

Implementasi Pengenalan Wajah Berbasis Algoritma *Nearest Feature Midpoint*

Diana Purwitasari, Rully Soelaiman, Mediana Aryuni dan Hanif Rahma Hakim
Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya
E-mail : rully@its-sby.edu

Abstrak

Sistem pengenalan wajah yang baik adalah sistem yang mampu mengatasi variasi yang timbul saat pengambilan citra wajah. Variasi ini bisa berupa ekspresi wajah, aksesoris yang dipakai, tingkat pencahayaan dan arah pengambilan citra. Variasi tersebut akan ditangkap oleh garis-garis maya yang dibuat dari setidaknya dua prototype dalam sebuah kelas. Garis maya tersebut akan mengeneralisasi variasi yang mungkin terjadi dari kedua prototype. Proses identifikasi wajah akan dilakukan dengan mencari jarak terpendek antara wajah yang akan dikenali dengan semua variasi hasil ekstrapolasi dan interpolasi prototype pada tiap kelas. Implementasi dari metode ini bisa mencapai tingkat akurasi lebih dari 90% dengan waktu eksekusi 0.5 detik pada kondisi optimal.

Kata Kunci: pengenalan wajah, *eigenface*, *principal component analysis*, *nearest feature line*, *nearest feature midpoint*

1. Pendahuluan

Pengenalan wajah adalah salah satu bidang kaji dalam pengenalan pola yang selalu mengalami pengembangan. Keandalan sebuah metode bisa dilihat dari proses perhitungan dengan biaya minimal dan hasil perhitungan dengan tingkat kesalahan yang relatif kecil. Sebuah sistem pengenalan wajah yang handal harus tetap bisa bekerja dan mampu menangani masukan citra wajah dengan berbagai variasi terutama dalam sudut pengambilan, ekspresi, pencahayaan dari citra yang dijadikan masukan. Dari ketiga variasi tersebut, variasi wajah yang sama dalam pencahayaan dan sudut pandang pada saat pengambilan citra biasanya jauh lebih besar dari pada ekspresi wajah yang sama.

Nearest Feature Midpoint (NFM) adalah salah satu metode dalam pengenalan wajah yang dinyatakan sebagai metode perbaikan dari pengenalan wajah dengan metode *Nearest Feature Line* (NFL) [1]. Diharapkan dengan menggunakan metode ini bisa dibangun sebuah

aplikasi pengenalan wajah yang relatif lebih baik dibandingkan dengan metode NFL.

Secara umum, klasifikasi menggunakan NFL dan NFM dilakukan dengan mencari jarak minimum antara *feature point* yang di-*query*-kan (wajah yang akan dikenali) dengan semua *feature line* yang ada [1]. *Feature line* adalah garis virtual yang menghubungkan dua *prototype* dalam sebuah kelas (satu orang) sedangkan *feature midpoint* adalah titik tengah antara dua *prototype* dalam sebuah kelas yang sama. Dengan demikian, perbandingan wajah akan diperbanyak dengan melakukan ekstrapolasi atau interpolasi *feature point* pada tiap *feature line* yang ada dalam *feature space* [2].

Untuk melakukan klasifikasi berbasis *feature line* ini, sebagai representasi awal dari citra akan digunakan metode pembentukan *eigenface*. Secara umum dengan membentuk *eigenface space*, dimensi-dimensi yang kurang signifikan dalam citra wajah akan direduksi sehingga hanya menyisakan dimensi yang penting saja. Pembentukan *eigenface space* tidak lepas dari penggunaan *Principal Component Analysis* (PCA) sebagai alat pereduksi dimensi.

2. *Principal Component Analysis* dan *Eigenface*

PCA adalah sebuah metode untuk mengidentifikasi pola dalam sebuah himpunan data untuk kemudian mengekspresikan data-data tersebut sedemikian hingga bisa terlihat perbedaan dan persamaan antara data-data tersebut. Keuntungan dari PCA adalah mengecilkan ukuran data dengan cara mengurangi dimensi dari data tanpa banyak menghilangkan informasi penting dari himpunan data sehingga sering digunakan dalam kompresi citra [3].

