

BAB I

PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Diharapkan dari penjelasan dalam bab ini gambaran Tugas Akhir secara umum dapat dipahami.

1.1 Latar Belakang

Teknologi Biometrik adalah metode untuk mengenali seseorang melalui ciri-ciri fisik, karakter, dan kebiasaan. Biasanya ciri-ciri fisik khas yang dijadikan indikator untuk mengenali seseorang adalah wajah, sidik jari, telapak tangan, retina, atau iris mata. Dapat pula mengenali seseorang melalui ciri-ciri lainnya seperti cara berjalan, tanda tangan, atau suara. Dengan menggunakan macam-macam ciri yang unik tersebut, dapat dijadikan solusi untuk mengidentifikasi dan mengenali seseorang.

Iris atau selaput pelangi adalah daerah berbentuk gelang pada mata yang dibatasi oleh pupil dan sklera (bagian putih dari mata). Dalam penerapannya pada teknik biometrik, iris memiliki pola yang sangat unik, berbeda pada tiap individu dan pola itu akan tetap stabil. Sebelum kelahiran, degenerasi terjadi sehingga menghasilkan pembukaan pupil dan acak, serta pola-pola unik dari iris. Walaupun genetik serupa, seseorang yang memiliki struktur iris yang unik dan berbeda, dapat memungkinkan untuk digunakan untuk tujuan pengenalan. Iris mata juga mudah untuk dicitrakan pada jarak yang sesuai dari subjek dengan penggunaan alat yang ada, misalnya kamera. Atas dasar inilah iris mata dapat dijadikan dasar bagi pengenalan biometrik [1].

Metode pengklasifikasian yang dapat digunakan adalah *Support Vector Machine* (SVM). Pengklasifikasian dengan metode ini terdiri dari dua modul utama, yaitu pembuatan

model dan klasifikasi data. Model SVM yang dibuat dimaksudkan untuk memisahkan satu kategori dengan satu kategori yang lain dengan menggunakan sebuah bidang *hyperlane*. Selanjutnya klasifikasi akan dilakukan dengan menggunakan model yang dibuat, membuatnya mampu memberikan kinerja yang lebih tinggi dalam hal akurasi dari algoritma klasifikasi lainnya [2].

Salah satu metode lainnya dalam digunakan dalam pengklasifikasian adalah *Hamming distance*. Metode ini merepresentasikan pola iris kedalam bentuk biner, karena lebih mudah untuk menentukan perbedaan antara dua kode biner daripada bilangan biasa. Metode ini pada dasarnya adalah fungsi eksklusif OR (XOR) antara dua pola bit.

Tugas Akhir ini mengimplementasikan pengklasifikasi tekstur iris mata dengan menggunakan masing-masing dua metode klasifikasi tersebut. Sebelum dilakukan klasifikasi, terlebih dahulu dilakukan praproses untuk identifikasi bagian iris pada citra mata. Selanjutnya dilakukan normalisasi iris pada *region of interest*, serta ekstraksi fitur untuk menghasilkan vektor fitur yang digunakan untuk proses klasifikasi.

Beberapa metode telah diaplikasikan dengan menggunakan iris sebagai teknologi biometrik. Pada proses identifikasi iris sampai normalisasi relatif sama. Metode ekstraksi fitur yang sudah pernah diaplikasikan adalah *Gabor filter*, *Bayesian*, dan yang lainnya untuk menghasilkan vektor fitur tekstur iris. Pada tugas akhir ini menggunakan metode *log-Gabor filter* dan *wavelet Haar* sebagai metode ekstraksi fitur untuk menghasilkan vektor fitur tekstur iris. Metode *log-Gabor filter* menutupi kelemahan yang terdapat pada *Gabor filter* yang membuat informasi tekstur iris berkurang banyak [3]. Pada metode *Bayesian* memakai parameter deformasi untuk menghasilkan vektor fitur tekstur iris [4]. Hal-hal tersebutlah yang membedakan antara pengimplementasian yang sudah dilakukan sebelumnya dengan apa yang diimplementasikan pada tugas akhir ini.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

- a) Bagaimana pengklasifikasian dengan *Support Vector Machine* pada pengenalan iris mata ?
- b) Bagaimana pengklasifikasian dengan *Hamming distance* pada pengenalan iris mata ?
- c) Bagaimana melakukan identifikasi bagian iris pada citra mata ?
- d) Manakah representasi klasifikasi yang lebih baik antara *Support Vector Machine* dan *Hamming distance* ?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan, yakni sebagai berikut.

1. Database yang digunakan adalah data citra mata dari basis data *Chinese Academy of Sciences Institute of Automation* (CASIA).
2. Implementasi menggunakan perangkat lunak MATLAB 8.3 dan *Image Processing Toolbox* didalamnya.
3. Data yang digunakan untuk data latih dan data uji berupa citra mata dari database CASIA.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah membandingkan dua metode klasifikasi dalam melakukan pengenalan iris mata. Selain itu, membuat sebuah implementasi perangkat lunak yang dapat melakukan pengenalan iris mata.

1.5 Manfaat

Pengerjaan tugas akhir ini dilakukan dengan harapan bisa memberikan kontribusi pada teknologi biometrik yang dapat melakukan pengenalan melalui iris mata.

