

## **PENGENALAN CITRA WAJAH MENGGUNAKAN METODE *TWO-DIMENSIONAL LINEAR DISCRIMINANT ANALYSIS* DAN *SUPPORT VECTOR MACHINE***

**\*Fitri Damayanti, Agus Zainal Arifin, Rully Soelaiman**

Program Magister Teknik Informatika, ITS

Jl. Raya ITS, Kampus ITS, Sukolilo, Surabaya, 60111

E-Mail: \*fitri2708@yahoo.com

### **Abstrak**

*Linear Discriminant Analysis* telah digunakan secara luas dalam pola linier pengenalan terhadap fitur ekstraksi dan pengurangan dimensi. Hal ini dimaksudkan untuk membuat seperangkat vektor proyeksi yang sangat berbeda untuk dipadatkan sepadat mungkin pada jenis yang sama. *Projection vector* bekerja dalam hitungan jenis  $S_w$  dan antara jenis *Sh matrix scatter*. Umumnya pada aplikasi pengenalan wajah jumlah dimensi data lebih besar dibandingkan jumlah sampelnya, hal ini menyebabkan tunggalnya isi jenis *scatter matrix*  $S_w$ , sehingga fitur wajah tidak diekstraksi dengan baik. Dalam penelitian ini digunakan metode *Two Dimensional Linier Discrimination Analysis (TDLDA)* untuk ekstraksi fitur, yang menilai secara langsung isi jenis *scatter matrix* tanpa pencitraan terhadap transformasi vektor, sehingga mengurangi masalah tunggal dalam isi jenis *scatter matrix*. Penelitian ini akan mengembangkan aplikasi pengenalan wajah yang dintegrasikan dengan metode *TDLDA* dan *SVM* untuk pengenalan wajah. Dengan kombinasi kedua metode tersebut terbukti dapat memberikan hasil yang optimal dengan tingkat akurasi pengenalan antara 84,18% sampai 100% dengan uji coba menggunakan basis data *ORL*, *YALE*, dan *BERN*.

Kata kunci: *Linear Discriminant Analysis, Two Dimensional Linear Discriminant Analysis, Support Vector Machine.*

### **Abstract**

*Linear Discriminant Analysis (LDA)* has been widely used in linear pattern recognition for feature extraction and dimension reduction. It aims to find a set of projection vector that separate the different as far as possible while compressing the same class as compact as possible. It works by calculated the within class  $S_w$  and between class  $S_b$  scatter matrices. In face recognition application, generally the dimension of data larger than the number of samples, this causes the within class scatter matrix  $S_w$  is singular, that can make the face features's not well extracted. *Two Dimensional Linear Discriminant Analysis (TDLDA)* is used on this research for feature extraction, that evalutes directly the within class scatter matrix from the image matrix without image to vector transformation, and hence dilutes the singular problem of within class scatter matrix. This research will develop a face recognition application that combined *Two Dimensional Linear Discriminant Analysis* and *Support Vector Machine*. The combination of two methods give optimal results that have high accuracy of recognition between 84.18% until 100% with the *ORL*, *YALE*, and *BERN* database.

Key words: *Linear Discriminant Analysis, Two Dimensional Linear Discriminant Analysis, Support Vector Machine.*

## **PENDAHULUAN**

Pengenalan wajah dewasa ini telah menjadi salah satu bidang yang banyak diteliti dan juga dikembangkan oleh para 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 lama, meskipun banyak teknik baru telah diajukan, akan tetapi 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.

Ekstraksi fitur adalah proses untuk mendapatkan ciri-ciri pembeda yang membedakan suatu sampel wajah dari sampel wajah yang lain. Bagi sebagian besar aplikasi pengenalan pola, teknik ekstraksi fitur yang handal merupakan kunci utama dalam penyelesaian masalah pengenalan pola. Metode Analisa Komponen Utama (*PCA*) untuk pengenalan wajah dikenalkan oleh Turk dan Pentland pada tahun 1991. Metode tersebut bertujuan untuk memproyeksikan data pada arah yang memiliki variasi terbesar (ditunjukkan oleh *vektor eigen*) yang bersesuaian dengan nilai *eigen* terbesar dari matrik kovarian. Kelemahan dari metode *PCA* adalah kurang optimal dalam pemisahan antar kelas. Pada tahun 1991, Cheng dkk memperkenalkan metode Analisa Diskriminan Linier (*LDA*) untuk pengenalan wajah. Metode ini mencoba menemukan sub ruang linier yang memaksimalkan perpindahan dua kelas pola menurut *Fisher Criterion*  $J_F$ . Hal ini dapat diperoleh dengan meminimalkan jarak matrik sebaran dalam kelas yang sama (*within-class*)  $S_w$  dan memaksimalkan jarak matrik sebaran antar kelas (*between-class*)  $S_b$  secara simultan sehingga menghasilkan *Fisher Criterion*  $J_F$  yang maksimal. *Diskriminan Fisher Linier* akan menemukan sub ruang dimana kelas-kelas saling terpisah linier dengan memaksimalkan *Fisher Criterion*  $J_F$ . Jika dimensi data jauh lebih tinggi daripada jumlah *sample training*, maka  $S_w$  menjadi *singular*. Hal tersebut merupakan kelemahan dari metode *LDA* [1].