Langkah-langkah penggunaan PCA adalah sbb:

(a) Mendapatkan himpunan data

Data yang akan menjadi bahan masukan PCA bisa berupa data numerik apapun yang telah disusun menjadi vektor-vektor data dengan dimensi sejumlah elemen pada sebuah vektor. Pada pembentukan *eigenface*, data ini berupa citra wajah dengan jumlah dimensi sama dengan jumlah pixel dalam citra.

(b) Menormalisasi data

Normalisasi data dilakukan dengan mencari vektor rata-rata data, kemudian mengurangkan vektor rata-rata tersebut pada himpunan data awal D_a (persamaan (1)).

$$D_a = D_a - \text{avg}(D_a) \quad (1)$$

(c) Menghitung matriks kovarian

Matriks kovarian $C^{m \times n}$ dibutuhkan untuk mengukur nilai keterhubungan antar dimensi pada himpunan data. Matriks kovarian didapat dengan mencari nilai kovarian untuk tiap dimensi terhadap semua dimensi dalam himpunan data pada persamaan (2).

$$C^{m \times n} = (c_{i,j} = \text{cov}(\text{Dim}_i, \text{Dim}_j)) \quad (2)$$

(d) Mencari *eigenvector* dan *eigenvalue*

Secara singkat, *eigenvector* x dari sebuah matriks A adalah sebuah vektor khusus yang memiliki sifat pada persamaan (3). Dengan λ adalah *eigenvalue* dari x sehingga bisa dicari dengan menyelesaikan persamaan (4). Dari matriks kovarian berukuran $N \times N$ akan didapatkan N buah *eigenvector* dan *eigenvalue*. Optimasi perhitungan tersebut pada persamaan (5) dilakukan dengan mencari *eigenvector* dari matriks $T = A^T \times A$ ordo $M \times M$ [4]. Optimasi sangat mempengaruhi kompleksitas perhitungan karena biasanya $M \ll N$.

$$Ax = \lambda x \quad (3)$$

$$(A - \lambda I)x = 0 \quad (4)$$

$$T = A^T \times A$$

$$Tv_i = \lambda_i v_i \quad (5)$$

$$A^T Av_i = \lambda_i v_i$$

$$AA^T Av_i = A \lambda_i v_i$$

$$AA^T (Av_i) = \lambda_i (Av_i) \quad (6)$$

(e) Memilih komponen utama data

Proses ini dilakukan dengan mengurutkan *eigenvector-eigenvector* tersebut sesuai dengan *eigenvalue*-nya dari yang terbesar sampai yang terkecil. Dengan demikian diperoleh himpunan *eigenvector* terurut berdasarkan tingkat kekuatan hubungan antar dimensinya. Dari sini, tentukan P *eigenvector* terbesar yang mewakili data.

P *eigenvector* yang dipilih kemudian dipisahkan untuk membentuk vektor fitur yaitu vektor-vektor yang digunakan untuk merepresentasikan data. Vektor-vektor ini dikumpulkan dalam matriks sebagai kolom-kolom dalam matriks (persamaan (7)).

$$\text{FeatureVector} = (\text{eig}_1, \text{eig}_2, \dots, \text{eig}_P) \quad (7)$$

Pada proses pengenalan wajah, *eigenvector-eigen-vector* inilah yang disebut sebagai *eigenface*. Dari percobaan yang dilakukan didapati bahwa pada penggunaan 15 *eigenface* dengan *eigenvalue* terbesar, proses pengenalan wajah mencapai nilai *error* yang bisa diterima.

(f) Membentuk himpunan data baru

Himpunan data baru didapat dari perkalian vektor fitur.

Yakni mengalikan vektor fitur yang ditranspose (vektor fitur menjadi vektor baris) dengan transpose data asli yaitu data yang telah dinormalisasi (persamaan (8)).

$$\text{dataBaru} = \text{FeatureVector}^T \times \text{DataAdjusted}^T \quad (8)$$

Operasi di persamaan (8) menghasilkan transpose data asli yang diproyeksikan ke vektor fitur terpilih. Pada sistem pengenalan wajah, operasi tersebut akan menghasilkan citra-citra wajah asal yang telah berubah karena terjadi pengurangan dimensi (informasi).