1.6 Metodologi

Metodologi yang dipakai pada pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Penyusunan Proposal Tugas Akhir

Tahap awal yang dilakukan dalam pengerjaan Tugas Akhir ini adalah penyusunan proposal Tugas Akhir. Di dalam proposal diajukan suatu gagasan untuk melakukan implementasi pengklasifikasian pada iris mata pada metode *Support Vector Machine* dan *Hamming distance*.

2. Studi Literatur

Pada tahap ini dilakukan pencarian, pengumpulan, pembelajaran dan pemahaman informasi dan literatur yang diperlukan untuk pembuatan implementasi pada tahap praproses citra mata, ekstraksi fitur serta metode-metode klasifikasi. Dasar informasi yang diperlukan pada pembuatan implementasi ini diantaranya mengenai citra secara umum dan algoritma tahap praproses. Serta metode pengklasifikasi pada iris mata. Literatur yang digunakan meliputi: buku referensi, jurnal, dan dokumentasi internet. Selain itu juga dipelajari database citra mata CASIA yang bisa didownload dari <http://biometrics.idealtest.org>.

3. Implementasi perangkat lunak

Pada tahap ini dilakukan implementasi perangkat lunak sesuai dengan rancangan perangkat lunak yang dibuat pada pengerjaan Tugas Akhir.

4. Uji coba dan Evaluasi

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat untuk mengetahui kemampuan algoritma yang dipakai, mengamati kinerja sistem. Ujicoba ini bertujuan untuk mengetahui performa algoritma ekstraksi fitur *wavelet Haar* dan *log-Gabor filter* pada metode klasifikasi *Support Vector Machine* dan *Hamming distance*.

5. Penyusunan Laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan pengerjaan Tugas Akhir yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil yang diperoleh selama pengerjaan Tugas Akhir.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

Bab II Dasar Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

Bab III Perancangan Perangkat Lunak

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *flowchart*. *Flowchart-flowchart* tersebut memperlihatkan diagram alir bagaimana proses-proses berlangsung selama pengimplementasian.

Bab IV Implementasi

Bab ini berisi implementasi dari perancangan yang telah dibuat sebelumnya. Penjelasan berupa code yang digunakan untuk proses implementasi.

Bab V Uji Coba dan Evaluasi

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

BAB II

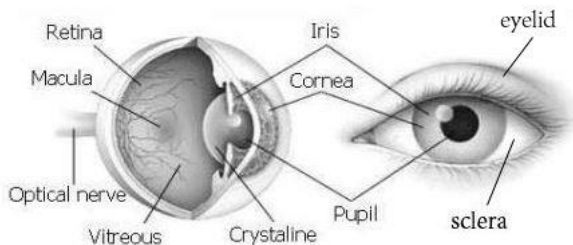
TINJAUAN PUSTAKA

Pada bab ini dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir.

2.1 Anatomi Mata

Bagian organ mata yang akan diteliti dalam tugas akhir ini adalah iris atau selaput pelangi. Bagian-bagian mata seperti yang tampak pada Gambar 2.1 dijelaskan sebagai berikut [5]:

1. Retina adalah bagian yang berfungsi untuk menangkap bayangan benda
2. Iris adalah bagian yang mempunyai pigmen untuk memberi warna pada mata. Bagian inilah yang akan diteliti pada Tugas Akhir ini
3. Kornea berfungsi untuk meneruskan cahaya yang masuk ke mata
4. Pupil merupakan bagian mata yang berbentuk lingkaran yang mengatur banyaknya cahaya yang masuk ke mata. Bagian ini akan diteliti juga untuk mendapatkan bagian iris
5. *Sclera* adalah bagian pelindung mata yang bewarna putih pada mata bagian luar
6. *Optical Nerve* adalah saraf mata yang meneruskan atau membelokkan sinar menuju ke otak.



Gambar 2.1 Anatomi Mata

2.2 Sistem Pengenalan Iris

Pengenalan iris adalah suatu proses untuk mengenal seseorang dengan menganalisa pola acak dari iris. Dimana pengenalan iris bertujuan untuk mengenali suatu objek dengan cara mengekstraksi informasi yang terdapat dalam suatu citra tersebut.

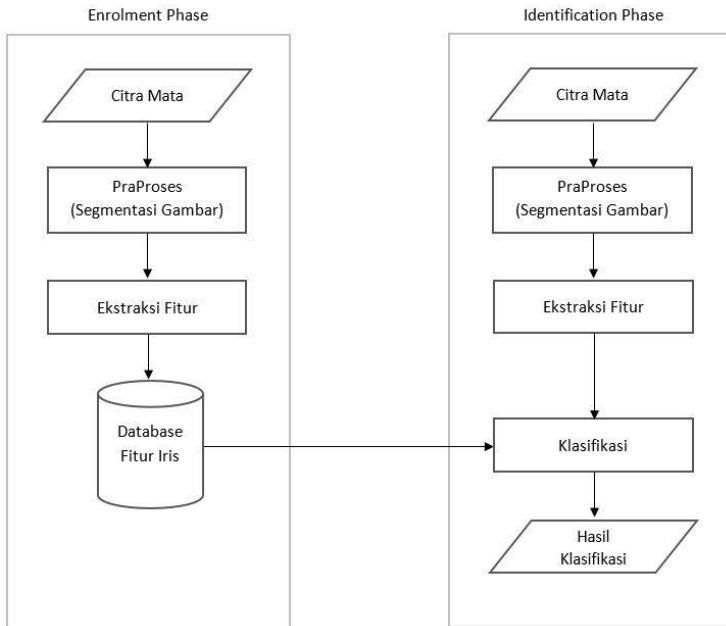
Dalam pengaplikasiannya, sudah banyak metode yang digunakan. Pada proses identifikasi iris sampai normalisasi relatif sama. Metode dengan menggunakan *wavelet* [1], *Laplacian of Gaussian Filter* [6], dan berbagai macam metode lainnya. Metode-metode tersebut digunakan pada proses ekstraksi fitur. Sedangkan pada proses pengenalan, ada pula klasifikasi *Support Vector Machine* [7], *Hamming distance* [8], dan metode-metode yang lainnya.