Telah banyak metode yang ditawarkan untuk mengatasi kovarian kelas yang sama

(*within class*) yang selalu *singular* karena *small sample size problem*. Pada tahun 1997, Belhumeur memperkenalkan metode *fisherface* untuk pengenalan wajah. Metode ini merupakan penggabungan antara metode *PCA* dan *LDA*. Proses reduksi dimensi dilakukan oleh *PCA* sebelum melakukan proses *LDA*. Hal ini dapat mengatasi *singular problem*. Tetapi kelemahan dari metode ini adalah pada saat proses reduksi dimensi *PCA* akan menyebabkan kehilangan beberapa informasi diskriminan yang berguna dalam proses *LDA* [1]. Metode-metode lainnya yang dapat mengatasi *singular problem* adalah *Direct-LDA*, *Null-space based LDA*, *Pseudo-inverse LDA*, *Two-stage LDA*, dan *Regularized LDA* [2]. Semua teknik *LDA* tersebut memakai model representasi data berdasarkan vektor yang menghasilkan vektor-vektor yang biasanya memiliki dimensi tinggi. Metode *Two Dimensional Linear Discriminant Analysis (TDLDA)* menilai secara langsung matrik *within-class scatter* dari matrik citra tanpa transformasi citra ke vektor, dan hal itu mengatasi *singular problem* dalam matrik *within-class scatter* [3]. *TDLDA* memakai *fisher criterion* untuk menemukan proyeksi diskriminatif yang optimal.

Dalam pengenalan 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 sebuah fungsi atau *hyperplane* untuk memisahkan dua buah kelas pola. *SVM* akan berusaha mencari *hyperplane* yang optimal dimana dua kelas pola dapat dipisahkan dengan maksimal.

Penelitian ini mengintegrasikan *TDLDA* dan *SVM* untuk pengenalan wajah. *TDLDA* sebagai metode ekstraksi fitur yang dapat mengatasi *singular problem* dan *SVM* sebagai metode klasifikasi yang mempunyai kemampuan generalisasi yang tinggi dibanding metode klasifikasi *KNN*.

#### ***Two-Dimensional Linear Discriminant Analysis (TDLDA)***

*TDLDA* adalah pengembangan dari metode *LDA*. Di dalam *LDA*, matrik 2D terlebih dahulu ditransformasikan ke dalam bentuk citra vektor

satu dimensi. Sedangkan pada *TDLDA* atau disebut teknik proyeksi citra secara langsung, matrik citra wajah 2D tidak perlu ditransformasikan ke dalam bentuk citra vektor. *Matrix scatter* citra dapat dibentuk langsung dengan menggunakan matrik citra aslinya.

$\{A_1, \dots, A_n\}$  adalah  $n$  matrik citra, dimana  $A_i$  ( $i=1, \dots, k$ ) adalah  $r \times c$  matrik.  $M_i$  ( $i=1, \dots, k$ ) adalah rata-rata citra pelatihan dari kelas ke  $i$ ,  $M$  adalah rata-rata citra dari semua data pelatihan dan  $X$  adalah matrik masukan. Menganggap  $\ell_1 \times \ell_2$  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 matrik  $L = [u_1, \dots, u_{\ell_1}]$  dan  $R = [v_1, \dots, v_{\ell_2}]$ .

Metode ekstraksi fitur adalah untuk menemukan  $L$  dan  $R$  sehingga ruang citra asli (*original image space*)  $A_i$  diubah ke dalam 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  $R$ , sedangkan jarak *between-class*  $D_b$  dan jarak *within-class*  $D_w$  didefinisikan dalam Persamaan (1) dan (2).

