

## BAB 2

### TINJAUAN PUSTAKA

#### 2.1 Citra Digital

Gambar (Citra) dapat didefinisikan sebagai fungsi dua dimensi.  $f(x,y)$ , di mana  $x$  dan  $y$  adalah spasial koordinat (plane). dan amplitudo  $f$  pada setiap pasangan koordinat  $(x,y)$  disebut intensitas tingkat abu-abu dari gambar pada titik tersebut [9]. Menurut Darma Putra [10], Citra Digital merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu.

Pada umumnya citra berbentuk empat persegi panjang dan dimensi ukurannya dinyatakan sebagai (tinggi x lebar). Citra dengan tinggi  $N$  piksel, lebarnya  $M$  piksel, dan memiliki intensitas  $f$  dapat direpresentasikan sebagai suatu matriks yang berukuran  $N$  baris dan  $M$  kolom, dengan  $x$  dan  $y$  adalah koordinat spasial, dan amplitudo  $f$  di titik koordinat  $(x, y)$  dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. apabila nilai  $x, y$  dan nilai amplitudo  $f$  secara keseluruhan berhingga (finite) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital.

Citra digital dapat ditulis dalam bentuk matrik pada Persamaan (2.1).

$$(x, y) = [f(0,0) f(1,0) : f(0,1) f(1,1) \quad (2.1)$$
$$\quad \quad \quad \vdots f(M - 1,0) f(M - 1,1) \quad \dots \dots$$
$$\quad \quad \quad \vdots f(0, N - 1) f(1, N - 1) : \dots f(M - 1, N - 1) ]$$

Indeks baris ( $x$ ) dan indeks kolom ( $y$ ) menyatakan suatu koordinat titik pada citra. Nilai pada suatu irisan antara baris dan kolom disebut dengan picture elements, image elements, pels, atau pixels, sedangkan  $f(x, y)$  merupakan intensitas (derajat keabuan) pada titik  $(x, y)$ .

#### 2.2 Ekspresi Wajah








Menurut Chiara Fiorentini dkk [11], ekspresi wajah merupakan cara pengungkapan atau proses menyatakan maksud tertentu seperti sedih, bahagia, terkejut, takut, marah dan *bad mood*. Menurut Z Abidin, Ekspresi wajah dan yang terkait dengan perubahan pola wajah

memberikan informasi mengenai keadaan emosional orang dan membantu mengatur percakapan dengan orang tersebut [12].

### 2.2.1 Jenis Ekspresi Wajah Yang Digunakan

Ekspresi wajah yang digunakan ada 7[2], berikut adalah gambar dari ekspresi wajah yang digunakan :

*Tabel 2.1 jenis wajah yang digunakan*

No	Ekspresi	Gambar Wajah
1	Marah	
2	Sedih	
3	Senang	
4	Terkejut	
5	Takut	
6	Jijik	
7	Netral	

## 2.3 Jenis Citra

Jenis citra yang digunakan pada pemrosesan citra, umumnya dibagi tiga. Ketiga citra tersebut yaitu Citra berwarna, Citra berskala keabuan dan Citra biner.

### 2.3.1 Citra Berwarna

Citra berwarna atau biasa disebut Citra RGB, Menyajikan tiga komponen warna, Yaitu: R(merah), G(hijau), dan B(biru). Komponen - komponen warna tersebut menggunakan 8 bit

dimana nilainya berkisar antara 0 sampai dengan 255. Sehingga, Nilai yang dihasilkan mencapai  $255 \times 255 \times 255$  atau 16.581.375 warna. menunjukkan contoh warna R, G dan B [13].

*Tabel 2.2 Citra Berwarna RGB*

Warna	R	G	B
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0
Putih	255	255	255
Kuning	0	255	255

### 2.3.2 Citra berskala keabuan

Citra berskala merupakan citra dengan komponen warna hitam dan putih, yang menghasilkan efek warna abu-abu. Dengan nilai komponennya 0 dan 255. Nilai 0 yang berarti hitam dan 255 adalah putih [13].

### 2.3.3 Citra Biner

Citra biner merupakan citra dengan piksel yang terbagi menjadi 2 Nilai yaitu nilai 0 dan 1. Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Citra biner sendiri sering digunakan dalam pemrosesan citra dalam memperoleh tepi objek [13].

## 2.4 Pengenalan Pola

Pola adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Pola bisa merupakan kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan dalam notasi vektor atau matriks. Secara umum pengenalan pola (pattern recognition) adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek (sitasi) [10]. Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. ciri yang baik adalah ciri yang memiliki daya pembeda yang tinggi sehingga pengenalan pola berdasarkan ciri yang dimiliki dapat dilakukan

dengan baik. pengenalan pola ini bersifat conceptually driven processing yang berarti bahwa proses dimulai dari pembentukan konsep pada objek yang dijumpai.