Secara garis besar, proses-proses tersebut dikelompokkan pada empat proses utama yaitu:

1. Akuisisi citra
2. Praproses citra (identifikasi objek iris dan normalisasi)
3. Ekstraksi fitur iris
4. Pencocokan iris.

Akuisisi citra adalah proses mendapatkan citra mata yang akan dilakukan pengenalan. Citra mata yang digunakan pada tugas akhir ini yaitu citra mata CASIA [9]. Proses selanjutnya adalah praproses citra atau melakukan pengolahan awal pada citra mata yang bertujuan untuk mengambil karakteristik tekstur iris mata dan melakukan normalisasi iris untuk mengatasi kondisi citra mata yang bervariasi pada setiap orang. Tekstur iris yang sudah terambil akan diekstraksi ciri atau fitur iris. Setiap fitur akan disimpan didalam sebuah database yang nantinya akan dibandingkan dengan fitur masukan dalam proses pencocokan iris. Proses pencocokan fitur masukan dengan seluruh fitur yang terdapat didalam database menghasilkan bobot atau tingkat kemiripan. Nilai tingkat kemiripan yang paling tinggi adalah citra yang dikenali

paling mirip dengan citra masukan. Gambar 2.2 menjelaskan pengenalan iris secara umum.



Gambar 2.2 Sistem Pengenalan Iris secara Umum

2.3 Representasi Citra Digital

Citra dapat didefinisikan sebagai $f(x,y)$, yaitu fungsi dua dimensi, dengan x dan y menyatakan koordinat spasial dan nilai f pada sembarang titik (x,y) disebut dengan intensitas atau tingkat keabuan dari citra pada koordinat tersebut. Jika nilai x,y dan nilai f adalah *finite*, bernilai diskrit atau terbatas, maka dapat disebut bahwa citra tersebut merupakan citra digital [10].

Sebuah citra digital $f(x,y)$ dapat direpresentasikan sebagai sebuah matriks yang indeks baris dan kolomnya mengidentifikasi sebuah titik pada citra dan nilai dari

elemen matriks yang bersangkutan merupakan tingkat warna pada titik tersebut, seperti yang ditunjukkan pada Gambar 2.3. Elemen tersebut disebut elemen citra, elemen gambar, pixels, atau pels. *Picture elements* atau pixel dapat didefinisikan sebagai elemen terkecil dari sebuah citra digital yang menentukan resolusi citra tersebut.

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,M) \\ f(1,0) & f(1,1) & \cdots & f(1,M) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Gambar 2.3 Representasi Citra Digital

2.4 Deteksi Tepi *Canny*

Deteksi Tepi adalah langkah awal dalam pemrosesan citra digital untuk mendapatkan informasi pada citra tersebut. Mendapatkan jenis informasi berbeda dari citra dengan jenis yang berbeda memerlukan metode yang berbeda pula. Tepi adalah perubahan nilai intensitas tingkat keabuan yang besar yang memperlihatkan rincian pada gambar. Tujuan deteksi tepi adalah untuk mengekstraksi fitur penting pada suatu citra, misalnya garis, lingkaran, lengkungan, ujung. Salah satu metode deteksi tepi yang terkenal adalah deteksi tepi *Canny*.

Deteksi tepi *Canny* sangat cocok digunakan untuk pengolahan awal citra digital. Deteksi tepi ini sangat banyak digunakan oleh peneliti karena memiliki keuntungan sebagai berikut:

1. Tepi yang dihasilkan akurat
2. Banyak tepi yang dihasilkan

Pada citra iris mata, deteksi tepi ini dilakukan dengan untuk mendeteksi citra iris mata dengan langkah-langkah berikut [11].

- a. Mengurangi noise dengan menggunakan filter Gaussian sebesar 13x13, standar deviasi iris dengan sclera, dan iris dengan pupil sebesar 2 [12].
- b. Menentukan intensitas gradien dari citra yaitu penentuan arah gradien ke suatu arah tertentu. Arah tersebut yaitu vertikal, horizontal, diagonal kiri&kanan. Perhitungan intensitas gradien dilakukan menggunakan persamaan (2.1).

$$G = \sqrt{G_x^2 + G_y^2} \quad (2.1)$$

dimana G_x = gradien ke arah sumbu x
 G_y = gradien ke arah sumbu y
 G = intensitas gradien

selanjutnya mencari arah gradien dengan persamaan (2.2)

$$\theta = \arctan \frac{G_y}{G_x} \quad (2.2)$$

dimana θ adalah arah gradien

Kemudian dilakukan pengelompokkan para arah gradien yang diperoleh dengan kondisi berikut.

