

# Jurnal SimanteC

- HIDAYATUL MUSTAFIDAH, YENI KUSTIYAHNINGSIH :**  
 Rancang Bangun E-Document Di Kantor Pelayanan  
 Perijinan Terpadu Kabupaten Bangkalan ..... 114-121
- DYAH SULISTYOWATI RAHAYU, CHASTINE FATICHAH,  
 RULLY SULAIMAN :**  
 Peningkatan Efisiensi Waktu Komputasi  
 dengan Metode Pemrograman Dinamis..... 122-131
- RIO BAYU AFRIANTO, HANDAYANI TJANDRASA,  
 ISYE ARIESHANTI :**  
 Prediksi Pergerakan Harga Saham menggunakan  
 metode Back Propagation Neural Network ..... 132-141
- AERI RACHMAD :**  
 Ekstraksi Fitur Menggunakan Metode LDA dan  
 Pemilihan Eigen Value pada Cacat Kertas Duplek..... 142-149
- ACHMAD JAUHARI , ERMA SURYANI :**  
 Pemodelan Sistem Dinamik untuk Mengurangi Susut  
 Teknis (Studi Kasus : PLN Rayon Pamekasan) ..... 150-157
- SIGIT SUSANTO PUTRO, ERMA SURYANI :**  
 Pemodelan Sistem Dinamik Untuk Efisiensi Anggaran  
 Administrasi Akademik Sesuai Standar  
 Pelayanan Minimum (Studi Kasus: Fakultas Teknik  
 Universitas XYZ) ..... 158 - 169
- WAHYUDI SETIAWAN, FITRI DAMAYANTI :**  
 Sistem Deteksi Retinopati Diabetika menggunakan  
 Support Vector Machine ..... 170 - 178



# Jurnal SimanteC

Vol. 3 No. 3 Desember 2013

ISSN 2088-2130

<b>DEWIYATUL MUSTAFIDAH, YENI KUSTIYAHNINGSIH</b> Rancang Bangun E-Document Di Kantor Pelayanan Perijinan Terpadu Kabupaten Bangkalan .....	114-121
<b>DEWI SULISTYOWATI RAHAYU, CHASTINE FATICHAH, RULLY SULAIMAN</b> Peningkatan Efisiensi Waktu Komputasi dengan Metode Pemrograman Dinamis .....	122-131
<b>DEWI BAYU AFRIANTO, HANDAYANI TJANDRASA, ISYE ARIESHANTI</b> Analisis Pergerakan Harga Saham menggunakan metode Back Propagation Neural Network .....	132-141
<b>DEWI RACHMAD</b> Integrasi Fitur Menggunakan Metode Lda Dan Pemilihan Eigen Value pada Cacat Kertas Duplek .....	142-149
<b>DEWI JAUHARI, ERMA SURYANI</b> Pemodelan Sistem Dinamik Untuk Mengurangi Susut Teknis (Studi Kasus : PLN Rayon Pamekasan) .....	150-157
<b>DEWI SUSANTO PUTRO, ERMA SURYANI</b> Pemodelan Sistem Dinamik Untuk Efisiensi Anggaran Administrasi Akademik Sesuai Standar Pelayanan Minimum (SPM) (Studi Kasus: Fakultas Teknik Universitas XYZ) .....	158-169
<b>DEWI YUDI SETIAWAN, FITRI DAMAYANTI</b> Sistem Deteksi Retinopati Diabetika menggunakan Support Vector Machine .....	170-177

## SISTEM DETEKSIRETINOPATI DIABETIKA MENGUNAKAN SUPPORT VECTOR MACHINE

Wahyudi Setiawan<sup>1)</sup>, Fitri Damayanti

Manajemen Informatika, Universitas Trunojoyo  
Jl. Raya Telang PO. BOX 2, Kamal, Bangkalan, Madura

<sup>1)</sup>email : wsetiawan.ok@gmail.com

### Abstrak

Retinopati Diabetika adalah efek samping dari Diabetes Mellitus. Hal ini dapat menyebabkan kebutaan jika tidak ditangani sedini mungkin. Sistem yang dibuat dalam penelitian ini adalah deteksi tingkat retinopati diabetik dari gambar yang diperoleh dari foto fundus. Ada tiga langkah utama untuk menyelesaikan masalah, preprocessing, ekstraksi fitur dan klasifikasi. Metode Preprocessing yang digunakan dalam sistem ini adalah Grayscale channel Hijau, Gaussian Filter, Contrast Limited Adaptive Histogram Equalization dan Masking. Two Dimensional Linear Discriminant Analysis (2DLDA) digunakan untuk ekstraksi fitur. Support Vector Machine (SVM) digunakan untuk klasifikasi. Hasil pengujian dilakukan dengan mengambil dataset dari Messidor dengan jumlah Images yang bervariasi untuk tahap pelatihan, citra yang lain digunakan untuk tahap pengujian. Hasil pengujian menunjukkan akurasi yang optimal adalah 84%.

Kata kunci: Retinopati Diabetik, Support Vector Machine, Two Dimensional Linear Discriminant Analysis (2DLDA), Messidor

### Abstract

Diabetic Retinopathy is side effect of Diabetes Mellitus. It can be a blindness if untreated settled as early as possible. System created in this study is the detection of diabetic retinopathy level of the image obtained from fundus photographs. There are three main steps to resolve the problems, preprocessing, feature extraction and classification. Preprocessing methods that used in this system are Grayscale Green Channel, Gaussian Filter, Contrast Limited Adaptive Histogram Equalization and Masking. Two Dimensional Linear Discriminant Analysis (2DLDA) is used for feature extraction. Support Vector Machine (SVM) is used for classification. The test result performed by taking a dataset of MESSIDOR with number of images that vary for the training phase, otherwise is used for the testing phase. Test result show the optimal accuracy are 84% .