Menurut Theodoridis, Pengenalan pola dapat diartikan sebagai proses klasifikasi dari objek atau pola menjadi beberapa kategori atau kelas dan bertujuan untuk pengambilan keputusan [14]. Ada banyak aplikasi yang bisa dijadikan implementasi dari pengenalan pola. diantaranya pengenalan wajah manusia, pengenalan gambar mata, pengenalan penyakit berdasarkan gejala-gejala yang ditemukan pada objek dan lain-lain. Pengenalan pola bisa dibuat melalui pendekatan pemrosesan citra yang mana tujuan akhir dari pemrosesan citra tersebut digunakan untuk pengelompokan objek sehingga menghasilkan output yang diinginkan.

## 2.5 Resizing

Resizing adalah proses perubahan ukuran citra, baik memperbesar maupun memperkecil resolusi citra [15]. Proses ini sering dilakukan pada pengolahan citra untuk normalisasi ukuran citra digital yang akan diproses sehingga memiliki ukuran yang sama dan mempercepat proses pengolahan citra ketika ukuran citra diperkecil.

Untuk merubah ukuran citra asli menjadi citra lebih kecil menggunakan perhitungan sebagai berikut:

$$ResizeW = w \times 0.5 \quad (2.2)$$

$$ResizeH = h \times 0.5 \quad (2.3)$$

Dimana:

$ResizeW$  = ukuran lebar citra baru (*width*)

$ResizeH$  = ukuran panjang citra baru (*Height*)

$w$  = ukuran lebar citra asli (*width*)

$h$  = ukuran panjang citra asli (*Height*)

## 2.6 Grayscale

*Grayscale* merupakan proses untuk mengubah citra berwarna (RGB) menjadi citra keabuan yang berfungsi untuk mengecilkan range warna menjadi 0 sampai dengan 255. Berikut adalah rumus konversi citra berwarna (RGB) menjadi grayscale pada Persamaan (2.4).

$$I = (0.2989 * R) + (0.5879 * G) + (0.1141 * B)$$

(2.4)

Dimana:

I = Nilai Grayscale

R= komponen nilai merah (Red) dari suatu titik pixel

G = komponen nilai hijau (Green) dari suatu titik pixel

B = komponen nilai biru (Blue) dari suatu titik pixel

## 2.7 Smoothing

Proses Smoothing digunakan untuk meningkatkan gambar yang *noise* (dengan mengorbankan keburaman). Filter ini menghasilkan rata-rata pada area 3 x 3 gambar. Teknik tersebut disebut juga moving window averaging.

$$\frac{1}{9} \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Gambar 2.1 gambar rumus matriks smoothing

## 2.8 Segmentasi Citra

Segmentasi citra bertujuan mendapatkan objek-objek citra dengan cara membagi citra ke dalam beberapa daerah yang memiliki kemiripan atribut [13].

Terdapat dua teknik segmentasi citra yaitu mendeteksi diskontinuitas dan similaritas. Pendekatan diskontinuitas membagi citra berdasarkan perubahan intensitas yang tiba-tiba seperti deteksi titik, deteksi garis, dan deteksi tepi. Sedangkan pendekatan similaritas memecah citra ke dalam daerah yang sama menurut beberapa kriteria yang sudah ditentukan, seperti proses thresholding, region growing, region splitting dan merging.

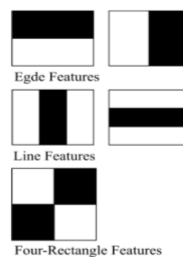
Pada penelitian ini segmentasi citra pada wajah memanfaatkan fitur *Haar like Feature* pada metode *viola and jones* untuk menentukan bagian wajah pada citra.

## 2.9 Deteksi Wajah (*Viola-Jones*)

Metode *Viola-Jones* merupakan metode pendeteksian objek yang memiliki tingkat keakuratan tinggi sekitar 93,7%. Metode ini diusulkan oleh Paul Viola dan Michael Jones pada tahun 2001 [16]. Proses-proses yang ada pada *Viola-Jones* meliputi *Haar-Like Feature*, *Integral Image*, *Adaboost*, dan *Cascade Classifier*. Adapun detail dari setiap tahap yang dilalui dari sebuah citra saat proses pendeteksian wajah menggunakan algoritma *Viola-Jones* sebagai berikut :

### 2.9.1 *Haar Like Feature*

*Haar Like Feature* adalah teknik yang dilakukan dengan cara menyeleksi citra dengan cara mengkotak-kotakkan setiap daerah pada citra dari mulai ujung kiri atas sampai kanan bawah. Proses ini dilakukan untuk mencari apakah ada fitur wajah pada area tersebut. Dalam algoritma *Viola and Jones*, ada beberapa jenis fitur yang bisa digunakan seperti *Edge-feature*, *Line feature*, dan *Four-rectangle feature*. dan berikut adalah contoh fitur yang terdapat dalam fitur *haar like feature*:



Gambar 2.2 Macam – macam variasi *Feature* pada *Haar*

Nilai *Haar-Like Feature* diperoleh dari selisih jumlah nilai piksel daerah gelap dengan jumlah nilai piksel daerah terang:

$$F (Haar) = \sum F_{White} - \sum F_{Black} \quad (2.5)$$

Dimana:

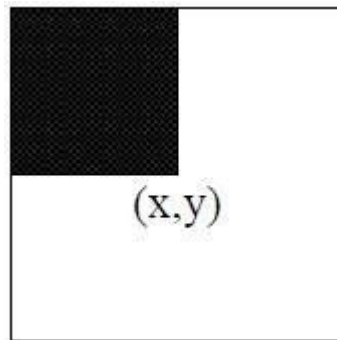
$F (Haar)$  = Nilai fitur total

$\sum F_{White}$  = Nilai fitur pada daerah terang

$\sum F_{Black}$  = Nilai fitur pada daerah gelap

### 2.9.2 *Integral Image*

*Integral image* sering digunakan pada algoritma untuk pendeteksian wajah. Dengan menggunakan *integral image* proses perhitungan bisa dilakukan hanya dengan satu kali scan dan memakan waktu yang cepat dan akurat. *Integral image* digunakan untuk menghitung hasil penjumlahan nilai piksel pada daerah yang dideteksi oleh fitur haar. Berikut adalah simulasi dari *integral image*.



*Gambar 2.3 Ilustrasi Integral Image*

Berdasarkan **Gambar 2.3** citra integral pada titik  $(x,y)$  ( $ii(x,y)$ ) dapat dicari menggunakan persamaan (2.3)

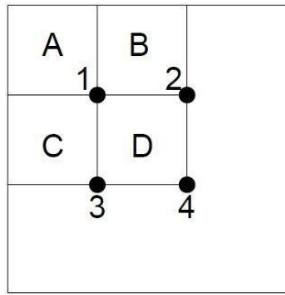
$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (2.6)$$

Dimana:

$ii(x,y)$  = Citra integral pada lokasi  $x, y$

$i(x',y')$  = Nilai piksel pada citra asli

Perhitungan nilai dari suatu fitur dapat dilakukan secara cepat dengan menghitung nilai citra integral pada empat buah titik sebagaimana disajikan dalam Gambar 2.4



Gambar 2.4 Ilustrasi perhitungan Integral Image

Jika nilai *integral image* titik 1 adalah A, titik 2 adalah A+B, titik 3 adalah A+C, dan di titik 4 adalah A+B+C+D, maka jumlah piksel di daerah D dapat diketahui dengan cara  $4+1-(2+3)$ .

Algoritma *Adaboost learning*, digunakan untuk meningkatkan kinerja klasifikasi dengan pembelajaran sederhana untuk menggabungkan banyak *classifier* lemah menjadi satu *classifier* kuat. *Classifier* lemah adalah suatu jawaban benar dengan tingkat kebenaran yang akurat.

### 2.9.3 Adaptive Booster

*Adaptive Booster* adalah proses untuk memperbaharui bobot dengan melakukan perhitungan menggunakan persamaan jumlah gambar negatif dan jumlah gambar positif. Algoritma tersebut mengkombinasikan *performance* banyak *weak classifier* untuk menghasilkan *strong classifier*. *Weak classifier* dalam hal ini adalah nilai dari *haar-like feature* [17].

Bobot awal:

$$w_{j_{1_{y_i}}} = \frac{1}{2m}, w_{j_{1_{y_i}}} = \frac{1}{2l} \quad (2.7)$$

Citra Positif:

$$\epsilon_t = \left( \sum_t w_{t,i} \right) |h_t(x) - y_i| \quad (2.8)$$

Citra Negatif:



$$\epsilon_j = \left( \sum_j^J w_{t,i} \right) |h_j(x) - y_i| \quad (2.9)$$

Jika,

$$\epsilon_t \forall \epsilon_j < 0 \quad (2.10)$$

Hentikan iterasi.

Dimana:

$w$  = *week classifier*

$m$  = jumlah citra positif

$l$  = jumlah citra negatif.

$t$  = indeks iterasi dari citra positif.

$j$  = indeks iterasi dari citra negatif.

$h_t(x)$  = nilai fitur citra positif.

$h_j(x)$  = nilai fitur citra negative

Hasil Akhir klasifikasi yang diharapkan pada citra positif adalah sebagai berikut:

$$H(x) = \begin{cases} 1 & \sum_{j=1}^J \alpha_j h_j \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{bukan objek.} \end{cases} \quad (2.11)$$

Dimana:

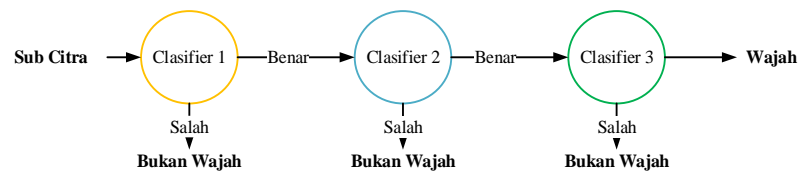
$$\alpha_j = \log \frac{1}{\beta_t}, \alpha_t = \log \frac{1}{\beta_t} \quad (2.12)$$

Jika posisi  $H(x)$  = Ketentuan 1 maka citra tersebut merupakan objek

Jika posisi  $H(x) = \text{Ketentuan } 0$  maka citra tersebut merupakan bukan objek  
 $H(x) = \text{Strong Classifier}$  atau klasifikasi yang menyatakan objek atau bukan  
 $\alpha_j =$  Tingkat pembelajaran citra positif.  
 $\alpha_t =$  Tingkat pembelajaran citra negatif.  
 $\beta_j =$  Nilai bobot setelah *error rate* pada citra negatif  
 $\beta_t =$  Nilai bobot setelah *error rate* pada citra positif  
 $H_j =$  *weak* atau *basic classifiers* (awal dari klasifikasi) citra negatif.  
 $H_t =$  *weak* atau *basic classifiers* (awal dari klasifikasi) citra positif.