3. Nearest Feature Line (NFL)

Algoritme klasifikasi NFL [2] mengasumsikan untuk tiap kelas setidaknya terdapat dua *prototype* yang berbeda. Oleh karena itu terdapat dua anggota yang berbeda dalam tiap kelas. Selanjutnya *prototype* akan disebut sebagai *point*. Metode NFL menggunakan sebuah model linier untuk menginterpolasi dan mengekstrapolasi tiap pasang *point* dari sebuah kelas yang sama. Dari kedua *point* ini, ditarik sebuah garis yang menggeneralisasi kapasitas representasi kedua *point* tersebut. Garis yang menghubungkan dua *point* dalam satu kelas yang sama ini disebut dengan *Feature Line* garis fitur.

Secara virtual garis fitur akan menyediakan *point-point* fitur yang tak terbatas dari kelas *point* tersebut sehingga kapasitas himpunan *prototype* dalam sebuah kelas akan bertambah. Untuk sebuah kelas c dengan jumlah anggota $N_c > 1$ akan terbentuk $K_c = N_c(N_c - 1) / 2$ buah garis yang bisa digunakan sebagai representasi dari kelas tersebut. Misalkan untuk lima *prototype* dalam sebuah kelas, maka representasi kelas tersebut bisa diperbanyak menjadi 10 jumlah garis fitur yang bisa dibangun. Jumlah total garis fitur dari M kelas yang akan digunakan dalam proses NFL ditunjukkan pada persamaan (9).

$$N_{\text{total}} = \sum_{c=1}^M K_c \quad (9)$$

Jika diimplementasikan pada citra wajah, perubahan-perubahan diantara dua *point* tersebut bisa berupa variasi posisi muka, pencahayaan dan ekspresi saat pengambilan citra. Proses klasifikasi dalam NFL dilakukan dengan menghitung jarak minimal antara *point* fitur yang diuji ke garis fitur. Hasil dari klasifikasi juga akan memberikan posisi relatif dari *point* fitur yang diuji terhadap dua *point* terdekat dalam kelas memben-tuk garis fitur.

Jarak Garis Fitur, *feature line distance*

Misalkan terdapat variasi wajah dari z_1 ke z_2 dalam ruang citra beserta variasi yang timbul karenanya dalam ruang fitur (ruang *eigenface*) dari x_1 ke x_2 . Besar dari perubahan tersebut bisa dihitung sebagai $\delta_z = \|z_2 - z_1\|$ atau $\delta_x = \|x_2 - x_1\|$. Ketika $\delta_z \rightarrow 0$ maka $\delta_x \rightarrow 0$. Setiap perubahan posisi x karena perubahan variasi *point* bisa

diperkirakan dengan cukup baik menggunakan sebuah garis lurus yang membentang antara x_1 dan x_2 . Sehingga untuk tiap perubahan variasi diantara kedua titik tersebut bisa dilakukan interpolasi sebuah titik pada garis tersebut. Hal ini juga berlaku untuk perubahan yang posisinya berada disisi luar dari garis x_1 dan x_2 . Pada kasus ini model linier akan mengekstrapolasi posisi perubahan terhadap garis fitur.

Sebuah garis lurus yang melewati x_1 dan x_2 pada kelas yang sama disebut garis fitur dari kelas tersebut. Fitur point x yang diuji akan diproyeksikan terhadap garis fitur pada titik p . Jarak antara garis fitur dan x didefinisikan pada persamaan (10) dengan $\|a\|$ merupakan nilai panjang dari vektor a . Kemudian titik proyeksi p dihitung melalui persamaan (11) dengan $\mu \in \mathbb{R}$ menunjukkan parameter posisi titik p pada garis fitur dari x_1 . Nilai μ bisa didapatkan dari nilai x , x_1 , atau x_2 .

