

BAB II

LANDASAN TEORI

2.1 Citra Digital

Citra adalah gambar pada bidang dwimatra (dua dimensi). Dalam tinjauan matematis, citra merupakan fungsi *continue* dari intensitas cahaya pada bidang dua dimensi. Ketika sumber cahaya menerangi objek, objek memantulkan kembali sebagian cahaya tersebut. Pantulan ini ditangkap oleh alat-alat pengindra optik, misalnya mata manusia, kamera, scanner dan sebagainya. Bayangan objek tersebut akan terekam sesuai intensitas pantulan cahaya. Ketika alat optik yang merekam pantulan cahaya itu merupakan mesin digital, misalnya kamera digital, maka citra yang dihasilkan merupakan citra digital. Pada citra digital, kontinuitas intensitas cahaya dikuantisasi sesuai resolusi alat perekam (Munir,2004).

Output dari suatu sistem perekaman dapat bersifat (Murni, 1992) :

1. optik, berupa foto,
2. analog, berupa sinyal video seperti gambar pada monitor televisi,
3. digital, berupa file yang dapat langsung disimpan dalam suatu memori.

Dalam hal ini citra yang dimaksud ialah citra diam (*still image*). Citra diam ialah citra tunggal yang tidak bergerak. Contoh dari citra diam dapat kita lihat pada Gambar 2.1.



Gambar 2.1 Contoh Citra Diam(sumber : Munir, 2004)

Selain citra diam juga ada citra bergerak (*moving images*). Citra bergerak ialah rangkaian dari citra diam yang ditampilkan secara sekuensial atau beruntun sehingga memberikan kesan pada mata sebagai gambar yang bergerak. Setiap citra dalam rangkaian tersebut disebut dengan *frame*. Seperti halnya yang biasa kita lihat pada film layar lebar atau televisi pada hakikatnya terdiri atas ratusan hingga ribuan *frame* (Munir, 2004).

Suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum hingga maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan kedalam citra integer (Putra, 2010).

Jenis citra terbagi atas 2, yaitu :

- Citra berwarna, ialah citra yang disusun oleh 3 komponen warna RGB (Red-Green-Blue). Intensitas suatu titik pada citra berwarna merupakan kombinasi dari intensitas merah, hijau dan biru. Setiap piksel dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna (Putra,2010). Setiap poin informasi (*RGB*) disimpan dalam 1 byte data, 8 bit pertama menyimpan nilai biru kemudian diikuti warna hijau pada 8 bit kedua dan 8 bit terakhir warna merah. Jumlah kombinasi jumlah kombinasi warna yang mungkin adalah 2^{24} atau lebih dari 1 juta warna. Variasi ini lebih dari cukup untuk memvisualkan seluruh warna yang dapat dilihat oleh manusia.

- Citra *Grayscale*, suatu citra dikatakan sebagai citra *grayscale* apabila sebuah citra tidak memiliki warna *RGB* atau dapat dikatakan sebuah citra yang memiliki nilai dari putih yang memiliki intensitas paling besar sampai hitam yang memiliki intensitas paling rendah. Citra *grayscale* terdiri dari x dan y dalam kordinat spasial dan memiliki nilai intensitas antara 0 (hitam) hingga 255 (putih) dalam citra 8 bit (Putra, 2010). *Grayscale* lebih mudah diproses karena mengandung nilai lebih sedikit, yaitu 8 bit warna dibandingkan *RGB* yang mengandung 24 bit warna. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R , G , dan B menjadi citra *grayscale* dengan nilai s , maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai r , g , dan b . Perubahan image dari format *RGB* menjadi format *grayscale* yang direpresentasikan dalam persamaan:

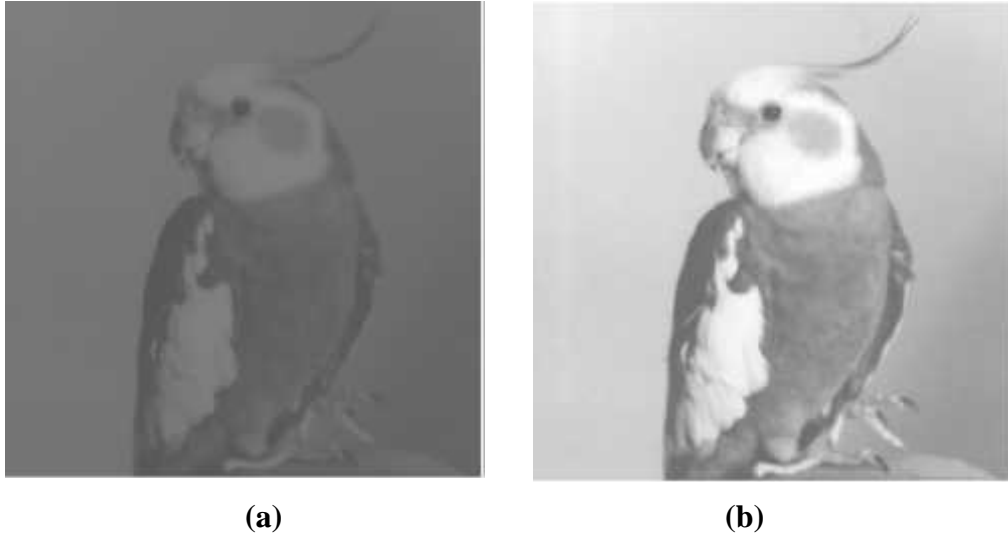
$$Grayscale : 0,299R + 0,587G + 0,114B \quad (2.1)$$

2.2 Pengolahan citra

Meskipun sebuah citra kaya akan informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau *noise*, warnanya terlalu kontras, kurang tajam, kabur (*blurring*), dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang (Munir, 2004).

Agar citra yang mengalami gangguan mudah diinterpretasi, maka citra tersebut perlu dimanipulasi menjadi citra lain yang memiliki kualitas lebih baik. Bidang studi yang menyangkut hal ini adalah pengolahan citra (*image processing*) (Munir, 2004).

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan computer, menjadi citra yang kualitasnya lebih baik. Sebagai contoh dapat dilihat pada Gambar 2.2.



Gambar 2.2 (a) citra burung nuri agak gelap, (b) telah diperbaiki kontrasnya(sumber : Munir, 2004)

Umumnya, operasi pada pengolahan citra ditetapkan pada citra bila (Jain, 1989) :

1. Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
3. Sebagian citra perlu digabung dengan bagian citra yang lain.

Pengolahan citra merupakan suatu proses yang dilakukan dengan input berupa citra dan output berupa citra. Proses ini dilakukan pada citra data *training* sebelum *feature extraction* (pengambilan ciri). Terdapat dua hal yang mendasar dalam memahami proses pembentukan citra, yaitu (Alfin, 2013) :

1. Geometri formasi citra yang menentukan lokasi suatu titik dalam pemandangan yang diproyeksikan pada bidang citra.
2. Fisik cahaya yang menentukan *brightness* suatu *pixel* citra sebagai fungsi pencahayaan.

Oleh karena itu, diperlukan penghubung notasi matematika untuk mengembangkan algoritma pengolahan citra dan notasi algoritma yang digunakan

untuk membuat program komputer yang disimpan kedalam sistem penyimpanan memori dua dimensi yang disebut larik (*array*).

Tujuan utama dari pengolahan citra adalah untuk perbaikan citra dengan menekan *noise* yang tidak diinginkan kedalam citra sehingga dapat diproses dan dianalisa lebih lanjut. Pada proses ini citra dinormalisasi agar lebih siap untuk proses selanjutnya. Proses ini memiliki tahapan-tahapan tertentu dengan berbagai metode dan kalkulasi tertentu sesuai kebutuhan penelitian (Alfin, 2013).

2.2.1. Ekualisasi histogram

Ekualisasi histogram merupakan suatu proses untuk meratakan histogram agar derajat keabuan dari yang paling rendah (0) hingga ke yang paling tinggi (255) memiliki kemunculan yang rata. Dengan histogram ekualisasi ini, hasil gambar yang memiliki histogram yang tidak merata atau distribusi kualitatif yang memiliki banyak loncatan gradasinya akan menjadi gambar yang lebih jelas karena derajat keabuannya tidak dominan gelap dan atau dominan terang, namun warna citra hitam atau putih yang merepresentasikan notasi biner. Penentuan pewarnaan hitam atau putih ditentukan oleh *threshold* yang menjadi acuan dominasi kedua warna tersebut. Proses ini menggunakan distribusi kumulatif, karena dalam proses ini dilakukan perataan *gradient* dari distribusi kumulatifnya (Kurniawan, agus, dkk. 2009).

