

BAB II

PENGGUNAAN *FUZZY-NEURO LEARNING*

***VECTOR QUANTIZATION* PADA SISTEM**

PENCIUMAN ELEKTRONIK

Penelitian ini merupakan bagian dari pengembangan Sistem Penciuman Elektronik. Sistem Penciuman Elektronik merupakan sebuah sistem yang dapat diharapkan mampu mengenali suatu zat berdasarkan aromanya. Sistem ini memiliki tiga bagian utama, yaitu Sistem Sensor, Sistem Elektronik, dan Sistem Jaringan Neural Buatan. [Budiarto, 1998] Sistem Sensor akan menerima aroma dari suatu zat dan mengubahnya menjadi gelombang listrik. Gelombang listrik ini akan diteruskan ke Sistem Elektronik untuk dicacah dan disimpan secara digital ke dalam sebuah komputer. Data digital ini merupakan data masukan bagi Sistem Jaringan Neural Buatan dan dapat bertindak sebagai data pelatihan atau menjadi data uji pengenalan aroma.

Beberapa penelitian telah dilakukan untuk memilih model Jaringan Neural Buatan terbaik yang dapat melakukan pengenalan aroma dengan tingkat keakuratan yang tinggi. Beberapa model Jaringan Neural Buatan yang telah diimplementasikan pada Sistem Penciuman Elektronik ini adalah JNB Propagasi Balik, *Learning Vector Quantization*, [Budiarto, 1998] dan *Fuzzy-Neuro Learning Vector Quantization*. [Irzal, 2000] Hasil penelitian menunjukkan bahwa sampai saat ini *Fuzzy-Neuro Learning Vector Quantization* (FNLVQ) memberikan tingkat keakuratan yang sangat tinggi, [Arsyad, 2004] oleh karena itu, penelitian terhadap model Jaringan Neural Buatan ini perlu dilanjutkan untuk lebih memaksimalkan keefektifan model Jaringan Neural Buatan ini.

Bab ini akan membahas tentang arsitektur dan algoritma dari Sistem Penciuman Elektronik. Pembahasan ini terutama menekankan pada penggunaan *Fuzzy-Neuro Learning Vector Quantization* sebagai sistem pengenalan aroma beserta data aroma yang digunakan dalam pengenalan.

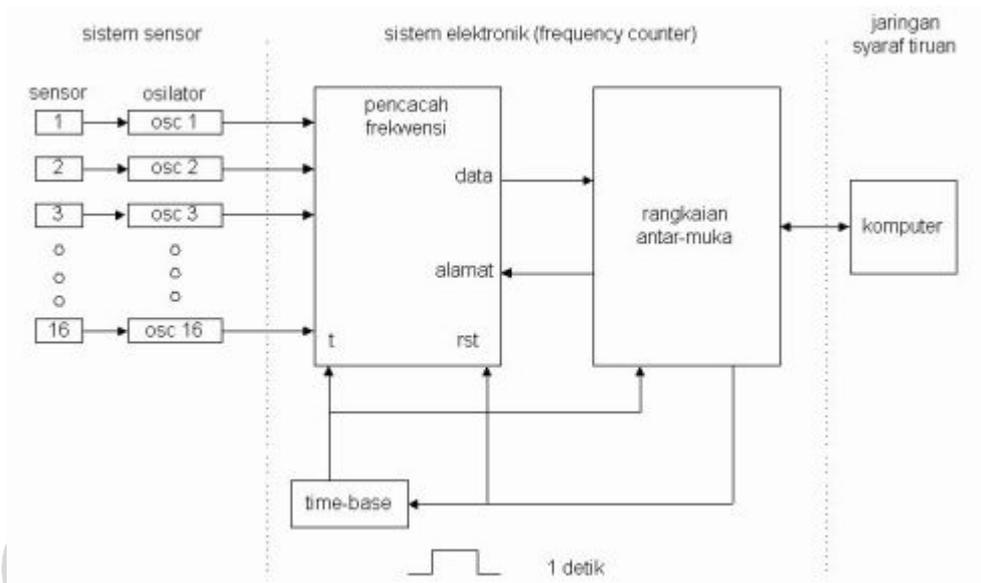
2.1 SISTEM PENCIUMAN ELEKTRONIK

Sistem Penciuman Elektronik dibagi menjadi 3 bagian utama, [Rivai, 1997] yaitu:

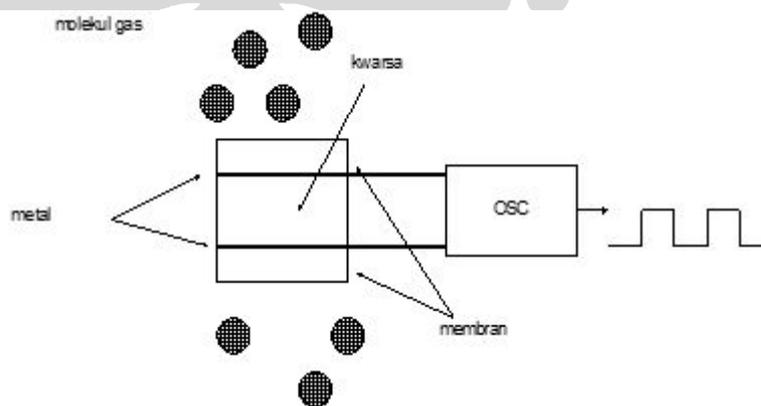
1. Sistem Sensor, berfungsi untuk mengubah besaran aroma menjadi besaran listrik / frekuensi.
2. Sistem Elektronik (*Frequency Counter*), berfungsi untuk mengukur besarnya perubahan frekuensi sensor dan disimpan dalam bentuk data digital.
3. Sistem Jaringan Neural Buatan, berfungsi untuk melakukan pemodelan data, sehingga sistem ini dapat melakukan pengenalan aroma.

2.1.1 Sistem Sensor

Sensor yang dipakai adalah sensor resonator kwarsa yang dilapisi membran sensitif. Sensor ini diletakkan di dalam osilator yang bertindak sebagai resonator pada frekuensi, seperti yang terlihat pada gambar 2.2. Prinsip kerja dari sensor hidung elektronik tersebut adalah menghitung nilai/besar perubahan frekuensi dari sensor resonator. Frekuensi sensor resonator akan menurun saat molekul gas *teradsorbsi* oleh membran, dan akan kembali normal setelah molekul tersebut mengalami proses *deadsorbsi*. Fenomena ini disebut efek pembebanan massa (*mass loading effect*) [Rivai, 1997]. Penelitian ini menggu-



Gambar 2.1: Diagram Sistem Penciuman Elektronik



Gambar 2.2: Prinsip Dasar Sistem Sensor

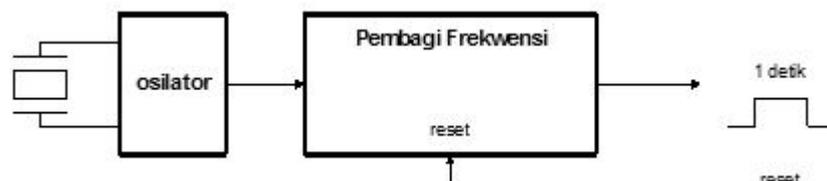
nakan sensor resonator kwarsa yang memiliki frekuensi dasar 20 MHz. Sensor yang digunakan berjumlah 16 buah, dimana masing-masing sensor dilapisi oleh membran-membran sensitif yang mempunyai karakteristik yang berbeda.

2.1.2 Sistem Elektronik (*Frequency Counter*)

Sistem elektronik disini berfungsi sebagai ADC (*analog digital converter*) yaitu mengubah besaran-besaran frekuensi yang disebabkan oleh perubahan massa dari gas yang dianalisa menjadi angka-angka yang akan disimpan dalam komputer. Angka-angka tersebut akan dijadikan dasar analisa oleh sistem. Sistem elektronik ini terbagi menjadi 3 sub-bagian rangkaian waktu-dasar, rangkaian pencacah frekuensi, dan rangkaian antar muka.

Rangkaian Waktu-dasar

Rangkaian waktu-dasar menggunakan rangkaian osilator dengan ketelitian tinggi yang stabil terhadap perubahan waktu dan suhu. Pada rangkaian tersebut dipasang pembagi frekuensi, dengan tujuan untuk menurunkan frekuensi osilator (orde Mhz) sehingga dapat menghasilkan waktu-dasar sebesar 1 detik. Penggunaan waktu-dasar sebesar 1 detik menyebabkan angka yang ditangkap oleh komputer adalah hasil pengukuran yang sebenarnya sehingga dari data tersebut dapat langsung dilakukan proses selanjutnya. Skematika rangkaian tersebut ditunjukkan pada gambar 2.3.



Gambar 2.3: Skematika Rangkaian Waktu-dasar

Rangkaian pencacah frekuensi

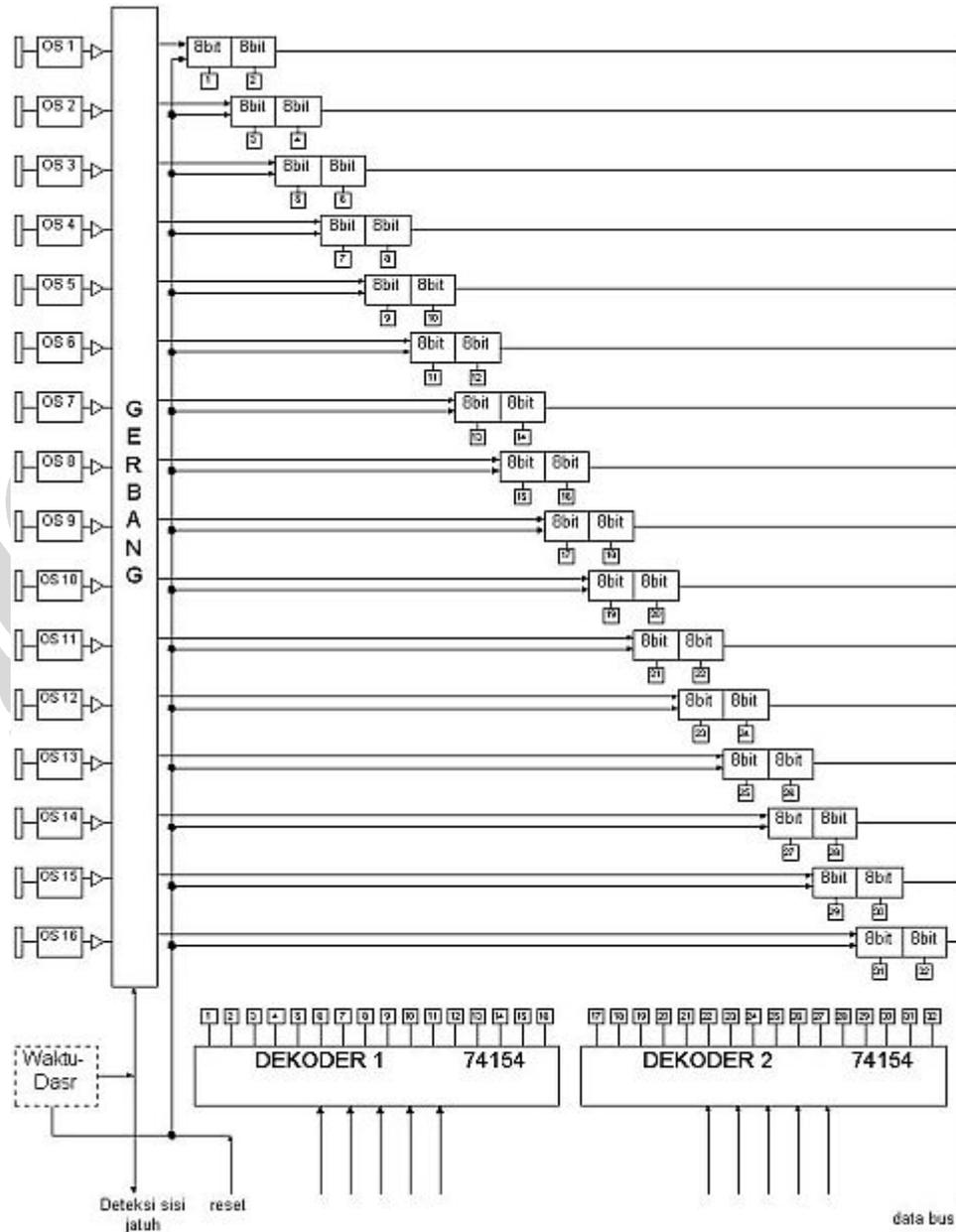
Pencacah frekuensi yang dikembangkan disini mempunyai ketelitian 16 bit, sehingga dapat memiliki kemampuan mencacah efek penurunan frekuensi hingga mencapai orde 76 Khz. Aliran data dari rangkaian pencacah frekuensi dilakukan dengan menggunakan dua buah pencacah 8 bit (16 bit) untuk masing-masing pencacah. Komputer akan mengambil data setiap byte (8 bit) dari pencacah secara berurutan lewat rangkaian antar muka. Proses pengambilan data tersebut berlangsung secara sangat cepat (mili detik), sehingga proses tersebut dapat dianggap secara paralel. Proses ini akan menghasilkan data dari masing-masing sensor pada kondisi (tekanan, volume, dan suhu) pencuplikan sampel yang sama. Skematika rangkaian tersebut ditunjukkan pada gambar 2.4.