*Keywords* : Diabetic Retinopathy, Support Vector Machine, Two Dimensional Linear Discriminant Analysis, MESSIDOR

**PENDAHULUAN**

Penelitian dan pengembangan aplikasi dengan berbagai metode dalam pencitraan medis telah berkembang sangat luas. Salah satu penelitian dalam pencitraan medis adalah klasifikasi citra retina untuk deteksi penyakit. Pada citra retina dapat dianalisa untuk mendapatkan informasi penting misalnya informasi tentang tingkat resiko penyakit retinopati diabetik.

Retinopati diabetika tidak bisa dideteksi langsung secara kasat mata karena tanda-tandanya berada di bagian syaraf retina. Tanda-tanda penyakit ini hanya dapat dilihat menggunakan foto fundus tetapi memerlukan waktu yang relatif lama untuk mengetahui hasilnya. Permasalahan tersebut diselesaikan dengan membangun sebuah sistem yang dapat mendeteksi tingkat resiko retinopati diabetik dengan waktu yang relatif cepat.

Sistem deteksi yang dibangun memerlukan sebuah model komputasi untuk mengubah piksel citra retina menjadi suatu ciri retina yang terindikasi retinopati diabetik. Tiga permasalahan utama pada sistem, yaitu prapengolahan, ekstraksi ciri dan teknik klasifikasi.

Prapengolahan berfungsi mempersiapkan citra agar dapat menghasilkan ciri yang lebih baik pada tahap berikutnya. Pada tahap ini sinyal informasi ditonjolkan dan sinyal pengganggu (derau) diminimalisasi [1].

Ekstraksi ciri adalah tahapan untuk memunculkan ciri dan mereduksi dimensi citra dari dimensi tinggi ke dimensi lebih rendah. Teknik ekstraksi ciri yang handal merupakan kunci utama dalam penyelesaian masalah pengenalan pola [2].

Proses klasifikasi sama pentingnya dengan proses ekstraksi ciri. Setelah ciri-ciri penting data citra retina yang dihasilkan pada proses ekstraksi ciri, ciri-ciri tersebut nantinya akan digunakan untuk proses klasifikasi.

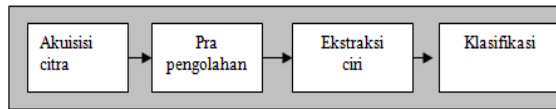
Penelitian ini mengintegrasikan proses prapengolahan, ekstraksi ciri dan klasifikasi. Proses prapengolahan menggunakan metode *Grayscale Green Channel*, *Filter Gaussian*, *Contrast Limited Adaptive Histogram Equalization (CLAHE)*, dan *Masking*. Proses ekstraksi ciri menggunakan metode *Two Dimensional Linear Discriminant Analysis (2DLDA)*, sedangkan proses klasifikasi menggunakan *Support Vector Machine (SVM)*.

**KERANGKA TEORI**

**Pengenalan Pola**

Pengenalan pola adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif citra atau sifat dari obyek. Pola sendiri merupakan suatu entitas yang terdefinisi, dapat diidentifikasi dan diberi nama. Pola

dapat berupa kumpulan hasil pengukuran atau pemantauan dan dapat dinyatakan dalam notasi vektor atau matriks [1]. Struktur sistem pengenalan pola terdapat pada Gambar 1.



Gambar 1. Struktur sistem pengenalan pola

**Akuisisi citra**

Akuisisi citra merupakan cara untuk mendapatkan citra yang akan digunakan dalam proses pengolahan citra. Dalam penelitian ini citra retina yang akan dilatih dan diuji berasal dari foto kamera fundus.

**Pra-pengolahan**

Pra-pengolahan bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dalam penelitian ini, proses pra pengolahan terdiri dari *grayscale*, filter *gaussian*, ekualisasi histogram dan *masking*.

**Grayscale**

Citra retina yang diterima adalah citra berwarna, sehingga terlebih dahulu perlu dilakukan proses *grayscale* untuk mendapatkan citra dengan aras keabuan. Jumlah warna pada citra *grey* adalah 256, karena citra *grey* jumlah bitnya adalah 8, sehingga jumlah warnanya adalah  $2^8=256$ , nilainya berada pada jangkauan 0-255. Untuk mendapatkan citra keabuan digunakan persamaan 1.

$$I(x, y) = \alpha.R + \beta.G + \gamma.B \dots\dots\dots(1)$$

**Filter Gaussian**

Filter *Gaussian* adalah salah satu filter linier dengan nilai pembobo untuk setiap anggotanya dipilih berdasarkan bentuk fungsi *Gaussian*. Filter ini sangat baik untuk menghilangkan derau (*noise*) yang bersifat sebaran normal. Secara alami derau juga memiliki sebaran *Gaussian*, sehingga secara teoritis akan menjadi netral jika dilawan dengan fungsi lain yang juga memiliki fungsi *Gaussian*, hal ini disebut sebagai *zero mean*. *Zero mean* dari fungsi *Gaussian* dengan nilai pembobotan 2 dimensi ditunjukkan pada persamaan 2 [3].

$$\frac{g(x,y)}{k} = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \dots\dots\dots(2)$$

dengan k = konstanta normalisasi dan  $\sigma$  menyatakan standar deviasi dari distribusi. Fungsi diatas diasumsikan memiliki *zero mean* (pusat distribusi pada garis  $x=0$ ). Semakin besar nilai  $\sigma$  maka kurva distribusi *Gaussian* semakin melebar dan puncaknya

menurun. Bentuk 2-D dari fungsi gaussian ditunjukkan pada persamaan 3.

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \dots \dots \dots (3)$$

**Contrast-Limited Adaptive Histogram Equalization (CLAHE)**

CLAHE dapat digunakan sebagai alternatif pengganti ekualisasi histogram. Ekualisasi histogram bekerja pada seluruh citra, sedangkan CLAHE beroperasi pada daerah kecil di citra yang disebut *tile*. Setiap *tile* ditingkatkan nilai kontrasnya, sehingga histogram dari wilayah sekitar cocok untuk histogram tertentu. Setelah melakukan pemerataan, CLAHE menggabungkan *tile* tetangga menggunakan interpolasi bilinear untuk menghilangkan batas-batas artifisial. CLAHE juga dapat digunakan untuk menghindari *noise* yang ada pada citra dengan membatasi kontras pada daerah homogen. CLAHE menghasilkan output citra yang memiliki nilai merata di seluruh bagian citra [4].

