

BAB II

SISTEM KLASIFIKASI DENGAN ALGORITMA PROPAGASI BALIK

Bab ini akan membahas tentang ekstraksi fitur dengan *Principal Component Analysis* (PCA) dan tentang algoritma propagasi balik serta kelemahan dari algoritma tersebut. Pada dasarnya dalam sistem klasifikasi terdapat dua tahapan utama yaitu ekstraksi fitur dan proses klasifikasi itu sendiri [SIR06]. Dalam penelitian ini, proses klasifikasi dilakukan dengan menggunakan algoritma propagasi balik.

2.1 Ekstraksi Fitur dengan *Principal Component Analysis* (PCA)

Ekstraksi fitur merupakan salah satu bagian yang penting dalam proses klasifikasi. Dengan ekstraksi fitur yang tepat maka proses klasifikasi akan lebih bagus hasilnya. Ukuran dimensi dari suatu data citra sangatlah besar, sehingga jika seluruh pixel dalam citra digunakan sebagai fitur maka biaya komputasi yang dibutuhkan dalam proses klasifikasi akan lebih besar.

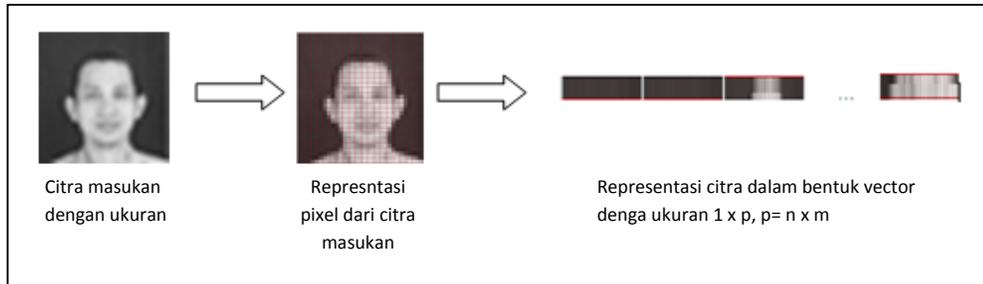
Selain biaya komputasi yang besar, dengan dimensi yang cukup besar memungkinkan adanya redundansi (penumpukan/ pengulangan) informasi dari suatu citra, sehingga besar kemungkinan terdapat informasi-informasi yang tidak penting yang dapat mengaburkan informasi yang penting dari data citra tersebut. Untuk mengatasi redundansi tersebut, maka dilakukan proses ekstraksi fitur dan dalam penelitian ini ekstraksi fitur dilakukan dengan menggunakan *Principal Component Analysis* (PCA) karena PCA merupakan salah satu cara untuk

mereduksi data set yang kompleks ke dimensi yang lebih kecil dengan tetap mempertahankan karakteristik dari data tersebut. PCA bekerja dengan melakukan pemetaan data dari sebuah ruang berdimensi- d ke ruang eigen berdimensi- c , dimana $c \leq d$.

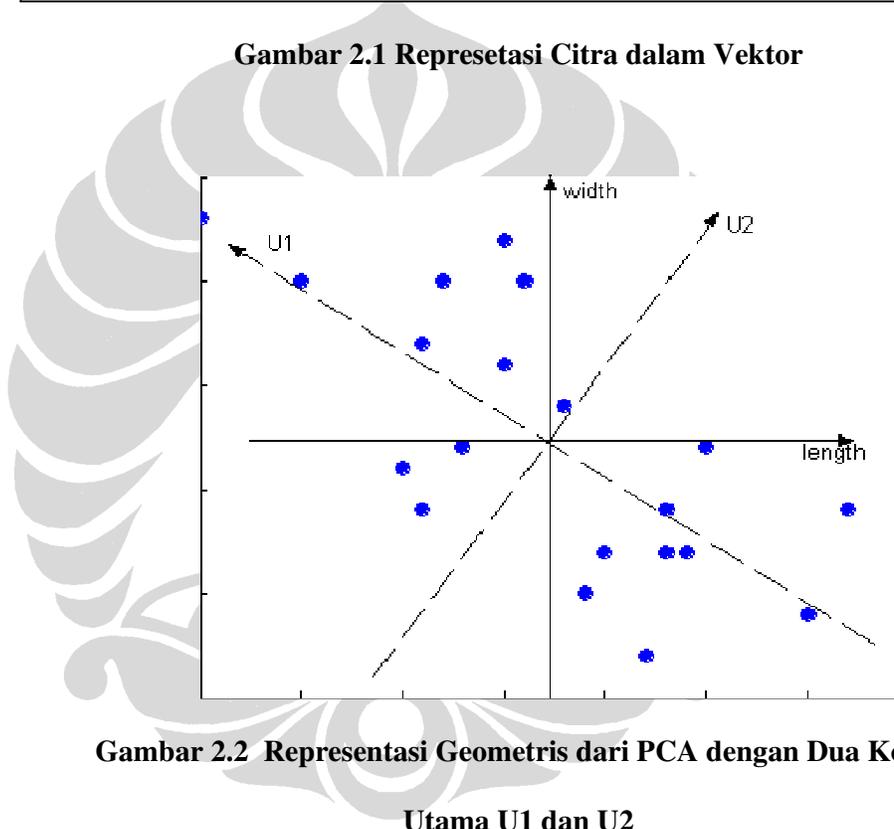
PCA pertama kali dikenalkan pada tahun 1901 oleh Karl Pearson. Metoda ini merupakan penurunan dari teknik *factor analysis* yang bertujuan untuk mengidentifikasi struktur dari banyak variable menjadi data yang lebih sederhana. PCA juga dikenal sebagai transformasi Karhunen-Loève atau transformasi Hotelling [MAR09]. PCA merupakan transformasi linier ortogonal yang mentransformasi data ke dalam koordinat sistem yang baru yang mana variasi yang paling banyak diperoleh dari proyeksi data pada koordinat prioritas utama (*principle component*) yang pertama, variasi kedua yang terbanyak pada koordinat prioritas utama yang kedua dan seterusnya. PCA biasanya digunakan untuk mengurangi dimensi dari himpunan data tetapi juga mempertahankan karakteristik dari himpunan data tersebut dengan menjaga beberapa *principle component* yang tinggi prioritasnya dan membuang beberapa *principle component* yang rendah prioritasnya.

Pada PCA data masukan adalah sebuah matriks yang berisi data citra pelatihan. Sebuah data citra pelatihan yang dinyatakan dalam satu matriks yang berisi nilai pixel dengan ordo $n \times m$, akan dikonversi menjadi sebuah vector dengan panjang $n \times m$. Vektor-vektor yang merupakan representasi data citra pelatihan akan dijadikan dalam sebuah matriks dengan ukuran $p \times q$, dimana p merupakan jumlah data pelatihan, dan $q = n \times m$ yang kemudian disebut dengan

dimensi dari citra. Representasi sebuah data citra wajah dalam sebuah vektor dapat dilihat pada ilustrasi gambar 2.1.



Gambar 2.1 Reresetasi Citra dalam Vektor



Gambar 2.2 Representasi Geometris dari PCA dengan Dua Komponen Utama U1 dan U2

Gambar 2.2 menunjukkan representasi komponen utama terhadap data asli. Dari gambar 2.2 dapat dilihat bahwa komponen utama U1 saling tegak lurus dengan komponen utama U2. Ada beberapa metoda yang umum digunakan untuk mendapatkan *principle component* pada metoda PCA, yaitu metoda kovarian,

metoda korelasi dan *singular value decomposition*. Prosedur PCA dengan menggunakan metoda kovarian adalah sebagai berikut [LIN06]:

1. Bentuk matriks data X yang berukuran $n \times d$, dimana n adalah jumlah data dan d adalah besarnya dimensi dari data tersebut.
2. Normalisasi data sehingga setiap vektor kolom memiliki rata-rata nol dan varian satu. Proses normalisasi data digunakan Z-score formula yang dinyatakan dalam persamaan 2.1

$$z_i = \frac{x_i - \mu}{s} \dots\dots\dots(2.1)$$

Dimana:

- z_i : data hasil normalisasi ke-1, $i=1, 2, 3, \dots, m$
- x_i : data ke- i
- μ : rata-rata kolom
- m : jumlah data
- s : standrart deviasi

3. Buat matriks kovarian C dari matriks hasil normalisasi tersebut.

$$C = E(XX') \dots\dots\dots (2.2)$$

Atau dapat dituliskan dalam persamaan 2.3

$$C \equiv \frac{1}{n - 1} \mathbf{X}\mathbf{X}^T \dots\dots\dots (2.3)$$

Dimana:

- C : matriks kovarian dari data.

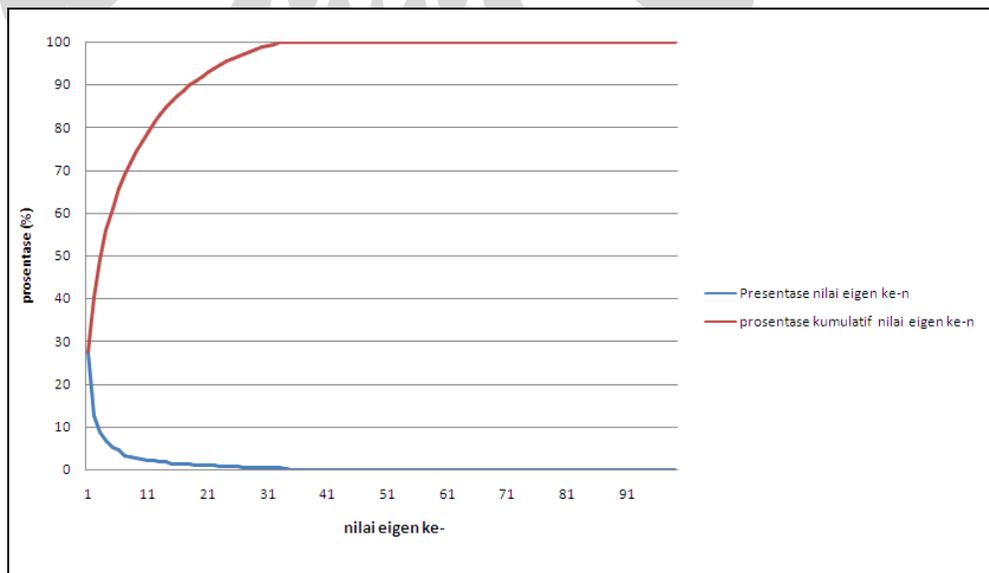
X : matriks yang berisi sekumpulan data citra masukan yang berukuran $n \times m$ dimana m adalah jumlah dimensi dan n adalah jumlah data citra input.

T : simbol transpose pada matriks.

n : jumlah data citra input.

4. Cari nilai eigen dan vektor eigen dari C . Vektor eigen inilah yang merupakan komponen utama dari data, sehingga untuk mengurangi dimensi kita tinggal memilih vektor eigen yang memiliki varian atau nilai eigen terbesar.

Dalam penelitian ini, Pemotongan dimensi dilakukan berdasarkan prosentase dari komponen utama. Prosentase komponen utama dihitung dari nilai eigen dibagi dengan jumlah total nilai eigen matriks C . Gambar 2.3 menunjukkan grafik prosentase dari komponen utama terhadap dimensi.



Gambar 2.3 Kurva Prosentase Komponen Utama



Gambar 2.4 Perbandingan Citra Wajah Setelah Ditransformasi Balik (berurut dari kiri ke kanan, direduksi dengan dimensi 10, 15, 20, 25, 30, 35, 40, 45, 50, 55) dengan Citra Wajah Asli

2.2 Algoritma Propagasi balik

Metode propagasi balik merupakan metode pembelajaran lanjut yang dikembangkan dari aturan perceptron. Hal yang ditiru dari perceptron adalah tahapan dalam algoritma jaringan. Metode propagasi balik ini dikembangkan oleh Rumelhart, Hinton dan Williams pada sekitar tahun 1986 yang mengakibatkan peningkatan kembali minat terhadap jaringan syaraf tiruan [JUA95]. Metode ini terdiri dari dua tahap yaitu tahap *feedforward* yang diambil dari perceptron dan tahap propagasi balik *error*. Salah satu hal yang membedakan antara back propagation dengan perceptron adalah pada arsitektur jaringannya. Perceptron memiliki jaringan lapis tunggal sedangkan propagasi balik memiliki lapisan lapis jamak, seperti pada gambar 2.5 yang merupakan jaringan neural lapis banyak (MLP) dengan satu lapis tersembunyi.

Langkah-langkah dalam proses metode propagasi balik jaringan syaraf tiruan adalah sebagai berikut :

Proses *Feedforward* :

- a. Menentukan nilai input pada layer input.

Nilai pada setiap node input didapatkan dari nilai tiap pixel pada citra yang akan dikenali.

- b. Menentukan nilai input untuk node-node layer hidden.

Nilai input untuk setiap node hidden didapatkan dengan menjumlahkan sinyal input terbobotnya.

- c. Menentukan nilai node-node layer hidden.

Setiap nilai node hidden didapatkan dengan menggunakan fungsi aktivasinya.

- d. Menentukan nilai input untuk node-node layer output.

Nilai input untuk setiap node output didapatkan dengan menjumlahkan sinyal input terbobotnya dari node hidden.

- e. Menentukan nilai node-node layer output.

Setiap nilai node output didapatkan dengan menggunakan fungsi aktifasinya.

Proses propagasi balik *of error* :

- a. Mengetahui nilai error bobot antara layer output dengan layer hidden.

Nilai error pada bagian ini digunakan untuk menghitung nilai koreksi bobot dan bias antara layer hidden dan layer output.

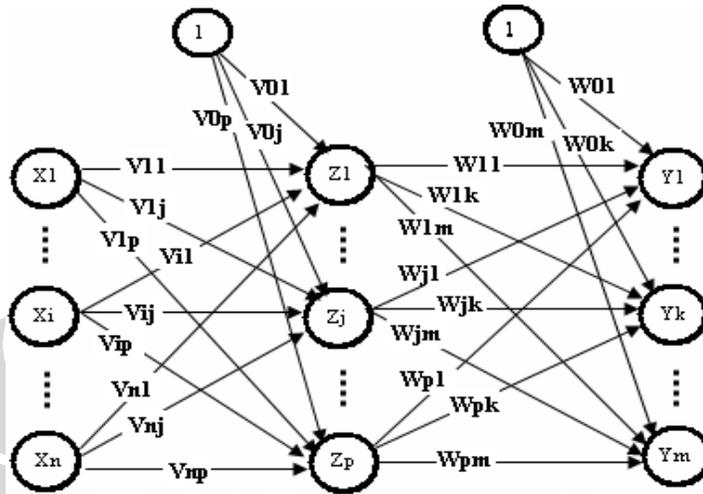
- b. Mengetahui nilai error bobot antara layer hidden dengan layer input.

Nilai error pada bagian ini digunakan untuk menghitung nilai koreksi bobot dan bias antara layer input dan layer hidden.

