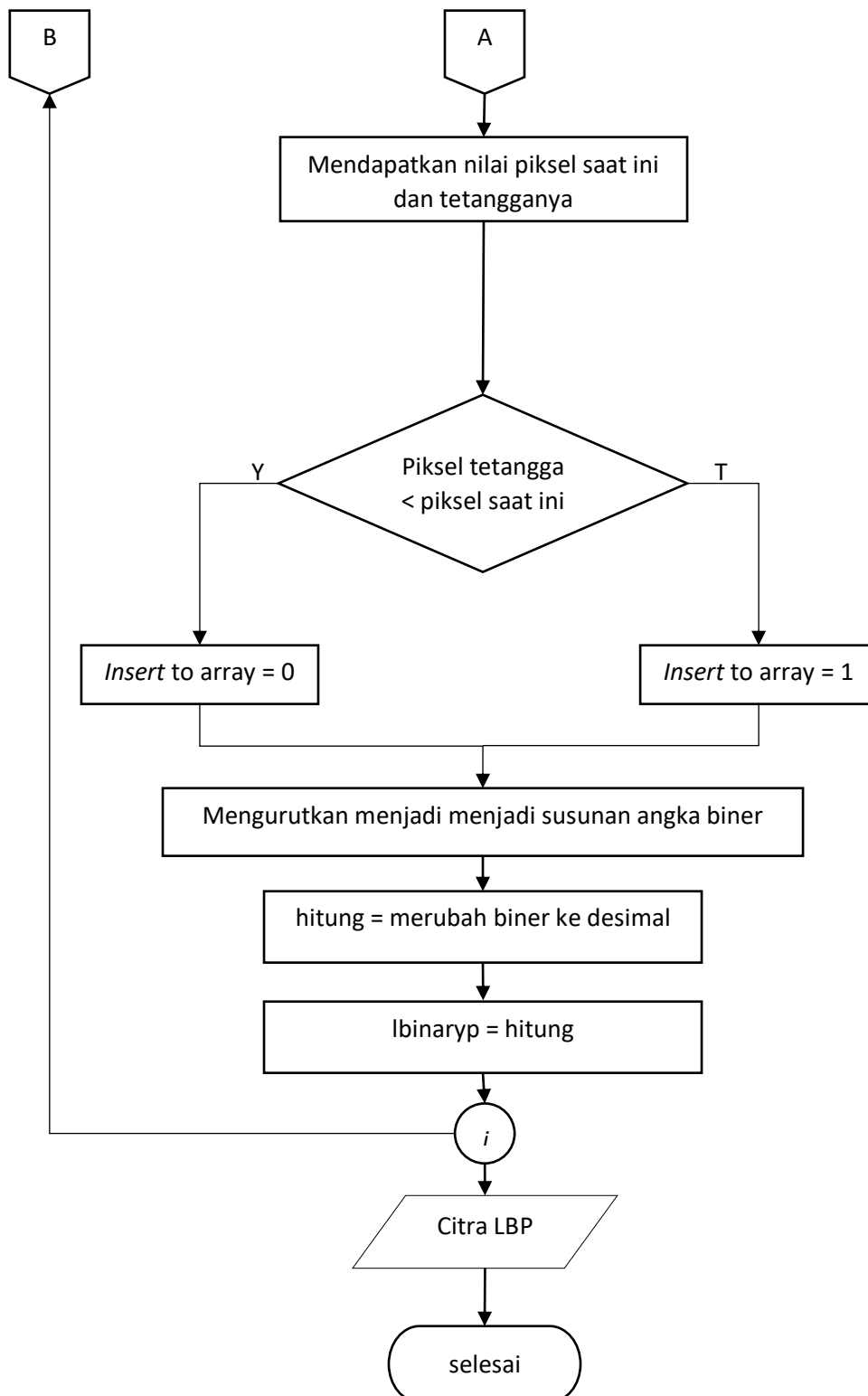


**Gambar 4.5** Diagram alir *face detection Haar Cascade*

#### 4.1.4 Ekstraksi fitur *Local Binary Pattern*

Proses ekstraksi fitur *Local Binary Pattern* memiliki konsep dasar yaitu membandingkan piksel yang dipilih dengan piksel tetangganya, kemudian melakukan *thresholding*. Piksel-piksel tetangga yang telah melalui proses *thresholding* kemudian diurutkan sehingga membentuk seperti bilangan biner, kemudian bilangan biner diubah menjadi bilangan desimal. Bilangan desimal yang didapat akan menggantikan piksel yang dipilih.