### 2.9.4 Cascade Classifier

Pada proses *cascade classifier* yang merupakan metode untuk mengkombinasikan *classifier* yang kompleks dalam sebuah struktur yang bertingkat dan dapat meningkatkan kecepatan pendeteksian sebuah objek pada citra yang memfokuskan pada daerah citra yang berpeluang saja, dan berikut adalah proses dari *cascade classifier*:



Gambar 2.5 Cascade Clasifier dengan N stages.

### 2.10 Thresholding

*Thresholding* mengubah citra keabuan menjadi citra biner bergantung pada nilai threshold (T) sehingga dapat diketahui daerah mana yang termasuk objek dan latar belakang [18]. Jika nilai pixel lebih besar dari threshold diatur menjadi 1 sebaliknya jika kurang dari threshold diatur menjadi 0. Secara umum proses thresholding ditunjukkan pada Persamaan (2.13).

$$g(x, y) = \{1, \text{jika } f(x, y) \geq T \ 0, \text{jika } f(x, y) < T \quad (2.13)$$

Persamaan (2.10) menjelaskan bahwa  $g(x, y)$  adalah citra biner dari citra aras keabuan  $f(x, y)$  dan T menyatakan nilai ambang.

## 2.11 Ekstraksi Fitur

Ekstraksi fitur adalah proses untuk mendapatkan fitur utama yang terdapat pada citra. Pada penelitian ini menggunakan Ekstraksi fitur bentuk berupa metode PCA.

### 2.11.1 Metode PCA

PCA adalah teknik statistik untuk menyederhanakan kumpulan data banyak-dimensi menjadi dimensi yang lebih rendah (extraction feature). Menurut Jolliffe, Gagasan utama dari PCA adalah untuk mengurangi dimensi suatu set data di mana ada sejumlah angka yang besar dari variabel yang saling terkait, sambil mempertahankan sebanyak mungkin variasi yang ada dalam set data [19].

Untuk mendapatkan nilai Eigenvector memerlukan beberapa tahap seperti menentukan nilai rata-rata, adjusted data, hitung nilai matriks kovarian, hitung nilai *Eigenvalue* dan *Eigenvector*, dan tahap terakhir yaitu hitung nilai PCA. Rumus perhitungannya sebagai berikut :

1. Hitung rata-rata seluruh sampel data diperoleh dengan menggunakan persamaan:

$$\bar{x}_j = \frac{\sum_{ij}^n x_{ij}}{n} \quad (2.14)$$

Dimana:

$\bar{x}$  = Rata-rata

$n$  = Banyak data

$x$  = Matrix data

2. Adjusted data (data yang telah disesuaikan) adalah hasil pengurangan dari setiap data dengan rata-rata setiap data yang diperoleh dengan rumusan berikut ini:

$$x' = x_{ij} - \bar{x}_j \quad (2.15)$$

Dimana:

$x'$  = Adjusted data

$\bar{x}$  = Rata-rata

$x$  = Matriks data

3. Mencari nilai matriks kovarian:

$$C = \frac{1}{M-1} x'^T x' \quad (2.16)$$

Dimana:

$C$  = Nilai kovarian matriks

$M$  = Banyak baris

$x'^T$  = Nilai transpose dari matrix  $x$

$x'$  = Adjusted data

4. Menghitung nilai *Eigen* dan *Eigenvector*:

$$Cv = \lambda Iv \quad (2.17)$$

$$Cv - \lambda Iv = 0$$

$$(C - \lambda I)v = 0 \quad (2.18)$$

$$|C - \lambda I| = 0$$

Dimana:

$C$  = nilai kovarian matriks

$v$  = vector eigen

$\lambda$  = nilai eigen

$I$  = matriks identitas

5. Hitung nilai eigen yang terbesar yang berkorespondensi terhadap nilai vector eigen yang terbesar dipilih menjadi Principal Component

$$v = (eig_1, eig_2, eig_3, \dots, eig_n) \quad (2.19)$$

6. Untuk mencari Principal Component dengan  $x'$  sebagai rata-rata

$$PC = X' x v \quad (2.20)$$

Dimana :