Pada ekualisasi histogram adaptif citra dibagi menjadi blok-blok (sub-image) dengan ukuran $n \times n$, kemudian pada setiap blok dilakukan proses ekualisasi histogram. Setiap ukuran blok nantinya akan memiliki hasil yang berbeda beda dikarenakan ukuran blok (n) dapat bervariasi dan setiap bloknya dapat saling tumpang tindih beberapa piksel dengan blok lainnya (Putra, 2010).

2.2.2. Operasi Geometri pada citra

Pada operasi geometri, koordinat piksel berubah akibat transformasi, sedangkan intensitasnya tetap. Operasi yang dilakukan misalnya translasi, rotasi, penskalaan citra, dan pencerminan citra (flipping) (Munir, 2004).

1. Translasi

Ini merupakan proses pergeseran citra dimana pergeseran koordinat piksel suatu citra dengan terhadap sumbu x dan sumbu y. Pergeseran terhadap sumbu x ialah pergeseran tiap baris pada suatu nilai matriks citra. Proses ini menggeser citra keatas dan kebawah. Sedangkan pergeseran terhadap sumbu y ialah pergeseran tiap kolom pada suatu nilai matriks citra. Proses ini menggeser citra kekiri dan kekanan.

2. Rotasi

Ini merupakan proses memutar citra. Pada proses ini perputaran citra diputar berdasarkan sudut tertentu dengan poros (0,0).

3. Penskalaan citra

Ini merupakan proses pengubahan ukuran citra. Proses ini lebih dikenal dengan nama *image zooming* (*zoom out* dan *zoom in*).

4. Flipping

Ini merupakan proses pencerminan citra (*image reflection*) berdasarkan cermin vertikal atau horizontal. Operasi pencerminan merupakan salah satu operasi geometri yang paling sederhana. Cermin vertikal (*vertical mirroring*) adalah pencerminan pada sumbu x, sedangkan cermin horizontal adalah pencerminan pada sumbu y.

2.2.3. *Fast Fourier Transform*

Fast Fourier Transform (FFT) adalah suatu algoritma yang digunakan untuk merepresentasikan sinyal dalam domain waktu diskrit dan domain frekuensi. Selain FFT juga terdapat IFFT, ini merupakan singkatan dari *Inverse Fast Fourier Transform*. IFFT merupakan pembalikan dari FFT. Mengenai FFT dan IFFT tentunya tidak terlepas dari DFT (*Discrete Fourier Transform*). DFT merupakan metode transformasi matematis untuk sinyal waktu diskrit ke dalam domain frekuensi. Secara sederhana dapat dikatakan bahwa DFT merupakan metode transformasi matematis sinyal waktu diskrit, sementara FFT adalah algoritma yang digunakan untuk melakukan transformasi tersebut. Jika kompleksitas DFT untuk mentransformasikan sebuah piksel sebesar $O(N^2)$, FFT memiliki kompleksitas sebesar $O(N \log_2 N)$. Sebagai perbandingan, jika N sama dengan 256 maka N^2 sama dengan 65.536, sedangkan $N \log_2 N$ menghasilkan 256

x 8 atau 2048. Jadi, FFT lebih cepat 32 kali dibandingkan DFT untuk ukuran citra seperti itu. Pada alihragam berdimensi dua, penghematan waktu akan lebih terasa (Abdul, 2012).

2.3 Pendeteksian Wajah (*Face Detection*)

Dewasa ini pendeteksian wajah telah menjadi salah satu bidang yang banyak diteliti dan dikembangkan oleh pakar pengenalan pola. Hal ini disebabkan karena semakin luasnya penggunaan teknik identifikasi wajah dalam aplikasi yang digunakan oleh masyarakat. Para peneliti telah melakukan penelitian terhadap teknik yang sudah ada dan mengajukan teknik baru yang lebih baik dari yang sebelumnya. Meskipun banyak teknik baru yang telah diajukan teknik teknik tersebut masih belum dapat memberikan akurasi yang optimal. Dua hal yang menjadi masalah utama pada identifikasi wajah adalah proses ekstraksi fitur dari sampel wajah yang ada dan juga teknik klasifikasi yang digunakan untuk mengklasifikasikan wajah yang ingin dikenali berdasarkan fitur-fitur yang telah dipilih (Budi, 2010).

Ekstraksi fitur adalah proses untuk mendapatkan ciri-ciri pembeda yang membedakan suatu sampel wajah dari sampel wajah lainnya. Bagi sebagian besar aplikasi pengenalan pola (*pattern recognition*), teknik ekstraksi fitur yang handal merupakan kunci utama dalam penyelesaian masalah pengenalan pola. Metode Analisa Komponen Utama (PCA) untuk pengenalan wajah diperkenalkan oleh Turk dan Pentland pada tahun 1991. Metode tersebut bertujuan untuk memproyeksikan data pada arah yang memiliki variasi terbesar yang bersesuaian dengan nilai *eigen* terbesar dari matriks kovarian. Kelemahan dari metode ini ialah kurang optimal dalam pemisah antar kelas. Pada tahun tahun 1991, Cheng dkk memperkenalkan metode Analisa Diskriminan Linier (LDA) untuk pengenalan wajah. Metode ini mencoba menemukan sub ruang linier yang memaksimalkan pemisahan dua kelas pola menurut *Fisher Criterion Jf*. Hal ini dapat diperoleh dengan meminimalkan jarak matriks sebaran dalam kelas yang sama (*within-class*) dan mamksimalkan jarak matriks sebaran antar kelas (*between-class*) secara simultan sehingga menghasilkan *Fisher Criterion Jf* yang maksimal. *Diskriminan Fisher Linier* akan menemukan sub ruang dimana kelas-

kelas salih terpisah linier dengan memaksimalkan *Fisher Criterion Jf*. Jika dimensi data jauh lebih tinggi daripada jumlah sampel *training*, maka *within-class* menjadi *singular*. Hal tersebut merupakan kelemahan dari metode LDA (Belheumeur, 1997).

Selain metode tersebut, untuk melakukan ekstraksi fitur juga dapat menggunakan metode Gabor. Pada beberapa tahun belakangan ini, banyak metode yang berbasis Gabor telah dikembangkan. Gabor menunjukkan karakteristik yang diinginkan dari lokalisasi dan orientasi secara selektif, dan representasi dari Gabor jika mendeteksi gambar wajah sangat kuat walaupun diberbagai pencahayaan dan ekspresi. Inilah yang membuat representasi dari wajah menggunakan fitur Gabor sangat ekstensif dan sukses dalam hal pendeteksian wajah. Tapi walaupun pun metode ini merupakan metode yang banyak sukses digunakan dalam hal merepresentasi wajah, terdapat batasan yang dapat digunakan dalam memproses atau mengekstraksi fiturnya. Filter yang digunakan biasanya hanya sanggup hingga 40 filter (5 skala dan 8 orientasi).

Dalam pendeteksian wajah, proses klasifikasi sama pentingnya dengan proses ekstraksi fitur. Setelah fitur-fitur penting data atau citra wajah dihasilkan pada proses ekstraksi fitur, fitur-fitur tersebut nantinya akan digunakan untuk proses klasifikasi. Metode klasifikasi yang digunakan adalah pengklasifikasi *Support Vector Machine (SVM)*. Pengklasifikasi SVM menggunakan fungsi atau *hyperplane* untuk memisahkan dua buah kelas pola. SVM akan berusaha mencari *hyperplane* yang optimal dimana dua kelas pola dapat dipisahkan dengan maksimal.