Rangkaian antar muka

Rangkaian ini menggunakan PIO (Programmable Input-Output) 8255 sebagai pengontrol rangkaian antar muka. Beberapa alasan dipakainya PIO 8255 diantaranya adalah : banyaknya jumlah port (3 port; 1 port mengontrol 8 pin), mudahnya pemrograman dan fleksibilitas alat untuk pengembangan lebih lanjut.

Alamat memori yang digunakan untuk mengontrol PIO 8255 pada sistem hidung elektronik ini adalah 300H-303H. Untuk lebih jelasnya pembagian alamat dan fungsi port dapat dilihat pada tabel 2.1.

Agar komputer dapat mengaktifkan PIO 8255 pada alamat-alamat tersebut dibutuhkan rangkaian dekoder untuk menterjemahkan alamat tersebut. Rangkaian dekoder tersebut akan digabung dengan PIO 8255 menjadi sistem antar muka yang dapat membaca data dari rangkaian pencacah frekuensi. Skematika rangkaian tersebut ditunjukkan pada gambar 2.5.



Gambar 2.4: Skematika Rangkaian Pencacah Frekuensi

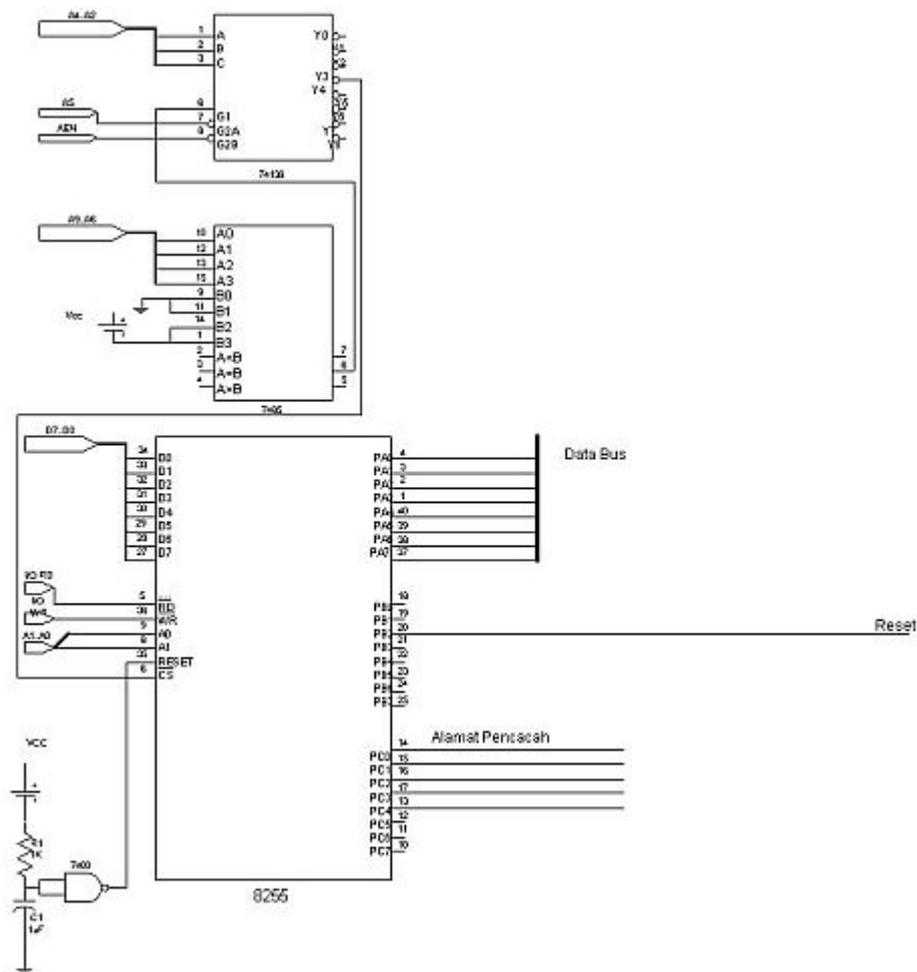
Alamat	Alat yang dikontrol	Keterangan
300H	Port A	Port A digunakan sebagai input ke komputer yaitu mengambil data dari rangkaian pencacah frekwensi
301H	Port B	Port B digunakan sebagai output dari komputer untuk memilih pencacah mana yang akan diaktifkan.
302H	Port C	Port C digunakan sebagai Input/Output untuk membaca sisi jatuh pewaktuan dan sisi reset dari sistem.
303H	<i>Control Word</i>	Digunakan sebagai pengatur konfigurasi sistem.

Tabel 2.1: Konfigurasi PIO8255

2.1.3 Sistem Jaringan Neural Buatan

Beberapa metoda pengenalan pola yang dapat diaplikasikan pada teknologi Hidung Elektronik adalah : Analisa Kluster, Analisa Fungsi Pembeda dan Jaringan Neural Buatan (JNB). Metoda jaringan neural buatan sering digunakan pada saat ini karena mempunyai algoritma proses pengenalan yang mudah dan hasil pengenalan aroma yang di dapat cukup baik. Dalam proses pengenalan aroma, JNB mempunyai dua tahapan proses. Pertama kali JNB dilatih dengan data-data aroma yang diketahui, dalam hal ini adalah nilai perubahan frekuensi yang ditangkap oleh komputer. Setelah proses pelatihan, jaringan ini dapat meramalkan jenis aroma yang dianalisa, tahapan ini disebut proses pengujian.

Ada beberapa metoda JNB yang telah berkembang. Pembagian metoda tersebut berdasarkan proses pembelajarannya yaitu proses pembelajaran dengan pengarahan (supervised) dan tanpa pengarahan (unsupervised). Beberapa metoda Jaringan Neural Buatan yang telah diimplementasikan pada Sistem Penciuman Elektronik ini adalah JNB Propagasi Balik, *Learning Vector Quantization*, [Budiarto, 1998] dan *Fuzzy-Neuro Learning Vector Quan-*



Gambar 2.5: Rangkaian Dekoder

tization. [Irzal, 2000] Hasil penelitian menunjukkan bahwa sampai saat ini *Fuzzy-Neuro Learning Vector Quantization* (FNLVQ) memberikan tingkat keakuratan yang sangat tinggi, [Arsyad, 2004] oleh karena itu, penelitian terhadap model Jaringan Neural Buatan ini perlu dilanjutkan untuk lebih memaksimalkan keefektifan model Jaringan Neural Buatan ini.

2.2 FUZZY-NEURO LEARNING VECTOR QUANTIZATION

Fuzzy-Neuro Learning Vector Quantization adalah sebuah model Jaringan Neural Buatan yang merupakan sebuah pengembangan dari *Learning Vector Quantization* (LVQ). [Irzal, 2000] *Learning Vector Quantization* merupakan sebuah model Jaringan Neural Buatan turunan dari *Self Organization Map* (SOM) yang menggunakan sistem kompetisi dalam algoritma pembelajarannya. [Fausett, 1994] Perbedaan utama LVQ dengan SOM terletak pada metoda pembelajaran yang digunakan. SOM menggunakan metoda pembelajaran tanpa pengarahan (*unsupervised learning*), sedangkan LVQ menggunakan metode pembelajaran yang membutuhkan pengarahan (*supervised learning*).

Berdasarkan sistem kompetisi, [Fausett, 1994] sebuah LVQ akan mencari sebuah vektor yang paling mirip dengan vektor pelatihannya. Vektor ini diberi nama *vektor pemenang*. Metode yang biasa digunakan dalam penentuan vektor pemenang ini adalah metoda *squared euclidian distance*. Setelah mendapatkan vektor pemenang, algoritma pembelajaran yang dimiliki LVQ akan membandingkan apakah vektor pemenang mempunyai kelas yang sama dengan vektor pelatihan. Apabila vektor pelatihan dan vektor pemenang memiliki kelas yang sama, maka vektor pemenang akan digeser mendekati vektor pelatihan agar tingkat kemiripan kedua vektor menjadi lebih tinggi, sebaliknya, apabila vektor pelatihan dan vektor pemenang bukan berasal dari kelas yang sama, maka vektor pemenang akan digeser menjauhi vektor pelatihan, agar tingkat kemiripan menjadi berkurang. Hasil dari proses pembelajaran ini adalah sebuah *vector codebook* yang berisi vektor-vektor perwakilan yang memiliki jarak terdekat dengan semua vektor pelatihan yang ada, sehingga dapat mewakili semua vektor pelatihan tersebut.

2.2.1 Penggunaan Teori *Fuzzy* pada FNLVQ

Pengembangan sistem *Learning Vector Quantization* dilakukan dengan memanfaatkan teori *fuzzy* pada vektor masukan, algoritma pembelajaran, dan penentuan kategori dari vektor masukan. [Budiarto, 1998] Hal ini bertujuan untuk mengakomodasikan perbedaan-perbedaan kecil pada vektor masukan, sehingga tingkat pengenalan dari sistem diharapkan akan menjadi lebih tinggi.

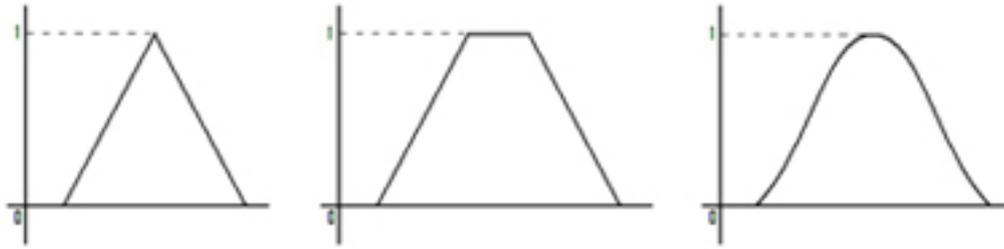
Penggunaan teori *fuzzy* mengubah struktur data dari neuron-neuron yang dimiliki oleh LVQ, dari neuron yang memiliki data berbentuk *crisp* menjadi neuron yang memiliki data yang berbentuk *fuzzy* (data *fuzzy*). Data *fuzzy* terdiri dari bilangan *fuzzy*. Bilangan *fuzzy* merupakan salah satu bentuk dari himpunan *fuzzy* yang telah diperkenalkan oleh Zadeh. [Zadeh, 1965]

Himpunan *Fuzzy*

Definisi himpunan *fuzzy* menurut Zadeh adalah

Definisi 1 *Sebuah himpunan fuzzy A yang merupakan himpunan bagian dari suatu himpunan universal X dikarakteristikan oleh sebuah fungsi keanggotaan, $\mu_A(x)$, yang mengasosiasikan setiap titik pada X dengan bilangan real dalam interval $[0, 1]$.*

Setiap nilai $\mu_A(x)$ menunjukkan tingkat keanggotaan dari x di dalam A , dengan $\mu_A(x) = 0$ menunjukkan tingkat keanggotaan yang paling rendah (dengan kata lain x bukan anggota dari A) dan $\mu_A(x) = 1$ menunjukkan tingkat keanggotaan yang paling tinggi. Untuk mempersingkat penulisan, pernyataan " $\mu_A(x)$ untuk semua $x \in X$ " dapat ditulis menjadi μ_A . Fungsi keanggotaan dapat berbentuk linear, seperti segitiga dan trapesium, maupun non-linear, seperti yang terlihat pada gambar 2.6.



Gambar 2.6: Fungsi Keanggotaan

Bilangan *Fuzzy*

Sebuah bilangan *fuzzy* adalah himpunan *fuzzy* yang memiliki himpunan bilangan riil, \mathbb{R} , sebagai himpunan universalnya. [Zadeh, 1965] Sebuah bilangan *fuzzy* juga harus memiliki sifat-sifat himpunan *fuzzy* yang konveks dan normal.

Definisi 2 Sebuah himpunan *fuzzy* A dikatakan konveks jika dan hanya jika

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2)) \quad (2.1)$$

untuk semua x_1 dan x_2 dalam X serta semua λ dalam $[0, 1]$.

Definisi 3 Suatu himpunan *fuzzy* A dikatakan normal apabila mempunyai nilai maksimum dari fungsi keanggotaan sebesar 1, atau $\mu_A^{max} = 1$.

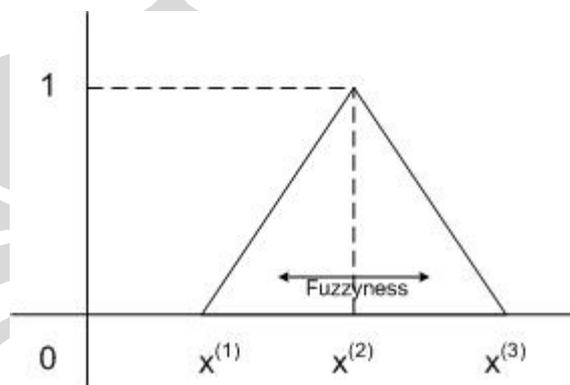
Untuk efisiensi komputasi, dipergunakan bilangan *fuzzy* dengan fungsi keanggotaan linear, seperti segitiga dan trapesium. Penelitian ini menggunakan bilangan *fuzzy* segitiga yang merupakan turunan dari definisi bilangan *fuzzy* trapesium [Dubois and Prade, 1988] yang diperkenalkan oleh Dubois dan Prade.

