

## BAB 2

### LANDASAN TEORI

#### 2.1 Bentuk Wajah

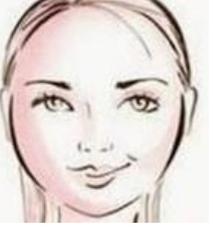
Bentuk wajah manusia merupakan salah satu ciri khas unik pada kepala yang dapat membedakan karakteristik pada manusia. Mengetahui bentuk wajah memiliki beberapa kegunaan yaitu dapat mengetahui bentuk kacamata yang cocok atau berguna untuk gaya hidup seperti riasan yang cocok ataupun rambut. Bentuk wajah juga dapat digunakan untuk observasi kebiasaan, mengetahui kepribadian, psikologi maupun kesehatan seseorang. [2]

Wajah sebagai objek dua dimensi digambarkan dengan berbagai macam illuminasi, pose dan ekspresi wajah untuk diidentifikasi berdasarkan citra dua dimensi dari wajah tersebut. Oleh sebab itu dengan melihat wajah, manusia dapat mengenali atau mengidentifikasi seseorang dengan mudah. [3]

Berdasarkan penelitian Wisuwat Sunhem dan Kitsuchart Pasupa [2], Bentuk wajah pada seseorang secara umum dibedakan dan diklasifikasi menjadi 6, yaitu berbentuk bulat, persegi, persegi panjang, hati, oval, berlian. Dengan ciri yang dapat dilihat pada tabel 2.1 berikut :

**Tabel 2. 1 Ciri Bentuk Wajah**

No	Gambar	Bentuk Wajah	Ciri
1		Hati	Lebar dahi lebih besar dari lebar garis rahang dan dagu mungkin lancip.
2		Persegi	Lebar dahi, tulang pipi, dan rahang sama dan ada garis rahang yang tajam.

3		Bulat	Lebar dahi, tulang pipi, dan rahang sama dan rahang sedikit bulat berlawanan dengan sudut.
4		Persegi Panjang	Ketinggian wajah sangat panjang dibandingkan dengan lebarnya
5		Oval	Rasio tinggi dan lebar adalah 3/2.
6		Berlian	Lebar dahi dan lebar rahang sama tetapi tulang pipinya lebih lebar dari mereka

Penelitian ini menggunakan 5 ciri yaitu bulat, oval, persegi, persegi panjang dan hati

## 2.2 Citra Digital

Citra digital adalah gambar dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) nilai digital yang disebut Pixel (picture elements). Pixel adalah elemen citra digital yang memiliki nilai yang menunjukkan intensitas warna. [14]

Berdasarkan cara penyimpanan atau pembentukannya, citra digital dapat dibagi menjadi dua jenis, jenis pertama adalah citra digital yang dibentuk oleh kumpulan pixel dalam array dua dimensi. Citra jenis ini disebut citra bitmap

(bitmap image) atau citra raster (raster image). Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor (vector graphic). [14]

Citra digital (diskrit) dihasilkan dari citra analog (kontinu) melalui digitalisasi. Digitalisasi citra analog terdiri atas penerokan (sampling) dan kuantisasi (quantization). Penerokan adalah pembagian citra ke dalam elemen elemen diskrit (pixel), sedangkan kuantisasi adalah pemberian nilai intensitas warna pada setiap pixel dengan nilai yang berupa bilangan bulat (Awcock, 1996) . [14]

Banyaknya nilai yang dapat digunakan dalam kuantisasi citra bergantung kepada kedalaman pixel, yaitu banyaknya bit yang digunakan untuk merepresentasikan intensitas warna pixel. Kedalaman pixel sering disebut juga kedalaman warna. Citra digital yang memiliki kedalaman pixel  $n$  bit disebut juga citra  $n$ -bit. [14]

### 2.3 Jenis Citra

Jenis citra dibagi menjadi tiga jenis citra yang umum digunakan untuk pemrosesan citra. Ketiga citra tersebut yaitu citra berwarna, citra berskala keabuan, dan citra biner.

