

## BAB 3 Modifikasi FNLVQ

Pada bab ini akan diuraikan metode pengklasifikasian baru yang merupakan penggabungan metode FNLVQ dengan *Particle Swarm Optimization* (PSO), yaitu metode FNLVQ-PSO serta *Swarm-FNLVQ*. Pada penelitian FNLVQ pada bab sebelumnya diketahui bahwa tingkat pengenalan FNLVQ untuk data aroma tiga campuran serta konfigurasi pelatihan 20:80 masih kurang baik, juga dengan validasi silang diketahui bahwa tingkat pengenalan dengan inisialisasi vektor referensi awal yang kurang baik dapat mengurangi tingkat pengenalan FNLVQ

### 3.1 Metode FNLVQ - PSO

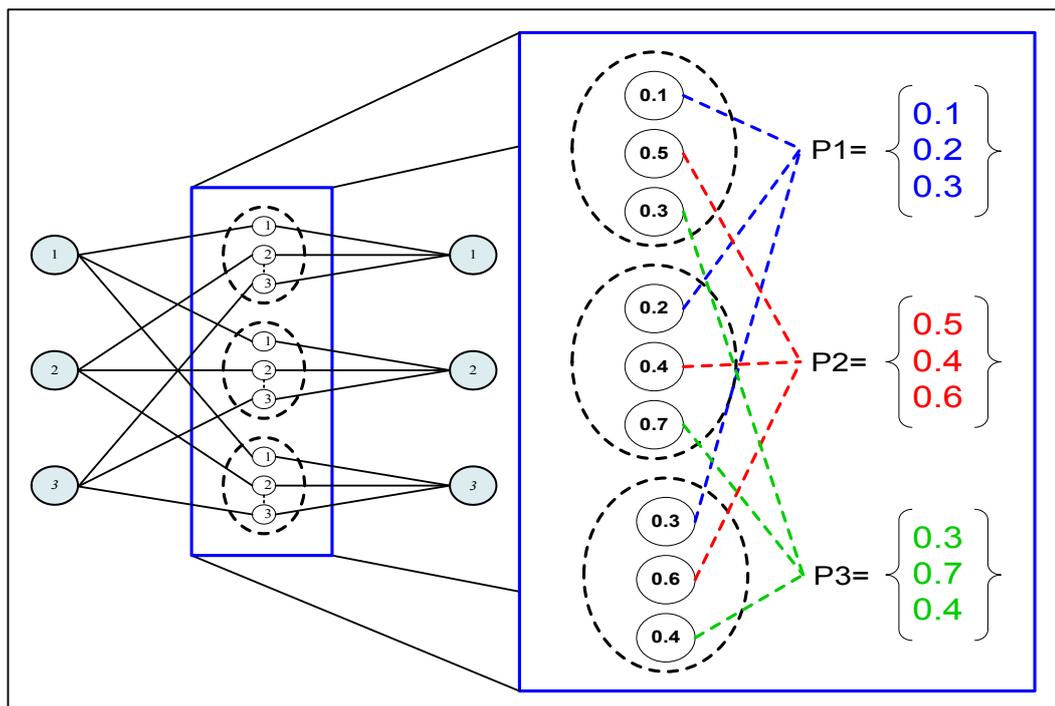
Untuk lebih meningkatkan kinerja dari FNLVQ dikembangkan metode FNLVQ-PSO, metode ini menggabungkan beberapa metode yaitu metode FNLVQ, MSA, serta PSO. Pada metode ini yang diambil dari FNLVQ adalah kecepatan pelatihannya, sedangkan MSA digunakan untuk menentukan nilai *fitness*, kemudian PSO digunakan untuk melakukan perubahan posisi vektor fuzzy agar menghasilkan posisi yang paling optimal.

Metode FNLVQ-PSO berusaha mengurangi ketergantungan FNLVQ terhadap inisialisasi vektor awal yang harus baik, hal ini dilakukan dengan membentuk vektor perwakilan secara acak sebanyak partikel yang dikehendaki. Setelah itu pada tahapan pelatihan; vektor perwakilan-vektor perwakilan tersebut diperbaharui posisinya dengan rumus PSO. Setelah sekian epoch vektor perwakilan tersebut akan semakin rapat satu sama lain dan pada akhirnya akan konvergen. Hal ini merupakan kelebihan dari metode pelatihan PSO yang akan konvergen ke suatu titik solusi permasalahan.