- c. Mengubah nilai tiap bobot dan bias.

Nilai setiap bobot diperbarui dengan menjumlahkan nilai bobot yang lama dengan nilai koreksi bobotnya.

Untuk lebih jelas tentang algoritma proses propagasi balik, maka dapat dilihat pada Gambar 2.5.



Gambar 2.5 Arsitektur Jaringan Propagasi balik

Keterangan :

- X_i : masukan (input).
- Z_k : node hidden layer
- V_{ij} : bobot pada lapisan tersembunyi.
- W_0 : bias keluaran.
- V_0 : bias pada lapisan tersembunyi.
- W_{jk} : bobot pada lapisan keluaran.
- Y_k : keluaran hasil.

2.2.1 Algoritma Pelatihan

Pelatihan suatu jaringan dengan algoritma propagasi balik meliputi dua tahap yaitu perambatan maju dan perambatan mundur. Selama perambatan maju, tiap unit masukan (x_i) menerima sebuah masukan sinyal ini ke tiap-tiap lapisan

tersembunyi z_1, \dots, z_p . Tiap unit tersembunyi ini kemudian menghitung aktivasinya dan mengirimkan sinyalnya (z_j) ke tiap unit keluaran. Tiap unit keluaran (y_k) menghitung aktivasinya (y_k) untuk membentuk respon pada jaringan untuk memberikan pola masukan. Selama pelatihan, tiap unit keluaran membandingkan perhitungan aktivasinya y_k dengan nilai targetnya t_k untuk menentukan kesalahan pola tersebut dengan unit itu.

Berdasarkan kesalahan ini, faktor δ_k ($k = 1, \dots, m$) dihitung, δ_k digunakan untuk menyebarkan kesalahan pada unit keluaran y_k kembali ke semua unit pada lapisan sebelumnya (unit-unit tersembunyi yang dihubungkan ke y_k). Selain itu δ_k juga digunakan untuk mengupdate bobot antara keluaran dan lapisan tersembunyi. Dengan cara yang sama, faktor ($j = 1, \dots, p$) dihitung untuk tiap unit tersembunyi z_j . Tidak perlu untuk menyebarkan kesalahan kembali ke lapisan masukan, tetapi δ_j digunakan untuk mengupdate bobot-bobot antara lapisan tersembunyi dan lapisan masukan.

Setelah seluruh faktor δ ditentukan, bobot untuk semua lapisan diatur secara serentak. Pengaturan bobot w_{jk} (dari unit tersembunyi z_j ke unit keluaran y_k) didasarkan pada faktor δ_k dan aktivasi z_j dari unit tersembunyi z_j . didasarkan pada faktor δ_j dan dan aktivasi x_i unit masukan. Untuk selengkapnya algoritma propagasi balik adalah sebagai berikut [FAU94]:

Langkah 0 : Inisialisasi bobot. (sebaiknya diatur pada nilai acak yang kecil),

Langkah 1 : Jika kondisi tidak tercapai, lakukan langkah 2-9,

Langkah 2 : Untuk setiap pasangan pelatihan, lakukan langkah 3-8,

Perambatan Maju :

Langkah 3 : Tiap unit masukan ($x_i, i = 1, \dots, n$) menerima sinyal x_i dan menghantarkan sinyal ini ke semua unit lapisan di atasnya (unit tersembunyi),

Langkah 4 : Setiap unit tersembunyi ($x_i, i = 1, \dots, p$) jumlahkan bobot sinyal masukannya,

$$z_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2.4)$$

v_{oj} = bias pada unit tersembunyi j aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya, $z_j = f(z_inj)$, dan kirimkan sinyal ini keseluruhan unit pada lapisan di atasnya (unit keluaran).

Langkah 5 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) jumlahkan bobot sinyal masukannya,

$$y_in_k = w_{ok} + \sum_{j=1}^n z_j w_{jk} \dots\dots\dots (2.5)$$

w_{ok} = bias pada unit keluaran k dan aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya, $y_k = f(y_ink)$.

Perambatan Mundur :

Langkah 6 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan informasinya,

$$\delta_k = (t_k - y_k) \dots\dots\dots (2.6)$$

hitung koreksi bobotnya (digunakan untuk memperbaharui wjk nantinya),

$$\Delta w_{jk} = \alpha \delta_k z_j + (\Delta w_{jk} * \text{momentum}) \dots \dots \dots (2.7)$$

hitung koreksi biasnya (digunakan untuk memperbaharui w_{jk} nantinya), dan kirimkan δ_k ke unit-unit pada lapisan dibawahnya,

Langkah 7 : Setiap unit lapisan tersembunyi ($z_j, j = 1, \dots, p$) jumlahkan hasil perubahan masukannya (dari unit-unit lapisan diatasnya),

$$\Delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \dots \dots \dots (2.8)$$

kalikan dengan turunan fungsi aktivasinya untuk menghitung informasi kesalahannya,

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \dots \dots \dots (2.9)$$

hitung koreksi bobotnya (digunakan untuk memperbaharui v_{oj} nanti),

Langkah 8 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) update bias dan bobotnya ($j = 0, \dots, p$):

$$w_{jk} (\text{baru}) = w_{jk} (\text{lama}) + \Delta w_{jk} \dots \dots \dots (2.10)$$

Tiap unit lapisan tersembunyi ($z_j, j = 1, \dots, p$) update bias dan bobotnya ($i = 0, \dots, n$):

$$v_{ij} (\text{baru}) = v_{ij} (\text{lama}) + \Delta v_{ij} \dots \dots \dots (2.11)$$

Langkah 9 : Test kondisi berhenti.

2.2.2 Algoritma Pengujian:

Setelah pelatihan, jaringan saraf propagasi balik diaplikasikan dengan hanya menggunakan tahap perambatan maju dari algoritma pelatihan. Prosedur aplikasinya adalah sebagai berikut :

Langkah 0 : Inisialisasi bobot (ambil dari algoritma pelatihan).

Langkah 1 : Untuk tiap vektor masukan, lakukan langkah 2-4.

Langkah 2 : for $i = 1, \dots, n$: atur aktivasi unit masukan x_i .

Langkah 3 : for $j = 1, \dots, p$:

$$z_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad \dots \dots \dots \quad (2.12)$$

$$z_j = f(z_in_j) \quad \dots \dots \dots \quad (2.13)$$

Langkah 4 : for $k = 1, \dots, m$:

$$y_in_k = w_{0k} + \sum_{j=1}^p z_j w_{jk} \quad \dots \dots \dots \quad (2.14)$$

$$y_k = f(y_in_k) \quad \dots \dots \dots \quad (2.15)$$

Langkah 5 : Jika $y_k \geq 0,5$ maka $y_k = 1$, else $y_k = 0$.

2.3 Kelemahan Algoritma Propagasi Balik

Algoritma propagasi balik sering diaplikasikan dalam beberapa sistem klasifikasi dan dinyatakan memiliki akurasi pengenalan yang cukup bagus [DAN95]. Selain proses pelatihan yang membutuhkan biaya komputasi yang cukup tinggi, propagasi balik juga memiliki kelemahan lainnya, yaitu ketidakmampuan algoritma ini mengenali data yang bukan merupakan bagian dari kelas data pelatihan dan dalam penelitian ini akan disebut dengan outlier.

Menurut Hair, dan rekan (1995) outlier adalah data yang muncul dan memiliki karakteristik unik yang jauh berbeda dari data observasi-observasi lainnya dan memiliki nilai ekstrim. Ferguson (1961) mendefinisikan outliers

sebagai suatu data yang menyimpang dari sekumpulan data yang lain [FRA03]. Secara khusus dalam penelitian ini outlier didefinisikan sebagai data yang memiliki kelas selain kelas yang terdapat dalam pelatihan algoritma propagasi balik.

Akurasi algoritma propagasi balik yang cukup tinggi terpenuhi jika sistem klasifikasi bekerja dalam kondisi ideal, yaitu data uji tidak meliputi data outlier. Jika data uji meliputi data outlier maka akurasi sistem akan menurun, semakin banyak data outlier maka semakin kecil pula akurasi dari algoritma propagasi balik dalam melakukan klasifikasi. Pada kenyataannya dalam aplikasi sebenarnya sangat mungkin terjadi dalam sistem klasifikasi data outlier muncul dan jumlahnya tidak dapat ditentukan sebelumnya, sehingga perlu penanganan agar algoritma propagasi balik memiliki kemampuan untuk menentukan data outlier dan bukan outlier.

Selama ini penelitian tentang pengembangan algoritma propagasi balik lebih pada sistem optimasi komputasinya, atau akurasinya tetapi masih dalam lingkup data yang ideal. Sedangkan publikasi penelitian tentang bagaimana mengatasi kelemahan propagasi balik yang tidak mampu menentukan data outlier dan bukan outlier belum dijumpai sampai saat penelitian dalam tesis ini dikembangkan. Untuk mengetahui pengaruh kelemahan propagasi balik terhadap akurasi pengenalannya akan dilakukan percobaan dengan data set pada tabel 2.1.

Dalam percobaan ini dilakukan pengenalan terhadap wajah seseorang dengan perbedaan sudut pandang, data citra wajah yang diujikan memiliki sudut pandang tidak diikutkan dalam proses pelatihan. Data outlier dalam percobaan ini adalah data wajah orang selain orang yang diikutkan dalam proses pelatihan

dengan semua sudut pandang (sudut yang diikutkan dalam proses pelatihan dan yang tidak diikutkan dalam proses pelatihan).

Tabel 2.1 Data Set Percobaan Sistem Klasifikasi dengan Algoritma Propagasi Balik

Data Pelatihan (°)	Data Uji (°)		Perbandingan (%)	
	Data Uji	Data Uji Oulier	data Pelatiha : DataUji	data Pelatihan : Data Uji All*
0, 30, 60,90, 120, 150, 180	15, 45, 75, 105, 135, 165	0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180	54 : 46	29 : 71
(5 data citra wajah JST lama)	(5 data citra wajah JST baru)	(3 data citra wajah JST baru + 4 data citra wajah JST lama)		

Pada tabel 2.1 data yang digunakan dalam percobaan ini adalah data wajah orang Indonesia dalam Lab Kecerdasan Komputasi FASILKOM Universitas Indonesia yang kemudian akan disebut dengan data JST. Data wajah ini terdiri dari data wajah duabelas orang dengan sudut pandang anantara 0° hingga 180° dengan perbedaan sudut rotasi 15° dengan ekspresi normal, jadi total data citra yang digunakan adaah 156. Dalam percobaan ini, data yang dilatih adalah data wajah lima orang dengan sudut mulai dari 0° hingga 180° dengan perbedaan sudut rotasi 30°, sedangkan data uji menggunakan data uji dengan sudut diantara data yang dilatih, yaitu data yang dilatih pada sudut 0°, 30°, 60°, 90°, 120°, 150°, 180° dan data yang diuji data dengan sudut 15°, 45°, 75°, 105°, 135°, 165°. Data outlier adalah data wajah JST selain data wajah orang yang digunakan dalam kelas target pada proses pelatihan, sudut pandang data yang digunakan adalah seluruh sudut pandang dari 0° hingga 180° dengan perbedaan sudut 15°.

Data set pada tabel 2.1 menggunakan *unkown pose* yaitu untuk mengetahui apakah sistem pengenalan masih dapat mengenali data-data dengan sudut yang berbeda dengan data pelatihan pada orang yang sama dan ekspresi yang sama. Data outlier jumlahnya lebih banyak dibandingkan dengan data yang bukan outlier untuk menunjukkan bahwa semakin banyak jumlah data outlier maka akurasi pengenalan propagasi balik semakin berkurang.

Tabel 2.2 Contoh Data Citra Wajah JST dengan Perbedaan sudut Pandang yang Digunakan dalam Percobaan

Sudut Pandang(°)												
0	15	30	45	60	75	90	105	120	135	150	165	180
												

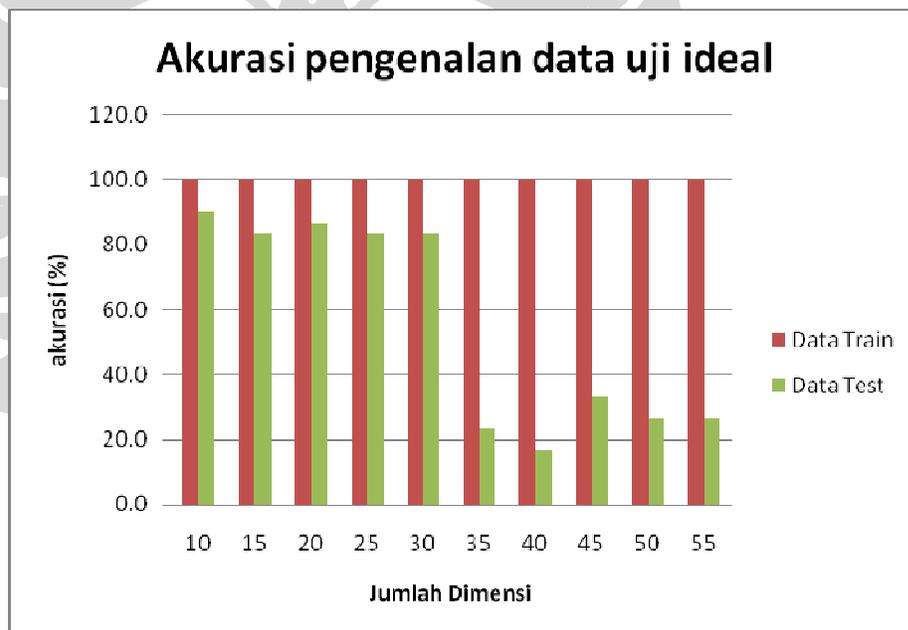
Tabel 2.2 menunjukkan contoh data citra wajah seseorang yang digunakan dalam percobaan ini dengan sudut pandang yang berbeda. Sedangkan tabel 2.3 menunjukkan contoh data citra wajah outlier dan bukan outlier yang digunakan dalam percobaan, data yang digunakan dalam tabel contoh adalah data dengan sudut 0°.