PC = Principal Component

$X'$  = rata – rata

## 2.12 SVM Classifier

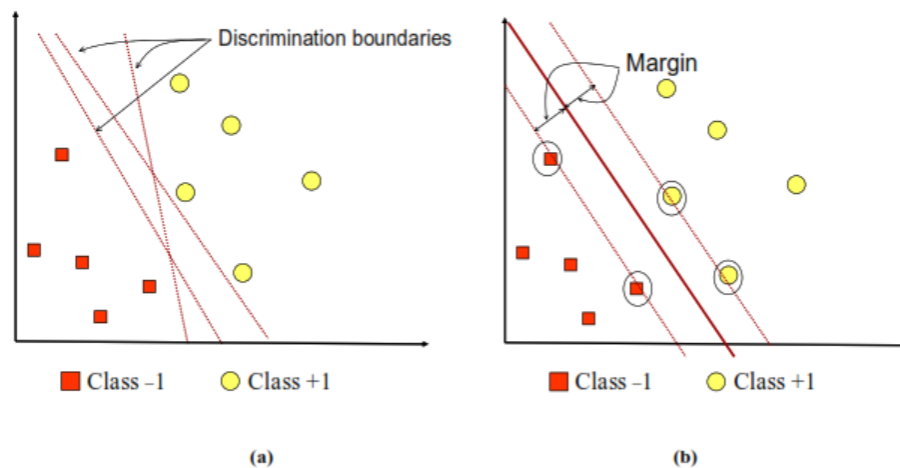
Support Vector Machine merupakan sebuah metode yang membandingkan suatu seleksi parameter standar nilai diskrit yang disebut kandidat set. Untuk mengklasifikasikan akurasi Support Vector Machine (SVM) diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992 [20].

Metode Support Vector Machine memiliki 5 komponen yang berfungsi di antaranya:

- SVM Linear
- SVM Polynomial
- Kernel RBF (*Radial Basis Function*)
- Kernel MLP (*Multi Layer Perceptron*)
- Tangent Hyperbolic (*sigmoid*)

SVM dapat mengklasifikasikan data linier dan non linier. Variabel-variabel predictor merupakan data input sedangkan variabel target yang saling bergantung merupakan output. SVM bertujuan untuk menemukan fungsi klasifikasi terbaik dan untuk membedakan antara anggota dari dua kelas dalam data latih. Matrik untuk konsep fungsi klasifikasi "terbaik" dapat diwujudkan secara geometris. Untuk dataset terpisah secara linear, fungsi klasifikasi linier berhubungan dengan hyperplane pemisah  $f(x)$  yang melewati tengah dua kelas, memisahkan keduanya [21].

Model algoritma SVM merupakan salah satu algoritma dari metode klasifikasi, yang bekerja dengan cara mencari suatu garis (hyperplane) untuk memisahkan dua kelompok data. Berikut ini adalah contoh berdasarkan pada Gambar 2.3 tentang bagaimana SVM mencoba menemukan hyperplane terbaik untuk memisahkan kelas -1 dan +1



Gambar 2.6 SVM mencoba untuk menemukan hyperplane terbaik yang memisahkan kedua class -1 dan +1

Gambar 2.3 menunjukkan beberapa pola yang merupakan anggota dari dua kelas, yaitu -1 dan +1. Pola di kelas -1 dilambangkan dengan warna merah (kotak), sedangkan pola di kelas +1 dilambangkan dengan warna kuning (bulat). Masalah klasifikasi dapat diselesaikan dengan mencoba mencari garis (hyperplane) yang memisahkan dua kelas. Gambar 1-a menunjukkan batas diferensiasi alternatif yang berbeda. Hyperplane dengan pemisah terbaik dapat ditemukan dengan mengukur margin dari hyperplane dan mencari titik maksimum. Kernel harus digunakan untuk mencapai keberhasilan banyak algoritma klasifikasi untuk permukaan linear [22].

Dalam linear SVM, pemisah merupakan fungsi linear. Data latih dinyatakan oleh  $(x_i, y_i)$  dan  $x_i = \{x_1, x_2, \dots, x_{iq}\}$  merupakan atribut (fitur) set untuk data latih kelas ke- $i$ . Untuk

$y_i \in \{-1, 1\}$  menyatakan label kelas. Pendefinisian persamaan suatu *hyperplane* pemisah yang dituliskan dengan :

$$w \cdot x_i + b = 0 \quad (2.21)$$

Data  $x_i$  yang terbagi ke dalam dua kelas, yang termasuk kelas -1 (sampel negatif) didefinisikan sebagai vektor yang memenuhi pertidaksamaan (2.22) berikut ini:

$$w \cdot x_i + b < 0 \text{ untuk } y_i = -1 \quad (2.22)$$

Sedangkan yang termasuk kelas +1 (sampel positif) memenuhi pertidaksamaan (2.23) berikut:

$$w \cdot x_i + b < 0 \text{ untuk } y_i = +1 \quad (2.23)$$

Dimana:

$x_i$  = data input

$y_i$  = label yang diberikan

$w$  = nilai dari bidang normal

$b$  = posisi bidang relatif terhadap pusat koordinat

Parameter  $w$  dan  $b$  adalah parameter yang akan dicari nilainya. Bila label data  $y_i = -1$ , maka pembatas menjadi persamaan (2.24) berikut ini :

$$w \cdot x_i + b \leq -1 \quad (2.24)$$

Bila label data  $y_i = +1$ , maka pembatas menjadi persamaan (2.25) berikut:

$$w \cdot x_i + b \geq +1 \quad (2.25)$$

Margin terbesar dapat dicari dengan cara memaksimalkan jarak antar bidang pembatas kedua kelas dan titik terdekatnya, yaitu  $\frac{2}{|w|}$ . Hal ini dirumuskan sebagai permasalahan *Quadratic Programming (QP) problem* yaitu mencari titik minimal persamaan (2.26) dengan memperhatikan persamaan (2.27) berikut :

$$\tau(w) = \frac{1}{2} \|w\|^2 \quad (2.26)$$

$$y_i(w \cdot x_i + b) - 1 \geq 0, (i = 1, \dots, n) \quad (2.27)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya *Lagrange Multiplier* dengan menggunakan persamaan pada (2.28), dan menyederhanakannya menjadi persamaan (2.29) berikut :

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i (y_i ((w^T x_i + b) - 1)) \quad (2.28)$$

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) + \sum_{i=1}^n a_i \quad (2.29)$$

Dimana  $a_i$  adalah *lagrange multiplier* yang bernilai nol atau positif ( $a_i \geq 0$ ). Nilai optimal dari persamaan (2.28) dapat dihitung dengan meminimalkan  $L$  terhadap  $w$ ,  $b$  dan  $a$  dapat dilihat pada persamaan (2.30) sampai (2.32) berikut :

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^n a_i y_i x_i = 0 \quad (2.30)$$



$$\frac{\partial L}{\partial b} = \sum_{i=1}^n a_i y_i = 0 \quad (2.31)$$

$$\frac{\partial L}{\partial a} = \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i = 0 \quad (2.32)$$

Maka masalah *Lagrange* untuk klasifikasi dapat dinyatakan pada persamaan (2.33) berikut :

$$\text{Min } L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n a_i y_i (w^T x_i + b) - \sum_{i=1}^n a_i \quad (2.33)$$

Dengan memperhatikan persamaan (2.34) dan (2.35) berikut ini :

$$w - \sum_{i=1}^n a_i x_i y_i = 0 \quad (2.34)$$

$$\sum_{i=1}^n a_i x_i y_i = 0 \quad (2.35)$$

Model persamaan (2.34) di atas merupakan model primal *Lagrange*. Sedangkan dengan memaksimalkan  $L$  terhadap  $a_i$ , persamaannya menjadi persamaan (2.36) berikut ini :

$$\text{Max } \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j^T x_i x_j^T \quad (2.36)$$

Dan memperhatikan persamaan (2.37) berikut ini

$$\sum_{i=1}^n a_i y_i = 0, a_i \geq 0 (i, j = 1, \dots, n) \quad (2.37)$$

Untuk mencari nilai  $x_i$  dan  $y_i$  dapat dilakukan ketika sudah didapatkan nilai dari pembobotan dari masing-masing metode seleksi fitur dan inisialisasi kelas. Hasil dari pembobotan diubah ke dalam bentuk format data  $svm(x)$ , sedangkan data kelas menjadi label data  $svm(y)$ .

Untuk mendapatkan nilai  $a_i$ , langkah pertama adalah mengubah setiap *feature* data *training* menjadi nilai vektor (*support vector*) =  $x y$ . Kemudian nilai vektor dari setiap data *training* dimasukkan ke persamaan (2.38) *kernel trick* phi ( $\varphi$ ) berikut ini :

$$\varphi[x y] = \begin{cases} \sqrt{x_n^2 + y_n^2} > 2, \text{ maka } [\sqrt{x_n^2 + y_n^2} - x + |x - y| \\ y|\sqrt{x_n^2 + y_n^2} - x + |x - y|] \sqrt{x_n^2 + y_n^2} < 2, \text{ maka } [x y] \end{cases} \quad (2.38)$$

Nilai  $x$  didapatkan dari persamaan *kernel linear* untuk  $x$  berikut :

$$\sum_{i=1}^n x_i^T x_j \quad (2.39)$$

Untuk mendapatkan jarak tegak lurus yang optimal dengan mempertimbangkan vektor positif, maka hasil perhitungan dari substitusi nilai  $x$  dan nilai  $y$  ke persamaan (2.31) diberi nilai bias =1. Kemudian cari parameter  $a_i$ , dengan terlebih dahulu mencari nilai fungsi setiap data *training* menggunakan persamaan (2.32), lalu mencari nilai  $a_i$  pada persamaan linear menggunakan persamaan (2.40) dengan memperhatikan  $i, j = 1, \dots, n$  berikut :

$$\sum_{i=1}^n a_i T_i^T T_j \quad (2.40)$$

$$\sum_{i=1}^n a_i T_i^T T_j = y_i \quad (2.41)$$

Setelah parameter  $a_i$  didapatkan, kemudian masukkan ke persamaan (2.42) berikut :

$$w = \sum_{i=1}^n a_i T_i \quad (2.42)$$

Terakhir akan dicari nilai  $w$  dan  $b$  untuk menemukan *hyperplane* sebagai patokan proses klasifikasi dengan persamaan sebagai berikut :

$$y = wx + b, w = \sum_i^n a_i s_i \quad (2.43)$$

Nilai  $s_i$  merupakan nilai *support vector* yang telah dihitung sebelumnya. Dengan demikian proses klasifikasi selesai dengan memperhatikan *hyperplane* nya