- Jika arah tepi yang berkisar antara 0° - 22.5° serta 157.5° - 180° , arah gradien diubah menjadi 0° (horizontal).
- Jika arah tepi yang berkisar antara 22.5° - 67.5° , arah gradien diubah menjadi 45° (diagonal kanan).
- Jika arah tepi yang berkisar antara 67.5° - 112.5° , arah gradien diubah menjadi 90° (vertikal).
- Jika arah tepi yang berkisar antara 112.5° - 157.5° , arah gradien diubah menjadi 135° (diagonal kiri).

Karena bagian iriss berada diantara tepi sclera dan tepi iris, serta tepi iris dan tepi pupil. Maka dari itu dilakukan pendeteksian secara berturut-turut dengan menggunakan arah gradien vertikal dan horizontal.

- c. Mengatur Gamma atau pengontrasan terhadap arah yang telah terdeteksi. Pengontrasan dilakukan menggunakan persamaan (2.3)

$$M = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \quad (2.3)$$

dimana I_{max} = nilai citra intensitas maksimum

I_{min} = nilai citra intensitas minimum

M = nilai citra intensitas rata-rata

Kemudian mendapatkan hasil akhir pengontrasan dengan persamaan (2.4)

$$C = M^{1/g} \quad (2.4)$$

dimana C = citra gamma

g = nilai gamma

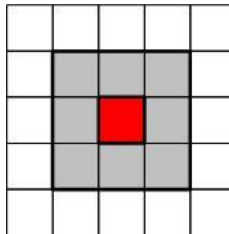
M = nilai citra intensitas rata-rata

Citra akan semakin gelap jika nilai gamma berada diantara 0-1, namun semakin terang jika nilai gamma diatas 1. Pengontrasan dilakukan untuk mempermudah dalam penelusuran tepi citra iris mata.

- d. Proses *non-maximum suppression* atau memperkecil garis tepi yang muncul sehingga menghasilkan garis tepian yang lebih ramping. Proses ini menjadikan nilai piksel yang dianggap tidak layak menjadi sebuah tepi dengan nilai nol dari citra tepi pengontrasan.

- e. Langkah terakhir adalah proses *hystheresis*, proses ini menerapkan dua buah bilangan threshold, yaitu *high threshold* (T1) dan *low threshold* (T2). Kondisi penelusuran tepi pada proses ini adalah sebagai berikut:
1. Semua piksel diatas T1 ditandai sebagai tepi.
 2. Semua piksel yang berdekatan dengan titik yang telah ditandai sebagai tepi dan mempunyai nilai diatas T2 juga ditandai sebagai tepi.

Menggunakan 8-konektivitas piksel untuk mengetahui titik yang berdekatan dengan titik yang telah ditandai sebagai tepi. Dalam Gambar 2.4 memperlihatkan ilustrasi 8-konektivitas.



Gambar 2.4 Ilustrasi 8-konektivitas

Warna merah pada gambar diatas menunjukan titik yang telah ditandai sebagai tepi jika memenuhi syarat satu, sedangkan warnal abu merupakan koordinat piksel yang akan ditandai sebagai tepi jika memenuhi syarat dua.

2.5 Transformasi Hough

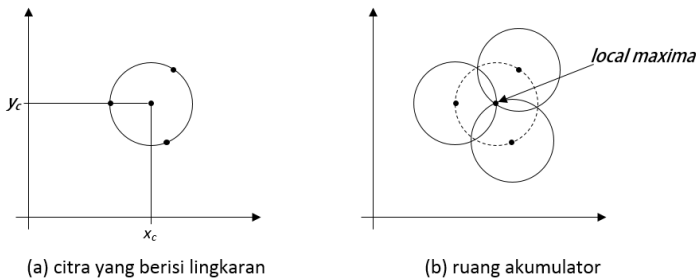
Transformasi Hough adalah suatu metode untuk mencari bentuk fitur tertentu seperti garis, lingkaran, atau bentuk sederhana lain dalam suatu citra. Pada umumnya, transformasi Hough diterapkan pada banyak permasalahan pada visi komputer karena kebanyakan citra mempunyai batas tepi yang membentuk sebuah garis. Implementasi transformasi

Hough menjelaskan sebuah pemetaan pada ruang akumulator. Dalam hal ini, transformasi Hough membahas pada pencarian bentuk lingkaran atau biasa disebut Transformasi Hough Lingkaran (*Circular Hough Transform*). Pencarian lingkaran ini menggunakan persamaan (2.5) untuk mencari titik pusat dan jari-jari pada lingkaran iris dan lingkaran pupil [13].

$$(x_c - a)^2 + (y_c - b)^2 = r^2 \quad (2.5)$$

dimana (x_c, y_c) = koordinat titik-titik lingkaran
 (a, b) = koordinat pusat lingkaran
 r = jari-jari lingkaran

Pada proses pendeteksian iris, transformasi Hough lingkaran digunakan untuk mendeteksi area iris dengan cara mencari radius dan titik pusat lingkaran pupil dan iris. Tahap awal dari proses ini adalah mendeteksi tepi citra. Tujuan dari deteksi tepi adalah untuk menurunkan jumlah titik dalam pencarian ruang bagi objek. Ketika titik tepi sudah ditemukan, transformasi Hough akan bekerja pada titik tersebut [11].



Gambar 2.5 Ilustrasi Transformasi Hough Lingkaran

Transformasi Hough lingkaran membentuk lingkaran sepanjang titik yang ditemukan dengan jari-jari sebesar r . Kemudian mencari *voting* untuk mencari titik yang sering dilewati (*local maxima*) dari lingkaran yang telah dibentuk dan

titik tersebut diasumsikan sebagai titik pusat lingkaran. Pada Gambar 2.5 adalah ilustrasi Transformasi Hough lingkaran. Hasil akhirnya adalah sebuah citra iris melalui transformasi Hough.