**Masking**

Citra hasil segmentasi diberikan masking agar nantinya background hitam pada citra retina, tidak dihitung sebagai obyek. Operator AND digunakan untuk mendapatkan citra hasil operasi dari dua citra yang berbeda.

**Two-Dimensional Linear Discriminant Analysis (2DLDA)**

2DLDA adalah pengembangan dari metode LDA. Didalam LDA pada pengenalan citra dengan matrik 2D terlebih dahulu ditransformasikan kedalam bentuk citra vektor satu dimensi. Sedangkan pada 2DLDA atau disebut teknik proyeksi citra secara langsung, matriks citra 2D tidak perlu ditransformasikan kedalam bentuk citra vektor namun secara langsung matriks scatter citranya dapat dibentuk langsung dengan menggunakan matriks citra aslinya.

2DLDA adalah pengembangan dari metode LDA. Didalam LDA pada pengenalan citra dengan matrik 2D terlebih dahulu ditransformasikan kedalam bentuk citra vektor satu dimensi. Sedangkan pada 2DLDA atau disebut teknik proyeksi citra secara langsung, matriks citra 2D tidak perlu ditransformasikan kedalam bentuk citra vektor namun secara langsung matriks scatter citranya dapat dibentuk langsung dengan menggunakan matriks citra aslinya.

$\{A_1, \dots, A_n\}$  adalah  $n$  matriks citra, dimana  $A_i$  ( $i=1, \dots, k$ ) adalah  $r \times c$  matriks.  $M_i$  ( $i=1, \dots, k$ ) adalah rata-rata citra pelatihan dari kelas ke  $i$  dan  $M$  adalah rata-rata citra dari semua data pelatihan.  $\ell_1 \times \ell_2$  adalah ruang dimensi (*dimensional space*)  $L \otimes R$ ,

dimana  $\otimes$  menunjukkan *tensor product*,  $L$  menjangkau  $\{u_1, \dots, u_{\ell_1}\}$  dan  $R$  menjangkau  $\{v_1, \dots, v_{\ell_2}\}$ , sehingga didefinisikan dua matriks  $L = [u_1, \dots, u_{\ell_1}]$  dan  $R = [v_1, \dots, v_{\ell_2}]$  [5].

Metode ekstraksi ciri adalah untuk menemukan  $L$  dan  $R$  sehingga ruang citra asli (*original image space*)  $A_i$  dirubah kedalam ruang citra dimensi rendah (*low-dimensional image*) menjadi  $B_i = L^T A_i R$ . Ruang dimensi rendah (*low-dimensional space*) diperoleh dengan transformasi linier  $L$  dan  $R$ , jarak *between-class*  $D_b$  dan jarak *within-class*  $D_w$  didefinisikan sebagai berikut :

$$D_b = \sum_{i=1}^k n_i \|L^T (M_i - M) R\|_F^2 \dots \dots \dots (4)$$

$$D_w = \sum_{i=1}^k \sum_{x \in \Pi_i} \|L^T (X - M_i) R\|_F^2 \dots \dots \dots (5)$$

dimana  $\| \cdot \|_F$  merupakan *Frobenius norm*.

Meninjau bahwa  $\|A\|_F^2 = \text{Ptrace}(A^T A) = \text{trace}(A A^T)$  untuk matriks  $A$ . Sedemikian sehingga persamaan (4) dan (5) dapat direpresentasikan lebih lanjut.

$$D_b = \text{trace} \left( \sum_{i=1}^k n_i L^T (M_i - M) R R^T (M_i - M)^T L \right) \dots \dots \dots (6)$$

$$D_w = \text{trace} \left( \sum_{i=1}^k \sum_{x \in \Pi_i} L^T (X - M_i) R R^T (X - M_i)^T L \right) \dots \dots \dots (7)$$

Sama halnya dengan LDA, metode 2DLDA adalah untuk menemukan matriks  $L$  dan  $R$ , sedemikian hingga struktur kelas dari ruang orisinal tetap didalam ruang proyeksi, sehingga patokan (*criterion*) dapat didefinisikan sebagai:

$$J(L, R) = \max \frac{D_b}{D_w} \dots \dots \dots (8)$$

Hal tersebut jelas bahwa persamaan (8) terdiri dari matriks transformasi  $L$  dan  $R$ . Matriks transformasi optimal  $L$  dan  $R$  dapat diperoleh dengan memaksimalkan  $D_b$  dan meminimumkan  $D_w$ . Bagaimanapun, sangat sulit untuk menghitung  $L$  dan  $R$  yang optimal secara simultan. Dua fungsi optimasi dapat didefinisikan untuk memperoleh  $L$  dan  $R$ . Untuk sebuah  $R$  yang pasti,  $L$  dapat diperoleh dengan menyelesaikan fungsi optimasi sebagai berikut :

$$J_2(L) = \max_{\text{trace}}((L^T S_W^R L)^{-1} (L^T S_b^R L)) \dots \dots \dots (9)$$

dimana

$$S_b^R = \sum_{i=1}^k n_i (M_i - M) R R^T (M_i - M)^T \quad (10)$$

$$S_W^R = \sum_{i=1}^k \sum_{x \in \Pi_i} (X - M_i) R R^T (X - M_i)^T \quad (11)$$

Dengan catatan bahwa ukuran matriks  $S_W^R$  dan  $S_b^R$  adalah  $r \times r$  yang lebih kecil daripada ukuran matriks  $S_w$  dan  $S_b$  pada LDA klasik.