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

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

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

Meninjau bahwa  $\|A\|_F^2 = Ptrace(A^T A) = trace(AA^T)$  untuk matrik  $A$ . Sedemikian hingga Persamaan (1) dan (2) dapat direpresentasikan lebih lanjut sebagai Persamaan (3) dan (4).

$$D_b = trace\left(\sum_{i=1}^k n_i L^T (M_i - M) R R^T (M_i - M)^T L\right) \quad (3)$$

$$D_w = trace\left(\sum_{i=1}^k \sum_{x \in \Pi_i} L^T (X - M_i) R R^T (X - M_i)^T L\right) \quad (4)$$

Sama halnya dengan *LDA*, metode *TDLDA* digunakan untuk menemukan matrik  $L$  dan  $R$ , sedemikian hingga struktur kelas dari ruang orisinil tetap di dalam ruang proyeksi. Sehingga patokan (*criterion*) dapat didefinisikan sebagai Persamaan (5).

$$J_i(L, R) = \max \frac{D_b}{D_w} \quad (5)$$

Hal tersebut jelas bahwa Persamaan (9) terdiri dari matrik transformasi  $L$  dan  $R$ . Matrik transformasi optimal  $L$  dan  $R$  dapat diperoleh dengan memaksimalkan  $D_b$  dan meminimumkan  $D_w$ . Namun, sangatlah 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 pada Persamaan (6).

$$J_2(L) = \max trace((L^T S_W^R L)^{-1} (L^T S_b^R L)) \quad (6)$$

dimana

$$S_b^R = \sum_{i=1}^k n_i (M_i - M) R R^T (M_i - M)^T \quad (7)$$

$$S_w^R = \sum_{i=1}^k \sum_{x \in \Pi_i} (X - M_i) R R^T (X - M_i)^T \quad (8)$$

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

Untuk sebuah  $L$  yang pasti,  $R$  dapat diperoleh dengan menyelesaikan fungsi optimasi pada Persamaan (9).

$$J_3(R) = \max trace((R^T S_w^L R)^{-1} (R^T S_b^L R)) \quad (9)$$

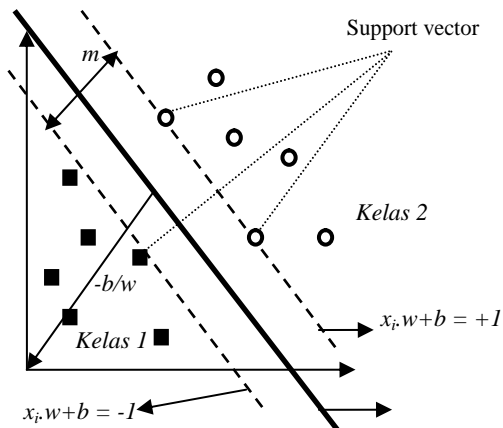
Dimana

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

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

Ukuran matrik  $S_w^L$  dan  $S_b^L$  adalah  $c \times c$  yang lebih kecil daripada ukuran matrik  $S_w$  dan  $S_b$  pada *LDA* klasik.

Secara khusus, untuk sebuah  $R$  yang pasti,  $L$  yang optimal dapat diperoleh dengan menyelesaikan *generalized eigenvalue problem* dari Persamaan (6). Demikian pula,  $R$  dapat diperoleh dengan menyelesaikan *generalized eigenvalue problem* dari Persamaan (9) pada  $L$  yang pasti.

Gambar 1. *Hard Margin Hyperplane*.

### Support Vector Machine (SVM)

SVM berusaha menemukan *hyperplane* yang terbaik pada *input space*. Prinsip dasar SVM adalah *linear classifier* yang selanjutnya dikembangkan agar dapat bekerja pada problem *non-linear* dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi [4].

SVM dapat melakukan klasifikasi data yang terpisah secara linier (*linearly separable*) dan *non-linier* (*nonlinear separable*) [5]. *Linearly separable* data merupakan data yang dapat dipisahkan secara linier. Misalkan  $\{x_1, \dots, x_n\}$  adalah *dataset* dan  $x_i \in \mathcal{R}^d$ , serta  $y_i \in \{+1, -1\}$  adalah label kelas dari data  $x_i$ . Anggap ada beberapa *hyperplane* yang memisahkan sampel positif dan negatif, maka  $x$  yang berada pada *hyperplane* akan memenuhi persamaan  $w \cdot x + b = 0$ . Untuk permasalahan data linier, algoritma *support vector* hanya mencari *hyperplane* dengan margin yang terbesar (jarak antara dua kelas pola). *Hard margin hyperplane* ditunjukkan pada Gambar 1. *Hyperplane* terbaik tidak hanya dapat memisahkan data dengan baik tetapi juga yang memiliki margin paling besar. Data yang berada pada bidang pembatas ini disebut *support vector*.

Untuk menyelesaikan permasalahan data *non-linier* dalam SVM adalah dengan cara memetakan data ke ruang dimensi lebih tinggi (ruang fitur atau *feature space*) [5], dimana data pada ruang tersebut dapat dipisahkan secara linier, dengan menggunakan transformasi  $\Phi$  pada Persamaan (12).

$$\Phi: \mathcal{R}^d \mapsto H \quad (12)$$

Dengan demikian algoritma pelatihan tergantung dari data melalui *dot product* dalam  $H$ . Sebagai contoh  $\Phi(x_i) \cdot \Phi(x_j)$ . Jika terdapat fungsi *kernel*  $K$ , sedemikian hingga  $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ , maka algoritma pelatihan hanya memerlukan fungsi *kernel*  $K$ , tanpa harus mengetahui transformasi  $\Phi$  secara pasti.