#### 2.3.1 Citra Berwarna

Citra berwarna, atau biasa dinamakan Citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R(merah), G(hijau), dan B(biru). Tiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 sampai dengan 255). Dengan demikian, kemungkinan warna yang dapat disajikan mencapai  $255 \times 255 \times 255$  atau 16.581.375 warna. Tabel 2.2 menunjukkan contoh warna R, G dan B. [15]

**Tabel 2. 2 Warna 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 Grayscale

Citra grayscale atau citra berskala keabuan. Berguna untuk merubah citra berwarna menjadi citra berskala keabuan. Secara umum perubahan citra berwarna menjadi citra keabuan dengan menggunakan rumus:

$$I = 0.2989 * R + 0.5870 * G + 0.1141 * B \quad (2.1)$$

Dimana ( $R$ ) adalah nilai pada warna merah, ( $G$ ) adalah nilai pada warna hijau, dan ( $B$ ) adalah nilai pada warna biru. Sementara untuk (0.2989, 0.5870, 0.1141) adalah nilai mutlak.

### 2.3.3 Resize

Resize atau penskalaan adalah sebuah operasi geometri yang digunakan untuk memperbesar atau memperkecil ukuran dari sebuah citra sesuai dengan ukuran yang dibutuhkan. Pada penskalaan apabila variabel penskalaannya bernilai lebih besar dari 1, maka ukuran citra akan diperbesar, namun apabila variabel penskalaannya bernilai lebih kecil dari 1 maka ukuran citra akan diperkecil.

Proses skala dapat dilakukan dengan rumus :

$$x = \frac{pb * pp}{pa} \quad (2.2)$$

Keterangan:

$x$  = Nilai piksel baris baru

$pb$  = Ukuran panjang matriks baru

$pp$  = Posisi piksel baris

$pa$  = Ukuran panjang matriks lama

$$y = \frac{lb * lp}{la}$$

Keterangan:

$y$  = Nilai piksel kolom baru

$lb$  = Ukuran lebar matriks baru

$lp$  = Posisi piksel kolom

$la$  = ukuran lebar matriks lama

### 2.3.4 Deteksi Wajah (Viola – Jones)

Deteksi Wajah pada penelitian ini menggunakan metode yang dikemukakan oleh Viola & Jones, terbagi menjadi 4 komponen utama: Haar Like Feature, Integral Image, Adaptive Boosting dan Cascade of Classifier.

#### 2.3.4.1 Haar Like Feature

Teknik yang dilakukan yaitu 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-Jones, ada beberapa jenis fitur yang bisa digunakan seperti Edge-feature, Line feature, dan Four-rectangle feature. [16]



**Gambar 2. 1 Ilustrasi Fitur Haar Like Feature**

3 tipe kotak feature adalah:

1. Tipe two-rectangle feature (horisontal/vertikal) adalah perbedaan antara jumlah nilai piksel pada kotak terang dengan nilai piksel pada kotak gelap dan 14 untuk mendapatkan fitur dengan cara menjumlahkan nilai piksel pada kotak terang dan kotak gelap.
2. Tipe three-rectangle feature adalah dengan mempertimbangkan tiga nilai piksel pada kotak terang piksel pada kotak terang pertama dijumlahkan dengan nilai piksel pada kotak terang kedua dan kemudian ditambahkan dengan nilai piksel kotak gelap antara kotak terang ke satu dan kotak terang kedua.
3. Tipe four-rectangle feature adalah dengan mempertimbangkan nilai piksel kotak gelap sebesar 2x2 yang tersusun secara diagonal dan menjumlahkan seluruh nilai piksel kotak terang dan kotak hitam.

Pada proses pemilihan fitur Haar, fitur-fitur tersebut digunakan untuk mencari fitur wajah seperti mata, hidung, dan mulut pada citra. Pada setiap kotak-kotak fitur tersebut terdiri dari beberapa piksel dan akan dihitung selisih antara nilai piksel pada kotak terang dengan nilai piksel pada kotak gelap. Apabila nilai selisih antara daerah terang dengan daerah gelap di atas nilai ambang (threshold), maka daerah tersebut.