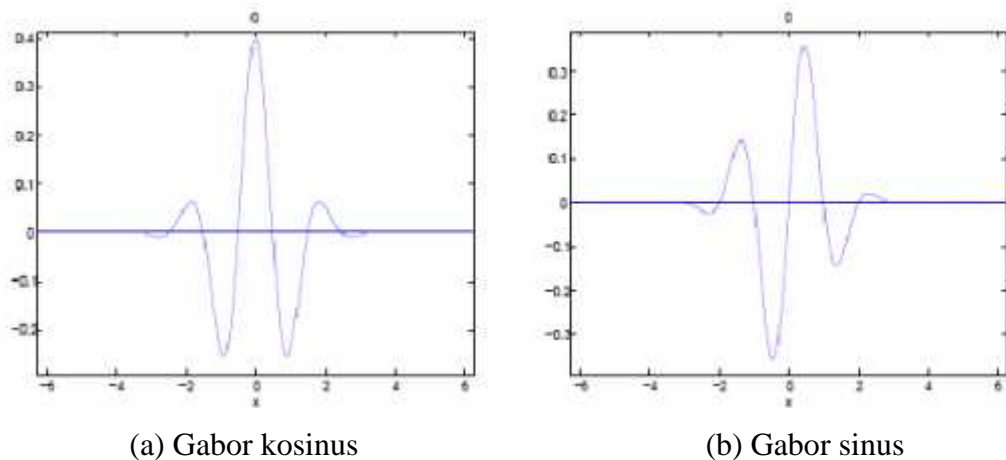
Support vector machine (SVM) adalah suatu teknik klasifikasi yang terbilang baru. Dalam hal klasifikasi metode SVM sangat populer belakangan ini. SVM berada satu kelas dengan metode ANN (Neural Network) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Keduanya masuk dalam kelas *supervised learning*. Baik para ilmuwan dan praktisi telah banyak menerapkan teknik ini dalam menyelesaikan masalah masalah nyata dalam kehidupan sehari hari. Baik dalam masalah gene expression analysis, finansial, cuaca hingga bidang kedokteran. Terbukti dalam banyak implementasinya, metode SVM memberikan hasil yang lebih baik disbanding metode ANN, terutama dalam hal solusi yang

dicapai. ANN menemukan solusi berupa *local optimal* sedangkan SVM menemukan solusi yang *global optimal*. Tidak heran bila setiap menjalankan ANN solusi dari training selalu berbeda. Inilah yang menyebabkan ANN disebut *local optimal* karena solusi yang dicapai tidak selalu sama.

SVM selalu memiliki hasil solusi yang sama untuk setiap *running*. Dalam teknik ini, kita berusaha untuk menemukan fungsi pemisah (klasifier) yang optimal yang bisa memisahkan dua set data dari dua kelas yang berbeda. Teknik ini menarik orang dalam bidang data mining maupun *machine learning* karena performasinya yang meyakinkan dalam memprediksi kelas suatu data baru (Budi, 2010).

2.4 Gabor Filter

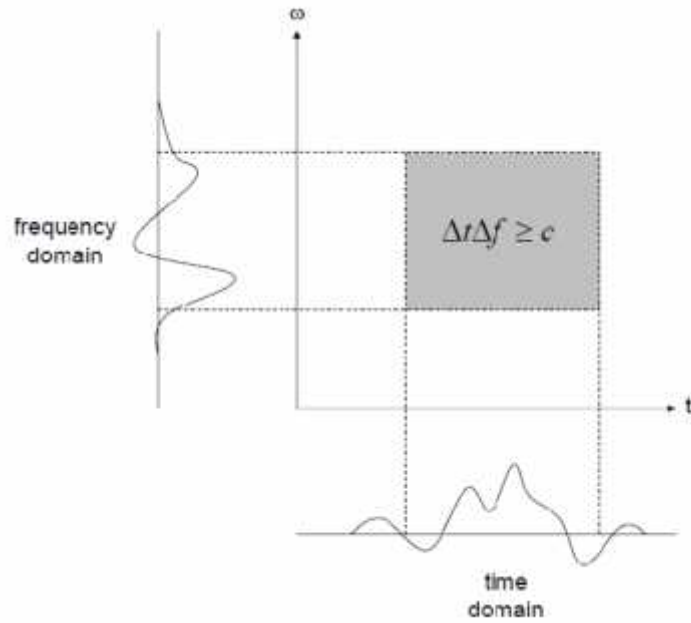
Pada mulanya gabor filter diperkenalkan oleh Dennis Gabor (Gabor, 1946). Filter gabor satu dimensi didefinisikan sebagai perkalian gelombang kosinus/sinus dengan Gaussian. Seperti yang dapat dilihat pada Gambar 2.3.



Gambar 2.3 Filter Gabor satu dimensi (Sumber : Derpanis, 2007)

Hal ini dapat dijelaskan sebagai berikut. Kekuatan spektrum gelombang sinus adalah 2 impuls yang terletak pada $\pm \omega_0$ dan kekuatan spektrum Gaussian adalah (non-normal) Gaussian. Perkalian dalam domain temporal (spasial) setara dengan konvolusi dalam domain frekuensi (Oppenheim, 1997).

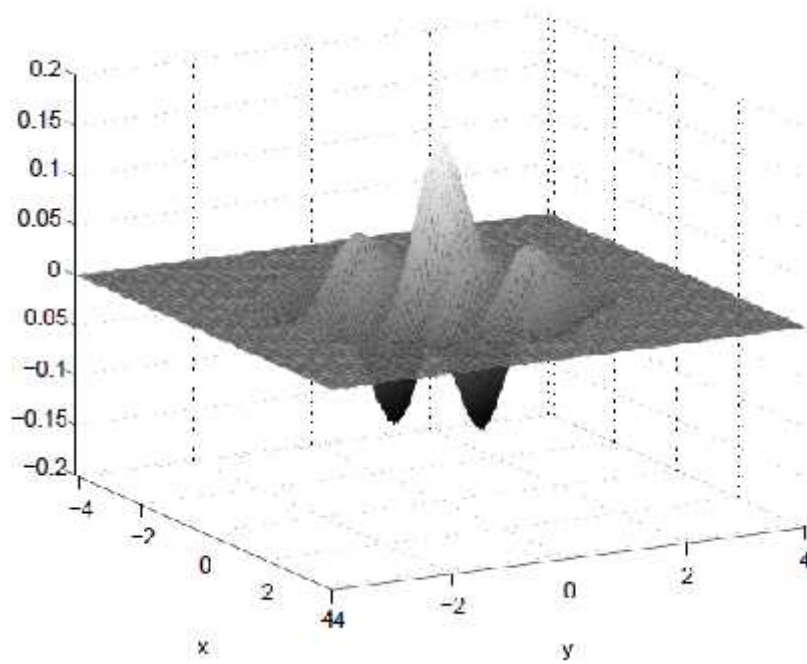
Kaidah tak tentu lainnya menyatakan bahwa hasil dari penyebaran sinyal dalam lingkup waktu dan frekuensi harus melebihi atau sama dengan konstanta yang ditetapkan. Hal ini dapat dilihat pada gambar 2.4.



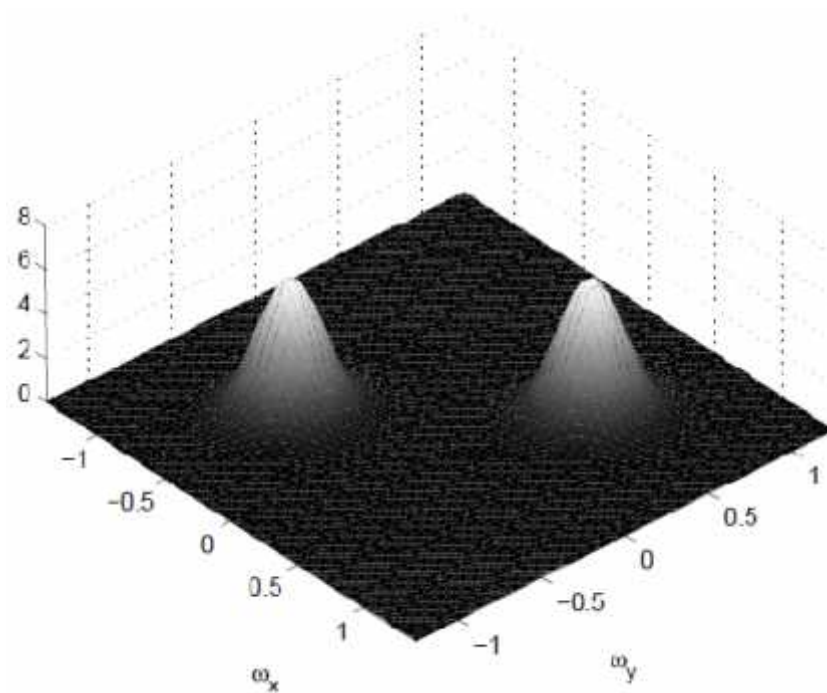
**Gambar 2.4 Gabungan lokalisasi sinyal pada domain waktu dan frekuensi
(Sumber : Derpanis, 2007)**

Implikasi dari kaidah ini ialah bahwa ketepatan salah satunya mengukur sinyal dalam satu batas domain dapat dicapai dari ketepatan dalam pengukuran domain lainnya Gabor (Gabor, 1946).