SVM pertama kali dikembangkan oleh Vapniks untuk klasifikasi biner, namun selanjutnya dikembangkan untuk klasifikasi *multiclass* (banyak kelas). Pendekatannya adalah dengan membangun *multiclass classifier*, yaitu dengan cara menggabungkan beberapa SVM biner. Pendekatan ini terdiri dari metode satu lawan semua (*One Against All*) dan metode satu lawan satu (*One Against One*) [6].

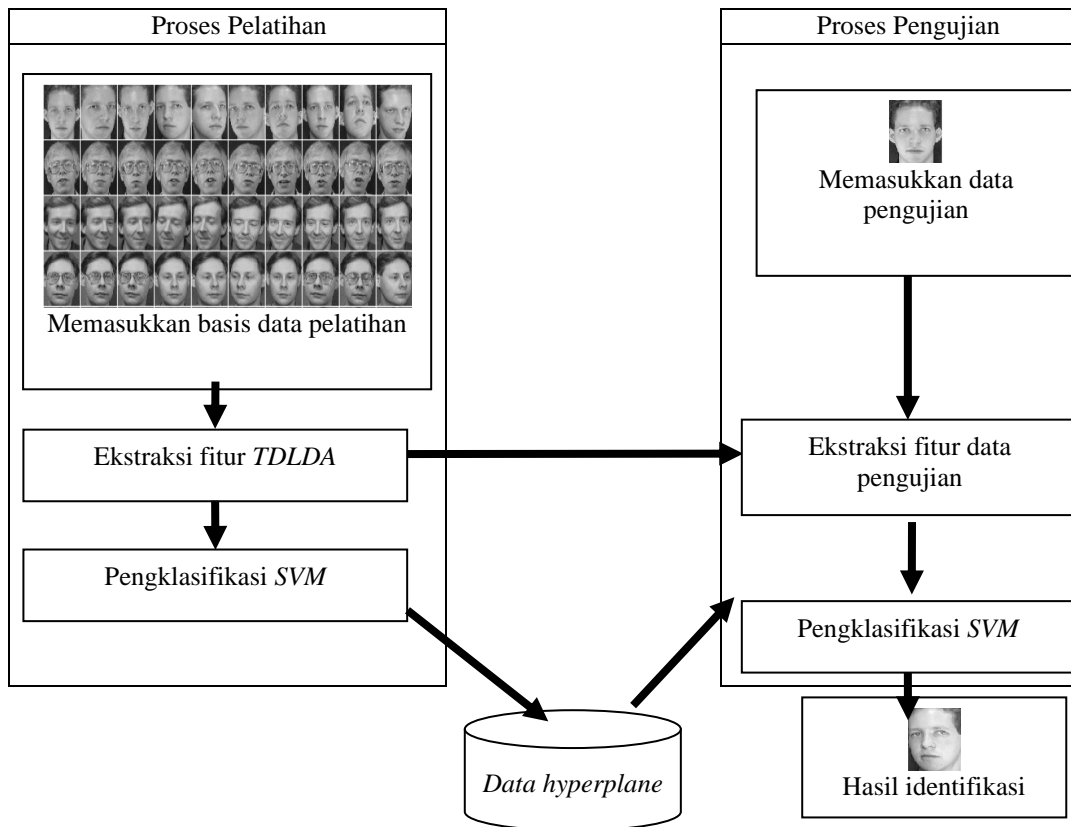
## PERANCANGAN SISTEM

Secara garis besar sistem terdiri dari dua bagian, yaitu proses pelatihan citra dan proses pengujian. Gambar 2 merupakan gambaran garis besar sistem pengenalan wajah. Pada proses pelatihan terdapat proses *TDLDA* yang digunakan untuk mengekstraksi fitur. Fitur-fitur yang terpilih pada saat proses pelatihan digunakan dalam proses klasifikasi dan juga digunakan untuk mendapatkan fitur-fitur yang terpilih pada data uji coba. Masing-masing basisdata wajah yang digunakan dibagi menjadi dua, yaitu satu bagian digunakan untuk proses pelatihan (*training*) dan sisanya digunakan untuk proses pengujian (*testing*).

### Ekstraksi Fitur

Ekstraksi fitur pada proses pelatihan dilakukan dengan menggunakan metode *TDLDA*. Tahap ini bertujuan untuk mendapatkan fitur-fitur yang terpilih dari masukan data-data pelatihan. Fitur-fitur yang terpilih nantinya digunakan untuk proses klasifikasi pelatihan dan digunakan untuk ekstraksi fitur data pengujian.