### 2.12.1 Kernel Trick

*Feature space* merupakan dimensi yang lebih tinggi dari vektor input (*input space*). Hal ini mengakibatkan komputasi pada *feature space* mungkin sangat besar, karena ada kemungkinan *feature space* dapat memiliki jumlah *feature* yang tidak terhingga. Selain itu, sulit mengetahui fungsi transformasi yang tepat. Untuk mengatasi masalah ini, pada SVM digunakan "kernel trick". Fungsi kernel yang umum digunakan adalah sebagai berikut [23]:

#### 1. Kernel linier

$$K(x_i, x) = x_i^T x \quad (2.44)$$

#### 2. Polynomial kernel

$$K(x_i, x) = (\gamma \cdot x_i^T x + r)^p, \gamma > 0 \quad (2.45)$$

#### 3. Radial basis function (RBF)

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2), \gamma > 0 \quad (2.46)$$

#### 4. Sigmoid kernel

$$K(x_i, x) = \tanh(\gamma x_i^T x + r) \quad (2.47)$$

### 2.12.2 SVM Multiclass

SVM hanya dapat mengklasifikasikan data ke dalam dua kelas (klasifikasi biner). Namun, penelitian lebih lanjut untuk mengembangkan SVM sehingga bisa mengklasifikasi data yang memiliki lebih dari dua kelas, terus dilakukan. Ada dua pilihan untuk mengimplementasikan SVM *multiclass* yaitu dengan menggabungkan beberapa SVM biner atau menggabungkan semua data yang terdiri dari beberapa kelas ke dalam sebuah bentuk permasalahan optimasi. Namun, pada pendekatan yang kedua permasalahan optimasi yang harus diselesaikan jauh lebih rumit. Berikut ini adalah metode yang digunakan untuk mengimplementasikan *multiclass* SVM dengan pendekatan Metode *one-against-all*. Contohnya, terdapat permasalahan klasifikasi dengan 7 buah kelas. Untuk pelatihan digunakan 7 buah SVM biner penggunaanya

dalam mengklasifikasi kelas pada data baru dapat dilihat pada persamaan sebagai berikut [24]:

$$\text{Kelas } x = \arg \max_{i=1..k} ((w^{(i)})^T \cdot \varphi(x) + b^{(i)}) \quad (2.48)$$

Dengan menentukan *hyperplane* terbesar pada nilai x maka akan mengklasifikasikan kelas tersebut.

Tabel 2.3 Contoh 7 SVM biner dengan metode One-against-all

$y_i = 1$	$y_i = -1$	Hipotesis
Kelas 1	Bukan kelas 1	$f^1(x)$ $= (w^1)x + b^1$
Kelas 2	Bukan kelas 2	$f^2(x)$ $= (w^2)x + b^2$
Kelas 3	Bukan kelas 3	$f^3(x)$ $= (w^3)x + b^3$

Kelas 4	Bukan kelas 4	$f^4(x)$ $= (w^4)x + b^4$
Kelas 5	Bukan kelas 5	$f^5(x)$ $= (w^5)x + b^5$
Kelas 6	Bukan kelas 6	$f^6(x)$ $= (w^6)x + b^6$
Kelas 7	Bukan kelas 7	$f^7(x)$ $= (w^7)x + b^7$

### 2.13 K-fold Cross Validation

K-fold Cross Validation adalah teknik untuk melakukan validasi pada dataset untuk menemukan akurasi yang baik. Teknik ini membagi dataset sebanyak k subset. Satu dari subset ini akan dijadikan sebagai data uji dan k-1 subset sisanya digunakan untuk proses data latih. Proses ini dilakukan sebanyak k kali sehingga setiap subset akan menjadi data uji dari model. Proses ini akan mendapatkan k buah nilai performa dari proses pembelajaran. Semua nilai performa ini akan dicari rata-ratanya dan nilai dengan rata-rata tertinggi akan dipilih sebagai model. k-fold cross validation memiliki kelebihan dapat mengklasifikasi dataset lebih efisien, namun metode ini memiliki kelemahan dalam proses komputasi yang digunakan akan lebih besar karena akan melakukan proses sebanyak k kali [24].

### 2.14 Akurasi

Sebuah sistem yang melakukan identifikasi atau klasifikasi diharapkan mampu melakukan klasifikasi semua data set dengan benar. Namun tidak dapat dipungkiri bahwa kinerja suatu sistem yang melakukan klasifikasi tidak akan selalu bisa 100% benar. Oleh karena itu, sistem harus diukur kinerjanya. Umumnya cara mengukur kinerja klasifikasi menggunakan *confusion matrix* [25].