Sebelumnya untuk menggunakan PSO kedalam FNLVQ dibutuhkan pentransformasian permasalahan pada FNLVQ, sehingga FNLVQ dapat diselesaikan menggunakan metode PSO. Caranya adalah dengan menentukan ruang pencarian FNLVQ sehingga bisa dicari nilai optimalnya dengan rumus PSO. Berikut adalah cara yang digunakan dalam penelitian ini untuk menentukan ruang pencarian pada FNLVQ sehingga bisa dioptimalkan dengan PSO.

### 3.1.1 Penentuan Ruang Pencarian PSO pada FNLVQ

Penentuan ruang pencarian PSO untuk dapat dioptimalkan dengan FNLVQ dilakukan dengan cara menggunakan koordinat pada vektor perwakilan di lapisan cluster menjadi koordinat pencarian pada PSO. Kemudian setelah ditentukan, koordinat ini akan dioptimalkan dengan PSO. Untuk lebih jelasnya berikut ilustrasi penggunaan koordinat FNLVQ, untuk dipergunakan sebagai koordinat pencarian pada PSO:

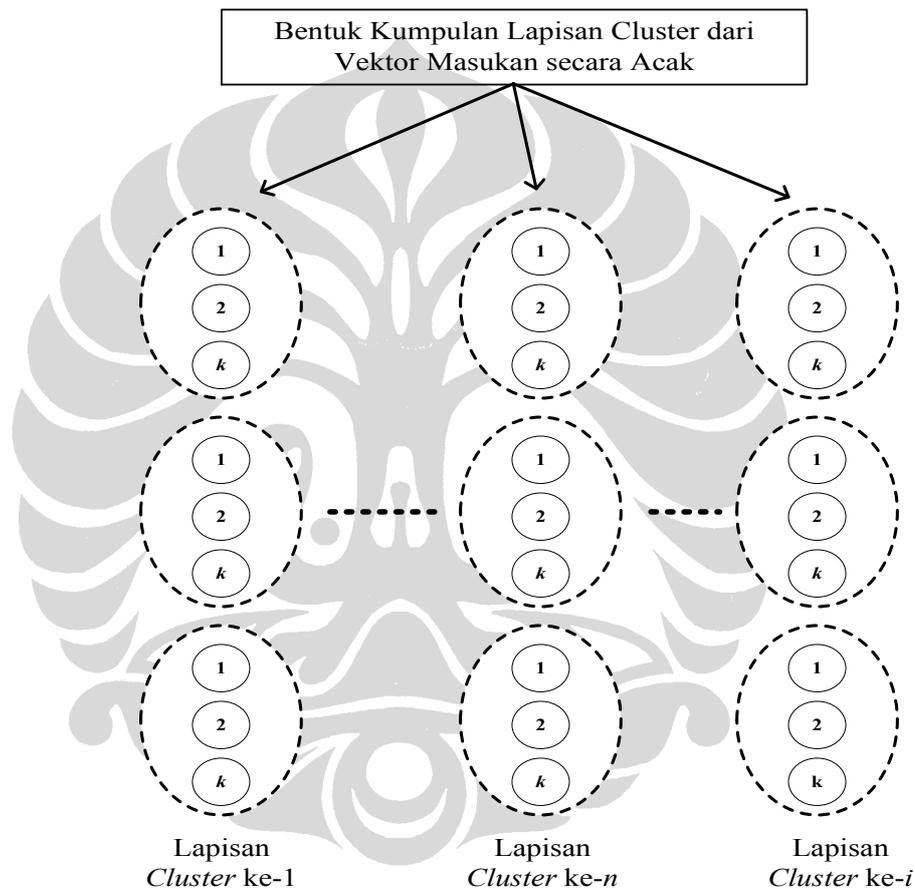


Gambar 3.1 Ilustrasi Penggunaan Koordinat FNLVQ untuk Pencarian dengan PSO