2.6 Transformasi Polar

Setelah daerah iris teridentifikasi dari citra mata, langkah berikutnya adalah melakukan transformasi citra iris sehingga memiliki ukuran yang konstan untuk memungkinkan pencocokan. Pada dasarnya, daerah iris akan menciut dan meregang disebabkan oleh pelebaran pupil karena faktor pencahayaan yang beragam. Faktor yang mempengaruhi pelebaran pupil yang lain juga seperti jarak citra, rotasi, dan derajat kemiringan. Proses normalisasi akan menghasilkan daerah iris yang memiliki ukuran yang konstan. Sehingga jika dilakukan pencocokan antara dua daerah iris akan lebih mudah [12].

Metode yang digunakan untuk melakukan normalisasi pada citra iris adalah mengubah citra iris pada bidang Cartesian ke dalam bidang polar dengan metode *Daugman's rubber sheet model* seperti yang terlihat pada Gambar 2.6. Metode ini memetakan daerah iris yang tersegmentasi pada koordinat (x, y) ke koordinat polar (r, θ) . Nilai r berada pada interval $[0, 1]$ dan nilai θ mempunyai interval derajat sudut $[0, 2\pi]$. Pemetaan daerah iris tersebut dimodelkan dengan persamaan 2.6 untuk mencari citra normalisasi, dan didalamnya memakai persamaan (2.7) dan persamaan (2.8) untuk mencari batas tepi pupil dan batas tepi iris.

$$I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta) \quad (2.6)$$

dengan

$$x(r, \theta) = (1 - r)x_p(\theta) + rx_i(\theta) \quad (2.7)$$

$$y(r, \theta) = (1 - r)y_p(\theta) + ry_i(\theta) \quad (2.8)$$

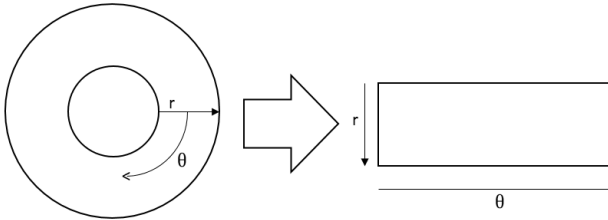
dimana $I(x, y)$ adalah citra iris

(x, y) adalah titik pada koordinat Cartesian

(r, θ) adalah koordinat pada polar normalisasi

x_p, y_p adalah koordinat batas tepi pupil

x_i, y_i adalah koordinat batas tepi.



Gambar 2.6 *Daugman's rubber sheet model*

Rubber sheet model pada Gambar 2.6 mengatasi pelebaran yang dilakukan pupil dan ketidakkonsistenan ukuran pada citra iris untuk menghasilkan representasi yang ternormalisasi dengan ukuran yang tetap. Dengan begitu, saat melakukan pencocokan dengan berbagai macam iris bisa lebih mudah dan fleksibel dengan batas luar tepi iris dan titik tengah pupil sebagai titik acuan.

2.7 Haar Wavelet

Wavelet adalah sebuah fungsi matematika yang mampu melakukan dekomposisi terhadap sebuah fungsi. Transformasi *wavelet* akan mengkonversi signal ke dalam sederetan gelombang singkat. Transformasi *wavelet* merupakan sebuah fungsi konversi yang dapat membagi data ke dalam komponen frekuensi yang berbeda-beda dan menganalisis komponen tersebut dengan resolusi tertentu sesuai dengan skalanya. *Wavelet* telah banyak diaplikasikan dalam berbagai bidang, salah satunya adalah pengolahan citra [14]. Pada citra,

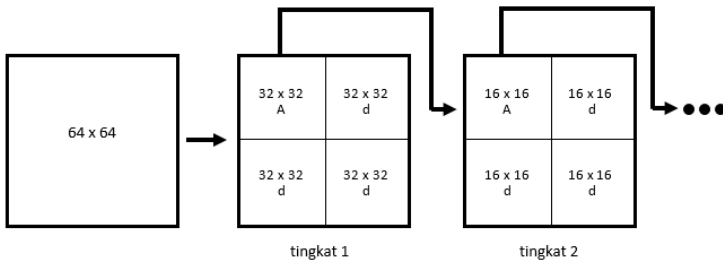
transformasi *wavelet* adalah transformasi yang melakukan filter terhadap suatu masukan. Filter yang digunakan dalam *wavelet Haar* adalah *high pass filter* dan *low pass filter*. Persamaan (2.9) dan persamaan (2.10) adalah *filter coefficient* dari *low pass filter* dan *high pass filter* pada *wavelet Haar* berturut-turut [10].