Catatan, notasi garis fitur pada x_1 dan x_2 adalah: $\overline{x_1 x_2}$.

$$d(x, \overline{x_1 x_2}) = \|x - p\| \quad (10)$$

$$p = x_1 + \mu(x_2 - x_1) \quad (11)$$

Persamaan (12) terbentuk karena kondisi tegak lurus $\overline{px} \perp \overline{x_1 x_2}$.

$$\begin{aligned} (p - x) \cdot (x_2 - x_1) &= 0 \\ (x_1 + \mu(x_2 - x_1) - x) \cdot (x_2 - x_1) &= 0 \\ \mu &= \frac{(x - x_1) \cdot (x_2 - x_1)}{(x_2 - x_1) \cdot (x_2 - x_1)} \end{aligned} \quad (12)$$

Notasi “ \cdot ” merupakan notasi untuk operasi *dot product*. Parameter μ menunjukkan posisi p relatif terhadap x_1 dan x_2 . Beberapa kemungkinan nilai μ : (1) $\mu = 0$ maka $p = x_1$, (2) $\mu = 1$ maka $p = x_2$, (3) $0 < \mu < 1$ maka p merupakan titik interpolasi antara x_1 dan x_2 , (4) $\mu > 1$ maka p merupakan ekstrapolasi maju pada sisi x_2 , dan (5) $\mu < 0$ maka p merupakan ekstrapolasi mundur pada sisi x_1 .

Perubahan variasi antara dua titik dalam sebuah kelas bisa diperkirakan dengan baik oleh garis fitur karena kemampuannya untuk menginterpolasi posisi parameter perubahan.

Klasifikasi Berbasis NFL

Klasifikasi berbasis NFL dilakukan dengan mencari jarak garis fitur terdekat antara x dan semua garis fitur yang mungkin. Misal x_i^c dan x_j^c merupakan dua *prototype* yang akan diuji. Proses klasifikasi menghasilkan N_{total} jarak garis fitur yang kemudian diurutkan menaik beserta atribut kelas, *point* yang membentuk garis fitur dan μ . Jarak NFL adalah urutan pertama dari jarak garis fitur yang telah terurut (lihat persamaan (13)).

$$d(x, \overline{x_i^c x_j^c}) = \min_{1 \leq c \leq M} \min_{1 \leq i < j \leq N_c} d(x, \overline{x_i^c x_j^c}) \quad (13)$$

Urutan pertama dari jarak garis fitur memberikan hasil klasifikasi NFL terdiri dari kelas tercocok c^* serta dua *prototype* i^* dan j^* yang merupakan *prototype* termirip dengan *prototype* uji. Parameter posisi μ^* menunjukkan posisi dari titik proyeksi p relatif terhadap:

$$\overline{x_{i^*}^{c^*} x_{j^*}^{c^*}}$$

4. Nearest Feature Midpoint (NFM)

NFM adalah metode klasifikasi yang merupakan perbaikan dari NFL. NFM mengasumsikan setidaknya akan terdapat dua *prototype* berbeda pada sebuah kelas. Didalam NFM sebuah sub ruang fitur dibentuk untuk tiap kelas yang terdiri dari titik tengah fitur (*feature midpoint*) antara tiap dua *prototype* pada kelas yang sama, x_1 dan x_2 , dan dinotasikan sebagai $m_{x_1 x_2}$. *Prototype* pada kelas yang sama akan digeneralisasi oleh titik tengah fitur untuk merepresentasikan variasi dari kelas sehingga kemampuan pengklasifikasi melakukan generalisasi juga akan meningkat. Jarak NFM adalah jarak *euclidean* terkecil antara objek yang diuji dengan semua titik tengah yang mungkin dibangun.

Semua titik pada garis fitur antara x_1 dan x_2 bisa ditunjukkan sebagai $x_1 + \lambda(x_2 - x_1)$ dengan $-\infty < \lambda < \infty$. Ketika $\lambda = 0.5$ maka $m_{x_1 x_2} = 0.5(x_1 + x_2)$ adalah titik tengah fitur dari garis fitur. Persamaan (14) adalah jarak titik tengah fitur antara *feature point* yang diuji x dan $m_{x_1 x_2}$ dengan $\|\cdot\|$ berarti operasi pencarian panjang vektor.