Daugman (Daugman, 1980; Daugman, 1985) melanjutkan filter Gabor untuk dua dimensi. Seperti yang ia jelaskan pada Gambar 2.5.



(a) Gabor kosinus



(b) Gabor Power Spektrum

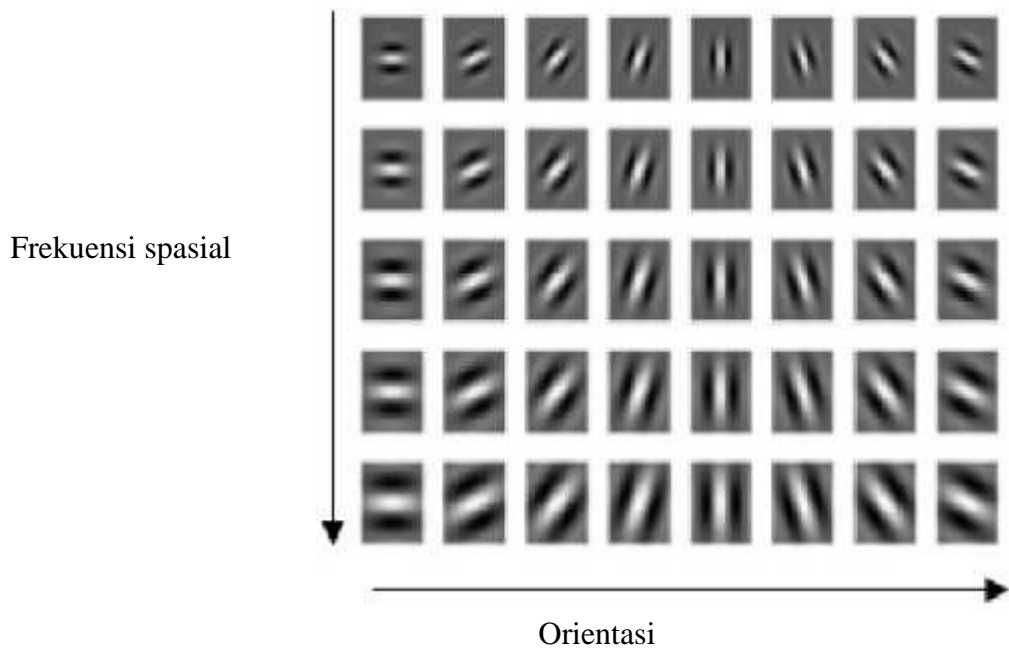
Gambar 2.5 Filter Gabor Dua Dimensi (Sumber : Derpanis, 2007)

Untuk tujuan pengekstraksian aliran optic, Heeger (Heeger, 1987) memanfaatkan filter Gabor tiga dimensi, Heeger (Heeger, 1987) menunjukkan

bahwa filter Gabor tiga dimensi (sama untuk kasus dua dimensi) dapat dibangun dari komponen satu dimensi yang terpisah. Mengingat filter Gabor dua dimensi, k menjadi ukuran konvolusi kernel dua dimensi dan n menjadi ukuran gambar (dalam piksel). Kerumitan dari konvolusi tak terpisah pada filter Gabor diturunkan dari $O k^2 n^2$ ke $O kn^2$.

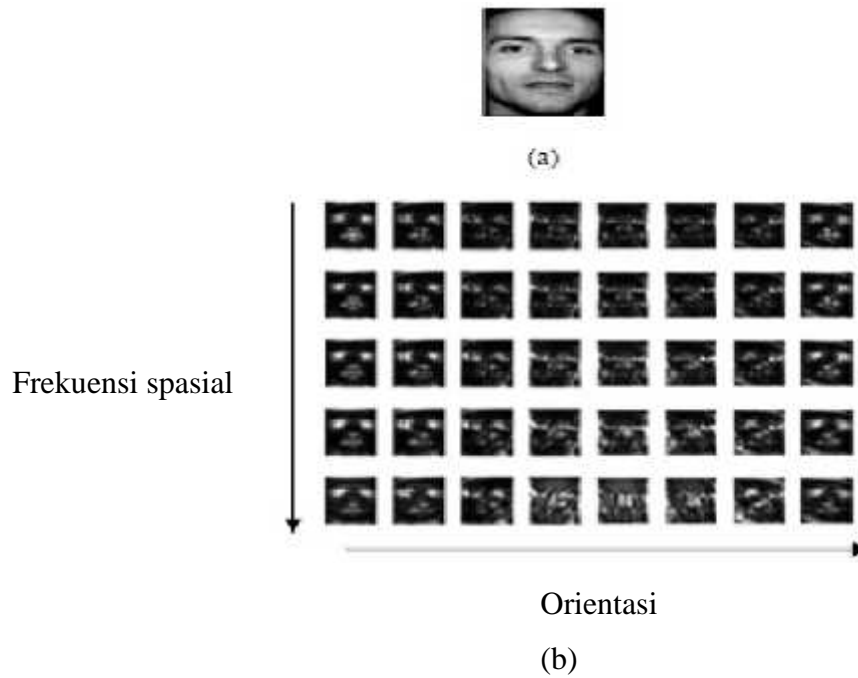
Sejak pengenalan wajah bukanlah hal yang sulit bagi manusia, pemilihan filter Gabor termotivasi biologis cocok untuk masalah ini. Filter Gabor, memodelkan respon sel sederhana dalam korteks visual primer yang bidang gelombang sederhananya dibatasi oleh fungsi Gaussian (L.Shen, 2006)

Sebuah gambar yang dapat direpresentasikan oleh transformasi *Gabor Wavelet* memungkinkan memberi keterangan tentang struktur frekuensi spasial dan hubungan spasial. Mengkonvolusikan gambar dengan kompleks filter Gabor dengan 5 frekuensi spasial ($v = 0, \dots, 4$) dan 8 orientasi ($\mu = 0, \dots, 7$) menangkap seluruh frekuensi spektrum, baik amplitude dan fase yang dijelaskan pada Gambar 2.6.



Gambar 2.6 Filter Gabor yang berhubungan dengan 5 frekuensi spasial dan 8 orientasi (Sumber : Gupta, 2010)

Sebuah masukan gambar wajah dan amplitude dari respon filter Gabor dapat dilihat pada Gambar 2.7.



Gambar 2.7 image wajah (a) beserta respon filter (b) (Sumber : Gupta, 2010)

Pada penelitian ini, secara umum *Gabor Wavelets* (kernel, filter) dapat dijelaskan pada Rumus 2.2 berikut :

$$\Psi_{x,y} = \sigma^{-2} \cdot e^{-\frac{-0,5 NK z_1^2 + z_2^2}{\sigma^2}} \cdot (e^{i K_{real} K_{imag} z} - e^{-\frac{\sigma^2}{2}}) \quad (2.2)$$

Rumus ini dijabarkan sebagai berikut :

$$W = [x \ y]$$

$$v = [0 \dots 4]$$

$$\mu = [0 \dots 7]$$

$$K_{max} = \frac{\pi}{2}$$

$$f = \sqrt{2}$$

$$\sigma = 2\pi$$

$$m = W(x)$$

$$n = W(y)$$

$$K = (K_{max}/f^v) \times (\exp(i \cdot \mu \cdot \frac{\pi}{8}))$$

$$K_{real} = \text{real}(K)$$

$$K_{imag} = \text{imag}(K)$$

$$NK = K_{real}^2 + K_{imag}^2$$

$$X = 1 \dots m$$

$$Y = 1 \dots n$$

$$Z = [x - m/2 ; y - n/2]$$

Dimana W merupakan ukuran *window* dari filter Gabor yang biasanya berukuran 32x32. ν dan μ menjelaskan skala dan orientasi dari Gabor. K_{max} , f , dan σ adalah parameter yang telah ditetapkan oleh Gabor.