Untuk sebuah  $L$  yang pasti,  $R$  dapat diperoleh dengan menyelesaikan fungsi optimasi sebagai berikut :

$$J_3(R) = \max_{\text{trace}}((R^T S_W^L R)^{-1} (R^T S_b^L R)) \quad (12)$$

dimana

$$S_b^L = \sum_{i=1}^k n_i (M_i - M)^T L L^T (M_i - M) \quad (13)$$

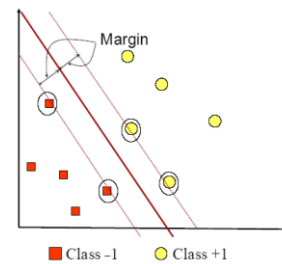
dan

$$S_W^L = \sum_{i=1}^k \sum_{x \in \Pi_i} (X - M_i)^T L L^T (X - M_i) \quad (14)$$

**Support Vector Machine (SVM)**

Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah class pada *input space*. *Hyperplane* dalam ruang vector berdimensi  $d$  adalah *affine subspace* berdimensi  $d-1$  yang membagi ruang vector tersebut ke dalam dua bagian, yang masing-masing berkorespondensi pada class yang berbeda.

*Hyperplane* pemisah terbaik antara kedua kelas dapat ditemukan dengan mengukur *margin hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada Gambar 2.2 menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua kelas, 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.



Gambar 2 SVM berusaha menemukan hyperplane terbaik yang memisahkan kedua class -1 dan +1

Data yang tersedia dinotasikan sebagai  $\vec{x}_i \in \mathfrak{R}^d$ , sedangkan label masing-masing dinotasikan  $y_i = \{+1, -1\}$  untuk  $i=1, 2, 3 \dots l$ . Yang mana  $l$  adalah banyaknya data. Diasumsikan kedua class -1 dan +1 dapat terpisah secara sempurna oleh hyperplane berdimensi  $d$ . yang didefinisikan

$$\vec{w} \cdot \vec{x} + b = 0 \quad (15)$$

Pattern  $\vec{w}$  yang termasuk class -1 (sampel negatif) dapat dirumuskan sebagai *pattern* yang memenuhi pertidaksamaan

$$\vec{w} \cdot \vec{x} + b \leq -1 \quad (16)$$

Sedangkan pattern  $\vec{w}$  yang termasuk class +1 (sampel positif)

$$\vec{w} \cdot \vec{x} + b \geq +1 \quad (17)$$

Margin terbesar dapat ditemukan dengan memaksimalkan nilai jarak antara *hyperplane* dan titik terdekatnya, yaitu  $\frac{1}{\|\vec{w}\|}$ . Hal ini dapat dirumuskan sebagai *Quadratic Programming* (QP) problem, yaitu mencari titik minimal persamaan (16), dengan memperhatikan constraint persamaan (17).

$$\min_{\vec{w}} \tau(w) = \frac{1}{2} \|\vec{w}\|^2 \quad (18)$$

$$y_i (\vec{x}_i \cdot \vec{w} + b) - 1 \geq 0, \quad \forall i \quad (19)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya *Lagrange Multiplier*.

$$L(\vec{w}, b, \alpha) = \frac{1}{2} \|\vec{w}\|^2 - \sum_{i=1}^l \alpha_i (y_i (\vec{x}_i \cdot \vec{w} + b) - 1) \quad (20)$$

dengan  $i = 1, 2, \dots, l$ .

$\alpha_i$  adalah *Lagrange multipliers*, yang bernilai nol atau positif ( $\alpha_i \geq 0$ ). Nilai optimal dari persamaan (20) dapat dihitung dengan meminimalkan  $L$  terhadap  $\vec{w}$  dan  $b$ , dan memaksimalkan  $L$  terhadap  $\alpha_i$ . Dengan

memperhatikan sifat bahwa pada titik optimal gradient  $L = 0$ , persamaan (20) dapat dimodifikasi sebagai maksimisasi problem yang hanya mengandung saja  $\alpha_i$ , sebagaimana persamaan (21).

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j \quad (21)$$

dimana  $\alpha_i \geq 0 (i = 1, 2, \dots, l) \quad \sum_{i=1}^l \alpha_i y_i = 0 \quad (22)$

Dari hasil dari perhitungan ini diperoleh  $\alpha_i$  yang kebanyakan bernilai positif. Data yang berkorelasi dengan  $\alpha_i$  yang positif inilah yang disebut sebagai *support vector* [6].

**METODOLOGI**

**Desain Sistem**

Sistem klasifikasi retina meliputi tahap pelatihan dan pengujian. Tahap pelatihan dimulai dengan menginputkan citra retina, selanjutnya pada citra akan dilakukan proses pra-pengolahan. Citra diubah terlebih dahulu kedalam format *grayscale green channel*, selanjutnya dilakukan operasi filter *Gaussian* untuk menghilangkan *noise*. Proses selanjutnya dilakukan ekualisasi histogram untuk mengubah sebaran tingkat keabuan citra, menggunakan *Contrast Limited Adaptive Histogram Equalization (CLAHE)*. Selanjutnya citra akan dilakukan operasi *masking* untuk memisahkan obyek dengan *background*.

Ekstraksi ciri pada proses pelatihan dilakukan dengan menggunakan metode 2DLDA. Tahap ini bertujuan untuk mendapatkan ciri-ciri yang terpilih dari masukkan data-data pelatihan. Ciri-ciri yang terpilih nantinya digunakan untuk proses klasifikasi pelatihan dan digunakan untuk ekstraksi ciri data pengujian.

Ekstraksi ciri pada proses pengujian dilakukan dengan mengambil hasil ekstraksi ciri pada proses pelatihan diterapkan pada data pengujian. Hasil ekstraksi ciri pada data pengujian ini nantinya digunakan sebagai inputan pada proses klasifikasi pengujian.

Proses klasifikasi pelatihan dilakukan setelah data-data pelatihan diambil ciri-ciri khusus, ciri-ciri khusus ini berupa vektor ciri yang dimensinya lebih kecil. Dalam penelitian ini menggunakan SVM *multiclass One Against All* dengan kernel *gaussian*. Pada proses klasifikasi pelatihan variabel *hyperplane* untuk setiap pengklasifikasi (*classifier*) 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 input, menggunakan *quadratic programming*.

Pada proses klasifikasi pengujian menggunakan hasil ekstraksi ciri 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 retina yang sesuai dengan nilai indeks dari fungsi keputusan yang terbesar hasil dari proses klasifikasi pengujian.