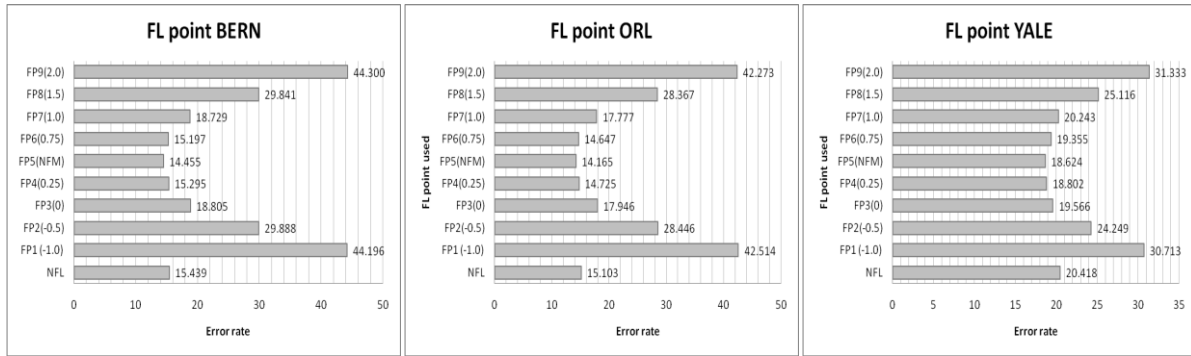
$$d(x, m_{x_1 x_2}) = \|x - m_{x_1 x_2}\| \quad (14)$$

Untuk sebuah kelas c dengan jumlah anggota $N_c > 1$ akan terbentuk $K_c = N_c(N_c - 1)/2$ titik tengah yang bisa digunakan sebagai representasi dari kelas tersebut. Sama dengan NFL, jumlah total titik tengah dari M kelas yang terbentuk ditunjukkan pada persamaan (9) serta proses klasifikasi dilakukan dengan menghitung jarak minimal antara *point* fitur yang diuji dan jarak titik tengah.

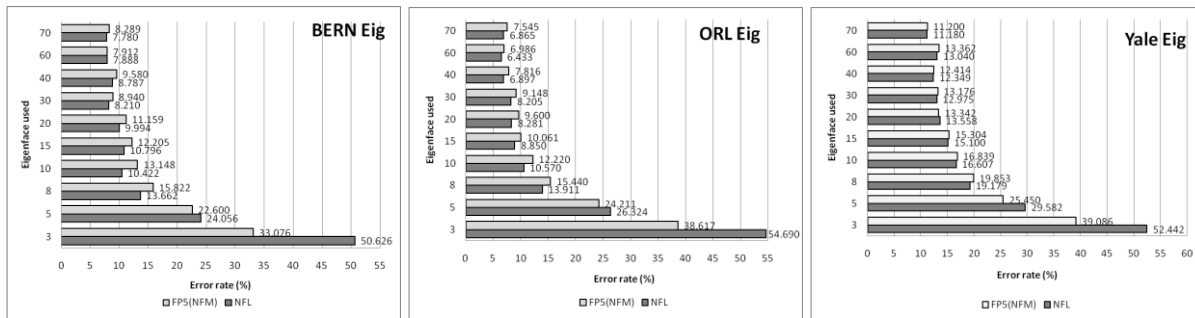
$$d(x, \overline{x_i^c x_j^c}) = \min_{1 \leq c \leq M} \min_{1 \leq i < j \leq N_c} d(x, m_{x_i^c x_j^c}) \quad (15)$$

Jarak titik tengah juga akan diurutkan secara menaik seperti klasifikasi dengan NFL pada persamaan (15).