$$h_0 = h_1 = \frac{1}{\sqrt{2}} \quad (2.9)$$

$$g_0 = \frac{1}{\sqrt{2}} \quad (2.10)$$

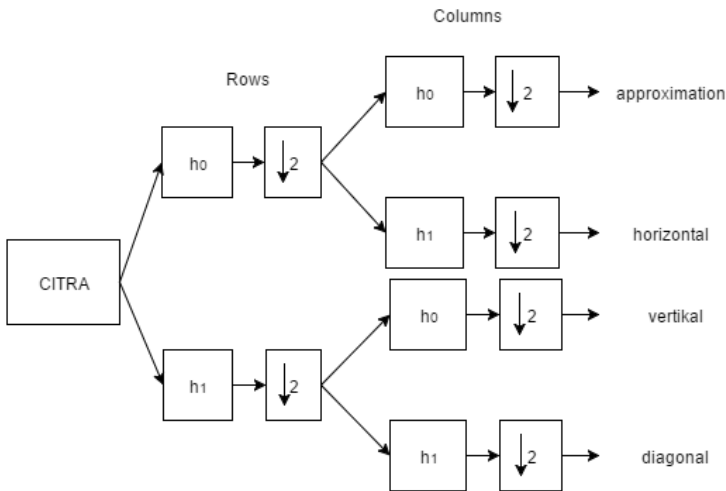
$$g_1 = -\frac{1}{\sqrt{2}}$$

Resolusi citra yang dilakukan *filter* akan ter-reduksi menjadi setengah dari citra awal. Hasil dari proses ini adalah empat subbidang dari citra awal. Keempat subbidang citra ini adalah citra aproksimasi, dan tiga citra detail. Citra detail yang didapat adalah detail horizontal, vertikal dan diagonal. Proses ini disebut dekomposisi, dekomposisi dapat dilanjutkan kembali dengan citra aproksimasi sebagai masukannya untuk mendapatkan tingkat dekomposisi selanjutnya seperti yang ditunjukkan pada Gambar 2.7 [1].



Gambar 2.7 Ilustrasi Dekomposisi *Wavelet*

Tahapan dari proses filter adalah melakukan filter terhadap baris dan kolom citra seperti pada Gambar 2.8. Untuk mendapatkan citra aproksimasi dan citra diagonal dilakukan dengan filter terhadap baris dan kolom. Kemudian melakukan filter terhadap baris saja untuk mendapatkan citra horizontal, dan terakhir melakukan filter terhadap kolom saja untuk mendapatkan citra vertikal [10].



Gambar 2.8 Filter pada *Wavelet*

2.8 Log-Gabor Filter

Log-Gabor filter adalah sebuah metode yang digunakan untuk memperoleh informasi frekuensi dan dapat menentukan lokasi bagian mana dari sinyal yang menghasilkan frekuensi tertentu. Metode ini banyak digunakan dalam karakteristik tekstur dari suatu citra dengan mencari representasi gabungan optimal pada domain spasial dan frekuensi. Metode ini juga menutupi kelemahan pada metode *Gabor filter* yang memiliki *komponen DC* (*direct current*) ketika bobotnya lebih dari satu oktaf [3]. *Komponen DC* adalah nilai hasil dari transformasi pada frekuensi awal. *Komponen*

DC akan bernilai nol pada bobot berapapun jika menggunakan skala logaritmik yang terdapat pada rumus frekuensi *log-Gabor filter* pada persamaan (2.11).

$$G(f) = \exp\left(-\frac{\left(\log\left(f/f_0\right)^2\right)}{2(\log(\sigma)^2)}\right) \quad (2.11)$$

dengan f adalah nilai frekuensi, f_0 adalah pusat frekuensi dengan nilai $1/\lambda$ dan σ adalah bobot filter. Berdasarkan penelitian Masek(2003) pada pengenalan citra iris mata, ditetapkan nilai λ sebesar 18, σ sebesar 0.5, dan interval f_0 sebesar $[0-0.5]$ [11].

Proses *log-Gabor filter* ini membutuhkan sinyal dari transform Fourier yang berasal dari setiap baris citra hasil normalisasi iris, karena akan direpresentasikan pada domain frekuensi. Kemudian dilakukan konvulsi antara sinyal citra iris dengan filter *log-Gabor* pada setiap bidang satu dimensi. Hasil tersebut akan diubah kembali kedalam bidang spasial dengan inverse transform Fourier. Tiap nilai piksel dari hasil tersebut memiliki komponen bilangan *real*, sehingga perlu dikuantisasi menjadi nilai yang terdiri atas bilangan biner dua bit, dengan ketentuan:

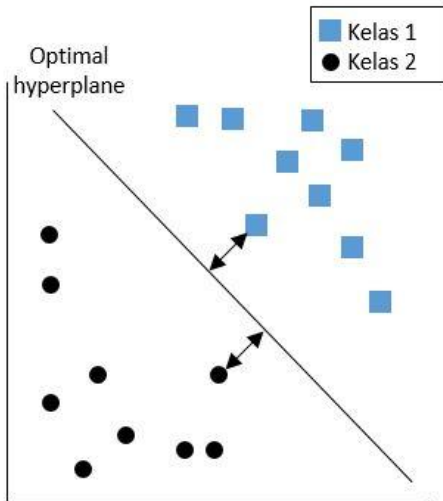
1. jika nilai *real* > 0 , nilai piksel menjadi 1,
2. jika nilai *real* < 0 , nilai piksel menjadi 0.

2.9 Support Vector Machine

Support Vector Machine (SVM) adalah metode klasifikasi yang hanya memfokuskan pada klasifikasi dua kelas, yaitu kelas +1 dan -1. Pada metode klasifikasi SVM, kedua kelas tersebut digambarkan ditaruh didalam sebuah bidang. Perlu untuk membagi bidang kedalam dua bagian dan masing-masing bagian merepresentasikan satu kelas. Dua kelas tersebut

dipisahkan dengan sebuah *hyperplane*. *Hyperplane* adalah garis pemisah yang memisahkan dua kelas yang berbeda.