Sebagai contoh dapat dilihat pada uraian dibawah ini.

Misalkan nilai yang diinputkan adalah sebagai berikut :

$W = [2 \ 2]$, *window* yang digunakan berukuran matriks 2x2

$\nu = 2$, nilai yang dicari pada skala ke-2

$\mu = 2$, nilai yang dicari pada orientasi ke-2

$$K_{max} = 1,5708$$

$$f = 1,4142$$

$$\sigma = 6,2832$$

$$M = 2$$

$$N = 2$$

$$K = (K_{max}/f^{\mu}) \times (\exp(i.\mu.\frac{\pi}{8}))$$

$$= (1,5708/1,4142^2) \times (\exp(i.2.\frac{3.14}{8}))$$

$$= 0,5554 + 0,5554i$$

$$K_{real} = 0,5554$$

$$K_{imag} = 0,5554$$

$$NK = K_{real}^2 + K_{imag}^2$$

$$= 0,5554^2 + 0,5554^2$$

$$= 0,6169$$

$$X = 1 \dots m$$

$$= 1, 2$$

$$Y = 1 \dots n$$

$$= 1, 2$$

$$Z = [x - m/2 ; y - n/2]$$

$$= \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Jika $x = 1, y = 1, z = \begin{matrix} 0 \\ 0 \end{matrix}$ maka,

$$\begin{aligned} \Psi_{x,y} &= \sigma^{-2} \cdot e^{-\frac{0,5 NK z_1^2 + z_2^2}{\sigma^2}} \cdot (e^{i \text{Kreal Kimag } z} - e^{-\frac{\sigma^2}{2}}) \\ &= 6,2832^{-2} \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 0^2 + 0^2}{6,2832^2}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \begin{matrix} 0 \\ 0 \end{matrix}} - e^{-\frac{6,2832^2}{2}}) \\ &= 0,0253 \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 0}{39,4784}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \begin{matrix} 0 \\ 0 \end{matrix}} - e^{-19,7392}) \\ &= 0,0253 \cdot e^0 \cdot (e^0 - 2,6753e^{-009}) \\ &= 0,0253 \cdot 1 \cdot (1 - 2,6753e^{-009}) \\ &= 0,0253 \cdot 1 \cdot 1 \\ &= 0,0253 \end{aligned}$$

Jika $x = 1, y = 2, z = \begin{matrix} 0 \\ 1 \end{matrix}$ maka,

$$\begin{aligned} \Psi_{x,y} &= \sigma^{-2} \cdot e^{-\frac{0,5 NK z_1^2 + z_2^2}{\sigma^2}} \cdot (e^{i \text{Kreal Kimag } z} - e^{-\frac{\sigma^2}{2}}) \\ &= 6,2832^{-2} \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 0^2 + 1^2}{6,2832^2}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \begin{matrix} 0 \\ 1 \end{matrix}} - e^{-\frac{6,2832^2}{2}}) \\ &= 0,0253 \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 1}{39,4784}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \begin{matrix} 0 \\ 1 \end{matrix}} - e^{-19,7392}) \\ &= 0,0253 \cdot e^{-0,0078} \cdot (e^{0 + 0,554i} - 2,6753e^{-009}) \\ &= 0,0253 \cdot 0,9922 \cdot (0,8497 + 0,5273i - 2,6753e^{-009}) \\ &= 0,0253 \cdot 0,9922 \cdot (0,8497 + 0,5273i) \\ &= 0,0253 + 0,0132i \end{aligned}$$

Jika $x = 2, y = 1, z = \begin{matrix} 1 \\ 0 \end{matrix}$ maka,

$$\begin{aligned}
\Psi_{x,y} &= \sigma^{-2} \cdot e^{-\frac{0,5 NK z_1^2 + z_2^2}{\sigma^2}} \cdot (e^{i \text{Kreal Kimag } z} - e^{-\frac{\sigma^2}{2}}) \\
&= 6,2832^{-2} \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 1^2 + 0^2}{6,2832^2}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \frac{1}{0}} - e^{-\frac{6,2832^2}{2}}) \\
&= 0,0253 \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 1}{39,4784}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \frac{1}{0}} - e^{-19,7392}) \\
&= 0,0253 \cdot e^{-0,0078} \cdot (e^{0 + 0,554i} - 2,6753e^{-009}) \\
&= 0,0253 \cdot 0,9922 \cdot (0,8497 + 0,5273i - 2,6753e^{-009}) \\
&= 0,0253 \cdot 0,9922 \cdot (0,8497 + 0,5273i) \\
&= 0,0253 + 0,0132i
\end{aligned}$$

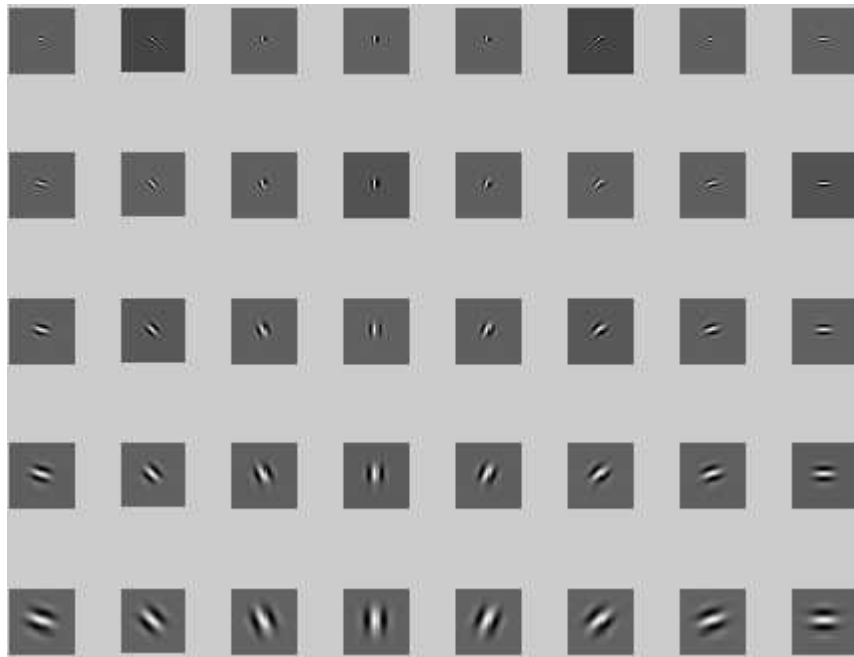
Jika $x = 2, y = 2, z = \frac{1}{1}$ maka,

$$\begin{aligned}
\Psi_{x,y} &= \sigma^{-2} \cdot e^{-\frac{0,5 NK z_1^2 + z_2^2}{\sigma^2}} \cdot (e^{i \text{Kreal Kimag } z} - e^{-\frac{\sigma^2}{2}}) \\
&= 6,2832^{-2} \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 1^2 + 1^2}{6,2832^2}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \frac{1}{1}} - e^{-\frac{6,2832^2}{2}}) \\
&= 0,0253 \cdot e^{-\frac{-0,5 \cdot 0,6169 \cdot 2}{39,4784}} \cdot (e^{i \cdot 0,5554 \cdot 0,5554 \cdot \frac{1}{1}} - e^{-19,7392}) \\
&= 0,0253 \cdot e^{-0,0156} \cdot (e^{0 + 1,1107i} - 2,6753e^{-009}) \\
&= 0,0253 \cdot 0,9845 \cdot (0,4440 + 0,8960i - 2,6753e^{-009}) \\
&= 0,0253 \cdot 0,9845 \cdot (0,4440 + 0,8960i) \\
&= 0,0111 + 0,0223i
\end{aligned}$$

Jadi, nilai keseluruhan Ψ jika $w = [2 \ 2]$ adalah

$$\begin{array}{cc}
0,0253 & 0,0213 + 0,0132i \\
0,0213 + 0,0132i & 0,0111 + 0,0223i
\end{array}$$

Dengan menerapkan teorema konvolusi, kita dapat memperoleh fitur Gabor melalui *Fast Fourier Transform*, yang mana dapat memproses lebih cepat dan tidak memakan banyak waktu. Aplikasi filter Gabor dalam menganalisa sinyal waktu-frekuensi local, khususnya *Fourier transform*, disebut dengan *Gabor transform*. Gambar 2.8 berikut adalah filter Gabor sesungguhnya yang digunakan dalam penelitian ini berdasarkan Rumus 2.2, ukuran *window* 32x32 dan parameter yang telah ditetapkan.