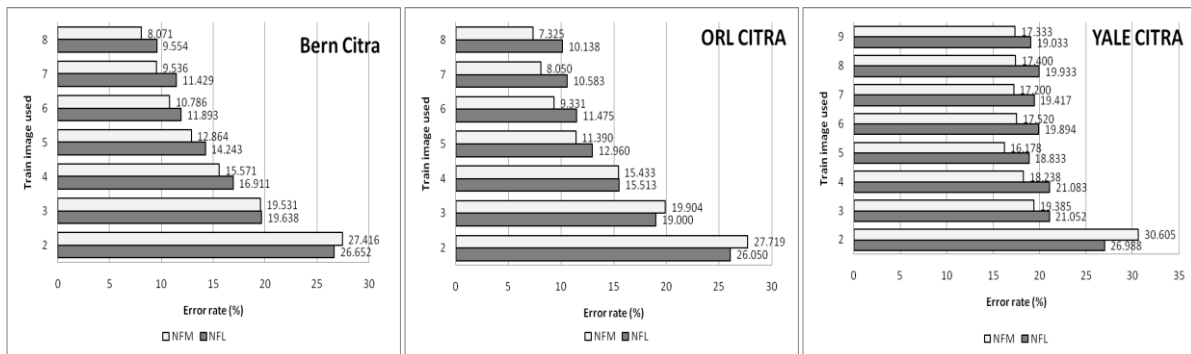
Jika ukuran populasi tak mendekati berhingga maka nilai ekspektasi dari populasi adalah nilai *mean*. Dianalogikan untuk penentuan titik proyeksi citra uji pada garis fitur, maka titik proyeksi pada proses pengenalan akan mendekati titik tengah dari garis fitur. Asumsi bahwa dimensi wajah sangat banyak. Oleh karena itu pada titik tengah (*midpoint*) klasifikasi berbasis NFL maupun NFM akan mendapatkan tingkat akurasi



Gambar 1. Grafik error rate penggunaan titik proyeksi beragam.



Gambar 2. Grafik hubungan penggunaan jumlah eigenface dengan rata-rata error rate.



Gambar 3. Grafik hubungan jumlah citra latihan yang digunakan dengan error rate.

yang relatif sama. Titik tengah tersebut digunakan pada NFM sebagai titik ukur jarak.

5. Implementasi dan Analisis Hasil

Uji coba dilakukan dengan dua skenario.

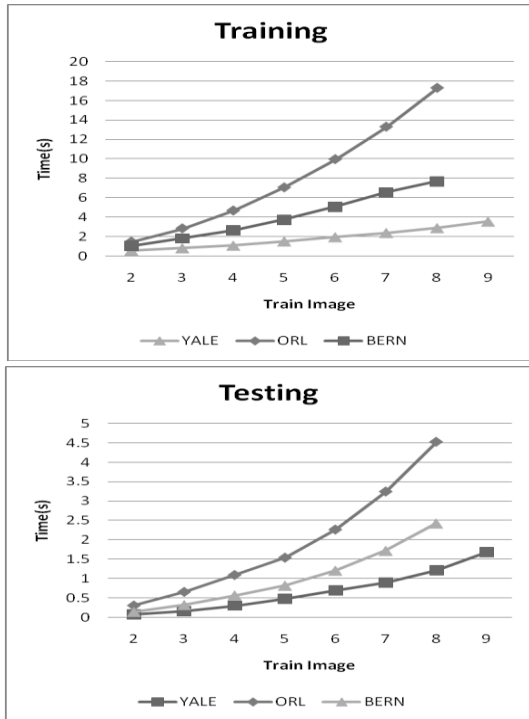
(1) untuk mendapatkan konfigurasi optimal dari sistem pengenalan wajah, dan (2) untuk membandingkan kinerja metode NFL dan NFM.