Dapat dilihat pada Gambar 3.1, untuk mencari koordinat diruang pencarian PSO, digunakan koordinat vektor perwakilan pada lapisan *cluster* FNLVQ. Untuk menentukan koordinat tersebut diambil nilai tengah *fuzzy* ( $f$ ) pada vektor perwakilan dari tiap *cluster*. Penentuan partikel dibuat sedemikian sehingga setiap vektor perwakilan pada urutan ke- $i$  akan dijadikan satu partikel dengan vektor perwakilan urutan ke- $i$  di cluster yang berbeda. Hal ini karena tiap vektor perwakilan dengan urutan sama pada cluster yang berbeda memiliki sifat saling mempengaruhi satu sama lain. Sehingga dibutuhkan suatu cara untuk meminimalisir sifat saling mempengaruhi tersebut. Caranya adalah dengan menggunakan MSA sebagai fungsi *fitness* dari FNLVQ-PSO.

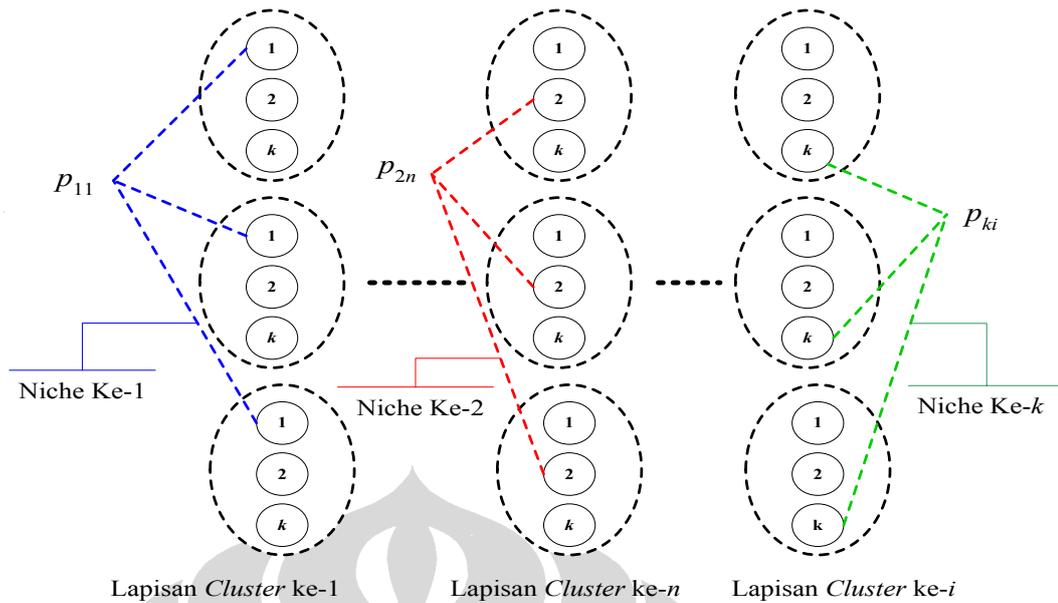
### 3.1.2 Inisialisasi Awal Vektor Perwakilan FNLVQ-PSO

Pada awal pelatihan, FNLVQ-PSO membentuk kumpulan lapisan cluster sebanyak jumlah partikel, kumpulan lapisan cluster ini diambil dari vektor masukan secara acak, hal ini bertujuan agar metode ini dapat menemukan vektor perwakilan awal yang paling baik dalam kumpulan lapisan cluster tersebut, pada Gambar 3.2 diilustrasikan proses pembentukan kumpulan lapisan cluster.