Ekstraksi fitur pada proses pengujian dilakukan dengan cara mengambil hasil ekstraksi fitur pada proses pelatihan untuk diterapkan pada data pengujian. Hasil ekstraksi fitur pada data pengujian ini nantinya digunakan sebagai masukan pada proses klasifikasi pengujian.



Gambar 2. Sistem Pengenalan Wajah.

**Klasifikasi**

Proses klasifikasi pelatihan dilakukan setelah data-data pelatihan diambil fitur-fitur khusus. Fitur-fitur khusus ini berupa vektor fitur yang dimensinya lebih kecil. Penelitian ini menggunakan SVM metode satu lawan semua 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 masukan (dalam hal ini digunakan *quadratic programming*).

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.

**Algoritma TDLDA**

Berikut ini adalah langkah-langkah dalam proses TDLDA terhadap suatu basisdata citra pelatihan:

1. Jika dalam suatu basisdata citra wajah terdapat himpunan sebanyak  $n$  citra pelatihan  $A_i = [A_{i1}, A_{i2}, \dots, A_{in}]$  ( $i = 1, 2, \dots, n$ ) dengan dimensi citra ( $r \times c$ ), maka himpunan total matrik dari semua citra tersebut adalah:

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

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

3. Tahapan berikutnya adalah perhitungan rata-rata citra pelatihan dari kelas ke- $I$  dengan menggunakan Persamaan (13).

$$M_i = \frac{1}{n_i} \sum_{X \in \Pi_i} X \quad (13)$$

4. Menghitung rata-rata semua citra pelatihan dengan menggunakan Persamaan (14).

$$M = \frac{1}{n} \sum_{i=1}^k \sum_{X \in \Pi_i} X \quad (14)$$

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

6. Menghitung matrik *between class scatter*  $R$  sesuai dengan Persamaan (7).

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

matriknya  $(r \times r)$ .

7. Menghitung matrik *within class scatter*  $R$  sesuai dengan Persamaan (8).

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

matriknya  $(r \times r)$ .

8. Hitung *generalized eigenvalue*  $(\lambda_i)$  dari  $S_b^R$  dan  $S_w^R$  menggunakan SVD sesuai dengan Persamaan (6).

$$J_4(L) = \max_{\text{trace}}((L^T S_w^R L)^{-1} (L^T S_b^R L)),$$

ukuran matriknya  $(r \times r)$ .

9. Ambil sebanyak  $\ell_1$  *eigenvector* dari langkah 8 sebagai matrik transformasi baris  $(L)$ .  $L = [\phi_1^L, \dots, \phi_{\ell_1}^L]$ , ukuran matriknya  $(r \times \ell_1)$ .

10. Menghitung matrik *between class scatter*  $L$  sesuai dengan Persamaan (10).

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

matriknya  $(c \times c)$ .

11. Menghitung matrik *within class scatter*  $L$  sesuai dengan Persamaan (11).

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

ukuran matriknya  $(c \times c)$ .

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

$$J_5(R) = \max_{\text{trace}}((R^T S_w^L R)^{-1} (R^T S_b^L R)),$$

ukuran matriknya  $(c \times c)$ .

13. Ambil sebanyak  $\ell_2$  *eigenvector* dari langkah 12 sebagai matrik transformasi kolom  $(R)$ .  $R = [\phi_1^R, \dots, \phi_{\ell_2}^R]$ , ukuran matriknya  $(c \times \ell_2)$ .

14. Hitung matrik fitur ekstraksi adalah  $B_i = L^T A_i R$ , ukuran matriknya  $(\ell_1 \times \ell_2)$ .

15. Keluaran: matrik fitur ekstraksi  $B_i$ , matrik transformasi baris  $L$ , dan matrik transformasi kolom  $R$ .

Tabel 1. Hasil Uji Coba Menggunakan TDLDA-KNN.

Basisdata	Prosentase Pengenalan		
	Uji 3	Uji 4	Uji 5
ORL	92,14 %	94,58 %	97,00 %
Yale	91,67 %	97,14 %	98,89 %
Bern	82,65 %	92,26 %	95,71 %

Tabel 2. Hasil Uji Coba Menggunakan TDLDA-SVM.