Definisi 4 Sebuah bilangan *fuzzy* segitiga, \tilde{x} , direpresentasikan sebagai

$$\tilde{x} = (x^{(1)}, x^{(2)}, x^{(3)}), \text{ dengan } x^{(1)}, x^{(2)}, x^{(3)} \in R \quad (2.2)$$

memiliki fungsi keanggotaan, $\mu_{\tilde{x}}(u)$,

$$\mu_{\tilde{x}}(u) = \begin{cases} 0 & , \text{jika } u \leq x^{(1)} \text{ atau } u \geq x^{(3)} \\ \frac{u - x^{(1)}}{x^{(2)} - x^{(1)}} & , \text{jika } x^{(1)} \leq u \leq x^{(2)} \\ 1 & , \text{jika } u = x^{(2)} \\ \frac{x^{(3)} - u}{x^{(3)} - x^{(2)}} & , \text{jika } x^{(2)} \leq u \leq x^{(3)} \end{cases} \quad (2.3)$$



Gambar 2.7: Bilangan Fuzzy Segitiga

2.2.2 Proses Fuzzifikasi

Sistem Elektronik dari Sistem Penciuman Elektronik menghasilkan data digital yang berbentuk *crisp*. Data *crisp* tersebut harus ditransformasikan menjadi data *fuzzy* agar dapat digunakan oleh *Fuzzy-Neuro Learning Vector Quantization*. Proses transformasi data *crisp* menjadi data *fuzzy* ini disebut Proses Fuzzifikasi

Algoritma dari Proses Fuzzifikasi ini adalah sebagai berikut.

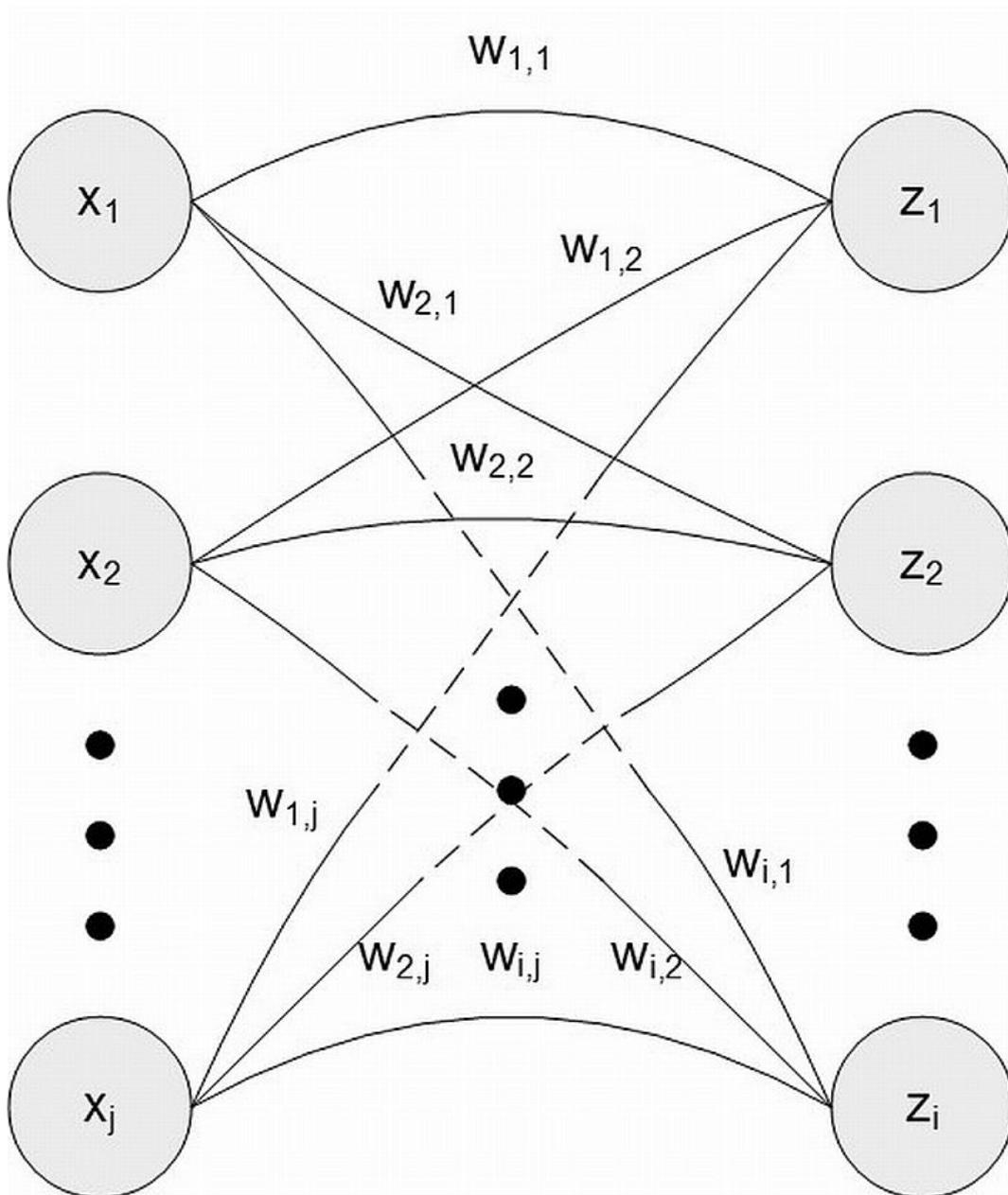
- Seluruh data *crisp* dibagi menjadi beberapa bagian yang sama besar.
- Untuk setiap bagian ini dicari nilai minimum, maksimum dan rata-ratanya.
- Sebuah bilangan *fuzzy* segitiga, \tilde{x} , dibentuk dengan $x^{(1)}$ adalah nilai minimum, $x^{(2)}$ adalah nilai rata-rata, dan $x^{(3)}$ adalah nilai maksimum.

Berdasarkan algoritma tersebut, banyaknya bilangan *fuzzy* segitiga yang terbentuk sesuai dengan banyaknya bagian data *crisp* yang dibuat.

2.2.3 Arsitektur FNLVQ

Arsitektur Jaringan Neural Buatan *Fuzzy-Neuro Learning Vector Quantization* mempunyai dua lapisan neuron yaitu lapisan masukan dan keluaran. Masing-masing neuron pada lapisan masukan akan terhubung dengan semua neuron pada lapisan keluaran. Setiap hubungan ini memiliki bobot. Bobot pada penghubung ini disebut juga vektor perwakilan. Langkah pertama yang dilakukan pada lapisan masukan adalah membandingkan besaran neuron yang masuk di lapisan masukan dengan bobot dari sisi penghubung ke lapisan keluaran. Perbandingan ini disebut juga sebagai pencarian nilai similaritas. Vektor perwakilan yang memiliki nilai similaritas terbesar akan menjadi vektor pemenang. Banyaknya neuron pada lapisan masukan sesuai dengan besarnya dimensi data masukan, kemudian banyaknya neuron pada lapisan keluaran sesuai dengan banyaknya kelas aroma. Banyaknya penghubung neuron lapisan masukan dan neuron lapisan keluaran merupakan perkalian dari banyaknya neuron pada lapisan masukan dengan banyaknya neuron pada lapisan keluaran. Gambar 2.8 menunjukkan sebuah Jaringan Neural Buatan *Fuzzy-Neuro Learning Vector Quantization* dengan i buah neuron pada lapisan masukan yang menunjukkan banyaknya sensor, j buah neuron pada lapisan keluaran yang menunjukkan banyaknya kelas atau kategori, dan $i \times j$ buah penghubung lapisan masukan dan lapisan keluaran.

Contoh, apabila data aroma memiliki dimensi sebanyak d , dengan kelas aroma sebanyak k , maka Fuzzy-Neuro LVQ akan memiliki d neuron pada lapisan masukan, k neuron pada lapisan keluaran, dan $d \times k$ neuron pada lapisan



Gambar 2.8: Arsitektur dari FNLVQ

tersembunyi.

Notasi-notasi yang digunakan pada jaringan Fuzzy-Neuro LVQ ini adalah sebagai berikut:

- \vec{x} merupakan notasi untuk vektor masukan, dengan

$$\vec{x} = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3, \dots, \tilde{x}_d) \quad (2.4)$$

\tilde{x}_i adalah neuron fuzzy ke- i .

d adalah besar dimensi data masukan.

- \vec{z} merupakan notasi untuk vektor keluaran, dengan

$$\vec{z} = (z_1, z_2, z_3, \dots, z_k) \quad (2.5)$$

z_i adalah nilai real pada kelas ke i .

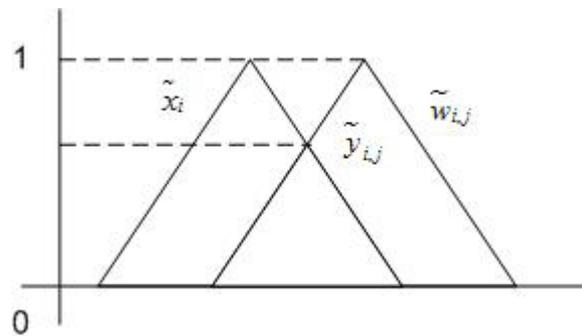
k adalah banyaknya kelas aroma.

- W adalah sebuah matriks bobot atau kumpulan vektor perwakilan, dimana setiap elemen $\tilde{w}_{i,j}$ menghubungkan neuron \tilde{x}_j di lapisan masukan dengan neuron z_i pada lapis keluaran.
- \vec{w}_i adalah vektor perwakilan dari kelas i dan merupakan vektor baris ke- i dari matriks W .
- Y adalah sebuah notasi matriks untuk nilai similaritas, dimana setiap elemen $y_{i,j}$ merupakan nilai similaritas neuron \tilde{x}_j dengan vektor perwakilan pada elemen $\vec{w}_{i,j}$.

2.2.4 Metode Pembelajaran Fuzzy-Neuro LVQ

Inisialisasi Awal

Sistem Jaringan Neural Buatan ini membutuhkan inisialisasi awal untuk vektor perwakilan dan laju pembelajaran (α) dalam memulai pelatihan. Inisi-



Gambar 2.9: Nilai Similaritas

alisasi awal untuk vektor perwakilan adalah rata-rata dari vektor masukan dan inialisasi untuk laju pembelajaran, $\alpha = 0,05$.

Perhitungan Nilai Similaritas

Perhitungan nilai similaritas dilakukan terhadap masing-masing neuron pada lapisan masukan, \tilde{x}_j , dengan penghubung ke neuron lapisan keluaran, z_i , yang memiliki bobot $\tilde{w}_{i,j}$. Hasil perhitungan disimpan dalam elemen sebuah matriks Y , $y_{i,j}$. Definisi dari nilai similaritas dari kedua bilangan fuzzy itu adalah sebagai berikut:

$$y_{i,j} = \max(\mu_{\tilde{x}_i} \cap \mu_{\tilde{w}_{i,j}}) \quad (2.6)$$

Perhitungan Lapisan Keluaran

Neuron z_i pada lapisan keluaran dihitung dengan mencari minimum dari $y_{i,j}$, atau

$$z_i = \min_{j=1}^d (y_{i,j}) \quad (2.7)$$

dimana d adalah dimensi data masukan.

Kelas pemenang adalah kelas yang mempunyai nilai similaritas z_i paling tinggi, atau

$$v = \max_{j=1}^d (z_i) \quad (2.8)$$

dengan v adalah nilai similaritas maksimum.

Apabila $v = 0$, maka kelas dari vektor masukan merupakan kelas yang *tidak dikenali* oleh Fuzzy-Neuro LVQ.

Aturan Pembelajaran

Pada Jaringan Neural Buatan, proses pembelajaran pada umumnya adalah proses modifikasi berulang-ulang terhadap *vector codebook* yang berisi vektor perwakilan untuk setiap kategori keluaran hingga vektor perwakilan tersebut menjadi cukup representatif. Modifikasi vektor perwakilan dalam Fuzzy-Neuro LVQ dilakukan dalam dua tahap:

- Melakukan pergeseran posisi vektor perwakilan
- Melakukan perubahan *fuzzyness* dari vektor perwakilan

Modifikasi tersebut dilakukan pada proses pembelajaran, hingga Fuzzy-Neuro LVQ mendapatkan *vector codebook* terbaik.

Perubahan *fuzzyness*, merupakan proses pelebaran atau penyempitan ukuran *fuzzyness*, dari bilangan fuzzy segitiga. Apabila bilangan fuzzy segitiga $\tilde{y} = (y^{(1)}, y^{(2)}, y^{(3)})$ merupakan hasil dari proses perubahan *fuzzyness* dari bilangan fuzzy segitiga $\tilde{x} = (x^{(1)}, x^{(2)}, x^{(3)})$, maka nilai \tilde{y} didefinisikan berdasarkan persamaan berikut:

$$\begin{aligned} y^{(1)} &= x^{(2)} - \beta \cdot (x^{(2)} - x^{(1)}) \\ y^{(2)} &= x^{(2)} \\ y^{(3)} &= x^{(2)} + \beta \cdot (x^{(3)} - x^{(2)}) \end{aligned} \quad (2.9)$$

Terdapat 3 kasus yang mungkin terjadi ketika proses pembelajaran pada Fuzzy-Neuro LVQ sedang berlangsung, yaitu:

1. Vektor perwakilan pemenang sama dengan vektor masukan

Pada kasus ini, vektor perwakilan pemenang digeser mendekati vektor masukan, kemudian vektor pemenang tersebut diperlebar *fuzzyness*-nya.