**Gambar 3.2 Ilustrasi Pembentukan Kumpulan Vektor Perwakilan FNLVQ-PSO**

Jaringan FNLVQ-PSO mengadaptasi metode PSO dengan *parallel niche*, yaitu tiap *niche* secara independen mencari solusi permasalahan masing-masing, pada FNLVQ-PSO banyaknya *niche* adalah sebanyak input, untuk lebih jelasnya pada Gambar 3.3 diilustrasikan penentuan partikel serta *niche* pada FNLVQ-PSO:



**Gambar 3.3 Ilustrasi Penentuan Partikel serta Niche pada FNLVQ-PSO**

Pada Gambar 3.3 diperlihatkan ilustrasi penentuan partikel pada jaringan FNLVQ-PSO, misalkan pada jaringan FNLVQ-PSO yang memiliki  $i$  partikel,  $j$  output/kelas dan  $k$  input, maka terdapat  $i \times j \times k$  vektor perwakilan dimana terdapat  $i$  partikel dengan  $j$  cluster yang tiap cluster terdiri dari  $k$  vektor perwakilan. Dengan  $w_{ijk}$  adalah vektor perwakilan pada partikel ke- $i$ , cluster ke- $j$  dan neuron ke- $k$ . Maka partikel ke- $k$  pada *niche* ke- $n$  dapat dinyatakan:

$$p_{nk} = (w_{n1k}, w_{n2k}, w_{n3k}, \dots, w_{nik}) \dots \dots \dots (3.1)$$

Setiap *niche* pada FNLVQ-PSO bertugas untuk mencari solusi permasalahannya masing-masing dengan memilih partikel yang terbaik pada kelompoknya, lalu partikel *niche* yang terbaik inilah yang dianggap sebagai *global best* dari *niche* tersebut, kemudian partikel ini digunakan untuk memperbaharui kumpulan vektor perwakilan/lapisan cluster FNLVQ sesungguhnya. Pada akhir pelatihan akan didapatkan solusi bersama berupa posisi vektor perwakilan yang paling representatif, yang diharapkan mampu mengenali data uji dengan baik.

### 3.1.3 Fungsi Fitness

FNLVQ-PSO menggunakan nilai *fitness* untuk merubah vektor perwakilan. nilai ini didapat melalui fungsi *fitness*, dimana pada FNLVQ-PSO fungsi *fitness*

adalah jumlah nilai diagonal pada MSA tiap partikel dikurangi nilai non-diagonal pada MSA, misalkan untuk matriks similaritas  $\mathbf{M}$  yang memiliki elemen  $m_{ij}$  dengan  $\forall j \in \{1, 2, \dots, n\}$  dan ordo matriks  $n \times n$ , *niche* ke- $h$  serta partikel ke- $k$  maka nilai *fitness* partikel tersebut dihitung dengan rumus :

$$fitness_{hk} = \sum_{i=1}^n m_{ii} - [\sum_{i=1}^n \sum_{j=1}^n m_{ij} \text{ if } i \neq j] \dots\dots\dots(3.2)$$

dimana,

- $fitness_{hk}$  = nilai fitness partikel ke- $k$  pada *niche* ke- $h$
- $m$  = elemen pada matriks  $\mathbf{M}$
- $n$  = banyaknya kelas aroma

Alasan penggunaan fungsi *fitness* ini adalah jika suatu vektor perwakilan memiliki MSA yang nilai diagonalnya besar sedangkan nilai non-diagonalnya kecil maka dapat dikatakan MSA itu semakin mendekati kondisi ideal matriks identitas. Kondisi ini adalah kondisi yang diharapkan pada proses pelatihan, karena tiap vektor perwakilan dengan urutan sama pada cluster yang berbeda, pengaruhnya satu sama lain sudah berkurang.

### 3.1.4 local best dan global best

Setiap *niche* memiliki nilai *global best*-nya masing-masing, nilai *global best* ini kemudian yang digunakan untuk memperbaharui posisi vektor perwakilan sesungguhnya, nilai *global best* didapatkan melalui penghitungan nilai *fitness*, jika suatu partikel pada *niche* memiliki nilai *fitness* terbaik dibandingkan dari partikel-partikel iterasi sebelumnya maka nilai tengah  $f()$  partikel itu akan menjadi *global best*. Pada metode PSO, tiap iterasi pelatihan menghasilkan nilai *local best*, namun pada kasus FNLVQ-PSO nilai *local best* merupakan nilai tengah  $f()$  vektor masukan, Dengan demikian perbedaan metode ini dengan pelatihan FNLVQ konvensional adalah metode ini selain mengarahkan vektor perwakilan kearah vektor masukan juga kearah vektor perwakilan yang terbaik sebelumnya. *Global best* untuk *niche* ke- $n$  dinyatakan dengan rumus :

$$Gbest_n = \max(fitness_n) \dots\dots\dots(3.3)$$

dan vektor perwakilan diperbaharui dengan,

$$W_{i,n} = N_{i,Gbest_n} \dots\dots\dots(3.5)$$

dimana,

$W_{i,n}$  = vektor perwakilan sesungguhnya dengan indeks :