Pada Gambar 3.1 merupakan tahapan proses sistem deteksi retinopati diabetik. Pada proses pelatihan terdapat metode 2DLDA yang digunakan untuk mengekstraksi ciri, ciri-ciri yang terpilih pada saat proses pelatihan digunakan dalam proses klasifikasi dan juga digunakan untuk mengekstraksi ciri pada data uji coba. Masing-masing dataset citra retina yang digunakan dibagi menjadi dua, sebagian digunakan untuk proses pelatihan (*training*) dan sisanya digunakan untuk proses pengujian (*testing*).

**Desain Algoritma**

**Desain Algoritma 2DLDA**

Desain algoritma 2DLDA dibagi menjadi dua subsistem yaitu subsistem pelatihan dan subsistem pengujian. Berikut ini adalah penjabaran masing-masing subsistem.

**Proses Pelatihan 2DLDA**

Untuk proses pelatihan 2DLDA dibagi menjadi tiga tahapan, yaitu : tahap pertama menghitung nilai rata-rata kelas dan rata-rata global, tahap kedua menghitung matriks *within class scatter* dan matriks *between class scatter*, dan tahap terakhir menghitung matriks ciri ekstraksi data-data pelatihan.

**Algoritma 2DLDA**

Inputannya adalah matriks data pelatihan, pada matriks data pelatihan tidak ditransformasikan kedalam vektor tetapi tetap berupa matriks. Inputan lainnya adalah jumlah kelas (*k*), jumlah data perkelas (*n<sub>i</sub>*), dan *n* (banyaknya data pelatihan).

1. Jika dalam suatu dataset citra retina terdapat himpunan sebanyak *n* citra pelatihan  $A_i = [A_1, A_2, \dots, A_n]$  ( $i = 1, 2, \dots, n$ ) dengan dimensi citra ( $r \times c$ ), maka himpunan total matriks dari semua citra tersebut adalah :

$$A_n = \begin{bmatrix} A_{(n)11} & A_{(n)12} & \dots & A_{(n)1c} \\ A_{(n)21} & A_{(n)22} & \dots & A_{(n)2c} \\ \dots & \dots & \dots & \dots \\ A_{(n)r1} & A_{(n)r2} & \dots & A_{(n)rc} \end{bmatrix}$$

Matriks ini digunakan sebagai data inputan. Data inputan lainnya adalah jumlah kelas (*k*), jumlah

data perkelas ( $n_i$ ), dan banyaknya data pelatihan ( $n$ ).

2. Tahapan berikutnya adalah menghitung rata-rata citra pelatihan dari kelas ke  $i$ :

$$M_i = \frac{1}{n_i} \sum_{X \in \Pi_i} X.$$

3. Menghitung rata-rata semua citra pelatihan :

$$M = \frac{1}{n} \sum_{i=1}^k \sum_{X \in \Pi_i} X.$$

4. Menentukan nilai  $\ell_1$  (dimensi proyeksi baris) dan  $\ell_2$  (dimensi proyeksi kolom). Nilai  $\ell_1 \leq r$  dan  $\ell_2 \leq c$ .

5. Menetapkan matriks transformasi  $R$  ukuran  $(c, \ell_2)$  yang diperoleh dari gabungan antara matriks identitas ukuran  $(\ell_2, \ell_2)$  dengan matriks nol ukuran  $(c - \ell_2, \ell_2)$ .

6. Menghitung matriks *between class scatter*  $R$  sesuai dengan persamaan

$$S_b^R = \sum_{i=1}^k n_i (M_i - M) R R^T (M_i - M)^T, \text{ ukuran matriksnya } (r \times r).$$

Ukuran matriks  $S_b^R$  lebih kecil dari ukuran matriks  $S_b$  pada LDA klasik (Dimensi x Dimensi).

7. Menghitung matriks *within class scatter*  $R$  sesuai dengan persamaan

$$S_w^R = \sum_{i=1}^k \sum_{X \in \Pi_i} (X - M_i) R R^T (X - M_i)^T,$$

ukuran matriksnya  $(r \times r)$ .

Ukuran matriks  $S_w^R$  lebih kecil dari ukuran matriks  $S_w$  pada LDA klasik (Dimensi x Dimensi).

8. Hitung *generalized eigenvalue* ( $\lambda_i$ ) dari  $S_b^R$  dan  $S_w^R$  sesuai dengan persamaan (17).

$$J_1(L) = \max \text{trace}((L^T S_w^R L)^{-1} (L^T S_b^R L)),$$

ukuran matriksnya  $(r \times r)$ .

9. Ambil sebanyak  $\ell_1$  *eigenvector* terbesar dari langkah 5 sebagai matriks transformasi baris ( $L$ ).  $L$

$$= [\phi_1^L, \dots, \phi_{\ell_1}^L], \text{ ukuran matriksnya } (r \times \ell_1).$$

10. Menghitung matriks *between class scatter*  $L$  sesuai dengan persamaan

$$S_b^L = \sum_{i=1}^k n_i (M_i - M)^T L L^T (M_i - M),$$

ukuran matriksnya  $(c \times c)$ .

Ukuran matriks  $S_b^L$  lebih kecil dari ukuran matriks  $S_b$  pada LDA klasik (Dimensi x Dimensi).

11. Menghitung matriks *within class scatter*  $L$  sesuai dengan persamaan

$$S_w^L = \sum_{i=1}^k \sum_{X \in \Pi_i} (X - M_i)^T L L^T (X - M_i),$$

ukuran matriksnya  $(c \times c)$ .

Ukuran matriks  $S_w^L$  lebih kecil dari ukuran matriks  $S_w$  pada LDA klasik (Dimensi x Dimensi)

12. Hitung *generalized eigenvalue* ( $\lambda_i$ ) dari  $S_b^L$  dan  $S_w^L$  sesuai dengan persamaan (9).

$$J_2(R) = \max \text{trace}((R^T S_w^L R)^{-1} (R^T S_b^L R)),$$

ukuran matriksnya  $(c \times c)$ .