Penggeseran dilakukan dengan mempergunakan persamaan 2.10 dan pelebaran *fuzzyness* dilakukan dengan mempergunakan persamaan 2.9, dimana nilai β lebih besar dari 1 ($\beta > 1$). Penelitian ini menggunakan nilai β sebesar 1.05.

$$\tilde{w}_{v,j}(t+1) = \tilde{w}_{v,j}(t) + \alpha(t) \cdot (1 - y_{v,j}(t)) \cdot (\tilde{x}_j(t) - \tilde{w}_{v,j}(t)) \quad (2.10)$$

$$0 \leq j < d$$

dimana

d = besarnya dimensi input

v = indeks vektor pemenang

$\tilde{w}_{v,j}(t+1)$ = elemen dari vektor pemenang setelah pergeseran

$\tilde{w}_{v,j}(t)$ = elemen dari vektor pemenang sebelum pergeseran

$\tilde{x}_j(t)$ = elemen dari vektor masukan

$\alpha(t)$ = laju pembelajaran

$y_{v,j}(t)$ = nilai similaritas dari vektor pemenang

2. Vektor perwakilan pemenang tidak sama dengan vektor masukan

Pada kasus ini, vektor perwakilan pemenang digeser menjauhi vektor masukan, kemudian vektor pemenang tersebut dipersempit *fuzzyness*-nya.

Penggeseran dilakukan dengan mempergunakan persamaan 2.11 dan penyempitan *fuzzyness* dilakukan dengan mempergunakan persamaan 2.9, dimana nilai β terletak antara 0 dan 1 ($0 < \beta < 1$). Penelitian ini menggunakan nilai β sebesar 1.05.

$$\tilde{w}_{v,j}(t+1) = \tilde{w}_{v,j}(t) + \alpha(t) \cdot y_{v,j}(t) \cdot (\tilde{x}_j(t) - \tilde{w}_{v,j}(t)) \quad (2.11)$$

$$0 \leq j < d$$

dimana

- d = besarnya dimensi input
- v = indeks vektor pemenang
- $\tilde{w}_{v,j}(t+1)$ = elemen dari vektor pemenang setelah pergeseran
- $\tilde{w}_{v,j}(t)$ = elemen dari vektor pemenang sebelum pergeseran
- $\tilde{x}_j(t)$ = elemen dari vektor masukan
- $\alpha(t)$ = laju pembelajaran
- $y_{v,j}(t)$ = nilai similaritas dari vektor pemenang

3. Nilai similaritas maksimum adalah 0 atau $v = 0$

Apabila nilai similaritas maksimum adalah 0, maka seluruh vektor perwakilan diperlebar fuzzyness-nya, dengan mempergunakan persamaan 2.12.

$$\tilde{w}_{i,j}(t+1) = \gamma(t) \cdot \tilde{w}_{i,j}(t) \quad (2.12)$$

dengan, γ adalah konstanta pelebaran. Penelitian ini menggunakan γ sebesar 1.1.

2.3 SPESIFIKASI DATA AROMA

Penelitian ini menggunakan data aroma yang berasal dari Sistem Penciuman Elektronik 16 sensor. Data aroma terdiri dari 3 macam aroma parfum

Alkohol	Martha Tilaar	Jeruk Kenanga	Jeruk Mawar	Mawar Kenanga
10%		JK-A10%	JM-A10%	MK-A10%
15%		JK-A115%	JM-A115%	MK-A115%
25%		JK-A125%	JM-A125%	MK-A125%
35%		JK-A135%	JM-A135%	MK-A135%
45%		JK-A145%	JM-A145%	MK-A145%
70%		JK-A170%	JM-A170%	MK-A170%

Tabel 2.2: Kelas-kelas pada data aroma

NO	Kelompok		
	I	II	III
1	JK-A10%	JM-A10%	MK-A10%
2	JK-A115%	JM-A115%	MK-A115%
3	JK-A125%	JM-A125%	MK-A125%
4	JK-A135%	JM-A135%	MK-A135%
5	JK-A145%	JM-A145%	MK-A145%
6	JK-A170%	JM-A170%	MK-A170%

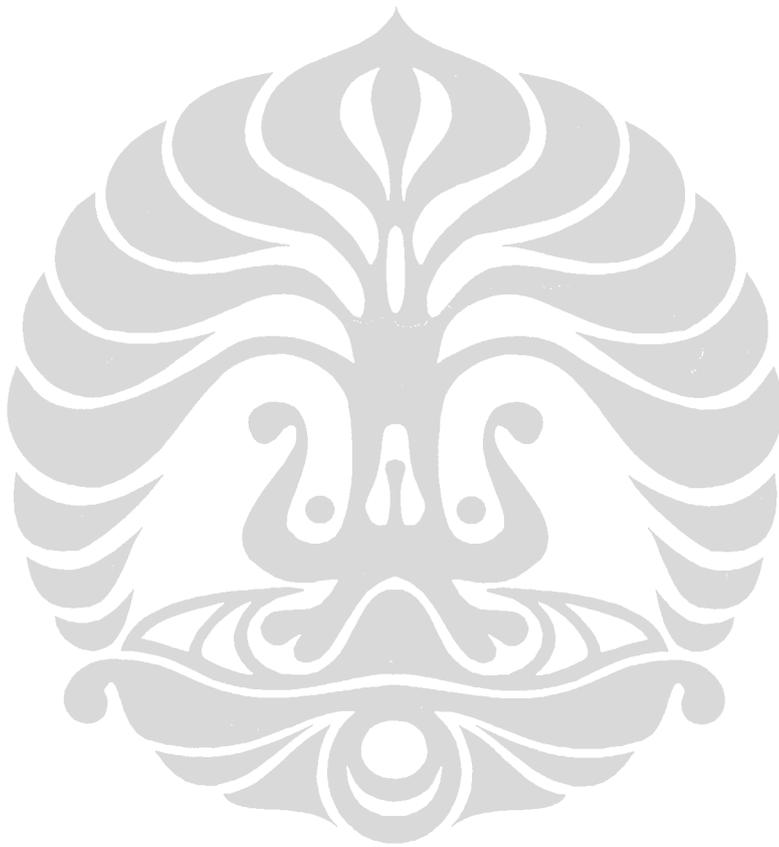
Tabel 2.3: Kelas aroma pada pengujian 6 kelas aroma

Martha Tilaar dan alkohol dengan 6 tingkat kepekatan yang berbeda. Aroma Martha Tilaar terdiri aroma jeruk(J), kenanga(K) dan mawar(M), sedangkan alkohol memiliki tingkat kepekatan 0%, 15%, 25%, 35%, 45%, dan 70%. Masing-masing aroma Martha Tilaar dicampur dengan aroma Martha Tilaar lain, hingga menghasilkan 3 aroma baru yang berasal dari campuran dua aroma Martha Tilaar, yaitu campuran jeruk dengan kenanga (JK), campuran jeruk dengan mawar (JM), dan campuran kenanga dengan mawar (KM). Aroma campuran ini dicampur lagi dengan semua variasi alkohol sehingga menghasilkan 18 aroma campuran baru, yang disebut sebagai Aroma 3 Campuran. Tabel 2.2 memberikan daftar keseluruhan Aroma 3 Campuran.

NO	Kelompok		
	I	II	III
1	JK-A10%	JK-A10%	JM-A10%
2	JK-A115%	JK-A115%	JM-A115%
3	JK-A125%	JK-A125%	JM-A125%
4	JK-A135%	JK-A135%	JM-A135%
5	JK-A145%	JK-A145%	JM-A145%
6	JK-A170%	JK-A170%	JM-A170%
7	JM-A10%	MK-A10%	MK-A10%
8	JM-A115%	MK-A115%	MK-A115%
9	JM-A125%	MK-A125%	MK-A125%
10	JM-A135%	MK-A135%	MK-A135%
11	JM-A145%	MK-A145%	MK-A145%
12	JM-A170%	MK-A170%	MK-A170%

Tabel 2.4: Kelas aroma pada pengujian 12 kelas aroma

Pengujian dilakukan dengan membagi data menjadi 6, 12, dan 18 kelas aroma. 6 kelas aroma terdiri dari 3 kelompok, tabel 2.3. 12 kelas aroma terdiri dari 3 kelompok, tabel 2.4. 18 kelas aroma melibatkan keseluruhan kelas aroma, sehingga hanya memiliki 1 kelompok. Keseluruhan data akan menghasilkan 7 kelompok kelas aroma.



BAB III

FUZZY PRINCIPAL COMPONENT ANALYSIS

Peningkatan teknik pengkoleksian data dan semakin besarnya daya tampung dari sistem penyimpanan, menyebabkan semakin banyak data pengamatan yang memiliki dimensi besar. Penambahan dimensi ini dilakukan karena adanya asumsi bahwa semakin besar dimensi dari data, maka semakin besar informasi yang dapat diberikan oleh data tersebut, yang berimplikasi pada semakin mudahnya proses pengklasifikasian data.

Peningkatan dimensi mempunyai efek samping, berupa bertambahnya tingkat kesulitan dalam komputasi pengolahan data tersebut. Selain itu, asumsi yang menyatakan bahwa penambahan dimensi berbanding lurus dengan penambahan informasi juga kurang tepat, karena tidak semua variabel-variabel data tersebut bersifat saling bebas. Adanya variabel-variabel yang tidak saling bebas ini, akan menghasilkan informasi yang *redundant*, yang tidak akan mempermudah proses pengklasifikasian data.

Pengurangan redundansi informasi dari data yang memiliki dimensi besar dapat, dilakukan dengan melakukan pengurangan dimensi. Teknik pengurangan dimensi ini harus dapat mengurangi variabel-variabel dependent tanpa mengurangi informasi yang terlalu besar. Teknik pengurangan dimensi ini, secara bahasa matematika dapat dituliskan sebagai berikut: diberikan sebuah variabel random dimensi- p , $x = (x_1, x_2, x_3, \dots, x_p)^T$, carilah representasi variabel tersebut dalam dimensi yang lebih rendah, $y = (y_1, y_2, y_3, \dots, y_q)^T$ dengan $q \leq p$, yang tetap mengambil isi dari data aslinya, berdasarkan suatu kriteria. Komponen s ini biasanya disebut sebagai komponen tersembunyi atau *feature* dari data x tadi.

Salah satu cara untuk mengurangi data *redundant* tersebut adalah dengan menemukan komponen-komponen yang dianggap penting dari sebaran data. Teknik ini disebut juga sebagai *Principal Component Analysis* (PCA). Bab ini akan membahas dasar-dasar dari metoda *Principal Component Analysis* dan modifikasinya sehingga metoda tersebut dapat digunakan pada data *fuzzy*.

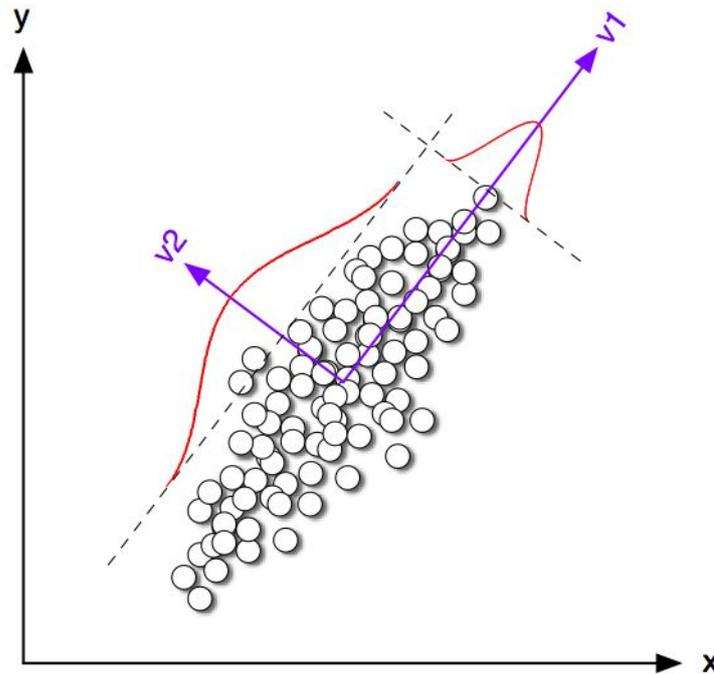
3.1 *PRINCIPAL COMPONENT ANALYSIS*

Principal Component Analysis yang disingkat menjadi PCA merupakan sebuah metoda pencarian komponen utama (*principal component*), secara linear, dari sekumpulan data. Komponen utama yang dihasilkan oleh metoda PCA ini berdasarkan variasi sebaran yang dimiliki oleh data yang bersangkutan. PCA dapat juga digunakan sebagai metode pengurangan dimensi, yaitu dengan melakukan pemetaan data dari suatu data yang memiliki ruang berdimensi- d ke ruang eigen yang berdimensi- e , dimana $e \leq d$. Pemetaan dilakukan dengan mempergunakan sebuah matriks transformasi yang terbentuk dari sekumpulan komponen utama (*principal component*) data tersebut.

Komponen utama dari sekumpulan data merupakan kombinasi linear orthogonal dari variabel-variabel data tersebut [Fodor, 2002], dimana komponen utama yang pertama memiliki varians yang paling besar. Komponen-komponen utama selanjutnya, memiliki varians yang terus mengecil, sehingga apabila k_m adalah komponen utama ke- m dan k_n adalah komponen utama ke- n , maka $Var(k_m) \geq Var(k_n)$, untuk semua m dan n dimana $m > n$.

Algoritma yang dilakukan untuk menerapkan metode *Principal Component Analysis* adalah sebagai berikut.

1. Langkah yang pertama adalah membentuk sebuah matriks data X yang berukuran $n \times d$, dimana n menunjukkan banyaknya data dan d adalah



Gambar 3.1: Interpretasi geometris dari PCA, v_1 dan v_2 adalah *principal component* dari data tersebut

besarnya dimensi dari data tersebut.