Tujuan *hyperplane* adalah mencari fungsi *hyperplane* yang terbaik karena fungsi *hyperplane* itu sendiri mempunyai jumlah kemungkinan yang tak terbatas. Fungsi *hyperplane* dirumuskan dengan persamaan (2.12). Karena tujuan akhirnya adalah mencoba untuk menggunakan *hyperplane* sebagai batas keputusan untuk membedakan dua kelas, sehingga dapat memilih *hyperplane* yang dapat membuat perbedaan yang lebih jelas. Oleh karena itu *hyperplane* yang dicari haruslah yang mempunyai jarak terjauh antara dua kelas tersebut. Jarak atau yang biasa dikenal dengan istilah *margin*. Hal itu dapat dilakukan dengan cara mencari nilai w dan b yang optimal, dilakukan dengan persamaan (2.13). Ilustrasi SVM dapat dilihat pada Gambar 2.9 [15].



Gambar 2.9 Ilustrasi *Support Vector Machine*

$$w \cdot x + b = 0 \quad (2.12)$$

$$\min \frac{1}{2} ||w||^2 + C \sum_{i=1}^l t_i \quad (2.13)$$

Pada persamaan diatas, variabel C adalah konstanta nilai pinalti dari kesalahan klasifikasi. Pencarian nilai w dan b dilakukan dengan menggunakan batasan yang ditulis menggunakan persamaan (2.14)

$$y_i(wx_i + b) + t_i \geq 1 \quad (2.14)$$

Persamaan (2.13) dilakukan untuk mencari nilai w dan b yang optimum. Fungsi tujuan Persamaan (2.13) berbentuk kuadrat pada variabel w . Oleh karena itu pengoptimasian pada persamaan (2.13) yang mempunyai nilai pinalti, bentuk tersebut ditransformasi dengan menggunakan pendekatan *Lagrange* atau bentuk *dual space*. Persamaan *dual space* dapat ditulis menggunakan persamaan (2.15).

$$\max \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (2.15)$$

dengan batasannya terdapat pada persamaan (2.16).

$$\alpha_i \geq 0, \sum_{i=1}^l \alpha_i y_i = 0 \quad (2.16)$$

Untuk mencari nilai α_i , digunakan *quadratic programming (QR)*. Setelah mendapatkan nilai α_i , parameter persamaan *hyperplane* menjadi seperti persamaan (2.17).

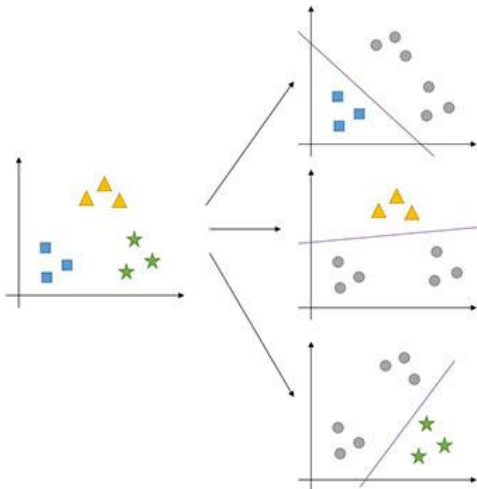
Kemudian hasilnya disubstitusikan ke fungsi *hyperlane* dan dirumuskan kedalam persamaan (2.18).

$$w = \sum_{i=1}^s \alpha_i y_i x_i^T \quad (2.17)$$

$$f = w^T z + b = \sum_{i=1}^s \alpha_i y_i x_i^T z + b \quad (2.18)$$

Dimana z adalah data training yang dimasukan. Pada banyak kasus, data training yang diklasifikasikan tidak bisa dipisahkan dengan garis yang linear.

Metode klasifikasi ini pula dapat digunakan dalam hal mengklasifikasi multi kelas. Pada klasifikasi multi kelas, output dari set data memiliki lebih dari dua kelas atau kategori. Secara garis besar tahapannya sama dengan cara klasifikasi dua kelas, namun berbeda konsep [16].



Gambar 2.10 Ilustrasi Pembuatan Model SVM Multi Kelas

SVM multi kelas membutuhkan proses pemodelan pada data training yang lebih banyak dibandingkan jika hanya dua kelas. Tahapan pertama metode multi kelas ini yaitu melakukan pembuatan model atau pemisahan setiap kelas dengan kelas yang lain. Hal itu dilakukan sebanyak jumlah kelas. Kemudian saat hendak memprediksi sebuah kelas data masukan, dibandingkan satu per satu terhadap model yang sudah dibuat. Ilustrasi SVM multi kelas dapat dilihat pada Gambar 2.10.

Akurasi model yang akan dihasilkan bergantung pada fungsi kernel serta parameter yang digunakan. Oleh karena itu digunakan metode kernel untuk mencari optimasi terbaik. Dengan metode kernel, suatu data x di *input space* dimapping ke fitur *space* F dengan dimensi yang lebih tinggi. Beberapa fungsi kernel yang terkenal digunakan pada model SVM dirumuskan pada persamaan (2.19) [11].

$$k(x, y) = \begin{cases} x \cdot y \\ (\gamma \langle x^T y \rangle + C)^p \\ \exp(-\gamma |x - y|^2) \end{cases} \quad (2.19)$$

Persamaan diatas adalah fungsi kernel linear, fungsi kernel polynomial dan radial basis function (RBF) secara berturut-turut. Persamaan diatas mempunyai parameter kernel yaitu p dan γ . Dimana variabel p adalah parameter yang menandakan *order*. Sedangkan variabel γ menandakan *scaling factor*. Dalam fungsi kernel ini, RBF adalah kernel yang paling bagus untuk digunakan karena alasan berikut [17]:

1. RBF kernel memetakan kedalam ruang dimensi yang lebih tinggi daripada linear kernel,
2. RBF kernel memiliki *hyperparameters* lebih sedikit daripada polynomial kernel,
3. RBF kernel memiliki kesulitan lebih sedikit dalam hal numerik.