#### 2.3.4.2 Integral Image

Integral Image digunakan pada algoritma untuk pendeteksian objek dimana proses perhitungan dengan menggunakan integral image memerlukan waktu yang singkat atau cepat dan hasil yang akurat. Proses untuk menghitung hasil penjumlahan nilai piksel pada daerah yang dideteksi oleh fitur haar. Nilai-nilai piksel yang akan dihitung adalah nilai-nilai piksel dari sebuah citra masukan yang dilalui oleh fitur haar pada saat pencarian fitur wajah. Pada setiap jenis fitur yang digunakan, pada setiap kotak-kotaknya terdiri dari beberapa piksel [16]. Dari nilai-nilai piksel yang didapatkan pada fitur tersebut, maka akan dihitung nilai integral image pada fitur tersebut dengan rumus sebagai berikut:

$$s(x, y) = i(x, y) + s(x, y) + s(x, y - 1) + s(x - 1, y) - s(x - 1, y - 1) \quad (2.4)$$

Dimana:

$s(x,y)$  = merupakan nilai hasil penjumlahan dari tiap-tiap piksel

$i(x,y)$  = merupakan nilai intensitas diperoleh dari nilai piksel dari citra masukan

$s(x-1,y)$  = merupakan nilai piksel pada sumbu x

$s(x,y-1)$  = merupakan nilai piksel pada sumbu y

$s(x-1,y-1)$  = merupakan nilai piksel diagonal

Kemudian selanjutnya adalah menghitung jumlah piksel pada daerah tertentu, adapun rumus untuk menghitung jumlah pikselnya adalah:

$$D = L1 + L4 - (L2 + L3) \quad (2.5)$$

### 2.3.4.3 Adaptive Booster (AdaBoost)

AdaBoost merupakan *ensemble learning* yang sering digunakan pada algoritma boosting. Boosting bisa dikombinasikan dengan classifier algoritma yang lain untuk meningkatkan performa klasifikasi. Tentunya secara intuitif, penggabungan beberapa model akan membantu jika model tersebut berbeda satu sama lain. Adaptive Boosting bertujuan untuk memperbaharui bobot dengan melakukan perhitungan menggunakan persamaan jumlah gambar negatif dan jumlah gambar positif. Adaptive Boosting mengkombinasikan performance banyak weak classifier untuk menghasilkan strong classifier. Weak classifier dalam hal ini adalah nilai dari haar-like feature [16].

$$\text{Bobot awal} = w_{j_1 y_i} = \frac{1}{2m}, w_{j_1 y_i} = \frac{1}{2l} \quad (2.6)$$

$$\text{Untuk citra positif: } \epsilon_t = (\sum_t^T W_{t,i}) |h_t(x) - (y_i)| \quad (2.7)$$

$$\text{Untuk citra positif: } \epsilon_j = (\sum_j^J W_{t,i}) |h_j(x) - (y_i)| \quad (2.8)$$

Jika  $\epsilon_t \in j < 0$ , hentikan iterasi

Dimana:

w = weak classifier

m = jumlah citra positif.

l = jumlah citra negatif.

t = indeks iterasi dari citra positif.

j = indeks iterasi dari citra negatif.

h(x) = nilai fitur citra positif.

h(x) = nilai fitur citra negatif.

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

$$H(x) = \left\{ \begin{array}{l} 1 \sum_{j=1}^j a_j h_j \geq \frac{1}{2} \sum_{t=1}^T a_t 0 \\ \text{bukan objek} \end{array} \right. \quad (2.9)$$

Dimana

$$a_j = \log \log \frac{1}{\beta_j}, a_t = \log \log \frac{1}{\beta_t} \quad (2.10)$$

Kondisi:

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

Jika posisi  $H(x) =$  Ketentuan 0 maka citra tersebut merupakan bukan objek

Keterangan:

$H(x)$  = Strong Classifier atau klasifikasi yang menyatakan objek atau bukan

$a_j$  = Tingkat pembelajaran citra positif

$a_t$  = Tingkat pembelajaran citra negatif

$\beta_t$  = Nilai bobot setelah error rate pada citra positif

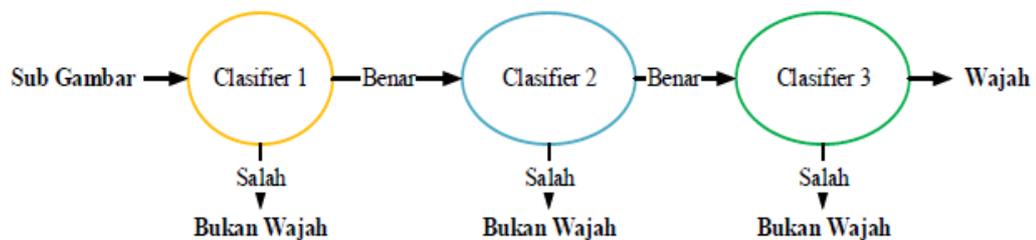
$\beta_j$  = Nilai bobot setelah error rate pada citra negatif

$H_t$  = weak atau basic classifiers (awal dari klasifikasi) citra positif

$H_j$  = weak atau basic classifiers (awal dari klasifikasi) citra negative

#### 2.3.4.4 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 [16].



Gambar 2. 2 Ilustrasi Cascade Classifier

### 2.3.5 Ekstraksi ciri

Ekstraksi ciri adalah proses pengukuran terhadap data yang telah dinormalisasi untuk membentuk sebuah nilai fitur. Nilai fitur digunakan oleh pengklasifikasi untuk mengenali unit masukan dengan unit target keluaran dan memudahkan pengklasifikasian karena nilai ini mudah untuk dibedakan.

#### 2.3.5.1 Scale-Invariant Fourier Transform (SIFT)

Pada tahun 1999, David G. Lowe seorang peneliti dari University of British Columbia memperkenalkan suatu metode baru dalam ekstraksi fitur dari suatu citra. Metode ekstraksi fitur ini disebut sebagai SIFT. Dengan menggunakan SIFT ini, suatu citra akan diubah menjadi vektor fitur lokal yang kemudian akan digunakan sebagai salah satu pendekatan untuk melakukan deteksi dan ekstraksi deskriptor fitur lokal tersebut [17].

Sebagai metode ekstraksi fitur pada pengenalan objek, SIFT ini memiliki kelebihan-kelebihan sebagai berikut:

1. Hasil ekstraksi fitur bersifat invariant terhadap ukuran, translasi dan rotasi dua dimensi.
2. Hasil ekstraksi fitur bersifat invariant sebagian terhadap perubahan iluminasi dan perubahan sudut pandang tiga dimensi.
3. Mampu meng-ekstrak banyak keypoint dari citra yang tipikal.
4. Hasil ekstraksi fitur benar-benar mencirikan secara khusus distinctive).

Dengan kelebihan-kelebihan tersebut, penggunaan metode SIFT banyak dikembangkan untuk aplikasi pengenalan objek. Secara garis besar, algoritma yang digunakan pada metode SIFT terdiri dari empat tahap, antara lain [18] :

#### 1. Mencari Nilai Ekstrim Pada Skala Ruang

Pencarian nilai ekstrim pada skala ruang merupakan tahap awal dalam penentuan keypoint dari suatu citra. Dengan menggunakan fungsi Gaussian-Blurred, citra pada skala ruang dapat didefinisikan sebagai fungsi  $L(x,y,\sigma)$ , yang diperoleh dari hasil konvolusi skala-variabel Gaussian,  $G(x,y,\sigma)$ , dengan citra masukan  $I(x,y)$ , sehingga diperoleh:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (2.11)$$