Tabel 2.3. Contoh Data Outlier dan non Outlier yang Digunakan dalam Percobaan

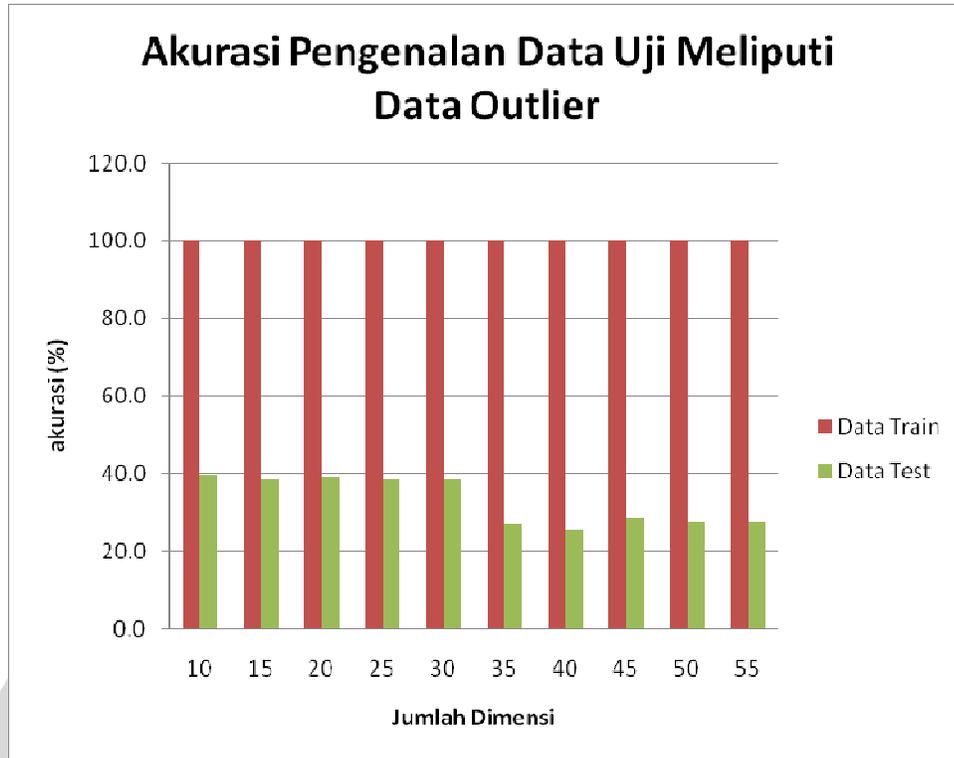
Ket	Citra wajah orang dari data JST dengan ekspresi normal sudut 0°						
Data Uji							
Data Outliers							

Tabel 2.4 Tingkat Keberhasilan Pengenalan pada Algoritma Propagasi Balik

Jumlah Dimensi	Prosentase Akurasi Pengenalan (%)		
	Data Train	Data Uji	Data Uji meliputi data Outlier
10	100	90	39.7
15	100	83.3	38.5
20	100	86.7	39.1
25	100	83.3	38.5
30.0	100	41.0	38.5
35.0	100	25.6	26.9
40.0	100	27.6	25.6
45.0	100	28.2	28.8
50.0	100	28.8	27.6
55.0	100	27.6	27.6



Gambar 2.6 Grafik Akurasi Pengenalan Sistem Klasifikasi dengan Algoritma Propagasi Balik

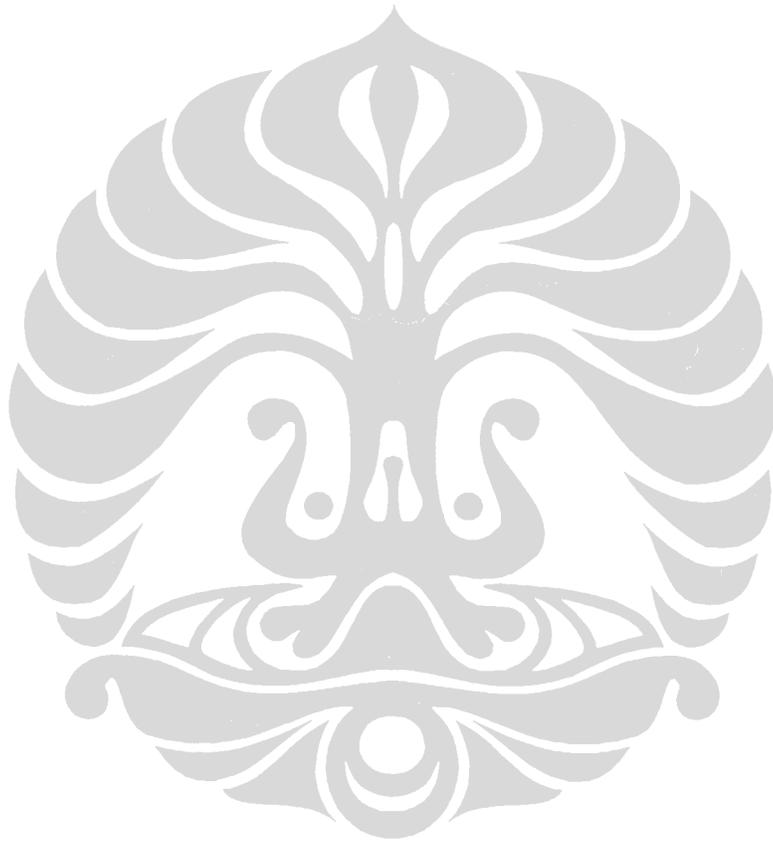


Gambar 2.7 Grafik Akurasi Pengenalan pada Data Uji Meliputi Data Outliers

Pada tabel hasil percobaan 2.4 dan gambar grafik pada gambar 2.6 dan gambar 2.7 dapat dilihat bahwa nilai akurasi sistem klasifikasi pada algoritma propagasi balik memiliki nilai akurasi tertinggi pada percobaan dengan dimensi 10 dan ketika dimensi diatas 25, nilai akurasi dari sistem klasifikasi propagasi balik menurun. Penurunan nilai akurasi pada dimensi diatas 25 dikarenakan pada dimensi 25 prosentase kumulatif dari *principal component* mendekati 100%, hal ini dapat dilihat pada gambar 2.3 yang merepresentasikan nilai prosentase terhadap dimensi pada PCA.

Dalam kondisi data ideal akurasi sistem pengenalan wajah dalam percobaan ini menunjukkan nilai akurasi pengenalan mencapai 90% pada dimensi 10, sedangkan jika sistem bekerja pada data yang mengandung outlier dapat dilihat

pada kolom ke-4 pada tabel 2.6 bahwa nilai akurasinya menurun drastis dari 90% menjadi 39.7% (akurasi sistem yang bekerja pada dimensi 10). Jika semakin banyak jumlah data outlier yang dimiliki pada data uji, maka semakin rendah pula akurasi pengenalan yang dihasilkan, untuk selengkapnya akan dibahas pada Bab IV.



BAB III

SISTEM PENENTU OUTLIER PADA ALGORITMA PROPAGASI BALIK

Pada Bab III akan dibahas tentang penentuan outlier pada algoritma propagasi balik dengan menggunakan perhitungan jarak mahalanobis dan perhitungan jarak fuzzy. Pengembangan algoritma penentu outlier pada algoritma propagasi balik diharapkan mampu mengurangi kelemahan algoritma propagasi balik, yaitu kelemahannya dalam menentukan outlier, sehingga algoritma propagasi balik dapat menghilangkan sifat ‘pemaksaan’ yang dimilikinya. Seperti yang telah dibahas dalam bab terdahulu, sifat pemaksaan yang dimaksud adalah kelemahan algoritma propagasi balik dalam mengklasifikasikan data outlier, jika terdapat data outlier pada data uji, sistem akan mengklasifikasikan data tersebut dalam salah satu kelas yang ada, walaupun sebenarnya data tersebut bukan termasuk dalam kelas manapun.

Penentuan outlier pada algoritma propagasi balik menggunakan perhitungan jarak fuzzy akan dibagi menjadi dua sub bahasan, yaitu penentuan outlier menggunakan perhitungan jarak fuzzy dan jarak euclidean serta penentuan outlier menggunakan jarak fuzzy dan jarak mahalanobis. Penggunaan jarak euclidean dan jarak mahalanobis dalam algoritma penentuan outlier menggunakan perhitungan jarak fuzzy adalah pada perhitungan jarak antara sebuah titik (representasi dari sebuah data uji) terhadap pusat data.

Sebelum membahas tentang algoritma penentu outlier terlebih dahulu akan dibahas tentang perhitungan jarak mahalanobis dan perhitungan jarak euclidean.

3.1 Perhitungan Jarak Mahalanobis

Jarak mahalanobis adalah salah satu teknik pengukuran jarak yang diperkenalkan oleh P. C. Mahalanobis pada tahun 1936. Secara umum jarak mahalanobis ($D_m(x)$) dari sekelompok data $x=(x_1, x_2, x_3, \dots, x_N)^T$ dengan rata-rata $\mu = (\mu_1, \mu_2, \mu_3, \dots, \mu_N)^T$ dan kovarian matriks S didefinisikan pada persamaan 3.1 [6].

$$D_m(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \quad \dots \dots \dots \quad (3.1)$$

Jarak mahalanobis dapat juga didefinisikan sebagai tingkat perbedaan (*dissimilarity measure*) antara dua random vector x dan y yang memiliki distribusi yang sama dengan kovarian matriks S .

$$d_m(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)} \quad \dots \dots \dots \quad (3.2)$$

Jika kovarian matrik merupakan matriks identitas, jarak mahalanobis dapat diturunkan menjadi jarak Euclidean. Jarak mahalanobis akan berguna jika:

- a. Kedua vector memiliki korelasi
- b. Kedua vector memiliki varian yang berbeda,
- c. Distribusi data gaussian atau mendekati distribusi gaussian.
- d. Kovarian matriks dari kedua vector tersebut merupakan matriks non-singular.

Dalam penelitian ini, sangat dimungkinkan bahwa data citra wajah yang digunakan sebagai data masukan memiliki distribusi yang non-gaussian, sehingga agar perhitungan jarak mahalanobis dapat bekerja dengan optimal maka data akan

dinormalisasi terlebih dahulu. Jika data dinormalisasi, dengan kata lain memiliki nilai rata-rata 0 maka, persamaan 3.1 dapat dinyatakan menjadi persamaan 3.3.

$$D_m(x) = \sqrt{Z^T S^{-1} Z} \quad \dots \dots \dots \quad (3.3)$$

Dengan tata nama sebagai berikut:

- $D_m(x)$: jarak mahalanobis
- Z : data yang telah dinormalisasi
- T : simbol *transpose* pada matiks
- S^{-1} : invers kovarian matriks

Nilai kovarian matriks yang singular menyebabkan jarak mahalanobis tidak dapat dihitung, karena matriks singular tidak memiliki nilai invers. Secara umum matriks kovarian merupakan matriks semi definit positif, namun untuk kasus data citra sering ditemukan adanya singularitas nilai matriks kovarian, sehingga untuk mengatasi singularitas ini akan digunakan regularisasi. Secara singkat regularisasi dapat dinyatakan dalam persamaan 3.4.

$$S_{baru} = S + \lambda I \quad \dots \dots \dots \quad (3.4)$$

Dengan penamaan sebagai berikut:

- S_{baru} : matriks kovarian setelah regularisasi
- λ : bilangan kecil mendekati nol
- I : matriks identitas

3.2 Jarak Euclidean

Secara umum jarak euclidean sering disebut sebagai jarak. Definisi matematis jarak euclidean biasa disebut dengan jarak antara dua titik yang dapat ditarik garis dan dihitung panjangnya dengan menggunakan penggaris, hal ini dapat

dibuktikan kembali dengan menggunakan teorema Pythagoras. Dengan menggunakan formula perhitungan jarak, maka *euclidean space* dapat menjadi *metric space*, yang bias disebut juga sebagai *euclidean norm*.

Secara umum jarak euclidean dari dua buah vector random x dan y dapat dinyatakan dalam persamaan 3.5 [ZHO07].

$$d(x,y) = \sqrt{(x - y)^2} \dots \dots \dots (3.5)$$

Dengan tatanama sebagai berikut:

$d(x,y)$: jarak euclidean antara vektor x dengan vektor y

x,y : vektor acak

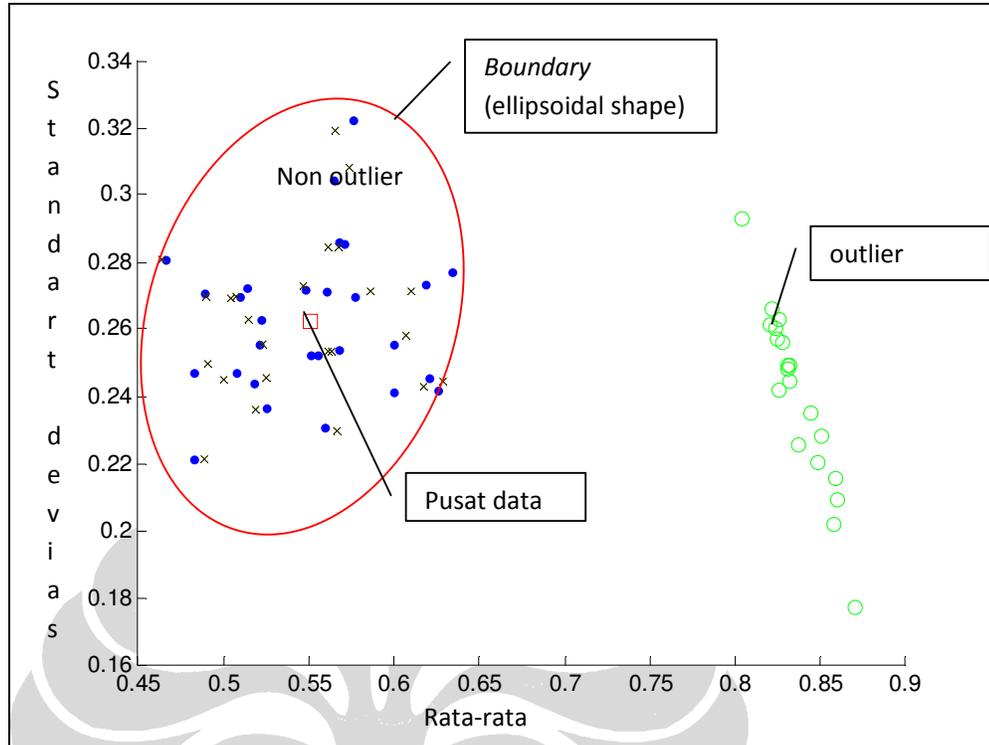
Jarak euclidean sering digunakan dalam berbagai macam aplikasi karena jarak euclidean merupakan salah satu cara menghitung jarak yang cukup sederhana dan mudah diaplikasikan.

3.3 Penentuan Outlier pada Algoritma Propagasi Balik Menggunakan Mahalanobis Distance Outlier Determination (MDOD)

Pada sub bahasan ini akan dibahas metode baru yang digunakan untuk membantu algoritma propagasi balik dalam mengatasi kelemahannya, yaitu menentukan outlier pada data uji. Pada dasarnya untuk memisahkan suatu data dengan data yang lain, dibutuhkan suatu batasan, representasi batasan ini dapat berupa garis atau suatu bidang. Dalam sistem penentu outlier berbasis pada perhitungan jarak mahalanobis, batasan untuk memisahkan data outlier dengan data bukan outlier merupakan suatu bidang yang berbentuk elipsoidal. Metode penentu outlier pada algoritma propagasi balik menggunakan perhitungan jarak mahalanobis yang dikembangkan dalam penelitian ini akan disebut dengan

Mahalanobis Distance Outlier Determination (MDOD) yang dalam pembahasan selanjutnya akan dituliskan dengan MDOD. Seperti yang didefinisikan sebelumnya outlier adalah data yang muncul dan memiliki karakteristik unik yang jauh berbeda dari data observasi-observasi lainnya dan memiliki nilai ekstrim. Ferguson (1961) mendefinisikan outliers sebagai suatu data yang menyimpang dari sekumpulan data yang lain [IRA05]. Karena memiliki nilai yang ekstrim dan menyimpang terhadap data lainnya (data non outlier) maka data outlier dapat dipisahkan dengan mudah dari data non outlier. Untuk memisahkan data outlier dengan data non outlier dibutuhkan suatu batasan yang akan disebut dengan *threshold*.