$i$  = urutan lapisan cluster

$n$  = urutan vektor perwakilan pada cluster ke- $i$

$W_{i,Gbest_n}$  = vektor perwakilan pemenang pada pelatihan FNLVQ-PSO dengan indeks:

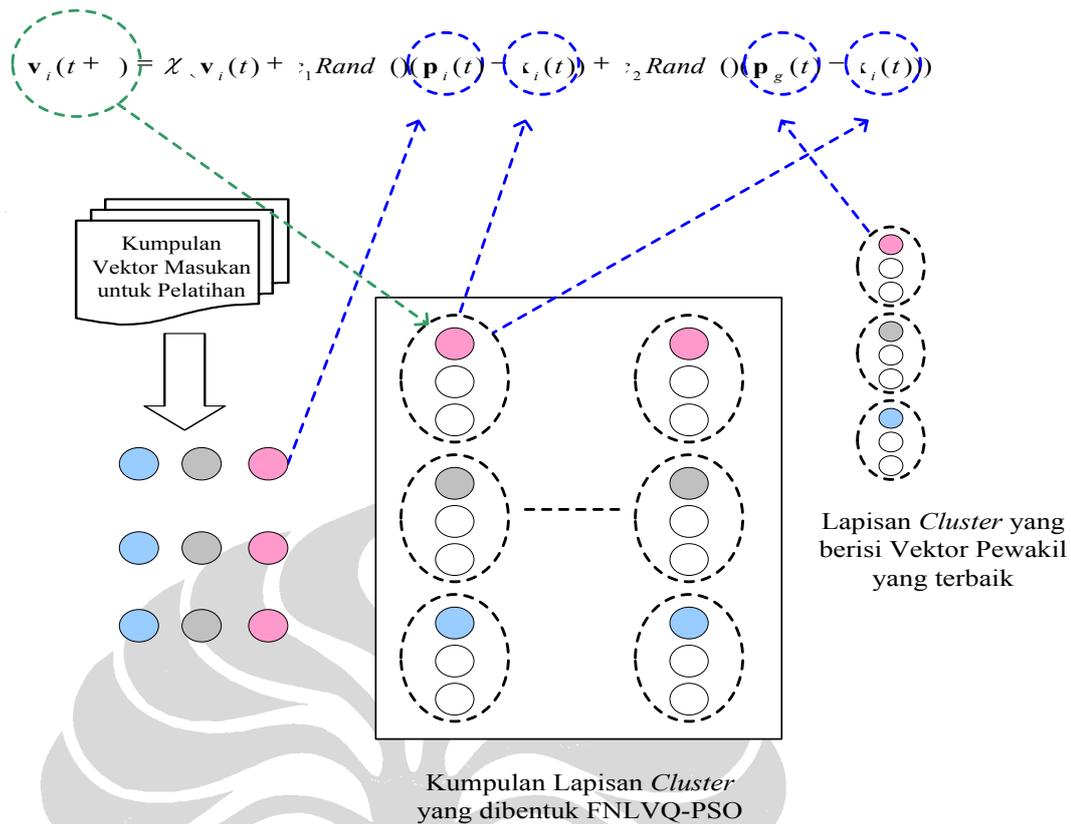
$i$  = urutan lapisan cluster pada partikel

$Gbest_n$  = indeks partikel yang memiliki nilai fitness tertinggi pada niche ke- $n$

Dengan demikian vektor perwakilan sesungguhnya  $W_{i,n}$  pada jaringan FNLVQ diperbaharui dengan nilai partikel  $W_{i,Gbest_n}$ , dimana partikel ini memiliki tingkat pengaruh antara vektor perwakilan yang terendah serta tingkat representasi yang tinggi, sehingga posisinya pada saat itu adalah yang paling baik. Dasar pemikiran dari penggunaan fungsi *fitness* ini adalah jika semakin besar selisih antara nilai diagonal terhadap nilai non-diagonal pada matriks similaritas, maka dapat dikatakan vektor perwakilan dengan matriks similaritas ini memiliki posisi yang paling representatif. Dengan kata lain tingkat ketergantungannya terhadap vektor perwakilan lain semakin mengecil.