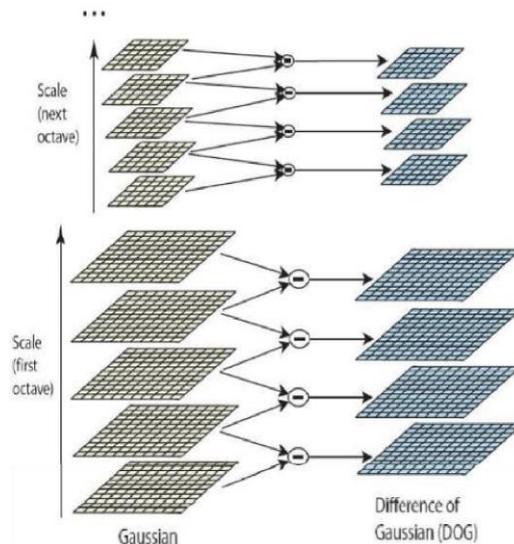
Dimana \* adalah operasi konvolusi antara  $x$  dan  $y$  dan  $G(x,y,\sigma)$  adalah skala-skala variabel Gaussian.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.12)$$

Citra hasil Difference-of-Gaussian,  $D(x,y,\sigma)$ , diperoleh dengan melakukan operasi konvolusi pada citra masukan dengan filter Difference-of-Gaussian, maka :

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \quad (2.13)$$

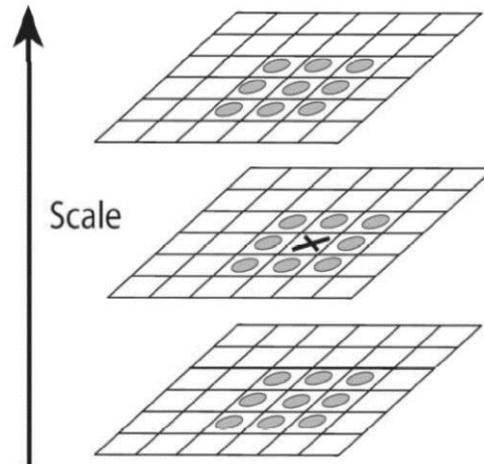
Dari persamaan (2.19) terlihat bahwa citra hasil Difference-of-Gaussian (DoG), sebenarnya merupakan selisih antara citra hasil pengkaburan Gaussian dengan nilai skala  $k$  yang berbeda. Proses ini diilustrasikan pada Gambar 2.3



**Gambar 2. 3 Proses konvolusi citra dengan filter DoG**

Citra hasil konvolusi dikelompokkan berdasarkan octave (satu octave setara dengan penggandaan besarnya nilai  $\sigma$ ), nilai  $k$  ditetapkan di awal sehingga diperoleh jumlah citra kabur yang sama pada setiap octave serta diperoleh citra hasil DoG yang sama untuk setiap octave. Setelah diperoleh citra DoG pada setiap octave, maka langkah selanjutnya adalah mencari kandidat keypoint. Kandidat keypoint dideteksi sebagai titik maksimum lokal atau titik maksimum lokal dari citra hasil DoG. Untuk mencari nilai maksimum dan minimum lokal maka masing-

masing pixel pada citra hasil DoG akan dibandingkan dengan 8 pixel disekitarnya yang berada pada skala yang sama dengan 9 pixel yang bersesuaian dengannya (pada Gambar 2.4). Jika pixel tersebut merupakan maksimum atau minimum lokal, maka pixel tersebut akan dijadikan sebagai kandidat *keypoint*.



**Gambar 2. 4 Ilustrasi pencarian maksimum atau minimum lokal**

## 2. Menentukan Keypoint

Setelah kandidat *keypoint* ditentukan melalui tahapan pertama, maka langkah selanjutnya adalah untuk mengambil detail dari kandidat *keypoint* tersebut. Detail yang diambil merupakan lokasi, skala dan rasio kelengkungan inti dari kandidat *keypoint*. Pada tahap ini akan terjadi pengurangan jumlah kandidat *keypoint*. Dimana setiap kandidat *keypoint* yang dianggap sangat rentan terhadap gangguan (*noise*) akan dihilangkan, yaitu kandidat *keypoint* yang memiliki nilai kontras yang rendah dan kandidat *keypoint* yang kurang jelas dan terletak di sepanjang tepi.