2. Kemudian normalisasikan data tersebut sehingga untuk setiap vektor kolom dari matriks X memiliki rata-rata nol dan varians satu.
3. Selanjutnya buatlah sebuah matriks kovarian C dari matriks hasil normalisasi tersebut.

Pembentukan matriks kovarian ini dapat dilakukan dengan menggunakan persamaan 3.1.

$$\Sigma = E(XX') \quad (3.1)$$

4. Langkah selanjutnya adalah mencari *eigenvalue* dan *eigenvector* dari Σ . Eigenvector inilah yang merupakan komponen utama dari data, sehingga untuk mengurangi dimensi data, kita cukup memilih beberapa *eigenvector* yang memiliki varians atau *eigenvalue* besar.

3.2 PENERAPAN PCA PADA DATA *FUZZY*

Data *fuzzy* memiliki karakteristik yang berbeda dengan data *crisp*, oleh karena itu metoda PCA tidak dapat diaplikasikan secara langsung terhadap data *fuzzy*. Data *fuzzy* yang digunakan pada penelitian ini merupakan data yang berasal dari data *crisp*. Data *crisp* tersebut diolah menjadi data *fuzzy* dengan menggunakan metoda fuzzifikasi seperti yang diterangkan pada bagian 2.2.2.

Pencarian ruang eigen dari data *fuzzy* ini dapat dilakukan dengan 2 cara. Cara pertama adalah dengan mencari ruang eigen dari data *crisp* sebelum data tersebut ditransformasikan menjadi data *fuzzy*. Pencarian ruang eigen ini dapat menggunakan metoda PCA biasa karena data yang diolah masih berbentuk data *crisp*. Data *crisp* yang telah berada pada ruang eigen inilah yang kemudian diubah menjadi data *fuzzy* menggunakan proses fuzzifikasi. Data *fuzzy* yang terbentuk merupakan data *fuzzy* yang telah berada pada ruang eigen.

Cara kedua adalah dengan mencari ruang eigen pada data *fuzzy* secara langsung. Metoda PCA biasa tidak dapat diaplikasikan pada cara ini, oleh karena itu dibutuhkan sebuah metoda yang dapat mencari ruang eigen dari data yang berbentuk bilangan *fuzzy*. Metoda ini akan dibahas secara lengkap pada bagian 3.3.

3.3 PENGGUNAAN *AUTO ASSOCIATIVE NEURAL NETWORK* SEBAGAI *FUZZY PRINCIPAL COMPONENT ANALYSIS*

Metoda PCA pada data *fuzzy* ini menggunakan sebuah Jaringan Neural Buatan yang dikenal sebagai *Auto Associative Neural Network*. Bourlard dan

Kamp [Bourlard and Kamp, 1988] telah menemukan bahwa terdapat sebuah hubungan yang menarik antara *Auto Associative Neural Network* dengan PCA. *Auto Associative Neural Network* merupakan sebuah jaringan yang memiliki neuron berbentuk data *crisp*. Penerapan *Auto Associative Neural Network* pada data *fuzzy* dilakukan dengan mengubah neuron pada jaringan tersebut menjadi neuron yang berbentuk bilangan *fuzzy*.

3.3.1 Operasi pada Bilangan Fuzzy

Pada bagian 2.2 telah dinyatakan bahwa *Fuzzy-Neuro Learning Vector Quantization* adalah sebuah Jaringan Neural Buatan yang merupakan pengembangan dari *Learning Vector Quantization*. Pengembangan ini berupa penggunaan bilangan *fuzzy* pada lapisan masukan, bobot, dan aturan pembelajarannya. *Fuzzy-Neuro Learning Vector Quantization* mempergunakan bilangan *fuzzy* yang berbentuk segitiga, yang definisinya dapat dilihat pada definisi 4 di halaman 17.

Salah satu sifat dari bilangan fuzzy ini adalah tertutup pada operasi penjumlahan, pengurangan, dan perkalian dengan bilangan real. Definisi operasi penjumlahan, pengurangan, dan perkalian dengan bilangan real adalah sebagai berikut:

Definisi 5 Untuk semua $\tilde{x} = (x^{(1)}, x^{(2)}, x^{(3)})$ dan $\tilde{y} = (y^{(1)}, y^{(2)}, y^{(3)})$ berlaku

$$\tilde{x} + \tilde{y} = (x^{(1)} + y^{(1)}, x^{(2)} + y^{(2)}, x^{(3)} + y^{(3)}) \quad (3.2)$$

$$\tilde{x} - \tilde{y} = (x^{(1)} - y^{(3)}, x^{(2)} - y^{(2)}, x^{(3)} - y^{(1)}) \quad (3.3)$$

$$\forall a \in \mathbb{R} \mid a \cdot \tilde{x} = \begin{cases} (a \cdot x^{(1)}, a \cdot x^{(2)}, a \cdot x^{(3)}) & , \text{ jika } a \geq 0 \\ (a \cdot x^{(3)}, a \cdot x^{(2)}, a \cdot x^{(1)}) & , \text{ jika } a < 0 \end{cases} \quad (3.4)$$

3.3.2 *Auto-associative Neural Network*

Auto-associative Neural Network adalah sebuah Jaringan Neural Buatan yang memiliki bentuk *feedforward*. Jaringan ini memiliki tiga lapisan neuron, yaitu lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Banyaknya neuron pada lapisan masukan sama dengan banyaknya neuron pada lapisan keluaran. Banyaknya neuron pada lapisan tersembunyi adalah lebih sedikit atau sama dengan banyaknya neuron di lapisan masukan atau lapisan keluaran. Jadi, bila lapisan masukan dan lapisan keluaran memiliki neuron sebanyak d serta lapisan tersembunyi memiliki neuron sebanyak q , maka $q \leq d$.

Setiap neuron pada lapisan masukan terhubung dengan semua neuron pada lapisan tersembunyi. Keterhubungan ini memiliki bobot dan diimplementasikan pada sebuah matriks A yang berukuran $q \times d$. Sedangkan keterhubungan antara semua neuron pada lapisan tersembunyi dengan semua neuron pada lapisan keluaran diimplementasikan pada sebuah matriks B yang berukuran $d \times q$.

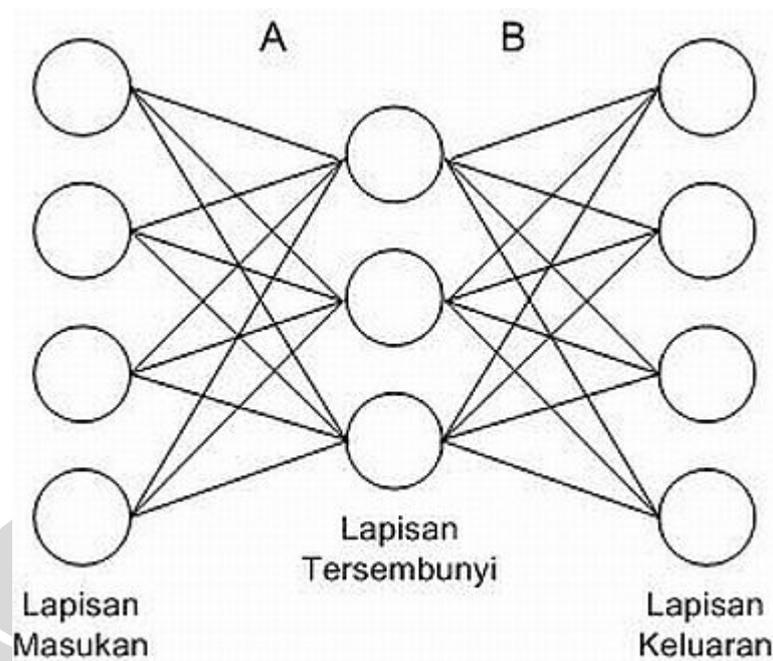
Apabila x adalah vektor pada lapisan masukan, y adalah vektor pada lapisan tersembunyi, dan z adalah vektor pada lapisan keluaran, maka keterhubungan antara ketiga vektor tersebut dapat dilihat pada persamaan berikut:

$$y = A \cdot x \quad (3.5)$$

$$z = B \cdot y \quad (3.6)$$

Metode pembelajaran pada jaringan ini disebut sebagai *auto-associative*, yaitu data yang masuk pada lapisan keluaran harus sama dengan data pada lapisan keluaran.

Algoritma perbaikan error mempergunakan *backpropagation*, dengan fung-



Gambar 3.2: Arsitektur *Auto-associative Neural Network*

si error diperkenalkan oleh Baldi dan Hornik [Baldi and Hornik, 1989], yaitu

$$E(A, B) = \sum_{p=1}^n e(x^p, y^p) \quad (3.7)$$

dimana n adalah banyaknya data dan $e(x^p, y^p)$ adalah rekontruksi error untuk pola p .

$$e(x^p, y^p) = \sum_{k=1}^d (x_k^p - y_k^p)^2 \quad (3.8)$$

Telah dibuktikan pada [Denœux and Masson, 2004] bahwa dengan mendefinisikan $A = B'$, maka pada lapisan tersembunyi akan menghasilkan vektor y yang mempunyai keterhubungannya dengan *principal component*.

Pendefinisian $A = B'$ mengakibatkan nilai z dapat dicari dengan

$$z = B' Bx \quad (3.9)$$

persamaan di atas dapat ditulis dalam notasi skalar sebagai:

$$z_k = \sum_{j=1}^q B_{kj} \sum_{i=1}^d B_{ij} x_i, k = [1, d] \quad (3.10)$$

3.3.3 Fuzzy Auto Associative Neural Network

Fuzzy Principal Component Analysis pada penelitian ini adalah sebuah teknik untuk mencari *principal component* dari data fuzzy dengan mempergunakan *Auto Associative Neural Network* yang memiliki neuron-neuron *fuzzy*. Dengan mengubah semua neuron pada arsitektur *Auto Associative Neural Network* di atas menjadi data *fuzzy*, maka persamaan 3.9 berubah menjadi:

$$\tilde{z} = B'B\tilde{x}, \quad (3.11)$$

dimana \tilde{x} dan \tilde{z} merupakan vektor yang berisi data fuzzy. Berdasarkan persamaan 3.6, nilai-nilai pada lapisan tersembunyi dapat didefinisikan sebagai:

$$\tilde{y} = B\tilde{x}, \quad (3.12)$$

Persamaan 3.11 juga dapat ditulis dalam notasi skalar, yaitu:

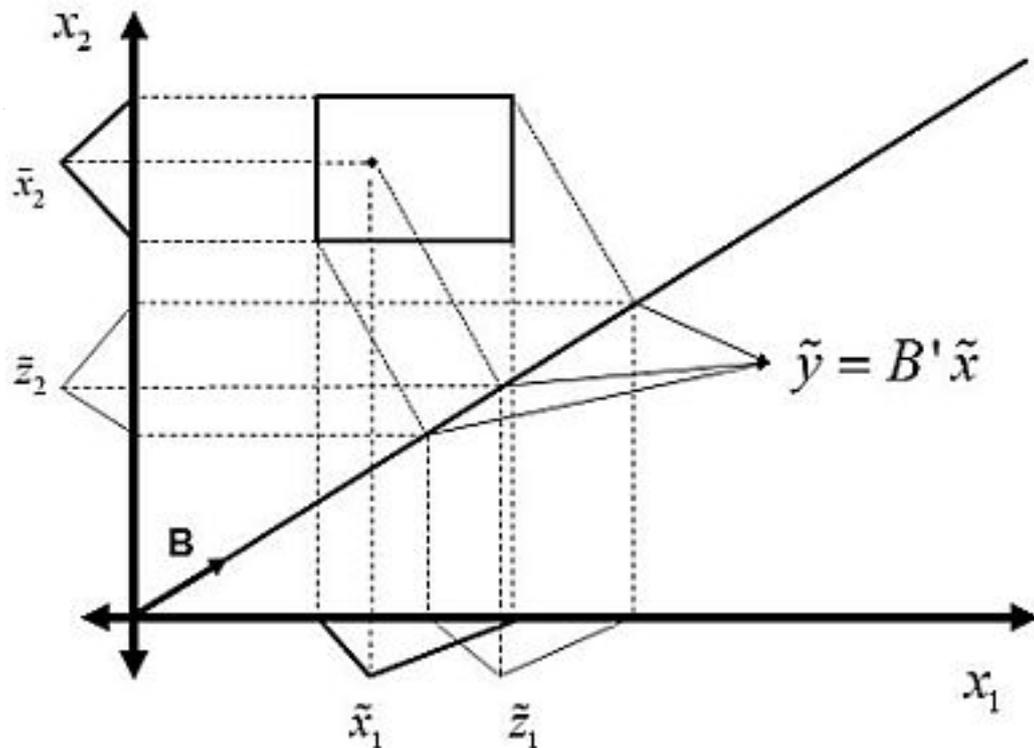
$$\tilde{z}_k = \sum_{j=1}^q B_{kj} \sum_{i=1}^d B_{ij} \tilde{x}_i, \quad k = [1, d] \quad (3.13)$$

Sebuah fungsi similaritas, untuk mengetahui tingkat kesamaan antara vektor masukan dengan vektor keluaran, didefinisikan agar tingkat kesalahan dari vektor keluaran dapat diketahui. Fungsi similaritas, $e(\tilde{x}, \tilde{z})$, ini didefinisikan sebagai berikut:

$$e(\tilde{x}, \tilde{z}) = \sum_{k=1}^d e(\tilde{x}_k, \tilde{z}_k) \quad (3.14)$$

Dengan mempergunakan algoritma backpropagation, maka matriks B akan diubah-ubah sehingga fungsi $e(\tilde{x}, \tilde{z})$ menghasilkan nilai yang minimum.