13. Ambil sebanyak  $\ell_2$  *eigenvector* terbesar dari langkah 9 sebagai matriks transformasi kolom ( $R$ ).

$$R = [\phi_1^R, \dots, \phi_{\ell_2}^R], \text{ ukuran matriksnya } (c \times \ell_2).$$

14. Hitung matriks ciri ekstraksi adalah  $B_i = L^T A_i R$ , ukuran matriksnya  $(\ell_1 \times \ell_2)$ .

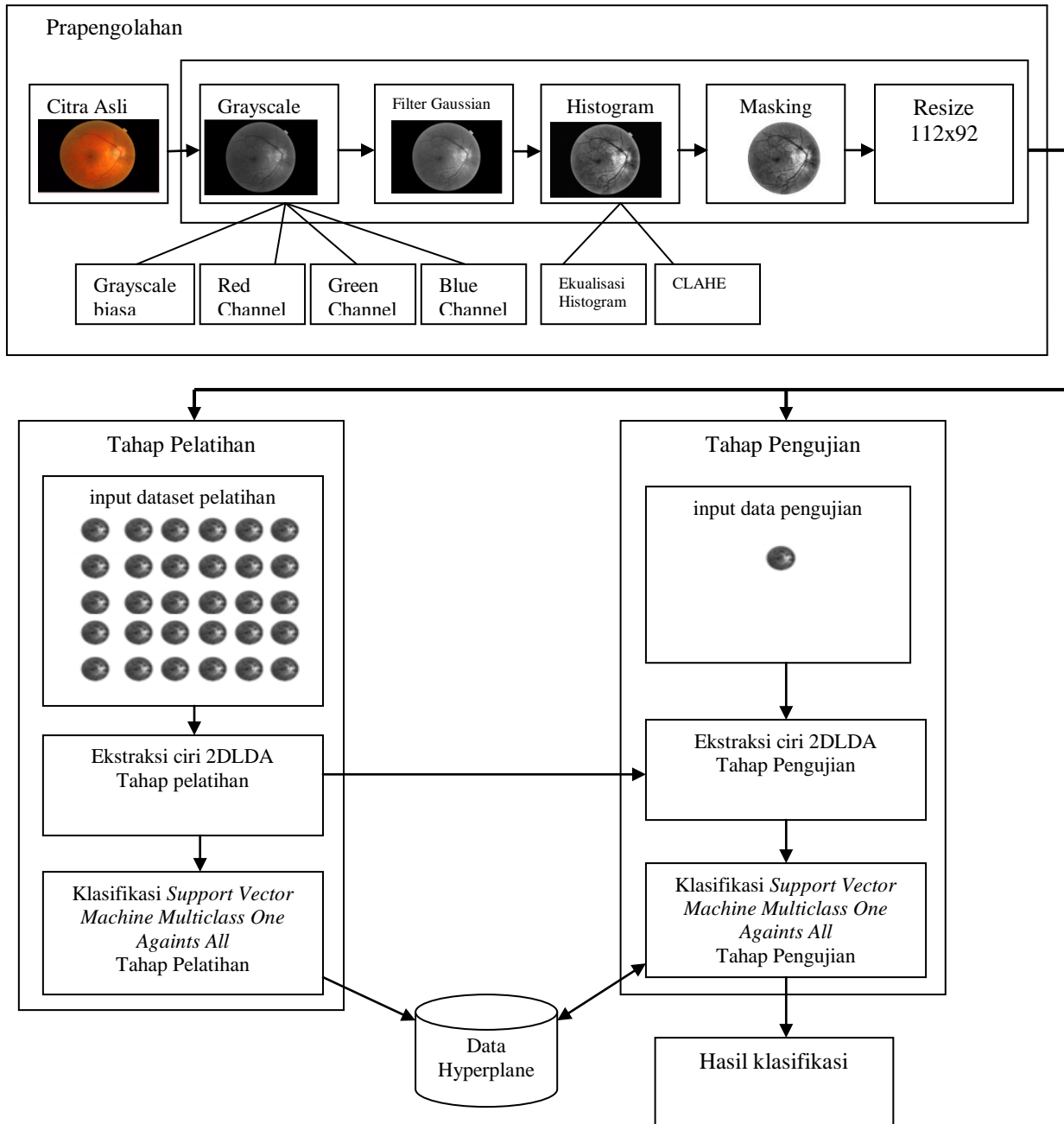
15. Output : matriks ciri ekstraksi  $B_i$ , matriks transformasi baris  $L$ , dan matriks transformasi kolom  $R$ .

### Desain Algoritma SVM

Pengklasifikasi SVM untuk *multiclass One Against All* akan membangun sejumlah  $k$  SVM biner ( $k$  adalah jumlah kelas). Fungsi keputusan yang mempunyai nilai maksimal, menunjukkan bahwa data  $x_d$  merupakan anggota dari kelas fungsi keputusan tersebut.

Pengklasifikasian dengan SVM dibagi menjadi dua proses, yaitu proses pelatihan dan pengujian. Pada proses pelatihan SVM menggunakan matriks ciri yang dihasilkan pada proses ekstraksi ciri pelatihan sebagai input. Sedangkan pada pengujian SVM memanfaatkan matriks ciri yang dihasilkan pada proses ekstraksi ciri pengujian sebagai input [7].





Gambar 3. Tahapan Proses Sistem Deteksi Retinopati Diabetik

**Proses Pelatihan SVM**

Algoritma pelatihan untuk masing-masing pengklasifikasi SVM biner dapat dituliskan sebagai berikut : input berupa matriks  $B$  (matriks hasil ekstraksi ciri pelatihan) dan vektor  $Y$  sebagai pasangan input-target dan outputnya adalah  $w, x, b$  (variable - variabel persamaan *hyperplane*). Langkah – langkahnya dijelaskan sebagai berikut :

1. Tentukan Input ( $Z = B$ ) dan Target ( $Y$ ) sebagai pasangan pelatihan dari dua kelas.

2. Hitung Kernel Gaussian  $K(Z, Z_i) = \exp \left( \frac{-|Z - Z_i|^2}{2\sigma^2} \right)$

3. Hitung Matriks Hessian  $H = K(Z, Z_i) * Y * Y^T$ .

4. Tetapkan  $c$  dan  $epsilon$ .

5. Tetapkan vektor  $e$  sebagai vektor satuan yang memiliki dimensi sama dengan dimensi  $Y$ .

6. Hitung solusi *quadratic programming*:

$$\min L(\alpha) = \frac{1}{2} \alpha^T H \alpha - e^T \alpha,$$

dimana  $y^T \alpha = 0$  dan  $0 \leq \alpha \leq c$ .