Pada sistem penentu outlier menggunakan jarak mahalanobis, *threshold* diperoleh dengan menghitung jarak mahalanobis data pelatihan terhadap pusat data pelatihan. Pusat data pelatihan diperoleh dengan menghitung nilai rata-rata data pelatihan. Setelah jarak mahalanobis antara data pelatihan dengan pusat data pelatihan dihitung, maka *threshold* dapat ditentukan dari jarak mahalanobis terbesar antara data pelatihan dengan pusat data pelatihan. Jarak mahalanobis terbesar akan dimiliki oleh titik (data pelatihan) yang berada paling jauh dari titik pusat data, dari titik tersebut dapat dibuat suatu bidang mahalanobis yang berupa ellipsoidal. Bidang ellipsoidal mahalanobis ini yang akan dijadikan *boundary* yang membatasi antara data outlier dengan data yang bukan outlier.



Gambar 3.1 Scater Plot Data

Proses penentuan threshold sebagai batas untuk memisahkan antara outlier dan bukan outlier dilakukan pada saat proses pelatihan pada algoritma propagasi balik, dan sistem ini berjalan parallel dengan proses algoritma propagasi balik. Data yang digunakan untuk menentukan threshold dalam algoritma MDOD adalah data pada proses pelatihan algoritma propagasi balik. Penentuan threshold dalam MDOD dibutuhkan perhitungan kovarian matriks dari sample data pelatihan, dalam hal ini dikatakan sample karena tidak semua data pelatihan dilibatkan dalam proses perhitungan nilai matriks kovarian. Pengambilan sample untuk menghitung matriks kovarian dikarenakan jika diambil semua data dalam perhitungan matriks kovarian maka nilai jarak mahalanobis adalah antara 0 hingga 1. Karena dalam proses perhitungan jarak mahalanobis data yang digunakan telah

dinormalisasi maka jarak titik dari pusat distribusinya tidak melebihi dari nilai variannya. Jika nilai mahalanobis terbesarnya adalah 1 maka thresholdnya adalah 1, sedangkan jika suatu data tidak dilibatkan dalam perhitungan nilai matriks kovarian, maka nilai jarak mahalanobisnya lebih besar dari 1, sehingga data tersebut akan dinyatakan diluar *boundary* atau disebut outlier.

Pada proses ujicoba, data ujicoba tidak dilibatkan dalam proses perhitungan nilai matriks kovarian, sehingga dipastikan semua data uji coba akan dinyatakan sebagai outliers, padahal dalam data uji coba terdapat data yang bukan outlier. Untuk mengatasi masalah ini, maka data pelatihan yang digunakan dalam proses pelatihan algoritma propagasi balik akan dibagi menjadi dua bagian, yaitu sebahagian pertama untuk data sample perhitungan nilai matriks kovarian dan sebahagian yang lain untuk perhitungan threshold.

Secara detil algoritma penentuan threshold dengan MDOD pada penelitian ini adalah sebagai berikut:

1. Diberikan :

- $X_{n \times f}$: data pelatihan
- n : jumlah data training
- f : feature (jumlah dimensi).

2. Hitung nilai rata-rata dari sampel data (\bar{x}) yang diambil dari data pelatihan dengan persamaan 3.6 dan standart deviasi data sampel (s) dengan persamaan 3.7.

$$\bar{x} = \frac{\sum_{i=1}^m \text{sampel}_i}{m} \dots\dots\dots (3.6)$$

$$s = \left(\frac{1}{m} \sum_{i=1}^m (x_i - \bar{x})^2 \right)^{1/2} \dots\dots\dots (3.7)$$

Dimana:

\bar{x} : rata-rata sampel data pelatihan

sample : data sampel dari data pelatihan

m : jumlah data sampel dari data pelatihan

s : standar deviasi sampel data dari data pelatihan.

3. Normalisasi semua data pelatihan dengan menggunakan Z-score yang dapat dinyatakan dalam persamaan 3.8

$$z_i = \frac{x_i - \bar{x}}{s} \dots\dots\dots (3.8)$$

Dimana :

z_i : data yang telah dinormalisasi

s : standart deviasi sampel data pelatihan

x : data pelatihan

\bar{x} : rata-rata sample data pelatihan

4. Hitung kovarian matriks dari data sample dengan persamaan 3.9.

$$cov = \frac{1}{n} \sum_{i=1}^m (z_i)(z_i)^t \dots\dots\dots (3.9)$$

5. Hitung jarak mahalanobis semua data pelatihan yang telah dinormalisasi menggunakan persamaan mahalanobis pada persamaan 3.3.

6. Dapatkan threshold dengan mencari nilai terbesar dari mahalanobis

$$threshold = \max(MDi) \dots\dots\dots (3.10)$$

Setelah threshold diperoleh maka untuk menentukan suatu data uji merupakan outlier atau bukan dapat dilakukan dengan prosedur sebagai berikut:

1. Normalisasi data uji menggunakan Z-score formula yang dinyatakan dalam persamaan 3.8
2. Hitung jarak mahalanobis data uji terhadap distribusi data sampel dari data pelatihan dengan persamaan 3.3.
3. Tentukan tiap-tiap data uji merupakan outlier ataukah bukan outlier dengan persamaan 3.11.

$$outliers = \begin{cases} 1, & MD > threshold \\ 0, & MD \leq threshold \end{cases} \dots\dots\dots (3.11)$$

3.4 Penentuan Outlier pada Algoritma Propagasi Balik Menggunakan *Fuzzy Distance Outliers Determination (FDOD)*

Algoritma penentuan outlier pada propagasi balik menggunakan perhitungan jarak fuzzy yang dalam penelitian ini disebut sebagai *Fuzzy Distance Outlier Determination (FDOD)* yang dalam bahasan selanjutnya dituliskan sebagai FDOD. Pada dasarnya algoritma FDOD memiliki proses yang hampir sama dengan MDOD, perbedaannya adalah FDOD memiliki tahapan proses fuzzifikasi dari nilai jarak dan perhitungan jaraknya dilakukan terhadap pusat data masing-masing kelas. FDOD dalam penelitian ini diadopsi dari proses kerja algoritma *Fuzzy C-Means (FCM)* dalam hal ini hanya proses inialisasi dari FCM yang digunakan, dan FDOD tidak bersifat iterative seperti FCM.

Pada algoritma FCM, inialisasi awal adalah untuk inialisasi bobot yang disebut dengan istilah *membership function*, kemudian dari inialisasi bobot awal dihitung center (pusat cluster) [FRA03]. Proses optimasi bobot terus dilakukan secara iteratif, sehingga proses akan berhenti *objective function*-nya telah lebih

kecil dibandingkan dengan target error yang ditentukan pengguna. Jika proses optimasi telah dipenuhi dapat dilihat dari pusat cluster yang tidak bergeser lagi. *Membership function* dalam FCM menyatakan derajat kemungkinan suatu data pada suatu cluster menjadi bagian dari cluster lainnya.

Dalam FDOD inisialisasi awal dimulai dari inisialisasi pusat kelas awal (*center* awal dari tiap kelas). Inisialisasi awal pusat kelas dari tiap kelas dilakukan dengan menghitung nilai rata-rata data pelatihan tiap kelas. Setelah diperoleh pusat dari tiap kelas, selanjutnya akan dihitung *membership function* dari data, proses perhitungannya mengadopsi persamaan *membership function* yang digunakan dalam FCM. Setelah diperoleh *membership function* akan dilakukan *update center* dengan formula yang juga diadopsi dari FCM. Setelah diperoleh pusat data akhir dari tiap kelas, akan dihitung jarak data dengan pusat data tiap kelas, dari jarak yang diperoleh akan dicari jarak maksimum untuk dijadikan sebagai threshold. Proses perhitungan jarak antara data dengan pusat data tiap kelas yang telah *update* dilakukan dengan dua teknik perhitungan yaitu dengan perhitungan jarak Euclidean dan perhitungan jarak mahalanobis. Untuk lebih detilnya masing-masing teknik perhitungan jarak FDOD akan dibahas pada sub bahasan selanjutnya.

3.4.1 Fuzzy Distance Outlier Determination dengan Perhitungan Jarak Euclidean

Seperti yang telah dibahas sebelumnya, FDOD dalam penelitian ini dilakukan dalam dua teknik perhitungan jarak yang berbeda. Dalam sub bahasan ini akan dibahas secara rinci algoritma FDOD dengan perhitungan jarak Euclidean.

Teknis algoritma FDOD dengan perhitungan jarak Euclidean untuk menentukan *threshold* yang menjadi batas antara outlier dan bukan outlier adalah sebagai berikut:

1. Diberikan data pelatihan $X_{m \times f}$.
2. Inisialisasi *center* awal dari kelas dengan menghitung rata-rata tiap kelas.
3. Hitung *membership function* dengan persamaan 3.12.

$$\mu_{ik} = \frac{\left[\sum_{j=1}^m (x_{jk} - \bar{x}_j)^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[\sum_{j=1}^m (x_{jk} - \bar{x}_j)^2 \right]^{\frac{-1}{w-1}}} \dots\dots\dots (3.12)$$

dimana :

- i : 1,2,...,n; dan $k=1,2,\dots,c$.
- x_{jk} : data kelas ke-k *feature* ke-j
- \bar{x}_j : rata-rata *feature* ke-j
- μ_{jk} : *membership function* ke-k dari data ke-j
- w : derajat kefuzzy-an

4. *Update center* (V) dengan menggunakan persamaan 3.13

$$V_{kj} = \frac{\sum_{i=1}^n (\mu_{ik})^w * X_{ij}}{\sum_{i=1}^n \mu_{ik}} \dots\dots\dots (3.13)$$

dimana :

- X_{ik} : data ke-i kelas ke-k.
- μ_{jk} : *membership function* ke-k dari data ke-j
- w : derajat kefuzzy-an

V_{kj} : center kelas ke-k feature ke-j

5. Hitung jarak Euclidean data pelatihan terhadap *center* kelas menggunakan persamaan 3.5.
6. Dapatkan *threshold* dengan mencari nilai terbesar dari jarak Euclidean.

$$threshold = \max(D_i) \dots\dots\dots (3.14)$$

Setelah *threshold* diperoleh maka untuk menentukan suatu data uji merupakan outlier atau bukan dapat dilakukan dengan prosedur sebagai berikut:

1. Hitung jarak euclidean data uji terhadap distribusi data sampel dari data pelatihan dengan persamaan 3.5.
2. Tentukan tiap-tiap data uji merupakan outlier ataukah bukan outlier dengan persamaan 3.15.

$$outliers = \begin{cases} 1, & D > threshold \\ 0, & D \leq threshold \end{cases} \dots\dots\dots (3.15)$$

3.4.2 Fuzzy Distance Outlier Determination dengan Perhitungan Jarak Mahalanobis

Algoritma FDOD dengan perhitungan jarak mahalanobis, memiliki prosedur yang hampir sama dengan FDOD dengan perhitungan jarak Euclidean, yang membedakan hanya teknik perhitungan jarak antara data dengan pusat kelasnya. Prosedur algoritma FDOD dengan perhitungan jarak mahalanobis untuk menentukan *threshold* yang menjadi batas antara outlier dan bukan outlier pada prosedur ke 1 sampai dengan 4 sama dengan FDOD dengan perhitungan jarak Euclidean, yang berbeda hanya pada point ke-5 yaitu teknis perhitungannya.

Detil prosedur algoritma FDOD menggunakan perhitungan jarak mahalanobis adalah sebagai berikut:

1. Diberikan data pelatihan X_{mxf} .
2. Inisialisasi *center* awal dari kelas dengan emnghitung rata-rata tiap kelas.
3. Hitung *membership function* dengan persamaan 3.12.
4. *Update center* (V) dengan menggunakan persamaan 3.13
5. Hitung jarak mahalanobis data pelatihan terhadap *center* kelas menggunakan persamaan 3.3.
6. Dapatkan *threshold* dengan mencari nilai terbesar dari jarak mahalanobis.

$$threshold = \max(MDi) \dots\dots\dots (3.16)$$

Setelah *threshold* diperoleh maka untuk menentukan suatu data uji merupakan outlier atau bukan dapat dilakukan dengan prosedur sebagai berikut:

1. Hitung jarak mahalanobis data uji terhadap distribusi data sampel dari data pelatihan dengan persamaan 3.3.
2. Tentukan tiap-tiap data uji merupakan outlier ataukah bukan outlier dengan persamaan 3.17.

$$outliers = \begin{cases} 1, & MD > threshold \\ 0, & MD \leq threshold \end{cases} \dots\dots\dots (3.17)$$

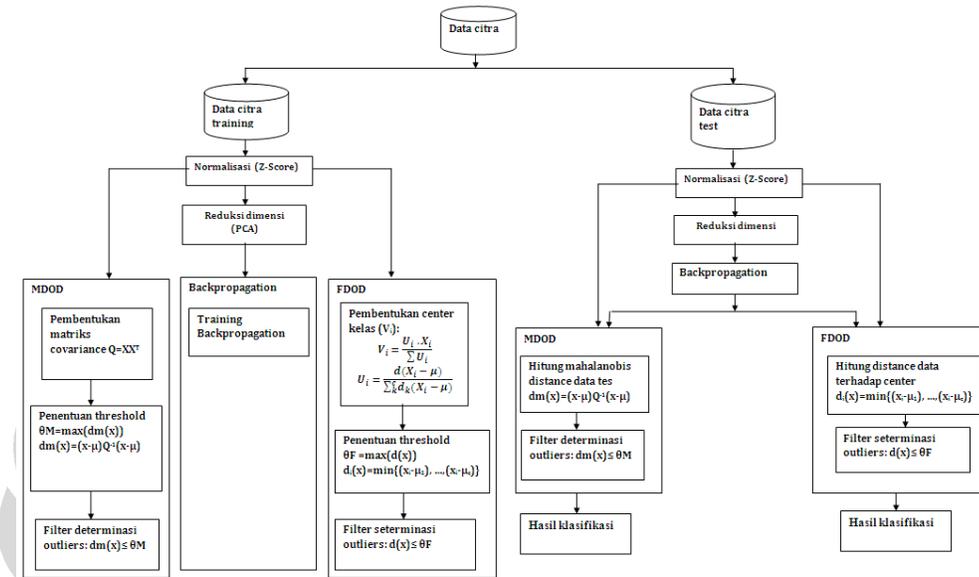
Untuk melihat membandingkan akurasi algoritma MDOD dan FDOD akan dibahas pada sub bahasan selanjutnya, yang membahas tentang hasil eksperimen yang dilakukan dengan data set pada tabel 2.1.