Gambar 2.8 Filter gabor yang digunakan dalam penelitian

Setelah filter Gabor didapatkan, selanjutnya dilakukan proses untuk mendapatkan nilai vector dari suatu citra inputan. Citra diolah berdasarkan filter Gabor dan melalui proses *Fast Fourier Transform*. Untuk lebih jelasnya dapat dilihat pada Algoritma 2.1.

Algoritma 2.1 proses image ke vector

```

Function image ke vector
(input : s,j of integer; Features :array [] of integer;
image : array [] of integer;
Output : imvector : array [] of integer);

Algoritma
Image = adapthisteq(image);
For s ← 1 to 5 do
  For j 1 to 8 do
Features{s,j}=abs(iff2(G{s,j}.*fft2(double(image),32,32)27,18))
  End for
Endfor
Features = Cell2mat(features 135x144)
Imvector = reshape(features45x48,[19440 1])

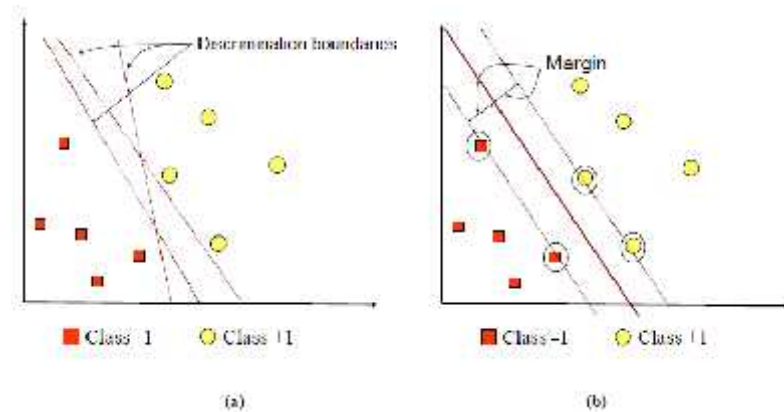
```



2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane* (Vapnik, 1995). kernel diperkenalkan oleh Aronszajn tahun 1950, dan demikian juga dengan konsep-konsep pendukung yang lain. Akan tetapi hingga tahun 1992, belum pernah ada upaya merangkaikan komponen-komponen tersebut. Berbeda dengan strategi neural network yang berusaha mencari hyperplane pemisah antar class, SVM berusaha menemukan hyperplane yang terbaik pada input space. Prinsip dasar SVM adalah linear classifier, dan selanjutnya dikembangkan agar dapat bekerja pada problem on-linear. dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi. Perkembangan ini memberikan rangsangan minat penelitian di bidang pattern recognition untuk investigasi potensi kemampuan SVM secara teoritis maupun dari segi aplikasi.

Dewasa ini SVM telah berhasil diaplikasikan dalam problema dunia nyata (*real-world problems*), dan secara umum memberikan solusi yang lebih baik dibandingkan metode konvensional seperti misalnya artificial neural network. Tulisan ini memperkenalkan konsep dasar SVM, dan membahas aplikasinya di bioinformatika, yang akhir-akhir ini merupakan salah satu bidang yang berkembang cukup pesat (Anto, 2003).



Gambar 2.9 SVM berusaha menemukan hyperplane terbaik yang memisahkan kedua class -1 dan +1 (sumber : Anto, 2003)

2.5.1 Pattern Recognition Memakai *Support Vector Machine*

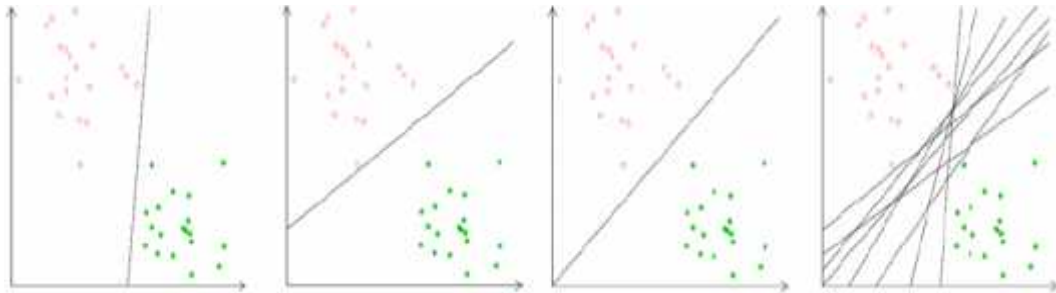
Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari hyperplane2 terbaik yang berfungsi sebagai pemisah dua buah class pada input space. Gambar 1a memperlihatkan beberapa pattern yang merupakan anggota dari dua buah class : +1 dan -1. Pattern yang tergabung pada class -1 disimbolkan dengan warna merah (kotak), sedangkan pattern pada class +1, disimbolkan dengan warna kuning(lingkaran). Problem klasifikasi dapat diterjemahkan dengan usaha menemukan garis (hyperplane) yang memisahkan antara kedua kelompok tersebut. Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar 1-a. Hyperplane pemisah terbaik antara kedua class dapat ditemukan dengan mengukur *margin* hyperplane tsb. dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane tersebut dengan pattern terdekat dari masing-masing class. Pattern yang paling dekat ini disebut sebagai *support vector* (Anto, 2003).

Garis solid pada gambar 1-b menunjukkan hyperplane yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua class, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah support vector. Usaha untuk mencari lokasi hyperplane ini merupakan inti dari proses pembelajaran pada SVM (Anto, 2003).

2.5.2 *Linierly Separable Data*

Pada *linierly separable data* mempertimbangkan setiap gambar menjadi satu titik pada gambar. Dan setiap titik warna yang berbeda menentukan kategori

yang berbeda. Disini kita memiliki dua gambar kategori dan kita harus menemukan batas pemisah dari dua kategori tersebut.



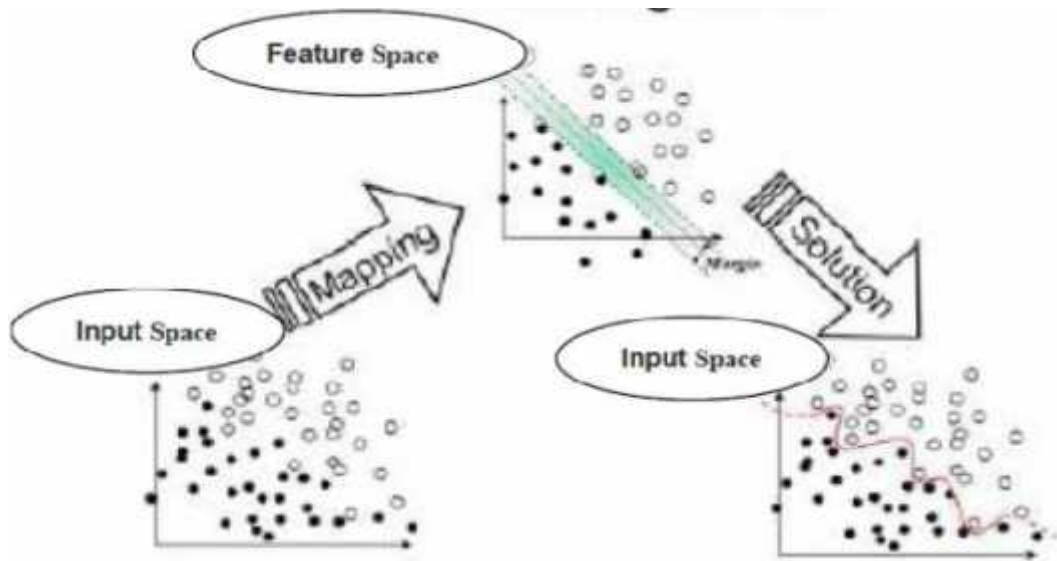
Gambar 2.10 SVM pada linierly separable data (Sumber : Parin, 2012)

Margin dari klasifier linier adalah lebar dimana panjang batasnya dapat ditingkatkan sebelum meletakkan titik data dari kategori yang berbeda. Garis tersebut akan mengambil margin tertinggi antara dua data set. Titik data yang terletak pada margin tersebut disebut dengan *Support Vector*.