Input matriks  $Z$  merupakan matriks ciri yang dihasilkan pada proses ekstraksi ciri dan vektor  $Y$  sebagai target. Contoh untuk Dataset MESSIDOR yang terdiri dari 5 kelas, maka jika digunakan sepuluh sampel tiap kelas dan dimensi proyeksi baris  $\ell_1 = 10$ , dimensi proyeksi kolom  $\ell_2 = 10$ , maka matriks  $Z$  yang dihasilkan adalah matriks ciri dengan dimensi  $50 \times 100$  (50 didapat dari jumlah sampel tiap kelas dikalikan dengan banyaknya kelas, 100 didapat dari perkalian dimensi proyeksi baris dengan dimensi proyeksi kolom). Vektor  $Y$  merupakan vektor kolom untuk pengklasifikasi pertama dimana semua citra retina dari kelas pertama akan disimbolkan dengan angka 1, semua citra retina dari kelas lainnya dengan angka -1. Vektor  $Y$  untuk pengklasifikasi kedua semua citra retina dari kelas kedua disimbolkan dengan 1 dan semua citra retina bukan kelas kedua disimbolkan dengan -1, demikian seterusnya untuk pengklasifikasi ketiga sampai ke  $k$ . Pada penelitian ini, digunakan fungsi *kernel gaussian* dengan nilai varian ( $\sigma$ ) = 1.

Langkah selanjutnya adalah menghitung matriks Hessian, yaitu perkalian antara *kernel gaussian* dengan  $Y$ .  $Y$  disini adalah berupa *vector* yang berisi nilai 1 dan -1. Dari contoh di atas jika *classifier* pertama yang dilatih, maka nilai  $Y$  untuk 10 elemen pertama (jika digunakan 10 sampel citra retina per kelas) akan bernilai 1 dan elemen lainnya bernilai -1. Jika *classifier* kedua dilatih, maka elemen 11, sampai dengan 20 bernilai 1, sedangkan sisanya bernilai -1. Matriks Hessian ini nantinya digunakan sebagai variabel input dalam *quadratic programming*.

Fungsi *quadratic programming* monqp memerlukan variabel  $c$  dan *epsilon*. Untuk itu tetapkan nilai  $c$  dan *epsilon* ( $c$  adalah batas atas nilai  $\alpha_i$ ) dari C-SVM. Vektor satuan  $e$  juga dibentuk dengan dimensi sama dengan vektor  $Y$ .

Penyelesaian  $\min L(\alpha) = \frac{1}{2} \alpha^T H \alpha - e^T \alpha$  dengan *quadratic programming*, merupakan implementasi dari pencarian solusi atas permasalahan  $\min \frac{1}{2} |w|^2 + C \left( \sum_{i=1}^n \xi_i \right)$ .

Jika diimplementasikan dalam bentuk matriks menjadi  $\min \frac{1}{2} w^T w + C \left( \sum_{i=1}^n \xi_i \right)$ , dengan  $y_i^T (w^T \Phi(x_i) + b) \geq 1 - \xi_i$ .

Jika formula dalam bentuk matriks tersebut diubah ke bentuk dual problem, maka formula tersebut menjadi  $\min L(\alpha) = \frac{1}{2} \alpha^T H \alpha - e^T \alpha$ , dimana  $y^T \alpha = 0$  dan  $0 \leq \alpha \leq C$ .

Disini  $H = y_i y_j K(x_i, x_j)$ , dan  $K(x_i, x_j) = \exp \left( \frac{-|x_i - x_j|^2}{2\sigma^2} \right)$

adalah fungsi kernelnya, dan  $e$  adalah vektor satuan yang dimensi sama dengan  $Y$ , sedangkan  $c > 0$  adalah batas atas dari nilai  $\alpha$ . Dalam penelitian ini digunakan nilai  $c = 1000$  dan  $\epsilon = 1 \times 10^{-7}$ . Hasil dari fungsi monqp (*quadratic programming*) adalah nilai variabel  $w$ ,  $x$ , dan  $b$  yang nantinya akan digunakan untuk proses pengujian. Diagram

alir untuk algoritma pelatihan SVM dapat dilihat pada Gambar 3.6.

### Proses Pengujian SVM

Setelah pada proses pelatihan didapat nilai variabel  $w$ ,  $x$ , dan  $b$  untuk masing - masing kelas. Nilai variabel  $w$ ,  $x$ , dan  $b$  untuk didefinisikan sebagai vektor  $w$ ,  $x$ , dan  $b$ . Untuk input data yang akan diklasifikasikan adalah matriks ciri  $D$  yang dihasilkan pada proses ekstraksi ciri pengujian. Matriks ciri  $D$  tersebut ditransformasikan dulu kedalam bentuk vektor menjadi  $1 \times (\ell_1 \times \ell_2)$  diberi nama  $T$ . Langkah - langkahnya sebagai berikut :

1. Input : vektor  $T$  (data pengujian), vektor  $w$ ,  $x$ ,  $b$ , dan  $k$  (jumlah kelas).
2. Hitung Kernel Gaussian  $K(T, x_i) = \exp \left( \frac{-|T - x_i|^2}{(2\sigma^2)} \right)$ .
3. Hitung  $f_i = K(T, x_i) w_i + b_i$ .
4. Ulangi langkah 2 dan 3 untuk  $i = 2$  sampai  $k$ .
5. Tentukan nilai  $f_i$  yang paling maksimal.
6. Kelas  $i$  dengan  $f_i$  terbesar adalah kelas dari vektor  $T$ .