Untuk setiap kandidat *keypoint* akan:

- Dilakukan interpolasi dengan data terdekat di sekitarnya untuk menentukan posisi yang tepat.
- Dibuang *keypoint* dengan kontras yang rendah.
- Diberikan orientasi tertentu pada *keypoint* tersebut

### 3. Penentuan Orientasi

Pada tahap ini, masing-masing *keypoint* yang diperoleh akan diberikan suatu orientasi yang tetap berdasarkan sifat-sifat lokal pada citra. Dengan adanya proses ini maka *keypoint* yang diperoleh dapat direpresentasikan relative terhadap orientasi ini sehingga *keypoint* yang dihasilkan tidak terpengaruh terhadap adanya rotasi pada citra. Untuk menentukan orientasi dari masing-masing *keypoint* maka dilakukan perhitungan terhadap besar nilai *gradient*,  $m(x,y)$ , dan arah orientasi,  $\theta(x,y)$ , dilakukan menggunakan persamaan berikut:

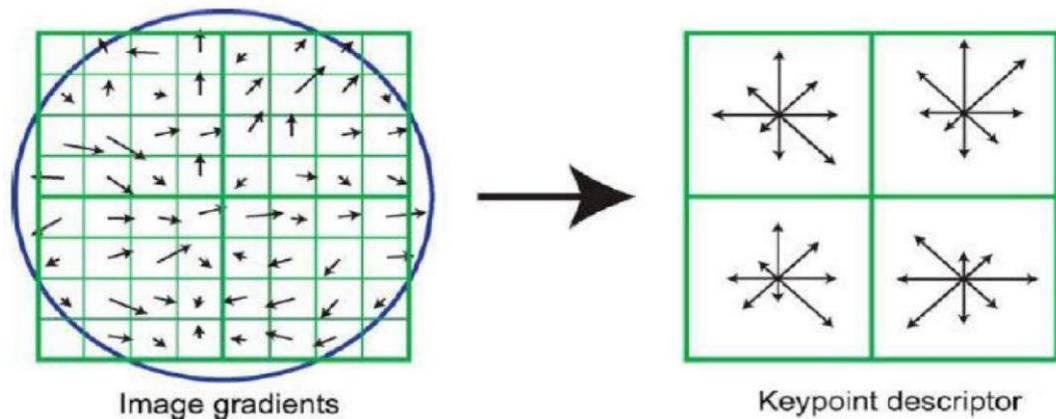
$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \quad (2.14)$$

$$\theta(x,y) = \tan^{-1} \left( \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \right) \quad (2.15)$$

### 4. Deskriptor Keypoint

Pada proses ini, masing-masing *keypoint* yang telah diorientasikan akan diberikan pencirian khusus (deskriptor). Proses ini bertujuan untuk mendapatkan *keypoint* yang invariant terhadap perubahan intensitas cahaya atau perubahan sudut pandang tiga dimensi.

Deskriptor akan diukur sebagai suatu histogram orientasi pada wilayah pixel dengan ukuran 4x4. Nilai orientasi diperoleh dari citra Gaussian yang memiliki skala terdekat dengan skala *keypoint* yang akan dihitung. Agar *keypoint* yang diperoleh invariant terhadap orientasi, maka koordinat dari deskriptor dan gradient orientasi akan dirotasi relative terhadap orientasi dari *keypoint*. Kemudian fungsi pembebanan Gaussian, dengan besar nilai  $\sigma$  satu setengah kali dari besar jendela deskriptor, akan digunakan sebagai pembeban pada setiap besaran nilai dari titik sampel. Proses ini ditunjukkan pada lingkaran yang terdapat pada Gambar 2.5 sebelah kiri.