Gambar 3.3 memberikan sebuah interpretasi geometris untuk data fuzzy yang memiliki vektor masukan berdimensi 2. Bilangan *fuzzy* masukan \tilde{x}_1 dan \tilde{x}_2 ditransformasikan pada garis B , yang merupakan sebuah matriks transformasi, menjadi bilangan *fuzzy* \tilde{y} . Kemudian, bilangan *fuzzy* \tilde{y} kembali ditransformasikan pada garis B menjadi bilangan *fuzzy* \tilde{z}_1 dan \tilde{z}_2 . Agar nilai \tilde{z}_1 dan \tilde{z}_2 mendekati nilai \tilde{x}_1 dan \tilde{x}_2 , maka garis B harus diubah-ubah.



Gambar 3.3: Interpretasi Geometris dari Fuzzy-PCA

Implementasi *Fuzzy Auto Associative Neural Network*

Prosedur perhitungan pada Fuzzy-AANN adalah sebagai berikut. Notasi-notasi untuk tiap-tiap komponen pada lapisan masukan \tilde{x} , lapisan tersembunyi \tilde{y} , dan lapisan keluaran \tilde{z} secara berturut-turut adalah \tilde{x}_i , \tilde{y}_j , dan \tilde{z}_k . Berdasarkan persamaan 3.12, \tilde{y}_j dapat dihitung dengan persamaan

$$\tilde{y}_j = (\tilde{y}_1, \tilde{y}_2, \tilde{y}_3) = \sum_{i=1}^d B_{ij} \tilde{x}_i, j = [1, q] \quad (3.15)$$

dan \tilde{z}_k dapat dihitung dengan persamaan

$$\tilde{z}_k = (\tilde{z}_1, \tilde{z}_2, \tilde{z}_3) = \sum_{j=1}^q B_{kj} \tilde{y}_j, k = [1, d] \quad (3.16)$$

Berdasarkan persamaan 3.4, perkalian dengan bilangan real positif dan real negatif memiliki perbedaan. Oleh sebab itu, matriks B perlu dipisahkan nilai-nilai real positif dan real negatif menjadi B^+ dan B^- . Dimana masing-masing

elemen ke- i dan j dari kedua matriks tersebut adalah:

$$B_{ij}^+ = \begin{cases} B_{ij} & , \text{jika } B_{ij} > 0 \\ 0 & , \text{lainnya} \end{cases} \quad (3.17)$$

$$B_{ij}^- = \begin{cases} B_{ij} & , \text{jika } B_{ij} < 0 \\ 0 & , \text{lainnya} \end{cases} \quad (3.18)$$

Dengan menggunakan persamaan di atas, maka komponen-komponen dari bilangan fuzzy \tilde{y}_j dan \tilde{z}_k dapat dihitung dengan persamaan berikut.

$$y^{(l)} = B^{+'} x^{(l)} + B^{-'} x^{(3-l+1)} \quad (3.19)$$

$$z^{(l)} = B^{+'} y^{(l)} + B^{-'} y^{(3-l+1)} \quad (3.20)$$

Turunan Fungsi Error

Algoritma backpropagation membutuhkan turunan dari fungsi error $e(\tilde{x}, \tilde{y})$ untuk memperbaiki matriks B , sehingga keluaran dari Fuzzy-AANN memiliki error minimum. Persamaan 3.21 adalah turunan fungsi error tersebut, yang telah dibuktikan oleh Denœux dan Masson [Denœux and Masson, 2004].

$$\begin{aligned} \frac{\delta e^{(l)}}{\delta B_{ij}} &= \delta_i^{(l)} \left(H(B_{ij}) y_j^{(l)} + \left(1 - H(B_{ij}) y_j^{(3-l+1)} \right) \right) \\ &+ \zeta_j^{(l)} \left(H(B_{ij}) x_i^{(l)} + \left(1 - H(B_{ij}) x_i^{(3-l+1)} \right) \right) \\ &+ \eta_j^{(l)} \left(H(B_{ij}) x_i^{(3-l+1)} + \left(1 - H(B_{ij}) x_i^{(l)} \right) \right), \end{aligned} \quad (3.21)$$

dengan

$$\begin{aligned} \delta_i^{(l)} &= 2 \left(z_k^{(l)} - x_k^{(l)} \right) \\ \zeta_j^{(l)} &= \sum_{k=1}^d \delta_k^{(l)} B_{kj}^+ \\ \eta_j^{(l)} &= \sum_{k=1}^d \delta_k^{(l)} B_{kj}^- \\ i = k &= [1, d], j = [1, q], l = [1, 3] \end{aligned} \quad (3.22)$$

BAB IV

PERCOBAAN DAN ANALISA HASIL

Bagian ini membahas tentang skenario dan hasil percobaan pengenalan data aroma, yang telah berada pada ruang eigen, pada Sistem Penciuman Elektronik. Metode pemetaan data aroma ke ruang eigen yang digunakan terdiri dari dua metode, yang telah dideskripsikan pada bab III. Spesifikasi dari data percobaan dapat dilihat pada bagian 2.3 yang terdapat di halaman 25.

4.1 SKENARIO PERCOBAAN

4.1.1 Metoda I

Percobaan dilakukan dengan langkah-langkah sebagai berikut:

1. Data awal merupakan data yang berbentuk crisp.
Data-data ini disusun menjadi sebuah matriks X yang berukuran $n \times d$, dimana n adalah banyaknya data dan d adalah besarnya dimensi.
2. Kemudian gunakan metode PCA, untuk memproyeksikan data crisp ke ruang eigen.
3. Data yang telah diproyeksikan ke ruang eigen, diubah ke bentuk bilangan fuzzy dengan menggunakan proses fuzzifikasi, sehingga menghasilkan data fuzzy di ruang eigen.
4. Gunakan Jaringan Neural Buatan *Fuzzy-Neuro Learning Vector Quantization* untuk melakukan pengenalan terhadap data fuzzy tersebut.
Pengenalan pola dilakukan tidak hanya pada dimensi data yang sebenarnya, tapi juga dilakukan pada dimensi yang lebih kecil. Pada data aroma

pengenalan dilakukan pada dimensi 2, 4, 6, 8, 10, 12, dan 14, sedangkan pada data wajah pengenalan dilakukan pada dimensi 1, 2, 3, 4, 8, dan 16.

4.1.2 Metoda II

Percobaan dilakukan dengan langkah-langkah sebagai berikut:

1. Data awal merupakan data yang berbentuk crisp.
Data-data ini disusun menjadi sebuah matriks X yang berukuran $n \times d$, dimana n adalah banyaknya data dan d adalah besarnya dimensi.
2. Lakukan prosedur fuzzyfikasi, sehingga menghasilkan data fuzzy
3. Proyeksikan data fuzzy tersebut ke ruang eigen, dengan menggunakan Fuzzy-PCA
4. Gunakan Jaringan Neural Buatan *Fuzzy-Neuro Learning Vector Quantization* untuk melakukan klasifikasi dari data fuzzy, yang telah berada di ruang eigen tersebut

Pengenalan pola dilakukan tidak hanya pada dimensi data yang sebenarnya, tapi juga dilakukan pada dimensi yang lebih kecil. Pada data aroma pengenalan dilakukan pada dimensi 2, 4, 6, 8, 10, 12, dan 14, sedangkan pada data wajah pengenalan dilakukan pada dimensi 1, 2, 3, 4, 8, dan 16.

Dim	6 Kelas			12 Kelas			18 Kelas
	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK
2	31%	14%	14%	6%	7%	4%	5%
4	50%	25%	14%	21%	21%	38%	3%
6	72%	47%	64%	43%	47%	46%	41%
8	89%	58%	86%	79%	82%	86%	64%
10	89%	89%	83%	88%	92%	86%	82%
12	89%	92%	89%	86%	92%	97%	89%
14	89%	89%	89%	86%	92%	97%	88%
16	89%	89%	89%	86%	93%	97%	87%

Tabel 4.1: Pengenalan Aroma setelah Pengurangan Dimensi oleh PCA

4.2 HASIL PERCOBAAN

4.2.1 Metoda I

Hasil pengenalan terhadap data aroma yang sebelumnya telah dipetakan ke ruang eigen, dapat dilihat pada tabel 4.1. Tabel tersebut memperlihatkan akurasi pengenalan dari setiap kelompok kelas aroma, berdasarkan banyaknya *principal component* yang digunakan. Setiap kolom merupakan hasil pengenalan untuk setiap kelompok kelas aroma berdasarkan variasi konsentrasi alkohol. Jadi, untuk setiap kelompok kelas terdiri atas 6 kelas aroma. Contoh, kolom JK menunjukkan hasil percobaan dari kelompok kelas yang berisi aroma campuran jeruk, kenanga, dan alkohol, yaitu JK-A10%, JK-A115%, JK-A125%, JK-A135%, JK-A145%, dan JK-A170%. Kolom-kolom terbagi menjadi 3 kelompok besar yaitu 6 kelas, 12 kelas, dan 18 kelas. Kolom yang tergabung dalam 6 kelas adalah data-data aroma yang memiliki kelas sebanyak 6 buah, demikian pula untuk 12 kelas dan 18 kelas.

Tabel 4.1 menunjukkan bahwa penggunaan dimensi yang kurang menyebabkan hasil pengenalan menjadi kecil. Selain itu, tabel tersebut juga menunjukkan bahwa terdapat dimensi minimum yang dapat digunakan agar tingkat

pengenalan menjadi cukup baik, sebagai contoh, pada kolom JK penggunaan dimensi sebesar 8 buah sudah cukup, karena penambahan dimensi tidak menambah tingkat pengenalan. Selain itu, juga terdapat kolom yang menunjukkan bahwa penambahan dimensi pada ruang eigen justru akan mengurangi tingkat pengenalan. Hal ini dapat dilihat pada kelompok kelas JM, dimana pada penggunaan dimensi sebanyak 12 buah justru menghasilkan tingkat pengenalan yang maksimum.

4.2.2 Metoda II

Berbeda dengan percobaan sebelumnya, pada percobaan ini data yang dipetakan ke ruang fuzzy tidak berbentuk data crisp, melainkan data yang sudah berbentuk fuzzy. Persentase pengenalan dari data hasil pemetaan ke ruang eigen oleh *Fuzzy Principal Component Analysis* ini dapat dilihat pada tabel 4.2. Seperti percobaan sebelumnya, tabel 4.2 ini, terdiri dari baris-baris yang menunjukkan banyaknya dimensi yang dipergunakan serta kolom-kolom yang menunjukkan kelompok-kelompok kelas dari data aroma.

Tingkat pengenalan yang diperoleh pada percobaan ini jauh lebih bervariasi apabila dibandingkan dengan hasil percobaan sebelumnya. Hasil percobaan ini juga menunjukkan bahwa penambahan banyaknya dimensi belum tentu meningkatkan persentase pengenalan. Beberapa kelompok kelas menunjukkan bahwa penambahan dimensi justru menurunkan tingkat pengenalan.

4.3 EVALUASI HASIL PERCOBAAN

Beberapa analisa dapat dilakukan terhadap hasil-hasil percobaan tersebut. Analisa yang dilakukan diantaranya tingkat kompleksitas dan kesalahan dari kedua metoda tersebut, beserta tingkat pengenalan *Fuzzy-Neuro Learning*

Dim	6 Kelas			12 Kelas			18 Kelas
	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK
2	58%	46%	42%	31%	33%	25%	24%
4	71%	63%	58%	54%	63%	46%	43%
6	63%	63%	75%	29%	42%	50%	31%
8	83%	58%	58%	71%	90%	50%	82%
10	100%	71%	67%	81%	94%	88%	71%
12	100%	75%	79%	79%	94%	79%	72%
14	100%	83%	88%	79%	94%	83%	72%
16	100%	79%	96%	81%	88%	85%	78%

Tabel 4.2: Pengenalan Aroma setelah Pengurangan Dimensi oleh Fuzzy-PCA

Vector Quantization terhadap data yang dihasilkan oleh kedua metoda tersebut.

4.3.1 Analisa Kompleksitas Metode Fuzzy-PCA

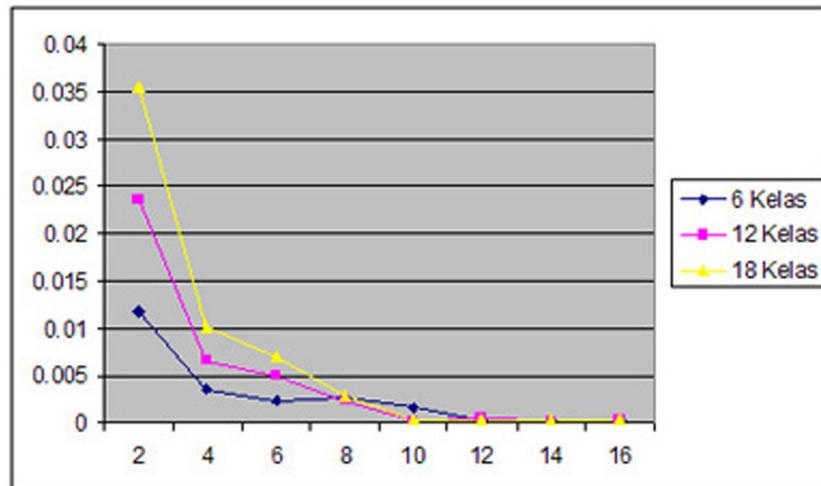
Menurut Dencœux dan Masson, [Dencœux and Masson, 2004] metode *Fuzzy Principal Component Analysis* ini mempunyai kompleksitas $O(ndq)$, untuk setiap iterasi(*epoch*), dimana n adalah banyaknya data, d adalah dimensi data, dan q adalah dimensi setelah direduksi. Dengan tingkat kompleksitas yang cukup rendah itu, kecepatan dari Fuzzy-PCA sangat dipengaruhi oleh banyaknya iterasi pembelajaran.