2.10 Hamming Distance

Hamming distance adalah metode untuk melakukan pencocokan antara nilai intensitas pada citra yang sudah melalui proses ekstraksi fitur dan menghasilkan vektor fitur. Vektor fitur direpresentasikan dengan bilangan biner dua bit. Penggunaan dengan bilangan biner lebih mudah untuk menentukan perbedaan antara dua kode biner daripada menggunakan bilangan biasa. Selain itu, bilangan boolean lebih mudah untuk dibandingkan dan dimanipulasi [13]. Tabel operasi eksklusif OR (XOR) pada Tabel 2.1 memperlihatkan jika dua bit memberikan nilai biner yang kembar (1 atau 0), maka operasi XOR memberikan nilai 0 pada perbandingan tersebut. Sebaliknya, jika dua bit yang dibandingkan itu berbeda, operasi akan memberikan nilai 1 pada perbandingan tersebut.

Tabel 2.1 Operasi XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Pada proses pengenalan iris, *Hamming distance* memberikan ukuran seberapa banyak bit yang sama antara dua vektor fitur. Metode ini pada dasarnya adalah fungsi eksklusif OR (XOR) antara dua pola bit. Setiap bit pada vektor fitur citra masukan dibandingkan dengan vektor fitur dalam database. Dalam melakukan perbandingan vektor fitur bit X dan Y , *Hamming distance*, HD , didefinisikan dengan persamaan (2.20) [8].

$$HD = \frac{1}{N} \sum_{j=1}^N X_j (XOR) Y_j \quad (2.20)$$

dimana N adalah dimensi dari vektor fitur

X_j adalah komponen ke- j dari vektor fitur

Y_j adalah komponen ke- j dari vektor fitur

Jika dua bit vektor fitur iris yang benar-benar berbeda jauh atau berasal dari iris yang berbeda, maka hasil dari jarak *Hamming* menjauhi nilai 0.0. Sebaliknya jika dua bit vektor iris berasal dari iris yang sama, jarak *Hamming* akan mendekati nilai 0.0 karena saling berkorelasi.

2.11 Bahasa Pemrograman Matlab 8.3

Matlab (*Matrix Laboratory*) adalah sebuah bahasa pemrograman komputer generasi keempat dengan performansi yang tinggi. Matlab menggabungkan komputasi, pemrograman, dan visualisasi pada lingkungan yang mudah digunakan. Permasalahan dan solusinya dinyatakan dalam notasi matematika yang sudah dikenal.

Matlab merupakan suatu sistem interaktif yang memiliki elemen data dalam suatu array sehingga tidak memerlukan dimensi. Hal ini memungkinkan untuk memecahkan banyak masalah teknis yang terkait dengan komputasi, khususnya yang berhubungan dengan matriks dan vektor. Matlab menyediakan akses mudah ke aplikasi matriks yang dikembangkan oleh proyek LINPACK dan EISPACK. Saat ini, mesin Matlab menggabungkan *library* LAPACK dan BLAS. *Tool* ini merupakan satu kesatuan dari sebuah seni tersendiri pada aplikasi untuk komputasi matriks.

Matlab berkembang dari tahun ke tahun dengan berbagai masukan dari penggunaanya. Bahasa ini adalah *tool* standar untuk memperkenalkan dan mengembangkan penyajian materi matematika, rekayasa dan kelimuan. Sistem Matlab terdiri dari lima bagian utama yaitu:

1. *Development Environment*. *Tool* dan fasilitas yang membantu penggunaan fungsi dan file Matlab. Sebagian

besar tool ini berbasis *graphical user interface*. Bagian ini terdiri dari Matlab *desktop*, *command window*, *command history*, *editor*, *debugger*, *help*, *workspace*, dan file

2. *The Matlab Mathematical Function Library*. Sekumpulan algoritma komputasi mulai dari fungsi-fungsi dasar seperti: *sum*, *sinus*, *cosinus*, dan aritmatika kompleks, sampai dengan fungsi-fungsi tingkat lanjut seperti *matriks eigenvalues* dan *fast Fourier transforms*.
3. *The Matlab Language*. Bahasa pemrograman matriks tingkat tinggi dengan *flow control* fungsi, struktur data, input/output.
4. *Graphics*. Matlab mempunyai fasilitas untuk menampilkan vektor dan matriks dalam bentuk grafik. Fungsi yang disediakan meliputi visualisasi data dua dan tiga dimensi, pengolahan citra, animasi, dan penampilan grafik.
5. *The Matlab Application Program Interface (API)*. Merupakan suatu library yang memungkinkan program dalam bahasa C dan Fortran mampu berinteraksi dengan MATLAB. Ini melibatkan fasilitas untuk pemanggilan MATLAB sebagai sebuah computational engine, dan untuk membaca dan menuliskan file berekstensi .MAT.