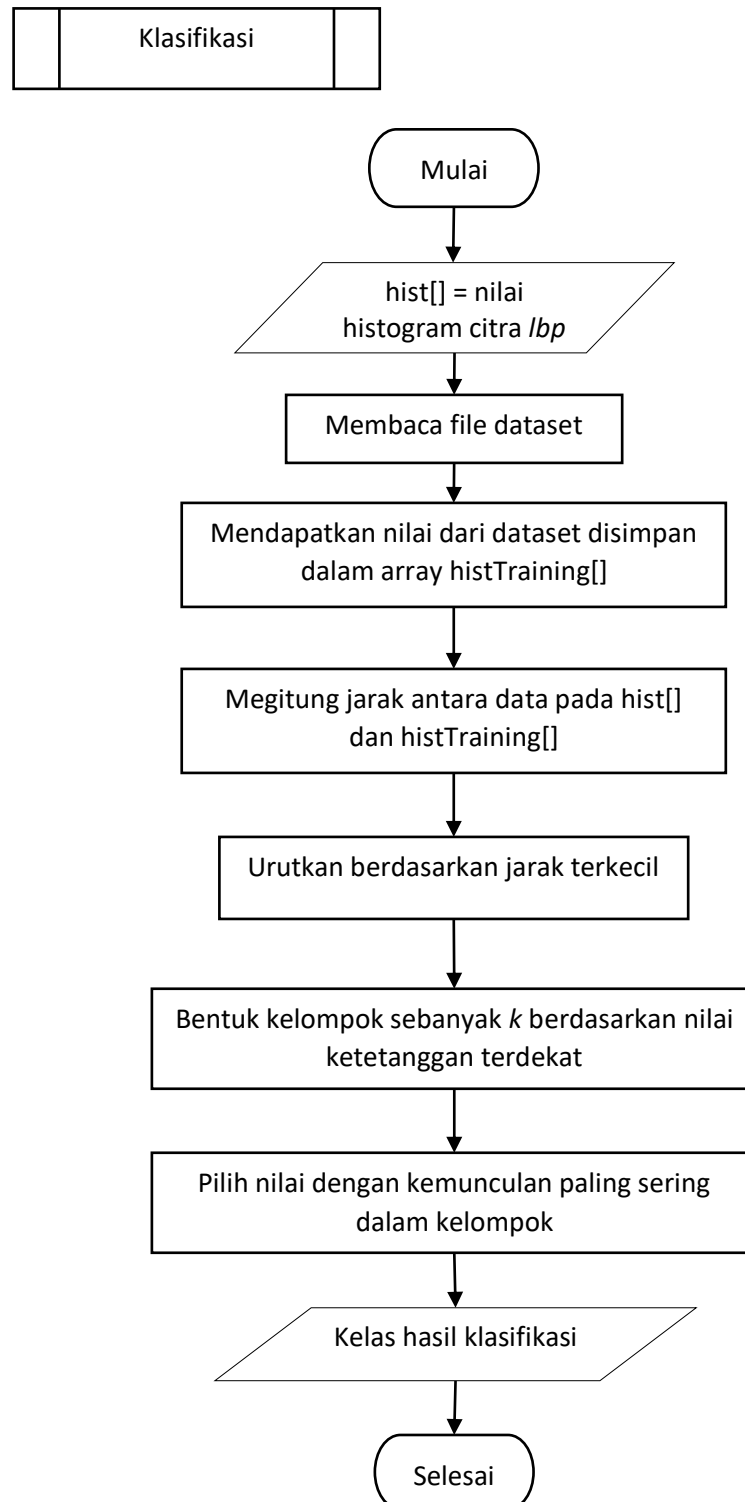




**Gambar 4.6** Diagram alir ekstraksi fitur *Local Binary Pattern*

#### 4.1.5 Klasifikasi K-NN

Klasifikasi K-NN adalah metode klasifikasi yang dilakukan dengan memilih jarak terdekat dari kelas-kelas yang ada untuk menentukan masuk kemana data yang baru. Alur klasifikasi K-NN ditunjukkan pada Gambar 4.7.



Gambar 4.7 Diagram alir klasifikasi K-NN



## 4.2 Perhitungan Manualisasi

Perhitungan manualisasi pada penelitian ini dilakukan pada citra yang telah melalui tahapan *pre-processing* seperti *grayscale* dan *face detection Haar Cascade*. Perhitungan ini dilakukan untuk mencari fitur dari citra yang digunakan untuk proses klasifikasi, menggunakan satu sampel dari dataset sebesar 200 piksel x 200 piksel yang ditunjukkan pada Gambar 4.8. Pada perhitungan manualisasi digunakan sampel sebesar 5 piksel x 5 piksel dikarenakan terlalu lebar dan banyak piksel pada citra asli.



Gambar 4.8 Sampel data citra ekspresi mikro

### 4.2.1 Perhitungan *Local Binary Pattern*

Langkah 1:

Menentukan radius dan jumlah ketetanggaan, dalam manualisasi ini digunakan radius dan jumlah ketetanggaan masing-masing 1 dan 8.

Langkah 2:

Melakukan *windowing* 3x3 piksel pada piksel citra *input*, *windowing* dimulai dari koordinat piksel (1, 1) sebagai titik tengahnya.