Langkah berikutnya adalah menemukan *hyperplane* terbaik yang memisahkan kedua kategori tersebut. SVM melakukannya dengan mengambil sebuah set poin dan membelahnya menggunakan aplikasi rumus matematika tertentu yang berbeda.

2.5.3 Non-Linearly Separable Data

Pada *non-linierly separable data*, data yang diinputkan adalah data yang tidak dapat dipisahkan oleh *linierly separable plane*. Kita akan memetakan semua titik untuk *feature space* menggunakan 25 jenis kernel tertentu untuk memisahkan data non-linier pada linier *plane*. Setelah memisahkan titik-titik dalam ruang fitur kita dapat memetakan titik kembali ke ruang input dengan menggunakan *curvy hyperplane*. Pada Gambar 2.11 menunjukkan aliran data SVM.



Gambar 2.11 SVM pada non-linearly separable data (Sumber : Parin, 2012)

Pada kenyataannya, kita akan menemukan bahwa bagian besar set data tidak sederhana dan berperilaku baik. Akan ada beberapa poin yang tidak diklasifikasikan dengan benar, titik-titik yang jauh dari kelas, atau titik-titik yang dicampur bersama dalam sebuah spiral atau pola kotak-kotak.

2.5.4 Metoda Kernel

Banyak teknik data mining dan *machine learning* yang dikembangkan dengan asumsi kelinieran. Sehingga algoritma yang dihasilkan terbatas untuk kasus-kasus linier. Karena itu, bila suatu kasus klasifikasi memperlihatkan ketidaklinieran, algoritma seperti perceptron tidak bisa mengatasinya. Secara umum kasus-kasus pada dunia nyata ialah kasus nonlinier yaitu data tersebut tidak bisa dipisahkan secara linier. Namun tidak semuanya harus diklasifikasi secara nonlinier, semua juga tergantung dari data yang diinputkan. Jika data yang diinputkan dapat dengan jelas dipisah menjadi dua kategori yang berbeda, data tersebut dapat diproses secara linier.

Fungsi kernel yang biasanya dipakai dalam literatur SVM (Budi, 2010):

- Linier : $x^T x_l$
- Polynomial : $x^T x_l + 1^p$
- Radial basis function (RBF) : $\exp - \frac{1}{2\sigma^2} \|x - x_l\|^2$
- Tangent Hyperbolic (sigmoid) : $\tan^{-1} \beta x^T x_l + \beta_1$

Fungsi kernel mana yang harus digunakan untuk substitusi *dot product* di *feature space* sangat bergantung pada data. Biasanya metoda *cross-validation* digunakan untuk pemilihan fungsi kernel ini. Pemilihan fungsi kernel yang tepat adalah hal yang sangat penting. Karena fungsi kernel ini akan menentukan *feature space* dimana fungsi klasifier akan dicari. Sepanjang fungsi kernel legitimate, SVM akan beroperasi secara benar meskipun kita tidak tahu seperti apa map yang digunakan. Fungsi kernel yang legitimate diberikan oleh teori Mercer dimana fungsi tersebut harus memenuhi syarat kontinu dan *positive definite*. Lebih mudah menemukan fungsi kernel daripada mencari map seperti apa yang tepat untuk melakukan *mapping* dari *input space* ke *feature space*. Pada penerapan kernel kita tidak perlu tahu map apa yang digunakan untuk satu per satu data, tetapi lebih penting mengetahui bahwa *dot product* dua titik di *feature space* bisa digantikan oleh fungsi kernel.

2.5.5 Klasifikasi *Support Vector Machine* (SVM)

Dalam pendeteksian wajah, proses klasifikasi sama pentingnya dengan proses ekstraksi fitur. Setelah fitur-fitur penting data atau citra wajah dihasilkan pada proses ekstraksi fitur, fitur-fitur tersebut nantinya akan digunakan untuk proses klasifikasi. Metode klasifikasi yang digunakan adalah pengklasifikasi *Support Vector Machine* (SVM). Pengklasifikasi SVM menggunakan fungsi atau *hyperplane* untuk memisahkan dua buah kelas pola. SVM akan berusaha mencari *hyperplane* yang optimal dimana dua kelas pola dapat dipisahkan dengan maksimal.

Proses klasifikasi pelatihan dilakukan setelah data-data pelatihan diambil fitur-fitur khusus. Fitur-fitur khusus ini berupa vektor fitur yang dimensinya lebih kecil. Pada proses klasifikasi, pelatihan variable *hyperplane* untuk setiap pengklasifikasi yang didapat akan disimpan dan nantinya akan digunakan sebagai data tiap pengklasifikasi dalam proses pengujian. Dengan kata lain proses klasifikasi pelatihan adalah untuk mencari *support vector* dari data masukan.

Pada proses klasifikasi pengujian menggunakan hasil ekstraksi fitur data pengujian dan hasil proses klasifikasi pelatihan. Hasil dari proses ini berupa nilai indeks dari fungsi keputusan yang terbesar yang menyatakan kelas dari data

pengujian. Jika kelas yang dihasilkan dari proses klasifikasi pengujian sama dengan kelas data pengujian, maka pengenalan dinyatakan benar. Hasil akhirnya berupa citra wajah yang sesuai dengan nilai indeks dari fungsi keputusan yang terbesar hasil dari proses klasifikasi pengujian (Budi, 2010).

2.5.6 Karakteristik Support Vector Machine (SVM)

Karakteristik SVM sebagaimana telah dijelaskan pada bagian sebelumnya, dirangkumkan sebagai berikut (Anto, 2003):

1. Secara prinsip SVM adalah linear classifier
2. Pattern recognition dilakukan dengan mentransformasikan data pada input space ke ruang yang berdimensi lebih tinggi, dan optimisasi dilakukan pada ruang vector yang baru tersebut. Hal ini membedakan SVM dari solusi pattern recognition pada umumnya, yang melakukan optimisasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi input space.
3. Menerapkan strategi *Structural Risk Minimization* (SRM)
4. Prinsip kerja SVM pada dasarnya hanya mampu menangani klasifikasi dua class.

2.5.7 Kelebihan dan Kekurangan Support Vector Machine (SVM)

Dalam memilih solusi untuk menyelesaikan suatu masalah, kelebihan dan kelemahan masing-masing metode harus diperhatikan. Selanjutnya metode yang tepat dipilih dengan memperhatikan karakteristik data yang diolah.

Dalam hal SVM, walaupun berbagai studi telah menunjukkan kelebihan metode SVM dibandingkan metode konvensional lain, SVM juga memiliki berbagai kelemahan. Kelebihan SVM antara lain sbb (Anto, 2003).

1. Generalisasi

Generalisasi didefinisikan sebagai kemampuan suatu metode (SVM, neural network, dsb.) untuk mengklasifikasikan suatu pattern, yang tidak termasuk data yang dipakai dalam fase pembelajaran metode itu. Vapnik menjelaskan bahwa generalization error dipengaruhi oleh dua faktor: error terhadap training set, dan satu faktor lagi yang dipengaruhi oleh dimensi VC (Vapnik-Chervokinensis). Strategi pembelajaran pada neural network dan umumnya metode learning machine difokuskan pada usaha untuk meminimalkan error pada training-set.

Strategi ini disebut *Empirical Risk Minimization* (ERM). Adapun SVM selain meminimalkan error pada training-set, juga meminimalkan faktor kedua. Strategi ini disebut *Structural Risk Minimization* (SRM), dan dalam SVM diwujudkan dengan memilih hyperplane dengan margin terbesar. Berbagai studi empiris menunjukkan bahwa pendekatan SRM pada SVM memberikan error generalisasi yang lebih kecil daripada yang diperoleh dari strategi ERM pada neural network maupun metode yang lain.