### 3.1.5 Penggeseran Vektor Perwakilan menggunakan PSO

Proses penggeseran/perubahan vektor perwakilan pada jaringan FNLVQ-PSO digeser dengan metode PSO, untuk lebih jelasnya proses penggeseran vektor perwakilan dengan diilustrasikan dengan cara ini:



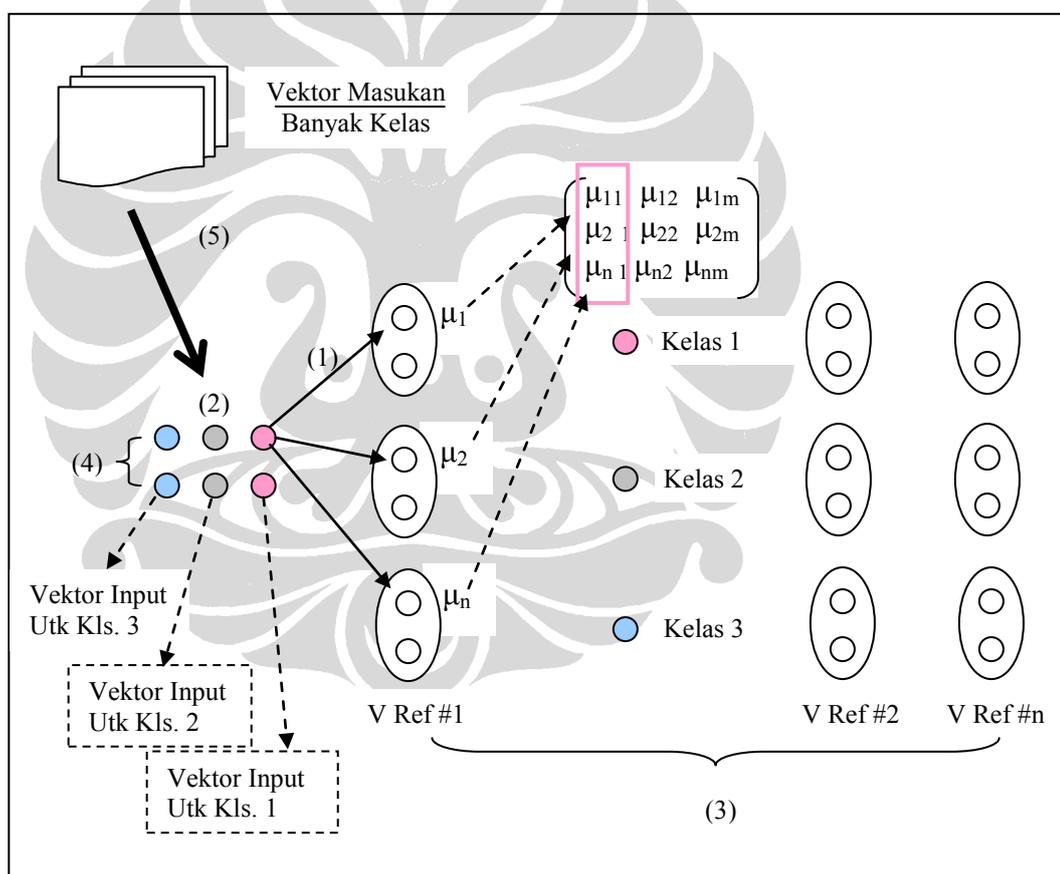
**Gambar 3.4 Ilustrasi Proses Pelatihan FNLVQ dengan PSO**