**Gambar 2. 5 Keypoint Descriptor**

Deskriptor *keypoint* pada Gambar 2.5 menunjukkan adanya 8 arah pada masing-masing histogram orientasi dengan panjang masing-masing anak panah sesuai dengan besar nilai dari histogram asal. Selanjutnya deskriptor *keypoint* yang telah diperoleh akan dinormalisasi untuk mengatasi pengaruh perubahan cahaya.

*Keypoint* ini yang kemudian menjadi fitur-fitur lokal pada suatu citra dan akan dicocokkan dengan *keypoint-keypoint* yang terdapat pada citra.

### 2.3.6 K-Means Clustering

Clustering merupakan faktor yang paling fundamental dalam pattern recognition. Masalah utama dari clustering adalah mendapatkan beberapa nilai vektor pusat yang dapat mewakili keseluruhan vektor dari hasil feature extraction. K-means clustering adalah salah satu metode yang digunakan untuk mempartisi vektor hasil feature extraction ke dalam k vektor pusat [2].

K-Means Clustering adalah proses memetakan vektor-vektor yang berada pada lingkup wilayah yang luas besar menjadi sejumlah tertentu (k) vektor. Wilayah yang terwakili oleh vektor pusat hasil dari proses kuantisasi disebut sebagai cluster. Sebuah vektor pusat hasil dari proses kuantisasi dikenal sebagai codewords. Sedangkan kumpulan dari vektor pusat dikenal sebagai codebooks [2].

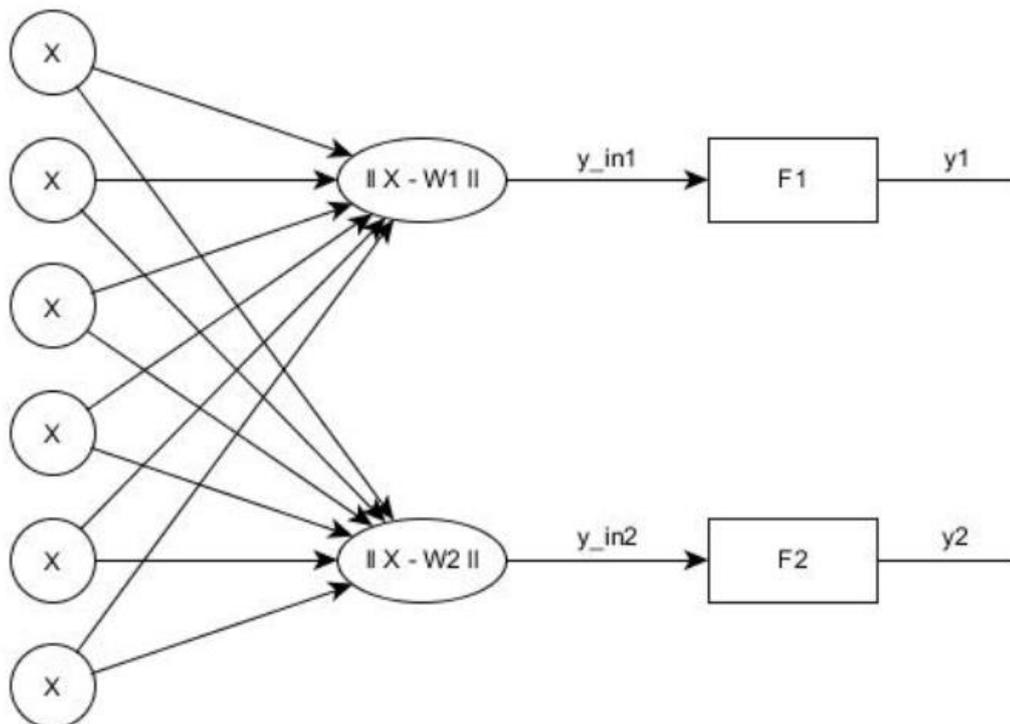
Algoritma K-Means adalah seperti berikut [14] :

1. Tentukan k (jumlah cluster) yang ingin dibentuk.
2. Bangkitkan k centroid (titik pusat cluster) awal secara random.
3. Untuk setiap record , temukan pusat cluster terdekat.