Basis data yang digunakan dalam pengujian adalah: basis data Bern¹ (28 subjek, 10 citra tiap subjek), ORL

(40 subjek, 10 citra tiap subjek), dan Yale² (15 subjek, 11 citra tiap subjek). Basis data Bern memiliki karakteristik adanya perubahan yang relatif kecil pada ekspresi wajah (*facial expression*) serta perubahan posisi kepala kearah kiri, kanan, atas dan bawah sebesar 30 derajat. Citra yang digunakan terlebih dahulu akan dinormalisasi dengan melakukan reduksi ukuran citra asal (*cropping*) menjadi citra berukuran 64x91. Subyek bervariasi terhadap jenis kelamin, ekspresi wajah, pencahayaan dan aksesoris wajah (misalnya kacamata) pada data Yale. Citra yang digunakan terlebih dahulu akan dinormalisasi

¹ <ftp://iamftp.unibe.ch/pub/Images/FaceImages/>

² <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>



Gambar 4. Grafik hubungan penggunaan jumlah citra latih dan waktu eksekusi pada proses training.

dengan melakukan reduksi ukuran citra asal menjadi berukuran 64x88. Sedangkan pada data ORL, subyek bervariasi terhadap orientasi wajah dan sedikit variasi pada ekspresi wajah. Citra yang digunakan terlebih dahulu akan dinormalisasi dengan melakukan reduksi ukuran citra asal menjadi berukuran 92x112.

Skenario 1.

Proses pengenalan wajah dilakukan untuk mencoba semua kemungkinan konfigurasi. Dilakukan perulangan untuk mendapatkan hasil yang representatif sebanyak 10 kali. Konfigurasi yang dimaksudkan adalah:

Tabel 1. Tingkat akurasi pengujian dari skenario 2.

Pengamatan		Basis data			Rata-rata (%)
		BERN	ORL	YALE	
Akurasi (%)	NFL	91.429	94.05	86.155	90.545
	NFM	90.386	92.98	86.733	90.033
	Perbaikan	98.859	98.862	100.67	99.464
Time (s)	NFL	0,305	0.785	0.173	0.4212
	NFM	0.129	0.3493	0.078	0.1855
	Perbaikan	42.318	44.495	45.002	43.938

(1) Titik proyeksi yang diuji $\mu = -1.0, \mu = -0.5, \mu = 0, \mu = 0.25, \mu = 0.5$ (NFM), $\mu = 0.75, \mu = 1.0, \mu = 1.5, \mu = 2.0$, dan menggunakan metode NFL (menghitung parameter μ).

(2) Jumlah citra latih dimulai dari dua (syarat pembentukan FL) sampai jumlah citra per kelas - 2 (sebagai citra uji).

(3) Jumlah *eigenface* yang diuji = 3, 5, 8, 10, 15, 20, 30, 40, 60 dan 70.

Untuk pengamatan atas titik proyeksi di Gambar 1 terlihat penggunaan FP6 (pada titik 0.75*FL), FP5 (NFM), FP4 (pada titik 0.25*FL) dan NFL (hasil pencarian parameter proyeksi citra uji pada FL) diperoleh tingkat kesalahan kurang dari 20%. Sebagai catatan, FL adalah feature point atau garis fitur.

Tampak dari grafik pada Gambar 2 bahwa untuk semua basis data dan proyeksi titik citra uji ke FL rata-rata *error*nya akan semakin kecil dengan bertambahnya *eigenface* yang digunakan. Tampak juga bahwa pertambahan akurasi pada penggunaan setidaknya lima belas *eigenface* tidak signifikan.

Pada Gambar 3 tampak bahwa pemakaian citra latih berpengaruh pada tingkat akurasi. Berkurangnya nilai *error* paralel dengan bertambahnya citra latih yang digunakan. Namun pengaruh ini kurang signifikan setelah penggunaan setidaknya lima citra latih. Sedangkan Gambar 3 menunjukkan bahwa waktu eksekusi untuk pelatihan akan bertambah seiring dengan bertambahnya citra latih yang digunakan. Fenomena itu juga terjadi untuk waktu eksekusi pengujian.

Skenario 2.

Pengujian pada skenario dua dilakukan hanya menggunakan konfigurasi optimal dan perulangan sebanyak 50 kali. Dari pengujian diperoleh prosentase tingkat akurasi yang ditunjukkan pada Tabel 1.