Pada Gambar 3.4 dapat dilihat proses perubahan/pergeseran nilai partikel pada kumpulan lapisan *cluster*, kecepatan partikel didapat dengan menggunakan rumus (2.25) dengan menggunakan nilai tengah  $f()$  dari vektor masukan (*local best*) serta nilai tengah  $f()$  vektor perwakilan terbaik (*global best*). Kemudian posisi vektor perwakilan diperbaharui dengan menjumlahkan nilai kecepatan partikel dengan nilai tengah  $f()$  vektor perwakilan. Pada PSO apabila faktor kognitif lebih besar, maka pergerakan partikel cenderung mencari solusi terhadap permasalahannya sendiri, namun bila faktor sosial yang tinggi maka partikel bergerak mencari solusi permasalahan bersama. Pada penelitian ini faktor kognitif (*local best*) serta faktor sosial (*global best*) keduanya dibuatimbang, artinya partikel akan bergerak seimbang mengikuti *local* dan *global best*.

### 3.1.6 Pelatihan pada FNLVQ-PSO

Proses pelatihan pada FNLVQ-PSO adalah dengan mencari posisi terbaik dari partikel, Gambar 3.5 memperlihatkan ilustrasi pelatihan FNLVQ-PSO.

Sebelum pelatihan dimulai perlu dilakukan pengurutan vektor masukan secara zig-zag, hal ini agar pada setiap iterasi seluruh kelas pada vektor masukan masuk kedalam proses pelatihan. Pada proses pelatihan metode FNLVQ-PSO mengoptimalkan tiap vektor perwakilan yang berada pada lapisan *cluster* dengan menggunakan rumus PSO, dimana vektor perwakilan dibentuk secara acak sesuai dengan banyaknya partikel. Lalu vektor perwakilan - vektor perwakilan tersebut dibandingkan dengan menggunakan rumus MSA yang kemudian vektor perwakilan yang memiliki nilai MSA terbaik yang dijadikan vektor perwakilan selanjutnya. Banyaknya partikel diatur sesuai dengan keperluan, semakin banyak partikel maka semakin lama proses komputasi FNLVQ-PSO, jika semakin sedikit maka proses komputasinya semakin cepat.



**Gambar 3.5 Ilustrasi Proses Pelatihan FNLVQ dengan PSO**

Berikut adalah tahapan pelatihan dengan menggunakan metode FNLVQ-PSO:

1. FNLVQ-PSO membentuk sejumlah kumpulan lapisan cluster sebanyak yang diinginkan. Pembentukan lapisan cluster ini dilakukan dengan cara mengambil nilai vektor masukan secara acak.

2. Kemudian setiap partikel akan merubah/menggeser posisi vektor perwakilan mendekati vektor masukan seperti telah dijelaskan pada sub bab 3.1.5, dan lebar segitiga *fuzzy* pada vektor perwakilan akan diubah dengan rumus :

$$fl(t+ ) = fl(t) - \{rand () * c * [f(t) - fl(t)]\} \dots\dots\dots(3.4)$$

dan

$$fr(t+ ) = fr(t) + \{rand () * c * [fr(t) - f(t)]\} \dots\dots\dots(3.5)$$

dimana,

$fl(t+ )$  = adalah nilai minimum vektor perwakilan setelah pergeseran.

$fl(t)$  = adalah nilai minimum vektor perwakilan sebelum pergeseran

$fr(t+ )$  = adalah nilai maksimum vektor perwakilan setelah pergeseran.

$fr(t)$  = adalah nilai maksimum vektor perwakilan sebelum pergeseran.

$f(t)$  = adalah nilai tengah vektor perwakilan.

$rand()$  = adalah bilangan acak, dengan kisaran  $0 < s.d. \leq 1$ .

$c$  = adalah nilai konstanta.

3. Lalu partikel akan menghitung nilai similaritas antara vektor masukan terhadap vektor perwakilan yang telah digeser tadi, setelah itu nilai similaritas pada MSA diperbaharui sesuai dengan nilai tersebut.
4. Jika setiap data vektor masukan telah dibandingkan dengan seluruh partikel, maka partikel yang memiliki nilai *fitness* tertinggi adalah partikel pemenang.
5. Vektor perwakilan pada partikel pemenang ini yang akan dijadikan vektor perwakilan iterasi selanjutnya.
6. Lakukan tahapan 2 – 5 sampai seluruh vektor masukan telah dilatihkan.
7. Diakhir pelatihan akan didapatkan vektor perwakilan yang paling representatif sebagai hasil akhir dari pelatihan, pelatihan sendiri akan berakhir bilamana memenuhi kriteria pemberhentian.