Piksel pada gambar

108	112	115	117	116
111	116	118	117	116
120	124	125	124	124
124	127	130	132	133
124	127	131	133	135

Piksel yang terpilih

108	112	115
111	116	118
120	124	125

Langkah 3:

Piksel yang terpilih kemudian dilakukan *thresholding* dengan ketentuan jika piksel pusat lebih besar dengan dari piksel tetangganya, maka piksel tetangga tersebut nilainya dirubah menjadi 0. Jika piksel pusat kurang dari atau sama dengan piksel tetangga, maka piksel tetangga tersebut dirubah menjadi 1. Perbandingan dilakukan bergantian pada setiap tetangga urutan mengikuti arah jarum jam.

Piksel terpilih

108	112	115
111	116	118
120	124	125



Hasil *thresholding*

0	0	0
0		1
1	1	1

Langkah 4:

Membaca piksel berurutan yang telah dilakukan proses *thresholding* hingga berbentuk urutan angka biner. Kemudian, angka biner tersebut diterjemahkan ke angka desimal. Nilai dari angka decimal akan mengganti nilai piksel yang dipilih saat melakukan *windowing*. Hal ini dilakukan pada semua piksel kecuali piksel yang berada ditepi. Pergeseran *windowing* terus dilakukan sampai piksel terakhir atau dalam kasus ini sampai piksel dengan koordinat (3, 3), tidak sampai tepi, karena piksel tepi akan dirubah semua nilainya menjadi 0.

Hasil *thresholding*

0	0	0
0		1
1	1	1



Pengurutan piksel

1	2	3
8		4
7	6	5



Angka biner

0 0 0 1 1 1 1 0



Angka desimal

30

Langkah 5:

Angka desimal yang didapat diletakkan dalam matriks 2 dimensi yang baru sebagai matriks LBP. Angka desimal yang didapat diletakkan sesuai dengan koordinat dari titik pusat hasil *windowing* di langkah 1. Jika semua piksel sudah dijelajahi maka akan menghasilkan piksel baru seperti berikut:

#### Matriks LBP

0	0	0	0	0
0	30	14	79	0
0	30	14	31	0
0	28	28	28	0
0	0	0	0	0

Piksel yang letaknya berada paling pinggir akan diberi nilai 0, dikarenakan perhitungan tidak sampai piksel paling akhir.

#### 4.2.2 Perhitungan K-NN

Langkah selanjutnya setelah didapatkan fitur melalui proses *Local Binary Pattern* adalah proses klasifikasi atau pengenalan emosi terhadap kelas yang diklaim. Permisalan manualisasi menggunakan 32 data latih yang terdiri dari kelas-kelas emosi yang terdiri dari masing-masing lima data latih perkelas kecuali kelas takut yang hanya menggunakan dua data latih.

Langkah 1:

Nilai  $k$  pada proses ini adalah 5. Nilai  $k$  berfungsi untuk menentukan jumlah tetangga terdekat yang akan dijadikan kelompok saat pemilihan kelas.

Langkah 2:

Menghitung jarak ( $d$ ) antara data uji dengan data latih menggunakan perhitungan *Euclidean distance* dengan persamaan sebagai berikut:

$$d(x, y) = \sqrt{\sum_{r=1}^n (x_i - y_i)^2} \quad (4.1)$$

Keterangan:

$d(x, y)$ : jarak *Euclidean* antara  $x_i$  dengan  $y_i$

$x_i$ : data pada  $x$  ke- $i$  yang akan dilakukan perhitungan

$y_i$ : data pada  $y$  ke- $i$  yang akan dilakukan perhitungan

**Tabel 4.1 Jarak *Euclidean* hasil perhitungan**

Data	$\sqrt{(x_1 - y_1)^2}$	$\sqrt{(x_2 - y_2)^2}$	...	$\sqrt{(x_{255} - y_{255})^2}$	$\sqrt{(x_{256} - y_{256})^2}$	$d(x, y)$	kelas
1	441	1156	...	95	379	78138	lainnya
2	25	1	...	111	398	41806	lainnya
...	...	...	...	...	...	...	...
30	679	95	...	86	118	192148	kesedihan
31	632	60	...	94	100	56456	takut
32	712	80	...	88	141	131664	takut

Langkah 3:

Setelah didapatkan jarak antara data uji dan data latih, selanjutnya data diurutkan secara *ascending* yaitu data diurutkan dari jarak terkecil. Kemudian dipilih sejumlah  $k$  data terkecil.

**Tabel 4.2 Data kelompok tetangga terdekat**

Data	$d(x,y)$	Kelas
4	14058	lainnya
11	23148	kebahagiaan
7	24070	Jijik
2	41806	Lainnya
3	42312	Lainnya

Langkah 4:

Penentuan kelas yang sering muncul pada kelompok tetangga terdekat. Dari Tabel 4.2 didapatkan bahwa kelas yang sering muncul ialah kelas jijik sehingga hasil dari program ialah ekspresi wajah yang dibentuk adalah ekspresi wajah lainnya.