Berdasarkan percobaan banyaknya iterasi pembelajaran, sangat dipengaruhi oleh sebuah variabel yang disebut sebagai laju pembelajaran α . Nilai α ini tidak hanya mempengaruhi waktu pembelajaran, tetapi juga konvergensi dari fase pembelajaran. Apabila nilai ini terlalu besar maka fase pembelajaran tidak akan konvergen, sedangkan apabila nilai α terlalu rendah maka waktu pembelajaran menjadi sangat lama. Penentuan nilai α ini sangat tergantung dari karakteristik data, sehingga untuk data yang berbeda akan membutuhkan nilai *alpha* yang berbeda juga.

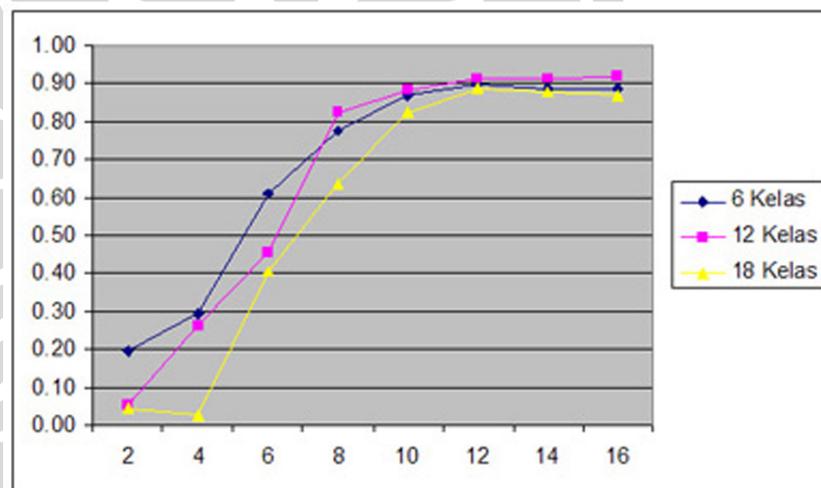
Setelah melakukan beberapa percobaan, maka nilai yang dianggap paling cocok untuk data aroma ini adalah 0.002. Nilai ini cukup kecil untuk membuat Fuzzy-PCA konvergen dan cukup besar sehingga waktu pembelajaran tidak terlalu lama, yaitu dibawah 10 iterasi.

4.3.2 Evaluasi Tingkat Kesalahan Fuzzy-PCA pada Data Aroma

Metode *Fuzzy Principal Component Analysis* mempergunakan sebuah jaringan *backpropagation* yang sangat mengandalkan pada penurunan tingkat kesalahan. Tingkat kesalahan ini diukur dari berapa besar perbedaan data yang masuk pada lapisan masukan dengan data yang keluar pada lapisan keluaran. Gambar 4.1 memperlihatkan tingkat kesalahan yang berhasil diperoleh Fuzzy-PCA dalam memetakan data aroma ke ruang eigen. Tabel tersebut terbagi atas kolom-kolom yang berisi banyaknya dimensi data fuzzy yang dipergunakan. Berdasarkan gambar 4.1 tersebut, terlihat bahwa tingkat kesalahan dipengaruhi oleh banyaknya kelas yang akan diklasifikasi dan banyaknya dimensi yang dipergunakan. Semakin banyak kelas yang harus diklasifikasi mengakibatkan bertambah besarnya tingkat kesalahan yang dihasilkan oleh *Fuzzy Principal Component Analysis*. Hal ini dimungkinkan karena fungsi kesalahan yang digunakan adalah fungsi penjumlahan dari selisih antara elemen-elemen data masukan dengan elemen-elemen data keluaran. Penambahan dimensi mengakibatkan tingkat kesalahan dari *Fuzzy Principal Component Analysis* menjadi semakin rendah. Hal ini terjadi karena penambahan dan pengurangan dimensi berpengaruh terhadap efektifitas sistem dalam melakukan pemampatan informasi.



Gambar 4.1: Tingkat Kesalahan pada Data Aroma

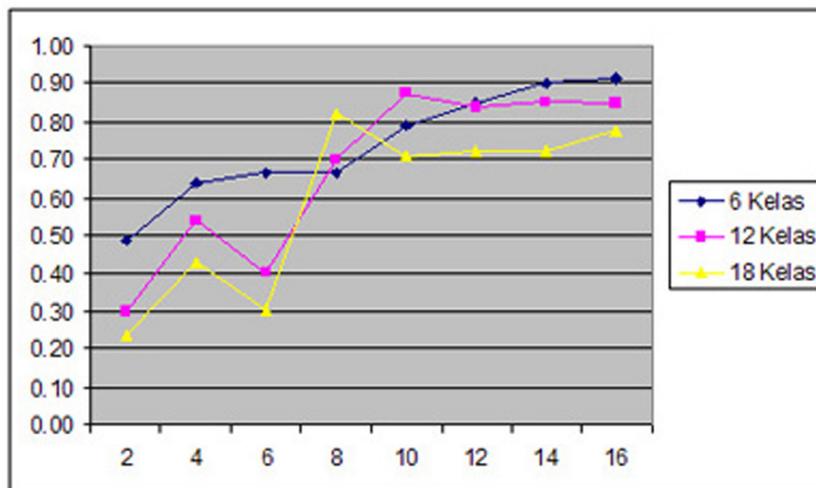


Gambar 4.2: Rata-rata Pengenalan Data Odor untuk Metoda PCA

4.3.3 Evaluasi Tingkat Pengenalan Data Aroma

Gambar 4.2 dan 4.3 menunjukkan tingkat pengenalan *Fuzzy-Neuro Learning Vector Quantization* terhadap data yang telah dipetakan ke ruang eigen oleh metode PCA biasa dan Fuzzy-PCA. Secara garis besar, grafik tersebut menunjukkan adanya korelasi antara banyaknya dimensi eigen yang digunakan dengan tingkat pengenalan aroma.

Pada metode *Principal Component Analysis* grafik peningkatan pengenalan aroma menunjukkan korelasi yang kuat antara banyaknya dimensi dan besar-



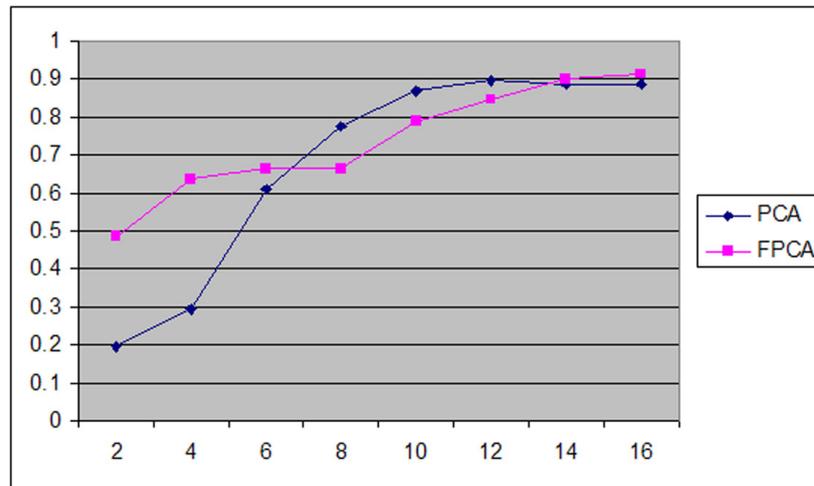
Gambar 4.3: Rata-rata Pengenalan Data Odor untuk Metode Fuzzy PCA

nya persentase pengenalan, sedangkan, pada metode Fuzzy PCA, terdapat data yang menunjukkan bahwa besarnya dimensi tidak berbanding lurus dengan tingkatan pengenalan. Sebagai contoh, untuk data yang memiliki banyak kelas sebesar 18 buah, memiliki tingkat penilaian yang lebih tinggi dari pada data sama yang menggunakan dimensi sebanyak 10 buah. Hal ini menunjukkan bahwa metode Fuzzy LVQ lebih berhasil memetakan data aroma tersebut ke ruang eigen berdimensi 8.

4.3.4 Evaluasi Perbandingan Pengenalan Data Aroma

Gambar 4.4–4.6 menunjukkan perbandingan kemampuan pengenalan data oleh *Fuzzy-Neuro Learning Vector Quantization*, berdasarkan data-data aroma di ruang untuk tiap-tiap kelompok kelas. Gambar 4.4 adalah grafik perbandingan tingkat pengenalan pada data 6 kelas. Gambar 4.4 adalah grafik perbandingan tingkat pengenalan pada data 12 kelas. Gambar 4.4 adalah grafik perbandingan tingkat pengenalan pada data 18 kelas.

Berdasarkan gambar 4.4 terlihat bahwa data yang diperoleh dari pemetaan ke ruang eigen oleh metode PCA dikenali lebih baik pada penggunaan dimensi

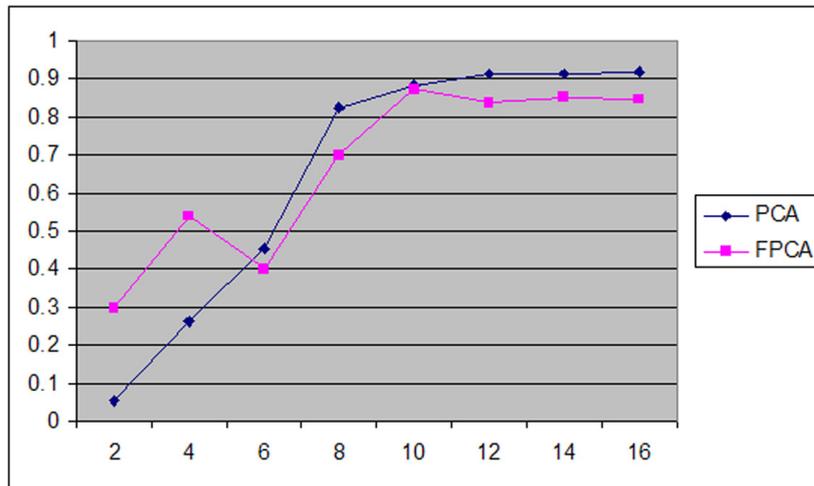


Gambar 4.4: Perbandingan Rata-rata Perbandingan Pengenalan untuk 6 Kelas Data Aroma

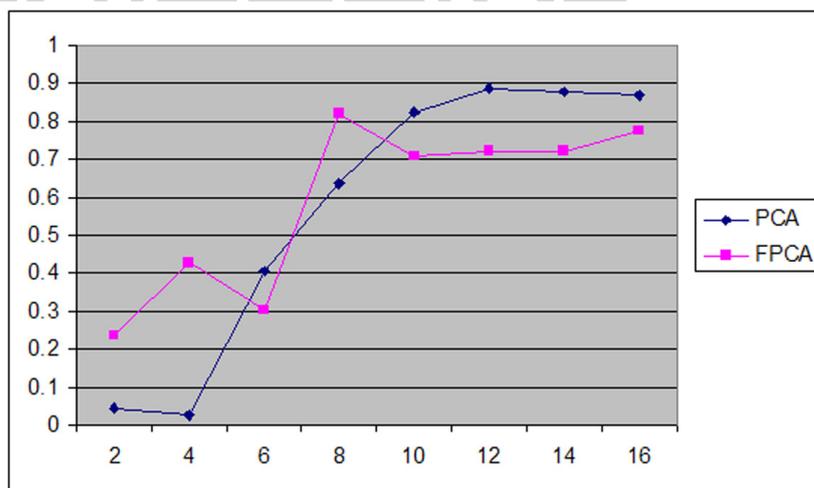
yang lebih besar. Untuk penggunaan dimensi lebih kecil, metode Fuzzy-PCA menghasilkan data aroma yang lebih mudah dikenali.

Gambar 4.5 menunjukkan perbandingan tingkat pengenalan pada 12 kelas aroma. Gambar ini sedikit berbeda dengan gambar 4.4, yaitu pada dimensi 6 dimana data yang berasal dari metode Fuzzy-PCA mengalami tingkat pengenalan yang rendah.

Gambar 4.6 menunjukkan perbandingan tingkat pengenalan pada 18 kelas aroma. Perbedaan utama pada grafik ini terletak pada dimensi 8, dimana tingkat pengenalan aroma mengalami peningkatan yang cukup drastis. Sehingga, untuk kelompok kelas 12 ini dapat disimpulkan bahwa pengurangan dimensi paling baik dilakukan pada dimensi 8.



Gambar 4.5: Perbandingan Pengenalan untuk 12 Kelas Data Aroma



Gambar 4.6: Perbandingan Pengenalan Untuk 18 Kelas Data Aroma

BAB V

PENERAPAN *FUZZY PRINCIPAL COMPONENT* *ANALYSIS* PADA DATA WAJAH

Data yang digunakan pada penelitian ini adalah data dari Sistem Penci-uman Elektronik dengan 16 sensor. Data ini memiliki 16 sensor dengan 18 variasi kelas. Penggunaan data tersebut pada penelitian yang berhubungan dengan pengurangan dimensi, seperti Fuzzy PCA, masih belum bisa membe-rikan kemampuan sebenarnya dari metode tersebut. Hal ini disebabkan oleh perbedaan dimensi, yang dihasilkan oleh metode pengurangan dimensi tadi, masih terlalu kecil.