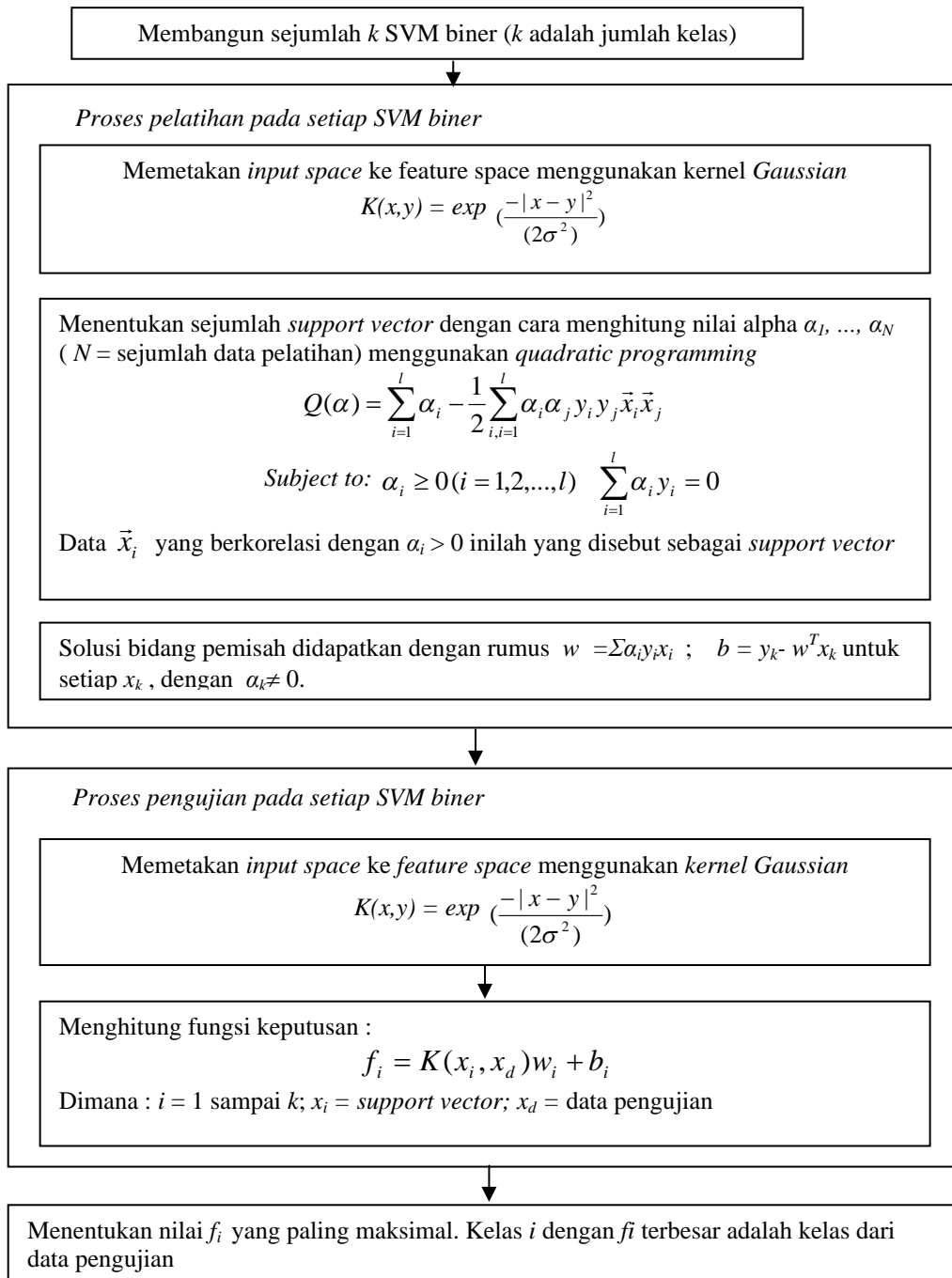
Basisdata	Prosentase Pengenalan		
	Uji 3	Uji 4	Uji 5
ORL	92,86 %	96,67 %	97,50 %
Yale	95,00 %	99,05 %	100 %
Bern	84,18 %	94,05 %	97,14 %

Tabel 3. Perbandingan Hasil Uji Coba dengan Basis Data ORL.

Variasi Pengujian	Prosentase Pengenalan			
	TDLDA-SVM	TDLDA-KNN	2DPCA*	Fisherface**
Uji ORL 3	92,86 %	92,14 %	91,80 %	84,50 %
Uji ORL 4	96,67 %	94,58 %	95,00 %	91,46 %
Uji ORL 5	97,50 %	97,00 %	96,00 %	95,15 %

Ket: \*) diperoleh dari sumber [7]

\*\*\*) diperoleh dari sumber [8].