Pengukuran terhadap kinerja suatu sistem klasifikasi merupakan hal yang penting. Kinerja sistem klasifikasi menggambarkan seberapa baik sistem dalam mengklasifikasikan data. Confusion matrix merupakan salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi. Pada dasarnya confusion matrix mengandung informasi yang

membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya. Pada pengukuran kinerja menggunakan confusion matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Nilai True Negative (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan False Positive (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, True Positive (TP) merupakan data positif yang terdeteksi benar. False Negative (FN) merupakan kebalikan dari True Positive, sehingga data positif, namun terdeteksi sebagai data negatif. Pada jenis klasifikasi biner yang hanya memiliki 2 keluaran kelas, confusion matrix dapat disajikan seperti pada Tabel 2.4.

*Tabel 2.4 Confusion Matrix*

Kelas		Prediksi	
		Positive	Negative
Target	Positive	TP	FP
	Negative	FN	TN

Berdasarkan nilai True Negative (TN), False Positive (FP), False Negative (FN), dan True Positive (TP) dapat diperoleh nilai akurasi. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan persamaan (2.49).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} * 100\% \quad (2.49)$$

## 2.15 Penelitian Terkait

Beberapa penelitian terdahulu yang berhubungan dengan penelitian ini, diantaranya :

Sebuah penelitian pengenalan citra ekspresi wajah menggunakan *Viola-Jones* sebagai deteksi wajahnya kemudian menggunakan LBP (Local Binary Pattern) sebagai pengenalan

tekstur dan menggunakan SVM sebagai classifier, berdasarkan hasil uji coba diperoleh hasil terbaik mencapai 80% [26]

Dalam sebuah penelitian pengenalan ekspresi citra wajah yang menggunakan *polar angular radial transform* sebagai metode dalam tahap preprocessing, yang dimana metode ini hampir menyerupai dari *Zernike* momen. Menggunakan PCA sebagai ekstraksi fiturnya dalam mengekstraksi karakteristik wajahnya kemudian menggunakan metode Support Vector Margin yang berasal dari Support Vector Machine sebagai classifier nya, berdasarkan hasil uji coba yang diperoleh dari hasil penelitian tersebut mendapatkan hasil terbaik dan mendapatkan akurasi sebesar 82,52% menggunakan metode yang sama dari dataset yang digunakan yaitu dari JAFFE [27].

Dalam sebuah penelitian pengenalan ekspresi citra wajah dengan menggunakan 2 model CNN menjadi metode utamanya yaitu fungsi PReLU (*Parametric Rectified Linear Unit*), dan fungsi *Loss*. Pada dasarnya metode ini digunakan untuk mereduksi ukuran gambar dataset agar lebih kecil, kemudian menggunakan *deep separable convolution layers* untuk proses ekstraksi fitur, berdasarkan hasil uji coba yang diperoleh dari hasil penelitian tersebut mendapatkan hasil sebesar 73,244% [28].

Dalam sebuah penelitian mengenai pengenalan ekspresi citra wajah dengan menggunakan *discrete wavelet transform* (DWT) sebagai ekstraksi fiturnya dan Fuzzy K-Nearest Neighbour sebagai classifier nya, DWT termasuk kedalam proses smoothing dalam ekstraksi fitur dengan memanfaatkan filter low-pass dan filter high-pass untuk dekomposisi gambar dataset yang kemudian diklasifikasi oleh Fuzzy KNN. Berdasarkan hasil uji coba dari penelitian tersebut mendapatkan nilai hasil rata – rata akurasi sebesar 77,49% dengan menggunakan 3 jenis ekspresi wajah dalam penelitian [29].

Dalam pengenalan penelitian mengenai pengenalan ekspresi citra wajah dengan menggunakan metode tapis gabor 2-D sebagai ekstraksi fitur dan SVM sebagai classifier. Dalam penelitian tersebut juga menggunakan metode *Viola-Jones* sebagai deteksi wajahnya, berdasarkan hasil uji coba dari penelitian tersebut mendapatkan nilai akurasi sebesar 85,92% Dari kategori akurasi [30].

Dari berbagai hasil penelitian tersebut sebagian besar peneliti yang meneliti tentang pengenalan ekspresi citra wajah memilih menggunakan metode K-Nearest Neighbour atau KNN dan Convolutional Neural Network atau CNN. Metode KNN dan CNN memang termasuk kedalam supervised learning yang merupakan proses pembelajaran yang terawasi.

Karena telah banyak peneliti sebelumnya yang telah membuktikan bahwa metode CNN dan KNN dapat digunakan untuk pengenalan ekspresi citra wajah, Adapun beberapa penelitian diatas menggunakan Support Vector Machine atau SVM untuk melakukan pengenalan ekspresi citra wajah akan tetapi dataset yang digunakan dari beberapa penelitian tersebut tidaklah banyak, maka pada penelitian menggunakan metode *SVM classifier* yang merupakan supervised learning juga akan tetapi menggunakan dataset yang lebih banyak dari penelitian yang sebelumnya. Oleh karena itu untuk proses pengenalan ekspresi citra wajah dalam penelitian ini akan menggunakan metode *SVM classifier* untuk proses pengenalan dan mengukur akurasi metode *SVM classifier* dalam mengenali ekspresi wajah.