3.5 Hasil Percobaan Sistem Penentu Outlier pada Algoritma Propagasi Balik dengan MDOD dan FDOD

Percobaan sistem penentuan outlier pada algoritma propagasi balik diaplikasikan pada sistem pengenalan wajah 3D dengan scenario *un-kown pose* dan data set seperti yang ditampilkan pada tabel 2.1. Dari percobaan yang dilakukan pada sistem klasifikasi dengan algoritma propagasi balik yang dibahas pada Bab II terlihat bahwa akurasi algoritma propagasi balik menurun drastis, sebanding dengan jumlah data outliers yang dijadikan sebagai masukan pada saat ujicoba. Sehingga dalam penelitian ini dikembangkan metode untuk meningkatkan akurasi algoritma propagasi balik walaupun data uji yang dimasukkan meliputi outlier. Percobaan pertama yang dilakukan dalam sistem penentu outlier, menggunakan data citra asli (data masih berada dalam ruang *image*, belum ditransformasi ke ruang *eigen*).

Untuk menghitung akurasi penentu outlier dengan algoritma MDOD maupun FDOD akan digunakan FAR, FRR, dan ACC. FAR (*false accept rate*) adalah kemungkinan suatu sistem salah dalam mendeterminasikan data outliers, artinya data outliers dinyatakan sebagai bukan outliers. FRR (*false reject rate*) adalah kemungkinan sistem salah dalam mendeterminasikan data bukan outliers dinyatakan sebagai data outlier. Sedangkan ACC (*accuracy*) adalah kemungkinan benar sistem melakukan determinasi terhadap data outlier maupun bukan outlier, atau secara matematis dapat dinyatakan ($ACC=100-(FAR+FRR)$).

Rancangan sistem pada percobaan ini dapat dilihat pada gambar 3.2., percobaan ini dilakukan dalam ruang *image*, jadi dimensi data masih belum direduksi dan belum ditransformasi ke ruang eigen.



Gambar 3.2 Diagram rancangan sistem percobaan 1

Hasil akurasi sistem MDOD dalam menentukan outlier dapat dilihat pada tabel 3.1 demikian juga hasil akurasi FDOD Euclidean distance dan FDOD mahalanobis distance. Percobaan yang ditampilkan dalam tabel 3.1 dilakukan dengan data set pada tabel 2.1.

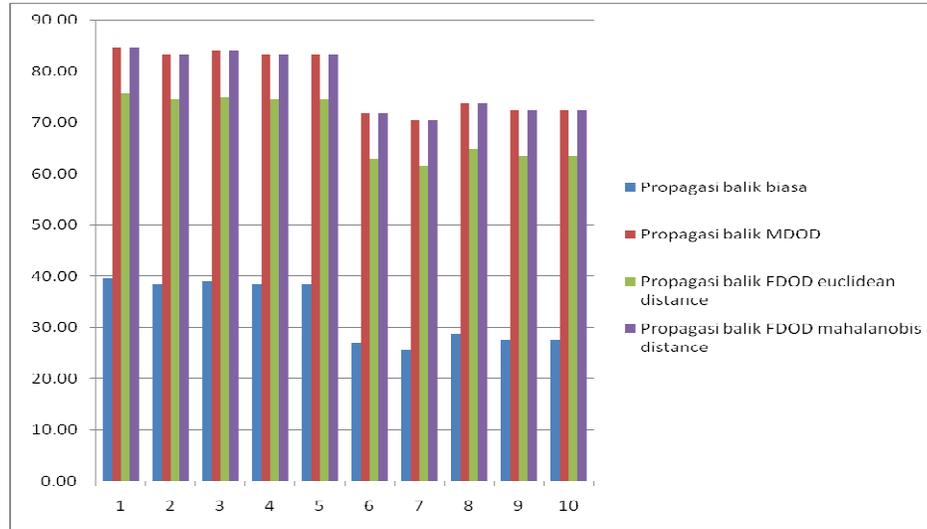
Tabel 3.1 Tabel Kemampuan Sistem Penentu outlier

Metode	FAR (%)	FRR(%)	ACC(%)
MDOD	14.0	3.3	82.6
FDOD euclidean distance	28.9	0.0	71.1
FDOD mahalanobis distance	0.0	24.8	75.2

Tabel 3.1 menunjukkan hasil percobaan dengan data set Tabel 2.1, kolom FAR menunjukkan prosentase banyaknya outlier yang dikenali sebagai bukan outlier, pada kolom ini, hasil terbaik dihasilkan oleh FDOD yang menggunakan perhitungan jarak mahalanobis. Kolom FRR menunjukan prosentase data bukan outlier yang dikenali sebagai outlier, pada kolom ini hasil terbaik dihasilkan oleh FDOD yang menggunakan perhitungan jarak Euclidean. Sedangkan accuracy (ACC) menunjukkan prosentase total tingkat pengenalan sistem penentu outlier, pada percobaan ini MDOD memiliki tingkat akurasi yang cukup tinggi dibandingkan dengan FDOD yang menggunakan perhitungan Euclidean ataupun dengan FDOD yang menggunakan perhitungan mahalanobis. Efektifitas algoritma propagasi balik dalam meningkatkan akurasi pengenalan algoritma propagasi balik jika data test meliputi data outliers, dapat dilihat dalam tabel 3.3 dan 3.4.

Tabel 3.2 Pengaruh Sistem penentu Outlier terhadap Akurasi Pengenalan Algoritma Propagasi Balik

Jumlah Dimensi	Akurasi Data test All(meliputi outliers)			
	Propagasi balik biasa	Propagasi balik MDOD	Propagasi balik FDOD euclidean distance	Propagasi balik FDOD mahalanobis distance
10	39.74	84.62	75.64	84.62
15	38.46	83.33	74.36	83.33
20	39.10	83.97	75.00	83.97
25	38.46	83.33	74.36	83.33
30	38.46	83.33	74.36	83.33
35	26.92	71.79	62.82	71.79
40	25.64	70.51	61.54	70.51
45	28.85	73.72	64.74	73.72
50	27.56	72.44	63.46	72.44
55	27.56	72.44	63.46	72.44



Gambar 3.3 Grafik Akurasi Pengenalan Algoritma Propagasi balik

Pada gambar 3.3 dan tabel 3.2 dapat diamati nilai akurasi pengenalan algoritma propagasi balik yang meliputi data uji outlier, akurasi pengenalan dengan sistem penentuan outlier dapat meningkatkan akurasi pengenalan data uji lebih dari dua kali akurasi pengenalan propagasi balik yang biasa. Dalam percobaan ini akurasi tertinggi diperoleh dari penggabungan FDOD mahalanobis dengan algoritma propagasi balik dan MDOD dengan algoritma propagasi balik. Akurasi tertinggi terjadi pada data yang direduksi dimensinya menjadi 10, dengan akurasi pengenalan propagasi balik sebelum dimodifikasi sebesar 39.74% menjadi 84.62% pada MDOD dan 84.62% pada FDOD mahalanobis.

3. 6 Kesimpulan

Dari hasil percobaan sementara dapat disimpulkan bahwa:

1. Tingkat akurasi pengenalan algoritma propagasi balik dapat ditingkatkan dengan cukup signifikan hingga 2 kali lipat dengan adanya pengembangan sistem penentu outlier pada algoritma propagasi balik untuk data test meliputi data outlier dan bukan outlier.

2. Untuk percobaan menggunakan data didalam ruang citra, diperoleh hasil bahwa MDOD dan FDOD mahalnobis memiliki akurasi yang lebih tinggi dibandingkan dengan FDOD menggunakan jarak Euclidean, tetapi tentunya dengan dimensi yang tinggi maka biaya komputasinya juga lebih besar. Jika bekerja di ruang citra maka dimensi yang bekerja sebesar 1024, sedangkan jika bekera di ruang eigen, dimensi data akan direduksi menjadi 1% hingga 5% dari dimensi data pada ruang citra sehingga waktu yang dibutuhkan untuk proses pelatihan maupun pengenalan juga akan berpengaruh lebih cepat jika dimensinya direduksi.

Sistem penentu outlier dapat bekerja dengan cukup baik pada ruang citra asli dalam peningkatkan akurasi algoritma propagasi balik, pada sistem data uji meliputi data outliers dan bukan outliers. Karena sistem bekerja dalam ruang citra asli, maka dimensi data masih belum direduksi sehingga biaya komputasi akan cukup besar, untuk itu akan dilakukan evaluasi akurasi terhadap reduksi dimensi. Proses evaluasi ini akan dilakukan dengan ujicoba sistem penentu outlier pada algoritma propagasi balik terhadap data yang telah direduksi dimensinya meggunakan PCA. Untuk lebih jelasnya akan di bahas pada Bab IV.

BAB IV

EVALUASI PENGARUH SISTEM OUTLIERS DETERMINATION DALAM ALGORITMA PROPAGASI BALIK

Ujicoba menggunakan data yang telah direduksi dimensinya dengan PCA menggunakan beberapa skenario percobaan, analisa hasil percobaan dan evaluasi serta kinerja sistem akan dibahas dalam Bab ini.

Pada Bab II dan Bab III telah dibahas bahwa algoritma propagasi balik memiliki tingkat akurasi yang cukup bagus dalam proses klasifikasi pada sistem pengenalan wajah 3D dengan kondisi data uji tidak meliputi data outlier. Untuk klasifikasi dengan data uji yang meliputi data outlier dan bukan data outlier, algoritma propagasi balik yang biasa memiliki akurasi yang cukup rendah, tingkat akurasinya masih dibawah 50%. Akurasi algoritma propagasi balik menjadi naik cukup signifikan dengan dikembangkannya sistem penentu outlier pada algoritma propagasi balik. Untuk mengetahui kinerja sistem penentu outlier pada algoritma propagasi balik akan dilakukan ujicoba lanjutan dengan variasi dimensi, variasi teknik reduksi dengan PCA dan variasi data set.

Percobaan ini dilakukan pada data wajah orang Indonesia (data wajah JST) yang terdiri dari 12 data wajah orang dengan ekspresi normal dan sudut mulai 0° hingga 180° dengan perseggaran sudut sebesar 15° . Data yang dilatih adalah data wajah 5 orang, sedangkan data outliernya adalah data wajah 7 orang yang lain. Data set eksperimen ini dapat dilihat pada tabel 4.1 dan contoh data yang digunakan dapat dilihat pada tabel 4.2.

Tabel 4.1. Tabel Data Set Percobaan

Data Set	Data Train (°)	Data Test (°)		Perbandingan (%)	
		DataTest	Data Test Ouliers	data train : DataTes	data Train : Data Test All*
1	0, 45, 90, 135, 180	15, 30, 60, 75, 105, 120, 150, 165	0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180	39: 61	16 : 84
	(5 data citra wajah JST lama)	(5 data citra wajah JST lama)	(3 data citra wajah JST baru + 4 data citra wajah JST lama)		
2	0, 30, 60,90, 120, 150, 180	15, 45, 75, 105, 135, 165	0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180	54 : 46	29 : 71
	(5 data citra wajah JST lama)	(5 data citra wajah JST lama)	(3 data citra wajah JST baru + 4 data citra wajah JST lama)		
3	0, 60, 120, 180	15, 30, 45, 75, 90, 105, 135, 150, 165	0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180	31: 69	13:87
	(5 data citra wajah JST lama)	(5 data citra wajah JST lama)	(3 data citra wajah JST baru + 4 data citra wajah JST lama)		

Tabel 4.2. Contoh Data Yang Digunakan dalam Percobaan

Ket	Citra wajah orang dari data JST dengan ekspresi normal sudut 0°						
Data Uji							
Data Outliers							

Dari data set pada tabel 4.1 sistem akan diuji coba pada data dengan skenario *unkown pose* dimana data akan dilatih pada sudut-sudut tertentu dan akan diuji coba pada data uji yang tidak meliputi data uji outlier dengan sudut antara data-data yang dilatih dengan pergeseran sudut 15° . Pada data set 1, data pelatihan menggunakan sudut dengan pergeseran 45° , sedangkan data pelatihannya menggunakan sudut antara data pelatihan. Pada data set 2 selisih sudut pada data pelatihan lebih besar yaitu 30° . Dan data set 3 selisihnya lebih besar lagi yaitu 60° sehingga data uji lebih banyak jumlahnya. Percobaan ini akan dilakukan dengan data pelatihan yang memiliki selisih sudut yang berbeda, sehingga akan dianalisa kemampuan sistem dalam mengenali wajah dengan sudut pandang yang berbeda-beda.

Sistem penentu outlier pada algoritma propagasi balik akan diimplementasikan pada prosesor Intel Core 2 Duo (1.66GHz, 667 MHz FSB, 2MB L2 cache) dengan memory 1.5 GB serta menggunakan Matlab 7.1 release 14. Karena dimensi yang cukup tinggi maka terlebih dahulu data citra masukan akan direduksi dimensinya dengan menggunakan algoritma PCA.

Pemotongan dimensi dilakukan berdasarkan prosentase dari komponen utama. Prosentase komponen utama dihitung dari nilai eigen dibagi dengan jumlah total nilai eigen data train. Hasil transformasi balik citra wajah data pelatihan setelah direduksi dimensinya dapat dilihat pada gambar 4.1



Gambar 4.1 Gambar data citra wajah train setelah ditransformasi balik(berurut dari kiri ke kanan, direduksi dengan dimensi 10, 15, 20, 25, 30, 35, 40, 45, 50, 55) dan data citra wajah yang masih asli.

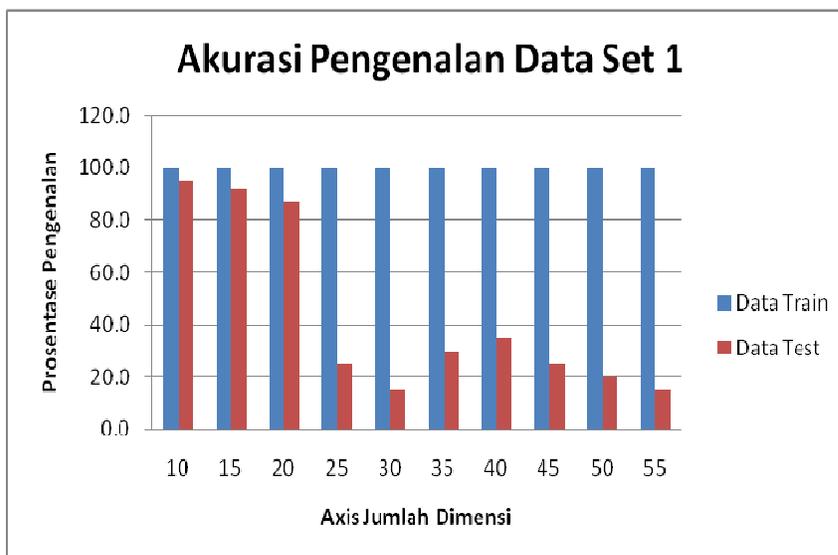
Gambar yang digunakan sebagai sampel hasil transformasi balik dari tari ruang eigen ke ruang *image* pada gambar 4.1 adalah data wajah pelatihan dengan sudut pandang 90°. Dari gambar terlihat bahwa secara kasat mata hasil transformasinya terlihat berbeda dengan gambar data wajah aslinya, tetapi polanya masih sama.