Untuk bisa melihat kemampuan metode Fuzzy PCA dalam melakukan pe-ngurangan dimensi, maka diadakan sebuah penelitian tambahan dengan mem-pergunakan data citra. Data citra merupakan data yang sangat cocok untuk diaplikasikan pada permasalahan pengurangan dimensi, karena data citra bi-asanya memiliki dimensi yang sangat besar.

5.1 SPESIFIKASI DATA WAJAH

Data wajah terdiri dari 4 kelas wajah. Masing-masing kelas wajah memiliki 4 variasi ekspresi wajah. Masing-masing citra memiliki resolusi 31×35 . Setiap variasi wajah tersebut dipotret dengan variasi sudut horizontal pemrotetan sebanyak 7 buah. Sudut-sudut pemrotetan itu adalah 0° , $\pm 5^\circ$, $\pm 10^\circ$, dan $\pm 15^\circ$. Setiap citra, diberikan fungsi noise gaussian dengan 50 variasi variabel. Data fuzzy didapatkan dari gabungan citra asli dengan citra-citra yang telah



Gambar 5.1: Sebagian dari Data Citra

ditambahkan noise.

Gambar 5.1 memberikan sebagian kecil data wajah yang digunakan pada percobaan. Baris yang berbeda memiliki kelas wajah yang berbeda, sedangkan kolom menunjukkan variasi sudut horizontal pemotretan.

5.2 SKENARIO PERCOBAAN

Langkah-langkah percobaan yang dilakukan terhadap data citra ini adalah sebagai berikut:

1. Pertama-tama data citra diubah dulu menjadi bentuk matriks X yang berukuran $n \times d$, dimana n adalah banyaknya citra yang digunakan dan d adalah banyaknya piksel pada citra
2. Gunakan skenario percobaan pada seksi 4.1.1, untuk mendapat data citra yang telah dipetakan ke ruang eigen menggunakan *Principal Component Analysis*, serta hasil pengenalannya pada *Fuzzy-Neuro Learning Vector Quantization*.
3. Gunakan skenario percobaan pada seksi 4.1.2, untuk mendapat data citra yang telah dipetakan ke ruang eigen menggunakan *Fuzzy Principal Component Analysis*, serta hasil pengenalannya pada *Fuzzy-Neuro Learning*

Dimensi \ Kelas	Wajah1	Wajah2	Wajah3	Wajah4
1	71%	14%	0%	71%
2	71%	14%	0%	71%
3	100%	43%	0%	86%
4	100%	43%	0%	100%
8	86%	14%	71%	71%
16	71%	14%	71%	71%

Tabel 5.1: Hasil Pengenalan Metoda I pada Data Wajah

Vector Quantization.

5.3 HASIL PERCOBAAN

5.3.1 Metoda I

Hasil percobaan menggunakan Metoda I terhadap data wajah dapat dilihat pada tabel 5.1. Tabel tersebut memperlihatkan akurasi pengenalan dari setiap kelompok kelas wajah, berdasarkan banyaknya *principal component* yang digunakan. Kelas wajah dibagi dalam kolom, sedangkan banyaknya dimensi dibagi dalam baris.

Tabel 5.1 menunjukkan bahwa tingkat pengenalan wajah mempunyai keakuratan yang rendah pada dimensi yang kecil. Tingkat pengenalan ini, umumnya ikut naik ketika dimensi data ditambah, hingga dimensi tertentu yang tingkat pengenalannya kembali menurun. Penurunan tingkat pengenalan wajah pada dimensi-dimensi besar, menunjukkan bahwa dimensi-dimensi tersebut memiliki informasi yang *redundant*, yang mempersulit FNLVQ dalam melakukan pengenalan.

Dimensi \ Kelas	Wajah1	Wajah2	Wajah3	Wajah4
1	86%	43%	0%	86%
2	86%	43%	43%	100%
3	86%	71%	43%	100%
4	100%	57%	43%	100%
8	100%	14%	43%	100%
16	86%	14%	43%	100%

Tabel 5.2: Hasil Pengenalan Metoda II pada Data Wajah

5.3.2 Metoda II

Hasil percobaan menggunakan Metoda II pada data wajah dapat terlihat pada tabel 5.2. Seperti percobaan sebelumnya, tabel 5.2 ini, terdiri dari baris-baris yang menunjukkan banyaknya dimensi yang dipergunakan serta kolom-kolom yang menunjukkan kelompok-kelompok kelas dari data wajah.

5.3.3 Rata-rata Pengenalan Data Wajah

Rata-rata pengenalan pada percobaan ini dapat dilihat pada tabel 5.3. Berdasarkan tabel tersebut, terlihat bahwa tingkat pengenalan masih berada dibawah 70%. Hal ini menunjukkan tingkat pengenalan yang masih kurang bagus. Penambahan dimensi, pada percobaan ini, belum tentu menghasilkan peningkatan pengenalan. Hal ini terlihat pada perbandingan data berdimensi 8 dan 16 pada kolom PCA serta data berdimensi 4 dan 8 pada kolom FPCA.

Dimensi	PCA	FPCA
1	0.54	0.54
2	0.43	0.57
3	0.57	0.64
4	0.61	0.64
8	0.61	0.61
16	0.57	0.61

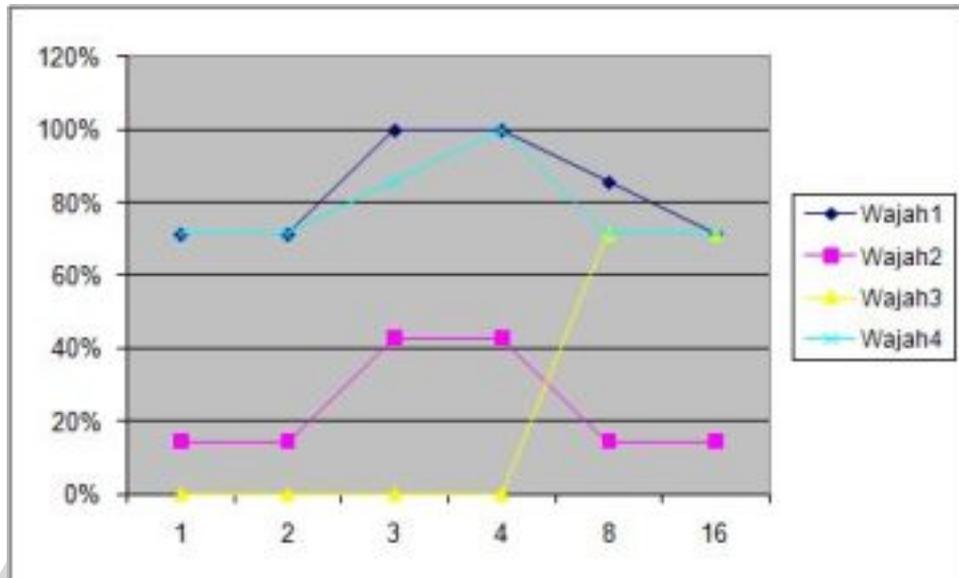
Tabel 5.3: Hasil Pengenalan Citra pada Kedua Metoda

5.4 ANALISA HASIL PERCOBAAN

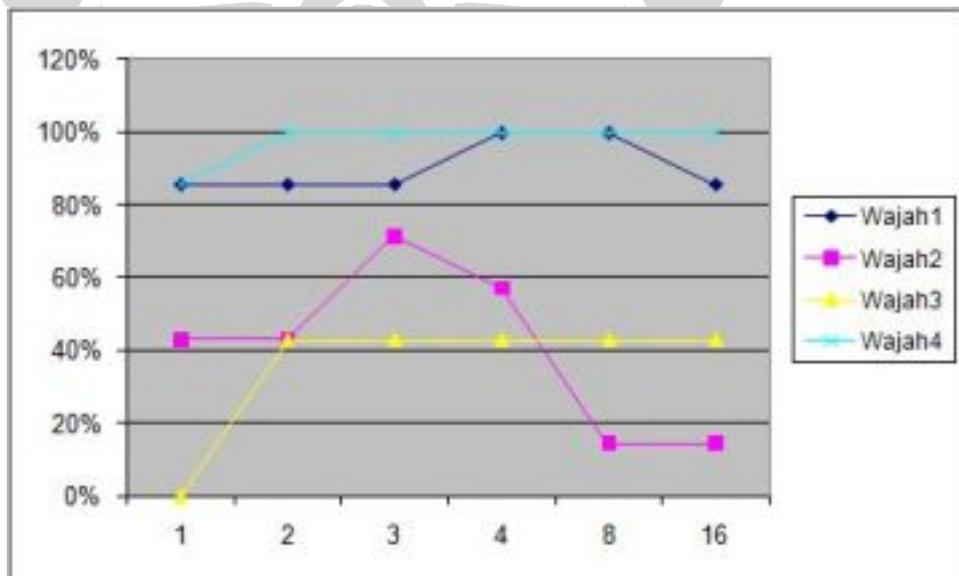
Analisa yang dilakukan terhadap data wajah terbatas pada tingkat pengenalan data wajah oleh *Fuzzy-Neuro Learning Vector Quantization* setelah data wajah tersebut ditransformasikan ke ruang eigen. Analisa ini bertujuan untuk melihat bagaimana data yang dihasilkan oleh kedua metoda dapat mempengaruhi tingkat pengenalan dari sistem *Fuzzy-Neuro Learning Vector Quantization*.

5.4.1 Evaluasi Tingkat Pengenalan Data Wajah

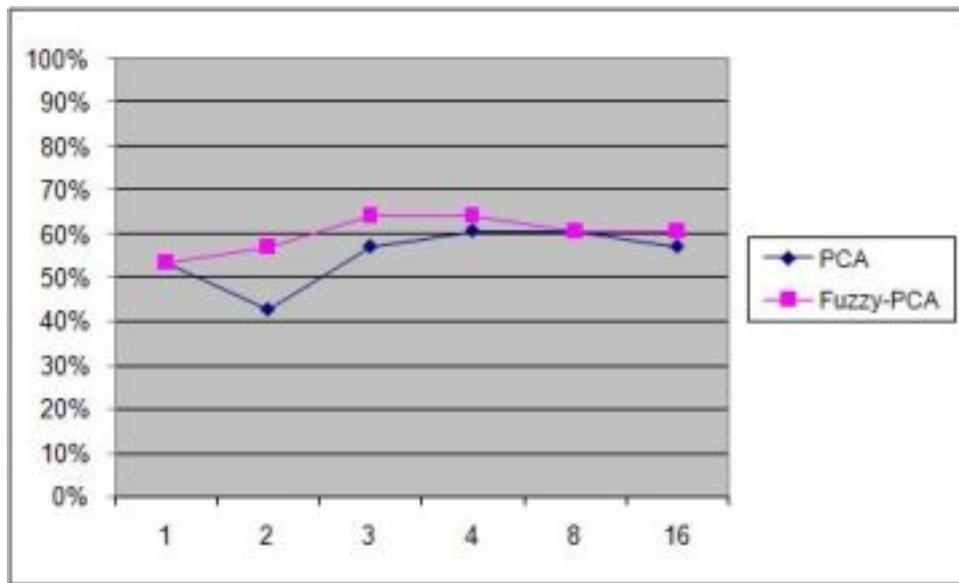
Percobaan dengan menggunakan data wajah, memberikan hasil yang berbeda apabila dibandingkan dengan percobaan menggunakan data aroma. Gambar 5.2 dan 5.3 secara berturut-turut memberikan gambaran tingkat pengenalan data wajah dengan menggunakan kedua metode tersebut. Berdasarkan gambar tersebut, dapat terlihat bahwa banyaknya dimensi berpengaruh terhadap tingkat pengenalan data wajah. Dimensi yang sangat kecil menyebabkan tingkat pengenalan menjadi rendah. Hal ini disebabkan oleh kurangnya data yang dibutuhkan untuk dapat mengenali data wajah tersebut. Penambahan dimensi tidak selalu meningkatkan keakuratan pengenalan. Hal ini terjadi karena data telah mencapai tingkat dimensi yang dibutuhkan untuk bisa membedakan antar satu kelas wajah dengan kelas lainnya, sehingga penambahan dimensi justru membuat FNLVQ kesulitan untuk melakukan pengenalan karena tingkat kerumitan data menjadi semakin tinggi.



Gambar 5.2: Tingkat Pengenalan Data Wajah pada Metoda PCA



Gambar 5.3: Tingkat Pengenalan Data Wajah pada Metoda Fuzzy PCA



Gambar 5.4: Perbandingan Tingkat Pengenalan Data Wajah

5.4.2 Evaluasi Perbandingan Pengenalan Data Wajah

Gambar 5.4 menunjukkan perbandingan tingkat pengenalan pada FNLVQ, setelah data wajah mengalami pengurangan metode menggunakan dua metoda yang telah diajukan pada penelitian ini. Gambar tersebut memperlihatkan metoda pertama, menghasilkan data yang lebih sulit dikenali oleh Sistem FNLVQ, apabila dibandingkan dengan metoda kedua. Hal ini terlihat dari tingkat pengenalannya yang lebih rendah. Berdasarkan gambar tersebut dapat diambil simpulan bahwa penggunaan AANN sebagai FPCA (METODA II) memberikan hasil yang lebih baik dibandingkan penggunaan PCA pada Metoda I.