4. Untuk setiap  $k$  cluster , temukan pusat cluster, dan update lokasi dari setiap pusat cluster dengan nilai centroid yang baru. Pusat cluster diperoleh dengan cara menghitung nilai rata-rata dari data-data yang berada pada cluster yang sama.
5. Kembali ke langkah 3 – 5 sampai konvergen.

### 2.3.7 Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) adalah metoda klasifikasi pola yang terawasi (supervised). Vektor input tersebut akan dikelompokkan dalam kelas yang sama. Jaringan LVQ ini memang mirip dengan jaringan yang telah dikembangkan oleh Prof. Teuvo Kohonen pada tahun 1982 (Heranurweni, 2010) sebagaimana diilustrasikan pada gambar 2.6.[19]



**Gambar 2. 6** Arsitektur Jaringan LVQ (6 Input dan 2 Cluster)

Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Jika dua vektor input mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor input tersebut ke dalam kelas yang sama. Setelah pembelajaran, lapisan LVQ membagi vektor input dengan penempatan lapisan LVQ ke kelas yang sama sebagai unit output yang mempunyai

vektor bobot (vektor referensi) terdekat dengan vektor input. Arsitektur jaringan syaraf LVQ pada dasarnya sama dengan Kohonen Self Organizing Map (tanpa suatu struktur yang diasumsikan untuk output). Jaringan syaraf tiruan LVQ terdiri dari layer input, layer kompetitif (Layer tersembunyi, hidden layer) dan layer output seperti yang terlihat pada masing-masing output mempunyai kelas yang telah diketahui. [19]

Adapun langkah yang digunakan untuk mendapatkan nilai kuantitas vektor pada metode LVQ dijabarkan seperti dibawah ini :

Langkah 0 : Inisialisasikan pengurangan rasio dan learning rate( $\alpha$ ).

Langkah 1 : Selama kondisi berhenti bernilai salah, kerjakan langkah 2 dan 3. (Kondisi berhenti jika perulangan mencapai maksimal perulangan dan perubahan rasio lebih kecil dari minimal eror)

Langkah 2 : Untuk masing-masing pelatihan vektor masukan , kerjakan:

a. Temukan J sehingga  $\|x - w_j\|$  bernilai minimum dengan menggunakan rumus *Euclidean Distance* sebagai berikut.

$$C_j = \sqrt{\sum_{i=1}^n (X_i - W_{ij})^2} \quad (2.16)$$

b. Perbaiki  $W_j$  dengan :

- Jika  $T=C_j$  maka

$$W_j(\text{baru}) = W_j(\text{lama}) + \alpha[x - W_j(\text{lama})] \quad (2.17)$$

- Jika  $T \neq C_j$  maka

$$W_j(\text{baru}) = W_j(\text{lama}) - \alpha[x - W_j(\text{lama})] \quad (2.18)$$

Langkah 3 : Kurangi  $\alpha$  dengan rumus :

$$\alpha = \alpha - (\text{pengurangan rasio} * \alpha) \quad (2.19)$$

Keterangan rumus:

T : Target

J : Jumlah selisih data dan bobot

- C : kelas selisih bobot terkecil  
 W : Bobot  
 $\alpha$  : Rasio pembelajaran  
 x : Data

### 2.3.8 Euclidean Distance

Euclidean distance adalah sebuah metode yang digunakan untuk mengukur jarak (distance). Euclidean distance sebenarnya merupakan generalisasi dari teorema pythagoras. Euclidean distance diformulasikan pada persamaan berikut:

$$d(x, y) = \sqrt{\sum_{i=1}^r (x_i - y_i)^2} \quad (2.20)$$

Keterangan rumus:

- d : Jarak antara x dan y  
 r : Jumlah nilai fitur pada vektor  
 $x_i, y_i$  : Nilai fitur i dari vektor x dan vektor y

Jarak minimal yang mungkin antara dua vektor data adalah 0. Selain itu, jarak dari x ke y akan sama dengan y ke x,  $d(x, y) = d(y, x)$ .