4.1. Pengaruh Reduksi Dimensi Pada Algoritma Propagasi Balik (konvensional)

Setelah data direduksi maka akan dilakukan evaluasi terhadap kinerja algoritma propagasi balik konvensional. Algoritma propagasi balik konvensional adalah algoritma propagasi balik sebelum dimodifikasi untuk mengatasi penentu outlier. Percobaan dengan menggunakan data set 1 dapat dilihat pada tabel 4.3 dan gambar 4.3.

Tabel 4.3. Akurasi Pengenalan Algoritma Propagasi balik pada Data Set 1

Jumlah Dimensi	Akurasi (%)	
	Data Train	Data Test
10	100.0	95.0
15	100.0	92.5
20	100.0	87.5
25	100.0	25.0
30	100.0	15.0
35	100.0	30.0
40	100.0	35.0
45	100.0	25.0
50	100.0	20.0
55	100.0	15.0



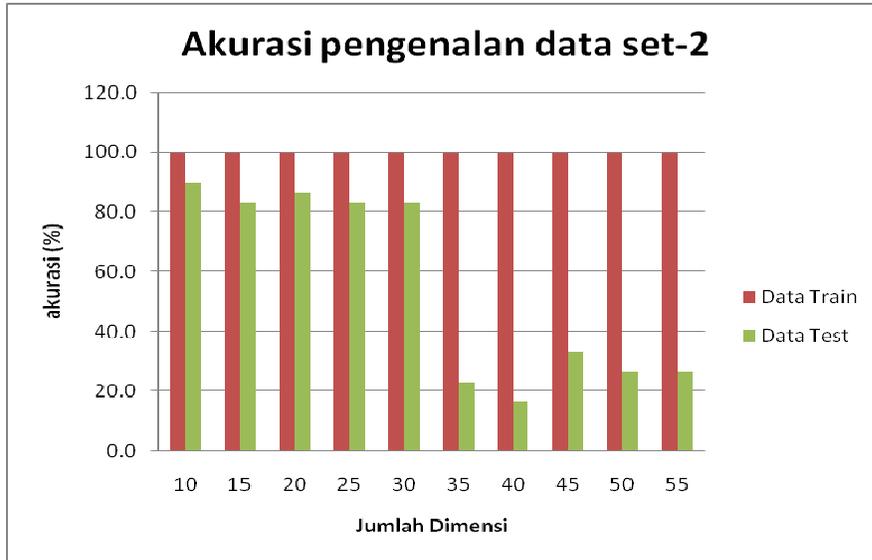
Gambar 4.2 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-1

Pada tabel 4.3 dan gambar grafik 4.2 terlihat bahwa nilai akurasi pengenalan algoritma propagasi balik cukup baik pada dimensi dibawah 25. Nilai akurasi data uji terbaik dihasilkan pada ujicoba dengan reduksi dimensi dari berdimensi 1024 menjadi berdimensi 10.

Percobaan dengan data set 2 dapat dilihat pada tabel 4.4 dan gambar 4.3.

Tabel 4.4 . Akurasi Pengenalan Algoritma Propagasi balik pada Data Set 2

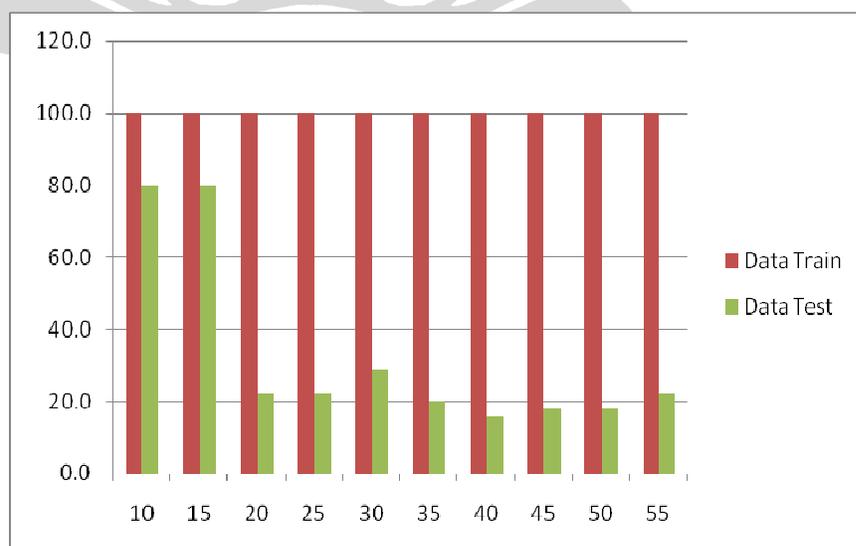
Jumlah Dimensi	Akurasi (%)	
	Data Train	Data Test
10	100.0	90
15	100.0	83.3
20	100.0	86.7
25	100.0	83.3
30	100.0	83.3
35	100.0	23.3
40	100.0	16.7
45	100.0	33.3
50	100.0	26.7
55	100.0	26.7



Gambar 4.3 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-2

Jika pada data set 1 nilai akurasi menurun pada dimensi diatas dimensi 20, pada data set 2 nilai akurasi masih bagus pada dimensi 30, pada dimesi 35 akurasi menurun drastis. Dari tabel 4.4 dan gambar 4.3 akurasi terbaik dari percobaab dengan data set 2 dihasilkan pada percobaan dengan dimensi 10 sama hanya dengan data set 1.

Percobaan dengan data set 3 dapat dilihat pada tabel 4.5 dan gambar 4.5.



Gambar 4.4 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-3

Tabel 4.5 . Akurasi Pengenalan Algoritma Propagasi balik pada Data Set 3

Jumlah Dimensi	Akurasi (%)	
	Data Train	Data Test
10	100.0	80.0
15	100.0	80.0
20	100.0	22.2
25	100.0	22.2
30	100.0	28.9
35	100.0	20.0
40	100.0	15.6
45	100.0	17.8
50	100.0	17.8
55	100.0	22.2

Jika pada data set 1 nilai akurasi menurun pada dimensi diatas dimensi 20, dan pada data set 2 nilai akurasi menurun pada dimensi 30, pada data set 3 akurasi menurun pada dimensi diatas dimensi 18. Dari tabel 4.5 dan gambar 4.4 akurasi terbaik dari percobaan dengan data set 3 dihasilkan pada percobaan dengan dimensi 10 sama hanya dengan data set 1 dan data set 2

Dari grafik dan tabel hasil ujicoba dengan data set-1, data set-2 maupun data set 3 menunjukkan bahwa semakin besar dimensi data tidak membuat akurasi sistem pengenalan wajah 3D menjadi lebih bagus, tetapi sebaliknya prosentase akurasi menurun pada batas dimensi tertentu, hal ini karena pada dimensi lebih tinggi kemungkinan redundansi informasi akan terjadi semakin besar sehingga memungkinkan informasi penting dari data menjadi kabur.

Sistem pengenalan wajah 3D memiliki hasil akurasi tertinggi pada dimensi 10. Hal ini dikarenakan informasi yang diberikan oleh data benar-benar informasi yang penting. Batas penurunan akurasi pada masing-masing data set tidak sama karena setiap data set memiliki pola yang berbeda, sehingga mempengaruhi

matriks transformasi dari ruang citra ke ruang eigen yang akhirnya juga berpengaruh terhadap hasil reduksi dimensi.

4.2. Pengaruh Reduksi Dimensi Pada Algoritma *Mahalanobis Distance Outliers Determination (MDOD)*

Pada sub bab ini akan ditunjukkan hasil percobaan yang dilakukan pada algoritma MDOD dengan variasi dimensi. Percobaan dengan data set-1 dapat dilihat pada tabel 4.6, data set-2 pada tabel 4.7 dan data set-3 pada tabel 4.8.

Tabel 4.6. Kinerja sistem MDOD dalam mendeteksi outliers pada Data Set -1

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	3.05	16.03	80.92
PCA tanpa reduksi	30.53	0.00	69.47
PCA 55	30.53	0.00	69.47
PCA 50	30.53	0.00	69.47
PCA 45	30.53	0.00	69.47
PCA 40	30.53	0.00	69.47
PCA 35	30.53	0.00	69.47
PCA 30	30.53	0.00	69.47
PCA 25	30.53	0.00	69.47
PCA 20	0.00	30.53	69.47
PCA 15	0.00	31.30	68.70
PCA 10	0.00	30.53	69.47

Tabel 4.7. Kinerja sistem MDOD dalam mendeteksi outliers pada Data Set -2

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	0.83	16.53	82.64
PCA tanpa reduksi	24.79	0.00	75.21
PCA 55	24.79	0.00	75.21
PCA 50	24.79	0.00	75.21
PCA 45	24.79	0.00	75.21
PCA 40	24.79	0.00	75.21
PCA 35	24.79	0.00	75.21
PCA 30	0.00	29.75	70.25
PCA 25	0.00	30.58	69.42
PCA 20	0.00	31.40	68.60
PCA 15	0.00	26.45	73.55
PCA 10	0.83	24.79	74.38

Tabel 4.8. Kinerja sistem MDOD dalam mendeteksi outliers pada Data Set -3

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	2.94	11.76	85.29
PCA tanpa reduksi	33.09	0.00	66.91
PCA 55	33.09	0.00	66.91
PCA 50	33.09	0.00	66.91
PCA 45	33.09	0.00	66.91
PCA 40	33.09	0.00	66.91
PCA 35	33.09	0.00	66.91
PCA 30	33.09	0.00	66.91
PCA 25	33.09	0.00	66.91
PCA 20	33.09	0.00	66.91
PCA 15	0.00	27.94	72.06
PCA 10	0.00	27.94	72.06

Dari hasil percobaan dapat dilihat bahwa tingkat akurasi MDOD akan lebih bagus jika sistem bekerja pada ruang image. Ketika data telah ditransformasi ke ruang eigen, hasil akurasi algoritma MDOD menjadi menurun. Hasil akurasi tertinggi pada data set-1 adalah 80.92% dan akurasi tertinggi pada data set ke-2 adalah 82.64% serta akurasi tertinggi pada data set ke-3 adalah 85.29%.

4.3. Pengaruh Reduksi Dimensi Pada Algoritma *Fuzzy Distance Outliers Determination (FDOD)*

Algoritma FDOD memiliki dua versi yaitu FDOD Euclidean distance dan FDOD Mahalanobis Distance.

a. *Fuzzy Distance Outliers Determination (FDOD) Euclidean Distance*

Algoritma FDOD Euclidean distance menggunakan formula perhitungan jarak euclidean untuk menentukan threshold dan determinasi outliersnya. Hasil percobaan FDOD Euclidean distance dapat dilihat pada Tabel 4.9, Tabel 4.10 dan Tabel 4.11.

Tabel 4.9. Kinerja sistem FDOD Euclidean Distance dalam mendeteksi outliers pada Data Set -1

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	0.00	26.72	73.28
PCA tanpa reduksi	0.00	16.79	83.21
PCA 55	0.00	21.37	78.63
PCA 50	0.00	21.37	78.63
PCA 45	0.00	21.37	78.63
PCA 40	0.00	21.37	78.63
PCA 35	0.00	21.37	78.63
PCA 30	0.00	21.37	78.63
PCA 25	0.00	21.37	78.63
PCA 20	0.00	22.14	77.86
PCA 15	0.00	22.14	77.86
PCA 10	0.00	22.14	77.86

Tabel 4.10. Kinerja sistem FDOD Euclidean Distance dalam mendeteksi outliers pada Data Set -2

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	0.00	28.93	71.07
PCA tanpa reduksi	0.00	17.36	82.64
PCA 55	0.00	23.14	76.86
PCA 50	0.00	23.14	76.86
PCA 45	0.00	23.14	76.86
PCA 40	0.00	23.14	76.86
PCA 35	0.00	23.14	76.86
PCA 30	0.00	23.14	76.86
PCA 25	0.00	23.14	76.86
PCA 20	0.00	23.14	76.86
PCA 15	0.00	23.14	76.86
PCA 10	0.00	23.14	76.86

Tabel 4.11. Kinerja sistem FDOD Euclidean Distance dalam mendeteksi outliers pada Data Set -3

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	0.00	25.74	74.26
PCA tanpa reduksi	0.00	15.44	84.56
PCA 55	0.00	19.12	80.88
PCA 50	0.00	19.12	80.88
PCA 45	0.00	19.12	80.88
PCA 40	0.00	19.12	80.88
PCA 35	0.00	19.12	80.88
PCA 30	0.00	19.12	80.88
PCA 25	0.00	19.12	80.88
PCA 20	0.00	19.12	80.88
PCA 15	0.00	19.12	80.88
PCA 10	0.00	19.12	80.88

Dari tabel hasil ujicoba data set 1, data set 2, dan data set 3 FDOD *Euclidean distance* menunjukkan akurasi tertinggi berada pada percobaan yang dilakukan dengan data citra telah ditransformasi ke ruang eigen, tetapi dimensi data tidak direduksi. Secara Umum FDOD *euclidean distance* lebih unggul dibandingkan dengan MDOD karena FDOD memiliki nilai akurasi yang lebih stabil. Untuk data set 1 akurasi tertinggi sebesar 83.21%, sedangkan akurasi terendahnya adalah 73.28%., dan untuk data set 2 akurasi tertinggi sebesar 82.64% sedangkan akurasi terendahnya 71.07% serta data set 3 memiliki akurasi tertinggi sebesar 84.56% dan akurasi terendah sebesar 74.26%.

b. Fuzzy Distance Outliers Determination (FDOD) Mahalanobis Distance

Algoritma FDOD mahalanobis distance menggunakan formula perhitungan jarak mahalanobis untuk menentukan threshold dan determinasi outliersnya. Hasil percobaan FDOD mahalanobis distance dapat dilihat pada Tabel 4.12, Tabel 4.13 dan Tabel 4.14.