Nilai  $T$  adalah transformasi matriks ciri  $D$  kedalam bentuk vektor. Langkah selanjutnya adalah dengan menghitung Kernel Gaussian  $K(T, x_i)$ , dengan  $T$  adalah data input dan  $x_i$  adalah support vector yang dihasilkan pada proses pelatihan SVM.

Fungsi keputusan  $f_i = K(T, x_i) w_i + b_i$  dihitung untuk masing-masing nilai  $i$ . dimana  $i = 1$  sampai  $k$  ( $k$  adalah jumlah kelas). Output dari algoritma ini berupa indeks  $i$  dengan  $f_i$  terbesar yang merupakan kelas dari vektor  $T$ . Diagram alir untuk algoritma pengujian dapat dilihat pada Gambar 3.7.

Dari proses pelatihan dan pengujian SVM dapat diringkas dalam bentuk blok diagram proses pelatihan dan pengujian yang ditunjukkan pada Gambar 3.8. Gambar 3.8 menjelaskan secara garis besar proses pelatihan dan pengujian pada SVM.

Data pelatihan yang sudah diproyeksikan oleh 2DLDA, selanjutnya menjadi data pelatihan SVM. Jika sebaran data yang dihasilkan pada proses 2DLDA mempunyai distribusi yang tidak linier, maka salah satu metode yang digunakan SVM untuk mengklasifikasikan data tersebut adalah dengan mentransformasikan data ke dalam dimensi ruang ciri (*feature space*), sehingga dapat dipisahkan secara linier pada *feature space*. Karena *feature space* dalam prakteknya biasanya memiliki dimensi yang lebih tinggi dari vektor input (*input space*). Hal ini mengakibatkan komputasi pada *feature space* mungkin sangat besar, karena ada kemungkinan *feature space* dapat memiliki jumlah *feature* yang tidak terhingga. Maka pada SVM digunakan "kernel trick". Fungsi kernel yang digunakan pada penelitian ini adalah Gaussian

$$K(x,y) = \exp\left(\frac{-|x-y|^2}{2\sigma^2}\right). \quad (23)$$

Sejumlah support vector pada setiap data pelatihan harus dicari untuk mendapatkan solusi bidang pemisah terbaik. Persoalan solusi bidang pemisah terbaik dapat dirumuskan :

$$Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j \quad (24)$$

dimana :  $\alpha_i \geq 0 (i = 1, 2, \dots, l)$   $\sum_{i=1}^l \alpha_i y_i = 0$ .

Data  $\bar{x}_i$  yang berkorelasi dengan  $\alpha_i > 0$  inilah yang disebut sebagai *support vector*. Dengan demikian, dapat diperoleh nilai yang nantinya digunakan untuk menemukan  $w$ . Solusi bidang pemisah didapatkan dengan rumus  $w = \sum \alpha_i y_i x_i$ ;  $b = y_k - w^T x_k$  untuk setiap  $x_k$ , dengan  $\alpha_k \neq 0$ .

Proses pengujian atau klasifikasi dilakukan juga pada setiap SVM biner menggunakan nilai  $w$ ,  $b$ , dan  $x_i$  yang dihasilkan pada proses pelatihan di setiap SVM biner. Fungsi yang dihasilkan untuk proses pengujian adalah

$$f_i = K(x_i, x_d) w_i + b_i \quad (25)$$

dimana :  $i = 1$  sampai  $k$ ;  $x_i = support\ vector$ ;  $x_d = data$  pengujian. Outputnya adalah berupa indeks  $i$  dengan  $f_i$  terbesar yang merupakan kelas dari data pengujian [7].

## HASIL DAN PEMBAHASAN

Tahapan pengujian terhadap sistem dilakukan dengan variasi data pelatihan dan variasi dimensi proyeksi. Hal ini dilakukan untuk mengetahui keakuratan dari sistem yang dibuat. Dataset yang digunakan adalah MESSIDOR, terdiri dari 125 citra retina, dibagi menjadi 5 kelas, masing-masing kelas terdiri dari 25 citra.

Dimensi proyeksi menggunakan nilai  $p$  dari 10 sampai dengan 20, nilai  $q$  dari 10 sampai dengan 20. Hasil pengujian ditunjukkan pada Tabel 4.1

Tabel 1. Tingkat keberhasilan pengenalan dataset MESSIDOR metode 2DLDA – SVM

Jumlah data latih	Jumlah data uji	Akurasi optimal
50	25	72%
75	25	64%
100	25	84%

## KESIMPULAN

Terdapat dua variabel penting yang mempengaruhi tingkat keberhasilan pengenalan, yaitu variasi urutan dari sampel pelatihan per kelas yang digunakan dan jumlah sampel pelatihan per kelas yang digunakan.

Dari hasil uji coba menggunakan metode 2DLDA - SVM didapatkan tingkat akurasi pengenalan optimal 84%.

## DAFTAR PUSTAKA

- [1] Putra, D., Suarjana, I.G., 2010. Segmentasi citra retina digital retinopati diabetes untuk membantu pendeteksian mikroaneurisma. Jurnal teknologi teknik elektro vol 9 no 1.
- [2] Purnomo, M.H., Muntasa, A., 2010. Konsep Pengolahan Citra Digital dan Ekstraksi Ciri. Graha Ilmu, Yogyakarta
- [3] Ahmad, U., 2005. Pengolahan Citra Digital dan Teknik Pemrogramannya, Graha Ilmu, Yogyakarta.
- [4] Zuiderveld, K., 1994. *Contrast Limited Adaptive Histogram Equalization*. Graphic Gems IV. Academic Press Professional, San Diego.
- [5] Liang, Z., Li, Y., Shi, P., 2008. "A note on two-dimensional linear discriminant analysis", Pattern Recognition. 2122-2128.
- [6] Nugroho, A.S., Witarto, A. B., Handoko, D., 2003. *Support Vector Machine* Teori dan aplikasinya dalam Bioinformatika, Kuliah umum Ilmu Komputer.com, website : <http://ilmukomputer.com>, diakses 20 Desember 2011.
- [7] Damayanti, F., Arifin, A.Z., Soelaiman, R., 2010. Pengenalan citra wajah menggunakan *support vector machine*. KURSUS Vol. 5, No. 3, 147-156.