2. *Curse of dimensionality*

Curse of dimensionality didefinisikan sebagai masalah yang dihadapi suatu metode pattern recognition dalam mengestimasi parameter (misalnya jumlah hidden neuron pada neural network, stopping criteria dalam proses pembelajaran dsb.) dikarenakan jumlah sampel data yang relatif sedikit dibandingkan dimensional ruang vektor data tersebut. Semakin tinggi dimensi dari ruang vektor informasi yang diolah, membawa konsekuensi dibutuhkan jumlah data dalam proses pembelajaran. Pada kenyataannya seringkali terjadi, data yang diolah berjumlah terbatas, dan untuk mengumpulkan data yang lebih banyak tidak mungkin dilakukan karena kendala biaya dan kesulitan teknis. Dalam kondisi tersebut, jika metode itu “terpaksa” harus bekerja pada data yang berjumlah relatif sedikit dibandingkan dimensinya, akan membuat proses estimasi parameter metode menjadi sangat sulit. Curse of dimensionality sering dialami dalam aplikasi di bidang *biomedical engineering*, karena biasanya data biologi yang tersedia sangat terbatas, dan penyediaannya memerlukan biaya tinggi. Vapnik membuktikan bahwa tingkat generalisasi yang diperoleh oleh SVM tidak dipengaruhi oleh dimensi dari input vector. Hal ini merupakan alasan mengapa SVM merupakan salah satu metode yang tepat dipakai untuk memecahkan masalah berdimensi tinggi, dalam keterbatasan sampel data yang ada.

3. *Feasibility*

SVM dapat diimplementasikan relatif mudah, karena proses penentuan support vector dapat dirumuskan dalam QP problem. Dengan demikian jika kita memiliki *library* untuk menyelesaikan QP problem, dengan sendirinya SVM dapat diimplementasikan dengan mudah. Selain itu dapat diselesaikan dengan metode sekuensial sebagaimana penjelasan sebelumnya.

Disamping kelebihanannya, SVM memiliki kelemahan atau keterbatasan, antara lain:

1. Sulit dipakai dalam problem berskala besar. Skala besar dalam hal ini dimaksudkan dengan jumlah sample yang diolah.
2. SVM secara teoritik dikembangkan untuk problem klasifikasi dengan dua class.

Dewasa ini SVM telah dimodifikasi agar dapat menyelesaikan masalah dengan class lebih dari dua, antara lain strategi One versus rest dan strategi Tree Structure. Namun demikian, masing-masing strategi ini memiliki kelemahan, sehingga dapat dikatakan penelitian dan pengembangan SVM pada *multiclass-problem* masih merupakan tema penelitian yang masih terbuka.

2.6 Akurasi Pengujian

Hasil dari penelitian ini akan diukur dengan menghitung tingkat akurasi dari hasil klasifikasi bentuk tulang daun dengan data uji. Tingkat akurasi ini dihitung dengan menggunakan rumus :

$$\text{Keakuratan pendeteksian (\%)} = \frac{A+B}{2} \quad (2.3)$$

Dengan :

$$A = \frac{\text{true positive}}{\text{face on image}}$$
$$B = \frac{\text{true positive}}{\text{face detected}}$$

Dimana :

True positive = jumlah objek terdeteksi yang benar merupakan wajah manusia.

Face on image = jumlah wajah manusia yang ada di dalam citra (jumlah wajah manusia yang seharusnya terdeteksi).

Face detected = jumlah seluruh objek yang terdeteksi, baik yang merupakan wajah maupun bukan (true positive + false positive).

2.7 Review Penelitian Sebelumnya

Terdapat beberapa penelitian tentang deteksi wajah seperti yang dilakukan oleh (Mudji, 2010) menjelaskan ciri emosi ekspresi wajah dinyatakan dengan menggunakan filter Gabor. Kemudian ciri yang telah didapatkan digunakan untuk melatih jaringan neural dengan algoritma pelatihan Backpropagation. Terakhir, ekspresi wajah diklasifikasi dengan jaringan neural. Menggunakan algoritma tersebut diperoleh hasil pengenalan yang tinggi.

Penelitian oleh (Putro, 2012) menggunakan metode *Viola-Jones* karena menganggap metode ini relatif mendapatkan hasil yang cepat, akurat, dan efisien dalam melakukan deteksi wajah pada gambar. Metode ini merupakan metode yang paling banyak digunakan untuk mendeteksi wajah. Metode ini terdiri atas 3 komponen penting yaitu integral *image* digunakan untuk menentukan data ada atau tidaknya fitur *Haar* tertentu pada sebuah gambar, metode *AdaBoostmachine learning* yang digunakan untuk memilih fitur *Haar* yang spesifik yang akan digunakan serta untuk mengatur nilai ambangnya (*threshold*), dan *cascade classifier* sebagai pengklasifikasi akhir untuk menentukan daerah wajah pada gambar dari metode ini. Urutan filter pada *cascade* ditentukan oleh bobot yang diberikan *AdaBoost*. Filter dengan bobot paling besar diletakkan pada proses pertama kali dengan tujuan untuk menghapus daerah bukan wajah secepat mungkin. Hasil penelitian ini mendapatkan nilai akurasi sistem deteksi wajah sebesar 90,9%. Hasil lain yang didapatkan adalah posisi wajah yang tegak/tidak tegak menentukan keberhasilan deteksi wajah tersebut.

Ergun dkk (Gumus, 2010) membahas tentang menggunakan berbagai macam metode untuk pengenalan wajah. Untuk ekstraksi fitur menggunakan teknik *wavelet decomposition* dan *Eigenfaces* yang mana berbasis *Principal Component Analysis* (PCA). Setelah menghasilkan vektor fitur, metode *Support Vector Machine* (SVM) digunakan sebagai langkah klasifikasi. Sebagai pengujian menggunakan database ORL yang mana merupakan database standar yang digunakan dalam pengenalan wajah. Pada akhirnya penelitian ini memiliki tingkat akurasi sebesar 98,1%.

Penelitian yang dilakukan Xiao-ming dkk (Wang, 2009) mengusulkan sebuah algoritma yang efektif untuk pengenalan wajah dengan menggunakan

metode Gabor dan *Support Vector Machine* (SVM). Pertama gambar gabor diperoleh dari representasi Gabor *Wavelet* yang merupakan konvolusi dari citra wajah dengan jenis kernel Gabor, dan kemudian metode 2D *Principal Component Analysis* (2DPCA) diterapkan pada gambar wajah Gabor untuk mengekstrak ruang fitur. Setelah itu *Support Vector Machine* (SVM) digunakan untuk proses klasifikasi. Hasil percobaan pada ORL database menunjukkan bahwa gambar Gabor memiliki informasi lebih diskriminan dan metode yang diusulkan dapat mencapai 99,5% nilai pengenalan pada dataset wajah penuh dan mencapai 98,0% nilai pengenalan pada dataset kesatuan.

Omid Sakhi pada tahun 2011 melakukan tahapan penelitian dasar dengan menggunakan metode filter Gabor dan *Support Vector Machine* (SVM). Proses dimulai dengan menggunakan filter Gabor untuk mengekstraksi fitur, setelah nilai fitur didapat kemudian dilakukan proses klasifikasi dengan menggunakan metode SVM. Pada penelitian Omid Sakhi ini tidak bertujuan untuk mendapatkan nilai akurasi tetapi hanya menerapkan tahapan dasar yang dirancangnya dengan metode filter Gabor dan SVM. Ini terbukti karena melihat dari penelitiannya tingkat akurasi yang dihasilkan pada beberapa citra yang diuji sangat rendah.

Berdasarkan penelitian yang dilakukan oleh Xioa-ming yang memiliki tingkat akurasi lebih tinggi dari metode lainnya, maka penelitian pada tugas akhir ini menggunakan metode yang sama yang digunakan oleh Xioa-ming dan Omid Sakhi yaitu metode filter Gabor dan *Support Vector Machine* (SVM). Tetapi tahapan yang digunakan melanjutkan tahapan yang dilakukan oleh Omid Sakhi pada penelitiannya. Hal ini dikarenakan tahapan penelitian yang dilakukan oleh Omid Sakhi tersebut masih bersifat dasar dimana Omid Sakhi hanya bertujuan untuk mencoba mengimplementasikan filter Gabor baru yang dibuatnya dan mengklasifikasikannya dengan menggunakan metode *Support Vector Machine* (SVM). Selain itu data fitur yang digunakan pada penelitiannya masih berupa data fitur biasa berupa tipe-tipe dasar wajah manusia yang tidak terlalu spesifik sehingga akurasi yang dihasilkan dari penelitian yang dilakukannya masih rendah. Oleh karena itu dilihat dari kekurangan pada tahapan penelitian yang dilakukan oleh Omid Sakhi, pada tugas akhir ini dilakukan penelitian lanjutan

untuk mengembangkannya dengan memperluas pengetahuan tentang fitur yang dibutuhkan pada pendeteksian wajah agar mendapatkan nilai akurasi lebih tinggi.