Tabel 4.12. Kinerja sistem FDOD Mahalanobis Distance dalam mendeteksi Outliers pada Data Set -1

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	4.58	16.03	79.39
PCA tanpa reduksi	30.53	0.00	69.47
PCA 55	30.53	0.00	69.47
PCA 50	30.53	0.00	69.47
PCA 45	30.53	0.00	69.47
PCA 40	30.53	0.00	69.47
PCA 35	30.53	0.00	69.47
PCA 30	30.53	0.00	69.47
PCA 25	30.53	0.00	69.47
PCA 20	0.00	29.77	70.23
PCA 15	0.00	31.30	68.70
PCA 10	0.00	30.53	69.47

Tabel 4.13. Kinerja sistem FDOD Mahalanobis Distance dalam mendeteksi outliers pada Data Set -2

Data	FAR (%)	FRR(%)	ACC(%)
NON PCA	0.83	16.53	82.64
PCA tanpa reduksi	24.79	0.00	75.21
PCA 55	24.79	0.00	75.21
PCA 50	24.79	0.00	75.21
PCA 45	24.79	0.00	75.21
PCA 40	24.79	0.00	75.21
PCA 35	24.79	0.00	75.21
PCA 30	24.79	0.00	75.21
PCA 25	24.79	0.00	75.21
PCA 20	16.53	1.65	81.82
PCA 15	11.57	8.26	80.17
PCA 10	6.61	10.74	82.64

Tabel 4.14. Kinerja sistem FDOD Mahalanobis Distance dalam mendeteksi outliers pada Data Set -3

Data	FAR (%)	FRR(%)	ACC(%)
------	---------	--------	--------

NON PCA	2.94	11.03	86.03
PCA tanpa reduksi	33.09	0.00	66.91
PCA 55	33.09	0.00	66.91
PCA 50	33.09	0.00	66.91
PCA 45	33.09	0.00	66.91
PCA 40	33.09	0.00	66.91
PCA 35	33.09	0.00	66.91
PCA 30	33.09	0.00	66.91
PCA 25	33.09	0.00	66.91
PCA 20	33.09	0.00	66.91
PCA 15	0.00	27.94	72.06
PCA 10	0.00	27.94	72.06

Hail percobaan dengan algoritma FDOD mahalanobis distance lebih variatif, untuk data yang direduksi dibawah 25 dimensi, menunjukkan kenaikan prosentase akurasi, namun hasil akurasi tertinggi untuk data set 1 dan data set 3 berada pada data citra asli yaitu data belum ditransformasi ke ruang eigen. Sedangkan data set 2 menunjukkan akurasi terbesar ada pada data yang telah direduksi dimensinya pada dimensi 10 dan pada data yang masih berada pada ruang citra asli. Untuk data set 1 akurasi tertinggi sebesar 79.39%, sedangkan akurasi terendahnya adalah 69.47%., dan untuk data set 2 akurasi tertinggi sebesar 82.64% sedangkan akurasi terendahnya 75.21% serta data set 3 memiliki akurasi tertinggi sebesar 86.03% dan akurasi terendah sebesar 66.91%.

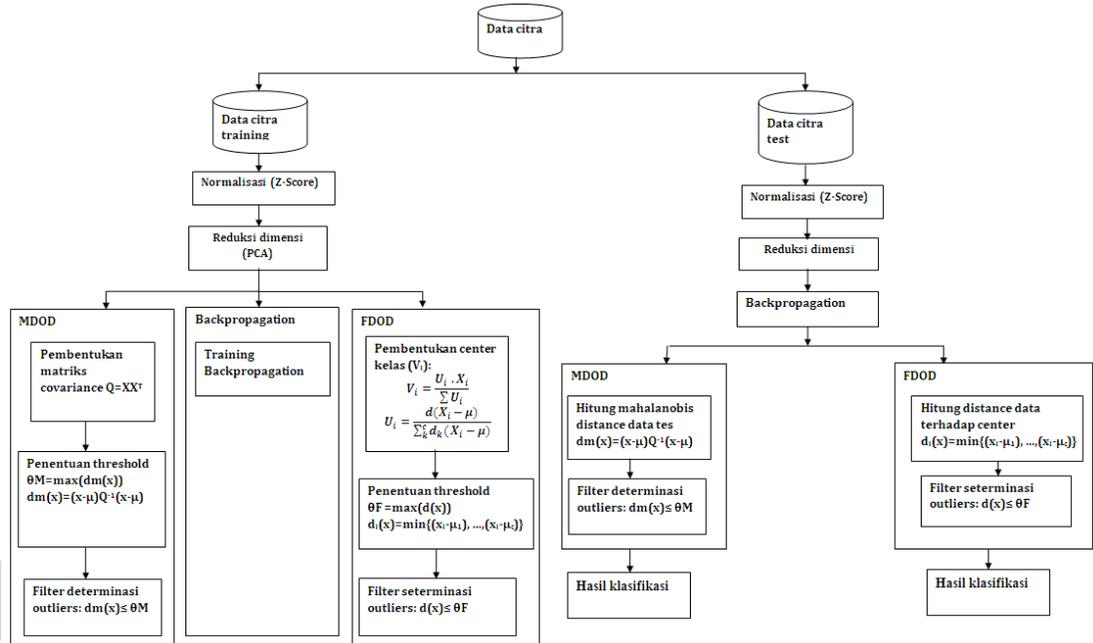
Percobaan dengan data set 1, data set 2, data set 3 dengan MDOD memiliki akurasi rata-rata terbaik pada percobaan yang menggunakan data asli, yaitu data yang masih bearada pada ruang citra asli dan belum ditransformasi ke ruang eigen, dengan akurasi 82.95%, sedangkan akurasi rata-rata terendahnya pada data yang telah direduksi dimensinya menjadi berdimensi 25 dengan nilai akurasi sebesar

68.32%. Pada FDOD *euclidean distance* akurasi rata-rata dari data set 1, data set 2, dan data set 3 terjadi pada percobaan yang bekerja pada ruang eigen tetapi jumlah dimensinya tidak direduksi, dengan akurasi sebesar 83.47% sedangkan akurasi terendahnya sebesar 78.53% terjadi pada percobaan dengan data yang telah direduksi dimensinya hingga menjadi berdimensi 10. Akurasi rata-rata percobaan data set 1, data set 2, data set 3, pada FDOD *mahalanobis distance* memiliki akurasi tertinggi pada percobaan yang bekerja pada ruang citra asli, dengan akurasi sebesar 82.69% dan akurasi terendah sebesar 70.53% pada data yang telah direduksi diatas 20. Secara keseluruhan dari percobaan dapat disimpulkan bahwa FDOD *euclidean distance* memiliki nilai akurasi rata-rata tertinggi dan memiliki nilai variasi akurasi yang cukup kecil, sehingga untuk kondisi data yang direduksi ataupun tidak dan data dalam ruang citra maupun ruang eigen hasil akurasinya masih bagus karena pada FDOD *euclidean distance* akurasi terendahnya adalah 78.53%.

4.4. Evaluasi Mahalanobis Distance Outliers Determination (MDOD) dan Fuzzy Distance Outliers Determination (FDOD) terhadap Kinerja algoritma Propagasi balik

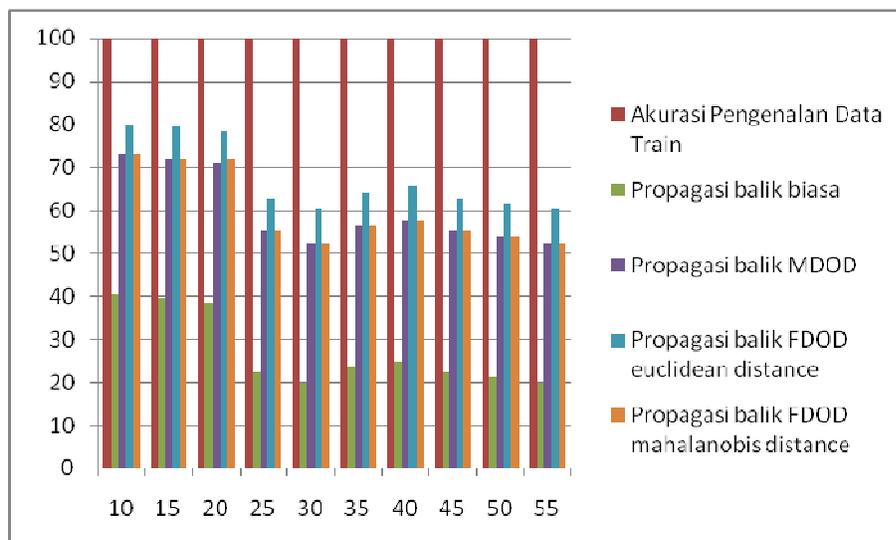
Sistem penentu outlier yang dikembangkan untuk menghilangkan kelemahan algoritma propagasi balik dalam melakukan determinasi outliers, akan diuji pada data set yang dapat dilihat detailnya pada tabel 4.1. Untuk ujicobanya dilakukan dua ujicoba, ujicoba pertama dilakukan dengan scenario percobaan sebagai berikut, data masukan pada algoritma propagasi balik langsung digunakan sebagai masukan dalam sistem outlier determination. Jadi data masukan pada sistem

penentu outlier sama dengan data masukan pada propagasi balik. Gambaran rancangan sistem pada proses ujicoba pertama dapat dilihat pada gambar 4.5.



Gambar 4.5 Rancangan sistem ujicoba ke-1

Hasil ujicoba dari rancangan sistem ujicoba ke-1 data set-1 dapat dilihat pada tabel 4.15 dan gambar grafik pada gambar 4.8.



Gambar 4.6 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-1

Tabel 4.15. Kinerja Sistem Penentu outlier pada Algoritma Propagasi balik menggunakan Data Set -1

jumlah Dimensi	Akurasi Pengenalan Data Train	Akurasi Propagasi balik pada data test yang meliputi outliers			
		propagasi balik konvensional	MDOD Propagasi balik	FDOD Euclidean distance Propagasi balik	FDOD mahalanobis distance propagasi balik
10	100	40.38	73.08	80.13	73.08
15	100	39.74	71.79	79.49	71.79
20	100	38.46	71.15	78.21	71.79
25	100	22.44	55.13	62.82	55.13
30	100	19.87	52.56	60.26	52.56
35	100	23.72	56.41	64.10	56.41
40	100	25.00	57.69	65.38	57.69
45	100	22.44	55.13	62.82	55.13
50	100	21.15	53.85	61.54	53.85
55	100	19.87	52.56	60.26	52.56

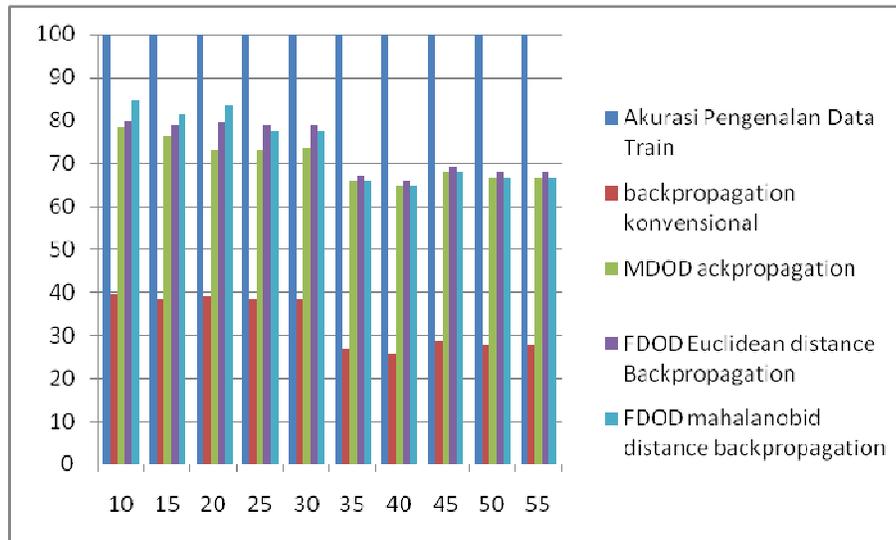
Dari hasil percobaan pada data set 1 dapat dilihat bahwa sistem penentu outlier mampu meningkatkan akurasi pengenalan algoritma propagasi balik. Tingkat peningkatan akurasinya cukup signifikan, hingga mencapai dua kali lipat.

Pada data set-1 dengan sekenario ujicoba ke-1 akurasi tertinggi pada FDOD Euclidean penentu outlier pada algoritma propagasi balik, yaitu 80.13%. Nilai ini meningkat dari akurasi 40.38% algoritma propagasi balik konvensional menjadi 80.13%. Pada ujicoba ini, hasil akurasi tertinggi pada data yang telah direduksi dari berdimensi 1024 menjadi berdimensi 10.

Hasil ujicoba dari rancangan sistem ujicoba ke-1 data set-2 dapat dilihat pada tabel 4.16 dan gambar grafik pada gambar 4.9.

Tabel 4.16. Kinerja Sistem Penentu outlier pada Algoritma Propagasi balik menggunakan Data Set -2

jumlah Dimensi	Akurasi Pengenalan Data Train	Akurasi Propagasi balik pada data test yang meliputi outliers			
		propagasi balik konvensional	MDOD Propagasi balik	FDOD Euclidean distance Propagasi balik	FDOD mahalnobid distance propagasi balik
10	100	39.74	78.21	80.13	84.62
15	100	38.46	76.28	78.85	81.41
20	100	39.10	73.08	79.49	83.33
25	100	38.46	73.08	78.85	77.56
30	100	38.46	73.72	78.85	77.56
35	100	26.92	66.03	67.31	66.03
40	100	25.64	64.74	66.03	64.74
45	100	28.85	67.95	69.23	67.95
50	100	27.56	66.67	67.95	66.67
55	100	27.56	66.67	67.95	66.67



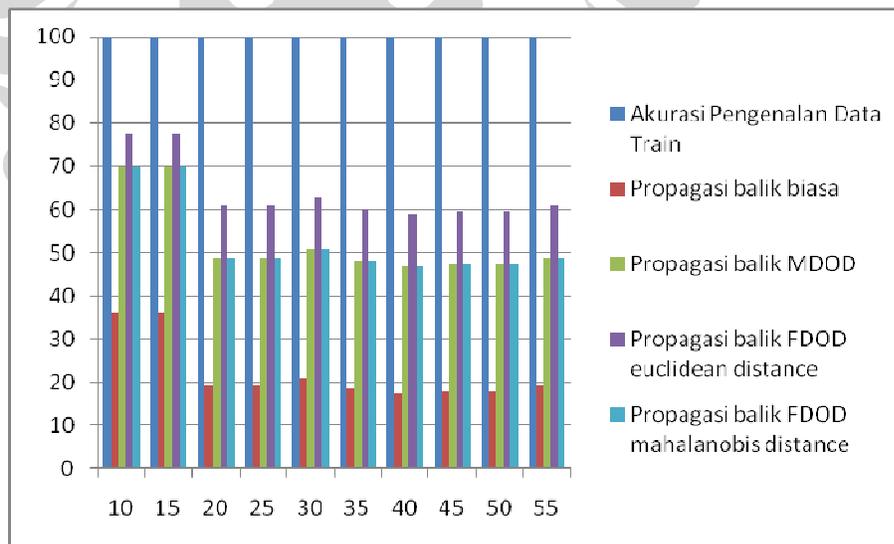
Gambar 4.7 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-2

Dari hasil percobaan pada data set 2 dengan sekenario ujicoba ke-1 akurasi tertinggi pada FDOD mahalnobis distance penentu outlier pada algoritma propagasi balik, yaitu 84.62%. Nilai ini meningkat dari akurasi 39.74% algoritma propagasi balik konvensional menjadi 84.62%. Pada ujicoba ini, hasil akurasi tertinggi pada data yang telah direduksi dari berdimensi 1024 menjadi berdimensi 10.