Untuk memastikan perbedaan tingkat akurasi dan waktu eksekusi metode NFL dan NFM, maka hasil akurasi dan waktu eksekusi kedua metode tersebut akan diuji dengan Uji-t. Diasumsikan ukuran sampel kurang dari 30, populasi berdistribusi normal dan terdiri dari dua sampel yang saling bebas dan berpasangan. Pengujian dilakukan pada taraf keberartian 0.05 atau *confidence interval* 95% dengan aplikasi bantu SPSS v.10.

Dari hasil analisis terlihat bahwa untuk hipotesa H_0 : rata-rata akurasi kedua metode identik, $t_{hitung}=0.028 < t_{(11,0.025)}=1.80$ dengan *df* (*degree of freedom*) $n-1 = 11$ dan 0.025 adalah setengah dari $\alpha(0.05)=0.025$.

Dikarenakan $t_{hitung}(0.028) < t_{tabel}(1.80)$ maka H_0 diterima sehingga pernyataan rataan akurasi kedua metode sama adalah benar. Keputusan juga bisa diambil dengan

melihat nilai *significant Sig.(2-tailed)* = 0.978 > setengah $\alpha = 0.025$ maka H_0 diterima.

Analisis uji H_1 : rata-rata waktu kedua metode identik menghasilkan nilai $t_{hitung}=2.617 > t_{(11,0.025)}=1.80$. Karena $t_{hitung}(2.617) > t_{tabel}(1.80)$ maka H_1 ditolak sehingga pernyataan rataan waktu eksekusi kedua metode sama adalah salah. Dikarenakan juga nilai *Sig.(2-tailed)* = 0.016 < setengah $\alpha = 0.025$ maka H_1 ditolak.

Berdasarkan analisa H_0 dan H_1 terbukti bahwa NFM memperbaiki waktu eksekusi dari algoritma NFL.

6. Simpulan

Pengenalan wajah bisa dilakukan menggunakan metode *Nearest Feature Line* yang diperbaiki dengan *midpoint* dari *Feature Line* atau garis fitur pada metode *Nearest Feature Midpoint*. Perbaikan yang diberikan oleh NFM adalah peningkatan kecepatan eksekusi 43.93% dari penggunaan NFL dengan tingkat akurasi hampir sama. Meskipun demikian, konfigurasi optimal dengan hasil tingkat akurasi yang bisa diterima (90.545% untuk NFL dan 90.033% untuk NFM) adalah sebagai berikut:

(a) titik proyeksi yang diuji adalah penggunaan proyeksi citra ke FL (NFL) dan titik tengah dari FL (NFM), (b) jumlah *eigenface* minimal yang digunakan adalah 15, (c) jumlah citra latih berpengaruh pada akurasi namun dengan pemakaian setidaknya lima citra latih sehingga akurasi yang dicapai bisa diterima dengan waktu eksekusi relatif cukup pendek.

Kedua metode masih sangat bergantung pada citra masukan. Variasi yang berlebih pada latar belakang dan besar ukuran citra akan mempengaruhi hasil akurasi. Kekurangan ini bisa diminimalkan dengan melakukan proses pendahuluan pada citra yang akan digunakan meliputi *cropping* dan *resizing*.

Daftar Pustaka

- [1] Zhou, Zonglin, Kwoh Chee Keong, "The Nearest Feature Midpoint: A Novel Approach for Pattern Classification", *Intl. Journal of Information Technology*, Vol. 11, No. 1.
- [2] Li, Stan Z, Lu Juwei, "Face Recognition Using the Nearest Feature Line Methode", *IEEE Transactions on Neural Networks*, Vol. 10, No. 2, hal. 439 – 443, 1999.
- [3] Soelaiman, Rully, "Sistem Pengenalan Wajah dengan Penerapan Algoritma Genetika pada Optimasi Basis Eigenface dan Proyeksi Fisherface", *Tesis master*, Depok, Fakultas Ilmu Komputer Universitas Indonesia, 2003.
- [4] Turk, M, Pentland A., "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, hal: 71–86, 1991.