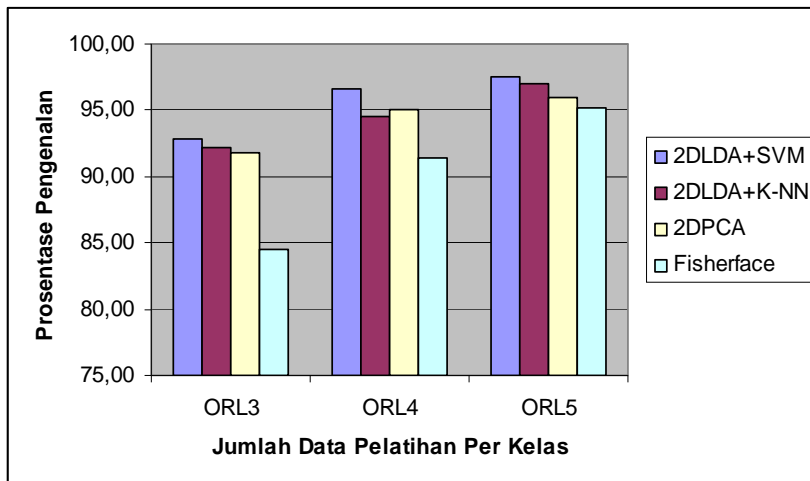


Gambar 3. Blok Diagram Proses Pelatihan dan Klasifikasi Menggunakan SVM.

**Rancangan Algoritma SVM**

Blok diagram proses pelatihan dan pengujian SVM dapat ditunjukkan pada Gambar 3. Pengklasifikasian dengan SVM dibagi menjadi dua proses, yaitu proses pelatihan dan proses pengujian. Pada proses pelatihan SVM menggunakan matrik fitur yang dihasilkan pada proses ekstraksi fitur sebagai masukan. Sedangkan pada pengujian SVM memanfaatkan

matrik proyeksi yang dihasilkan pada proses ekstraksi fitur yang kemudian dikalikan dengan data uji (sampel pengujian) sebagai masukan. Pengklasifikasian 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.



Gambar 4. Grafik Tingkat Keberhasilan Pengenalan untuk Tiap Variasi Pengujian pada Basisdata ORL Menggunakan Metode TDLDA-SVM dan Metode Lainnya.

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

Metode SVM menggunakan "kernel trick". Fungsi *kernel* yang digunakan pada penelitian ini adalah *Gaussian* (Persamaan (15)).

$$K(x, y) = \exp\left(\frac{-|x - y|^2}{(2\sigma^2)}\right) \quad (15)$$

Sejumlah *support vector* pada setiap data pelatihan harus dicari untuk mendapatkan solusi bidang pemisah terbaik. Persoalan solusi bidang pemisah terbaik dapat dirumuskan dalam Persamaan (16).

$$Q(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j \bar{x}_i \bar{x}_j \quad (16)$$

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  $w = \sum a_i y_i x_i$ ;  $b = y_k - w^T x_k$  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 didefinisikan dalam Persamaan (17).

$$f_i = K(x_i, x_d) w_i + b_i \quad (17)$$

Dimana:

$i = 1$  sampai  $k$

$x_i = support vector$

$x_d = data pengujian$ .

Keluarannya adalah berupa indeks  $i$  dengan  $f_i$  terbesar yang merupakan kelas dari data pengujian.

## HASIL DAN PEMBAHASAN

Uji coba terhadap sistem pengenalan wajah pada penelitian ini dilakukan pada tiga jenis basisdata wajah baku, yaitu *Olivetti Research Laboratorium* atau Basis Data ORL, dan *The Yale Face Database* atau Basisdata Yale, dan *The University of Bern* atau Basisdata Bern. Untuk masing-masing basisdata wajah, pelatihan menggunakan tiga wajah (uji 3), empat wajah (uji 4), lima wajah (uji 5). Sisa wajah yang tidak di-*training* digunakan sebagai data pengenalan.



Metode yang digunakan dalam pengujian ini ada dua kelompok. Kelompok pertama menggunakan metode *TDLDA* untuk ekstraksi fitur dan metode *SVM* untuk klasifikasi. Kelompok yang kedua menggunakan metode *TDLDA* sebagai ekstraksi fitur dan metode *K-Nearest Neighbor (KNN)* menggunakan *Euclidean Distance* sebagai klasifikasi. Tabel 1 dan Tabel 2 menunjukkan bahwa prosentase pengenalan *TDLDA-SVM* lebih tinggi dibandingkan dengan *TDLDA-KNN*.

Untuk melihat kelebihan algoritma *TDLDA-SVM*, dibuat perbandingan dengan beberapa metode lain. Perbandingan hasil uji coba antara metode *TDLDA-SVM* dengan *TDLDA-KNN*, *2DPCA*, dan *Fisherface* menggunakan basisdata *ORL* dapat dilihat pada Tabel 3. Tabel 3 menunjukkan bahwa prosentase pengenalan *TDLDA-SVM* lebih tinggi dibanding dengan metode lainnya (*TDLDA-KNN*, *2DPCA*, *Fisherface*).

Untuk mempermudah melihat perbedaan hasil uji coba antara metode *TDLDA-SVM* dengan metode lainnya digunakan diagram batang. Gambar 4 menunjukkan hasil uji coba terhadap basisdata *ORL* untuk metode *TDLDA-SVM* dengan metode lainnya.