Hasil ujicoba dari rancangan sistem ujicoba ke-1 data set 3 dapat dilihat pada tabel 4.17 dan gambar grafik pada gambar 4.10.

Tabel 4.17. Kinerja Sistem Penentu outlier pada Algoritma Propagasi balik menggunakan Data Set -3

jumlah Dimensi	Akurasi Pengenalan Data Train	Akurasi Propagasi balik pada data test yang meliputi outliers			
		propagasi balik konvensional	MDOD Propagasi balik	FDOD Euclidean distance Propagasi balik	FDOD mahalnobid distance propagasi balik
10	100	35.90	69.87	77.56	69.87
15	100	35.90	69.87	77.56	69.87
20	100	19.23	48.72	60.90	48.72
25	100	19.23	48.72	60.90	48.72
30	100	21.15	50.64	62.82	50.64
35	100	18.59	48.08	60.26	48.08
40	100	17.31	46.79	58.97	46.79
45	100	17.95	47.44	59.62	47.44
50	100	17.95	47.44	59.62	47.44
55	100	19.23	48.72	60.90	48.72

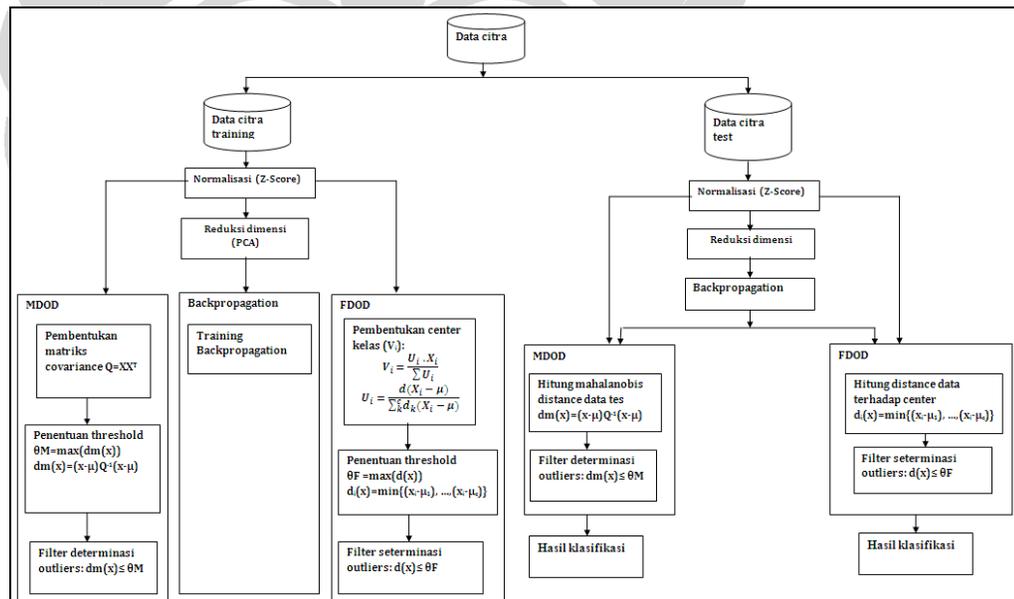


Gambar 4.8 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-3

Dari hasil percobaan pada data set 3 dengan skenario ujicoba ke-1 akurasi tertinggi pada FDOD euclidean distance penentu outlier pada algoritma propagasi

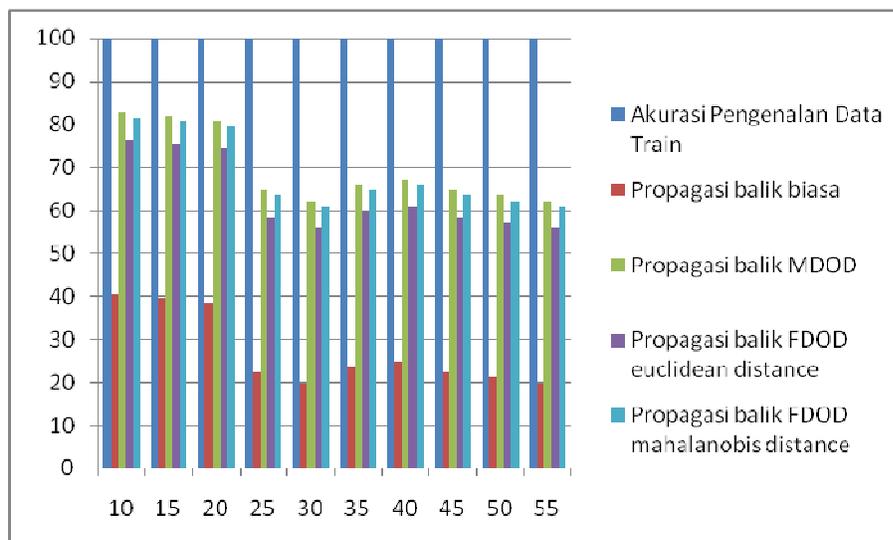
balik, yaitu 77.56%. Nilai ini meningkat dari akurasi 35.90% algoritma propagasi balik konvensional menjadi 77.56%. Pada ujicoba ini, hasil akurasi tertinggi pada data yang telah direduksi dari berdimensi 1024 menjadi berdimensi 10.

Ujicoba kedua memiliki skenario sebagai berikut, yaitu data input propagasi balik tidak sama dengan data input sistem penentu outlier. Data input pada algoritma propagasi balik terlebih dahulu ditransformasi ke ruang eigen untuk direduksi dimensinya, sedangkan data input pada sistem penentu outlier menggunakan data citra asli tanpa dilakukan transformasi ke ruang eigen ataupun reduksi dimensi. Rancangan sistem pada ujicoba ke-2 dapat dilihat pada gambar 4.9.



Gambar 4.9 Rancangan sistem ujicoba ke-2

Hasil ujicoba dari rancangan sistem ujicoba ke-2 data set-1 dapat dilihat pada tabel 4.18 dan gambar grafik pada gambar 4.10.



Gambar 4.10 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-1

Tabel 4.18. Kinerja Sistem Penentu outlier pada Algoritma Propagasi balik menggunakan Data Set 1

jumlah Dimensi	Akurasi Pengenalan Data Train	Akurasi Propagasi balik pada data test yang meliputi outliers			
		propagasi balik konvensional	MDOD Propagasi balik	FDOD Euclidean distance Propagasi balik	FDOD mahalanobid distance propagasi balik
10	100	40.38	82.69	76.28	81.41
15	100	39.74	82.05	75.64	80.77
20	100	38.46	80.77	74.36	79.49
25	100	22.44	64.74	58.33	63.46
30	100	19.87	62.18	55.77	60.90
35	100	23.72	66.03	59.62	64.74
40	100	25.00	67.31	60.90	66.03
45	100	22.44	64.74	58.33	63.46
50	100	21.15	63.46	57.05	62.18
55	100	7.44	44.27	50.19	56.30

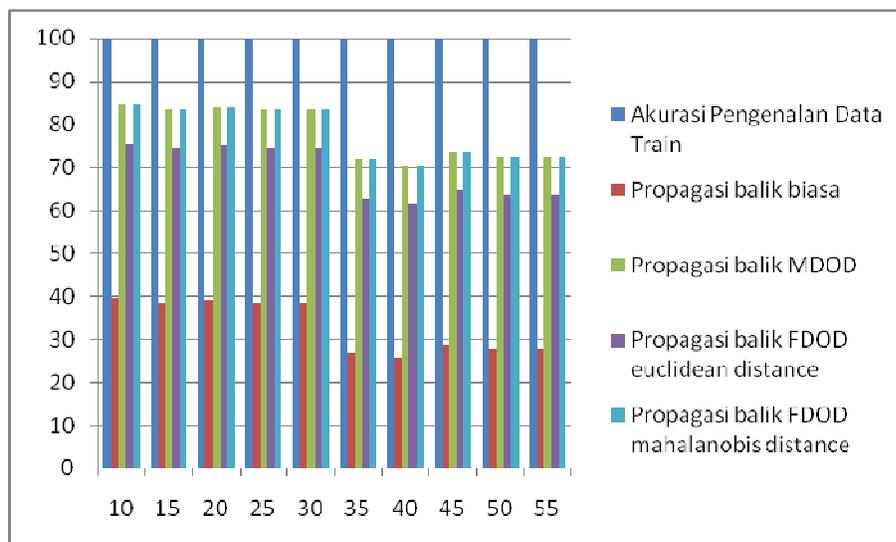
Pada data set 1 dengan skenario ujicoba ke 2 akurasi tertinggi pada MDOD penentu outlier pada algoritma propagasi balik, yaitu 82.69%. Nilai ini meningkat dari akurasi 40.38% algoritma propagasi balik konvensional menjadi 82.69%.

Pada ujicoba ini, hasil akurasi tertinggi pada data yang telah direduksi dari berdimensi 1024 menjadi berdimensi 10.

Hasil ujicoba dari rancangan sistem ujicoba ke-2 data set 2 dapat dilihat pada tabel 4.19 dan gambar grafik pada gambar 4.11.

Tabel 4.19. Kinerja Sistem Penentu outlier pada Algoritma Propagasi balik menggunakan Data Set -2

jumlah Dimensi	Akurasi Pengenalan Data Train	Akurasi Propagasi balik pada data test yang meliputi outliers			
		propagasi balik konvensional	MDOD Propagasi balik	FDOD Euclidean distance Propagasi balik	FDOD mahalnobid distance propagasi balik
10	100	39.74	84.62	75.64	84.62
15	100	38.46	83.33	74.36	83.33
20	100	39.10	83.97	75.00	83.97
25	100	38.46	83.33	74.36	83.33
30	100	38.46	83.33	74.36	83.33
35	100	26.92	71.79	62.82	71.79
40	100	25.64	70.51	61.54	70.51
45	100	28.85	73.72	64.74	73.72
50	100	27.56	72.44	63.46	72.44
55	100	27.56	72.44	63.46	72.44



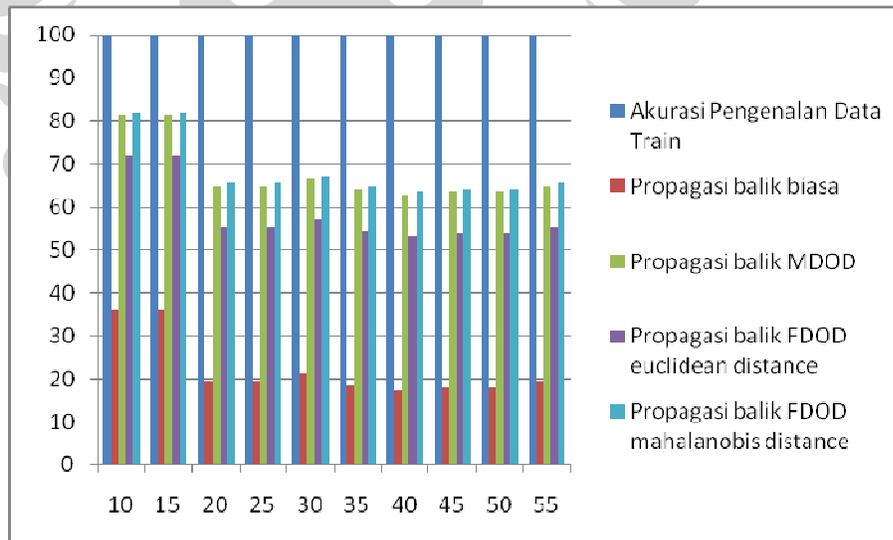
Gambar 4.11 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-2

Dari hasil percobaan pada data set-2 dengan sekenario ujicoba ke-2 akurasi tertinggi pada FDOD mahalanobis distance penentu outlier pada algoritma propagasi balik dan juga MDOD, yaitu 84.62%. Nilai ini meningkat dari akurasi 39.74% algoritma propagasi balik konvensional menjadi 84.62%. Pada ujicoba ini, hasil akurasi tertinggi pada data yang telah direduksi dari berdimensi 1024 menjadi berdimensi 10.

Hasil ujicoba dari rancangan sistem ujicoba ke-2 data set 3 dapat dilihat pada tabel 4.20 dan gambar grafik pada gambar 4.12.

Tabel 4.20. Kinerja Sistem Penentu outlier pada Algoritma Propagasi balik menggunakan Data Set -3

jumlah Dimensi	Akurasi Pengenalan Data Train	Akurasi Propagasi balik pada data test yang meliputi outliers			
		propagasi balik konvensional	MDOD Propagasi balik	FDOD Euclidean distance Propagasi balik	FDOD mahalanobid distance propagasi balik
10	100	35.90	81.41	71.79	82.05
15	100	35.90	81.41	71.79	82.05
20	100	19.23	64.74	55.13	65.38
25	100	19.23	64.74	55.13	65.38
30	100	21.15	66.67	57.05	67.31
35	100	18.59	64.10	54.49	64.74
40	100	17.31	62.82	53.21	63.46
45	100	17.95	63.46	53.85	64.10
50	100	17.95	63.46	53.85	64.10
55	100	19.23	64.74	55.13	65.38



Gambar 4.12 Grafik Akurasi Pengenalan Algoritma Propagasi balik pada Data Set-3

Dari hasil percobaan pada data set-3 dengan skenario ujicoba ke-3 akurasi tertinggi pada FDOD *mahalanobis distance* penentu outlier pada algoritma propagasi balik, yaitu 82.05%. Nilai ini meningkat dari akurasi 35.90% algoritma propagasi balik konvensional menjadi 82.05%. Pada ujicoba ini, hasil akurasi tertinggi pada data yang telah direduksi dari berdimensi 1024 menjadi berdimensi 10.

Percobaan dilakukan dengan menggunakan dua skenario yang berbeda, yaitu skenario 1 dengan rancangan pada gambar 4.5 dan skenario 2 dengan rancangan pada gambar 4.9. Dua skenario ini bekerja dengan perbedaan data masukan pada sistem penentu outlier, skenario pertama bekerja pada ruang eigen dengan data direduksi dimensinya, sehingga dimensi data masukan pada propagasi balik sama dengan dimensi data pada sistem penentu outlier. Pada skenario kedua data masukan pada propagasi balik berbeda dengan data masukan pada sistem penentu outlier, data masukan pada propagasi balik telah ditransformasi dalam ruang eigen dan dimensinya telah direduksi sedangkan pada sistem penentu outlier data masukan merupakan data yang berada dalam ruang citra asli. Dari percobaan pada skenario pertama dan skenario kedua dapat disimpulkan bahwa akurasi terbaik terjadi pada percobaan skenario kedua, yaitu data bekerja pada ruang citra asli.