Berikut adalah keunggulan metode *TDLDA-SVM* dibanding metode lainnya:

1. *TDLDA-SVM* dibanding dengan *TDLDA-KNN*.

*KNN* tidak memperhatikan distribusi dari data hanya berdasarkan jarak data baru itu ke beberapa data/tetangga terdekat. Boleh jadi, merupakan data/tetangga terdekat bukan kelompoknya, sehingga klasifikasi yang dihasilkan salah.

*SVM* memperhatikan distribusi data sehingga berusaha untuk menemukan fungsi pemisah (*classifier*) yang optimal yang dapat memisahkan dua *set* data dari dua kelas yang berbeda. Setiap kelas memiliki pola yang berbeda dan dipisahkan oleh fungsi pemisah. Sehingga, jika ada data baru yang akan diklasifikasikan maka akan diketahui kelas yang sesuai dengan data baru tersebut. Dengan demikian klasifikasi yang dihasilkan lebih sempurna dibanding dengan metode klasifikasi lainnya.

2. *TDLDA-SVM* dibandingkan dengan *2DPCA*.

*2DPCA* lebih fokus pada pengoptimalan representasi data daripada pengoptimalan diskriminan data, sehingga data-data tidak terpisah dengan sempurna.

*TDLDA* mengoptimalkan diskriminan data dengan lebih baik jika dibandingkan dengan *2DPCA*. Sehingga *TDLDA* dapat mengelompokkan vektor data dari kelas yang sama dan memisahkan kelas yang berbeda.

3. *TDLDA-SVM* dibandingkan dengan *Fisherface*.

Pada *Fisherface* prosedur *pre-processing* untuk mereduksi dimensi menggunakan *PCA* dapat menyebabkan kehilangan beberapa informasi diskriminan yang penting untuk algoritma *LDA* yang diterapkan setelah *PCA*.

*TDLDA* mengambil keuntungan penuh dari informasi yang diskriminatif dari ruang lingkup wajah (*face space*), dan tidak membuang beberapa subruang (*subspace*) yang mungkin berguna untuk pengenalan.

## SIMPULAN

Dari uji coba yang sudah dilakukan dapat diambil simpulan sebagai berikut:

1. Metode *TDLDA-SVM* mampu menunjukkan akurasi pengenalan yang optimal dibandingkan dengan metode lainnya (*TDLDA-KNN*, *2DPCA*, *Fisherface*). Hal ini dikarenakan *TDLDA* mampu mengatasi *singular problem*, mempertahankan keberadaan informasi diskriminatif, serta memaksimalkan jarak antar kelas dan meminimalkan jarak inter kelas. Sedangkan *SVM* mempunyai kemampuan menemukan fungsi pemisah (*classifier*) yang optimal.
2. Terdapat tiga variabel penting yang mempengaruhi tingkat keberhasilan pengenalan, yaitu variasi urutan dari sampel pelatihan per kelas yang digunakan, jumlah sampel pelatihan per kelas yang digunakan, dan jumlah dimensi proyeksi.
3. Dari hasil uji coba menggunakan metode *TDLDA-SVM* dengan memvariasi urutan data pelatihan didapatkan tingkat akurasi pengenalan antara 84,18% sampai 100% untuk basisdata *ORL*, *YALE*, dan *BERN*.

## DAFTAR PUSTAKA

- [1] Belhumeur PN, Hespanha JP and Kriegman DJ. Eigenfaces vs Fisherfaces Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 19: 711-720. 1997.
- [2] Kong H, Wang L, Teoh EK, Wang JG, and Venkateswarlu R. A framework of 2D Fisher Discriminant Analysis: Application to Face Recognition with Small Number of Training Samples. *IEEE Conf. CVPR*. 2005.
- [3] Quan XG, Lei Z and David Z. Face Recognition Using FLDA With Single Training Image Per Person. *Applied Mathematics and Computation*. 205: 726-734. 2008.
- [4] Nugroho AS, Witarto BA dan Handoko D. *Support Vector Machine - Teori dan Aplikasinya Dalam Bioinformatika*. Kuliah Umum Ilmu Komputer.com. 2003. URL: <http://www.ilmukomputer.com/anto-SVM.pdf>, diakses tanggal 16 Maret 2008.
- [5] Burges JC. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*. 2: 955-974. 1998.
- [6] Hsu CW and Lin CJ. A Comparison of Methods for Multi-class Support Vector Machines. *IEEE Transactions on Neural network*. 13: 415-425. 2002.
- [7] Yang J, Zhang D, Frangi AF, and Yang JY. Two-dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 26: 131-137. 2004.
- [8] Liang Z, Li Y, and Shi P. A Note on Two-Dimensional Linear Discriminant Analysis. *Pattern Recogn*. 29: 2122-2128. 2008.