## Algoritma FNLVQ-PSO

```

repeat
  for each InputAmount/Output
    for each Input
      Move Particle using equation (2.26) (3.1) (3.2)
      for each Particle
        for each Output
          for each Cluster
            temp = FuzzyIntersection()
            CopyToMSA(temp)
          end for
        end for
      update fitness using equation (3.3)
      find best particle using equation (3.4)
      update vector reference using equation (3.5)
    end for
  end for
end for
until stopping condition is true

```

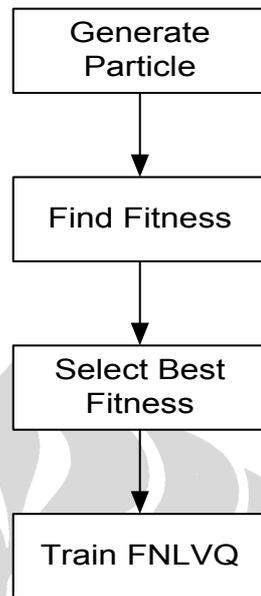
Dengan algoritma FNLVQ-PSO ini dapat ditentukan beberapa kriteria pemberhentian algoritma pada saat pelatihan, berikut adalah kriteria pemberhentian tersebut:

1. Algoritma ini dapat berhenti bilamana nilai *threshold* (ambang batas) dari MSA telah menyentuh nilai yang diinginkan, dengan anggapan bahwa bilamana nilai *threshold* sudah relatif tinggi maka dapat diharapkan nantinya tingkat pengenalan akan tinggi pula.
2. Yang terakhir adalah bilamana epoch untuk pelatihan terlampaui maka algoritma ini akan berhenti dengan sendirinya, meskipun algoritma ini belum menemukan solusi yang optimal.

### 3.2 Metode Swarm-FNLVQ

Metode Swarm-FNLVQ merupakan suatu metode untuk mengoptimasikan vektor perwakilan awal dari metode FNLVQ, sebagaimana terlihat dari Gambar 4.2 tingkat pengenalan FNLVQ untuk aroma tiga campuran dengan menggunakan inisialisasi acak berfluktuatif. Metode Swarm-FNLVQ berusaha mengambil tingkat pengenalan yang terbaik dari FNLVQ dengan cara mengidentifikasi vektor

pewakil yang terbaik dari vektor perwakilan yang ada. Berikut ilustrasi metode Swarm-FNLVQ.



Gambar 3.6 Ilustrasi Pelatihan FNLVQ dengan PSO

Berikut adalah metode pelatihan menggunakan Swarm-FNLVQ:

- Pertama kali metode ini membentuk kumpulan partikel, yang berisi kumpulan lapisan cluster yang dibentuk dari vektor masukan secara acak, dalam hal ini lapisan cluster dianggap sebagai sebuah partikel.
- Pada epoch pelatihan pertama, setiap partikel ini dilatih dengan metode pelatihan FNLVQ konvensional, selanjutnya nilai *fitness* didapatkan melalui penjumlahan nilai similaritas keluaran FNLVQ per-partikel dengan rumus:

$$fitness_k = \sum_{i=1}^n \mu_i \dots\dots\dots(3.6)$$

$fitness_k$  = nilai fitness partikel ke- $k$

$\mu_i$  = nilai similaritas keluaran dari FNLVQ

$n$  = banyaknya vektor masukan pada kelas aroma

- Partikel yang memiliki nilai fitness yang terbaik, diasumsikan sebagai partikel pemenang dengan rumus:

$$p_{win} = \max(fitness) \dots\dots\dots(3.7)$$

$p_{win}$  = partikel pemenang

- Dan selanjutnya partikel/lapisan cluster ini yang digunakan untuk pelatihan FNLVQ pada epoch selanjutnya.

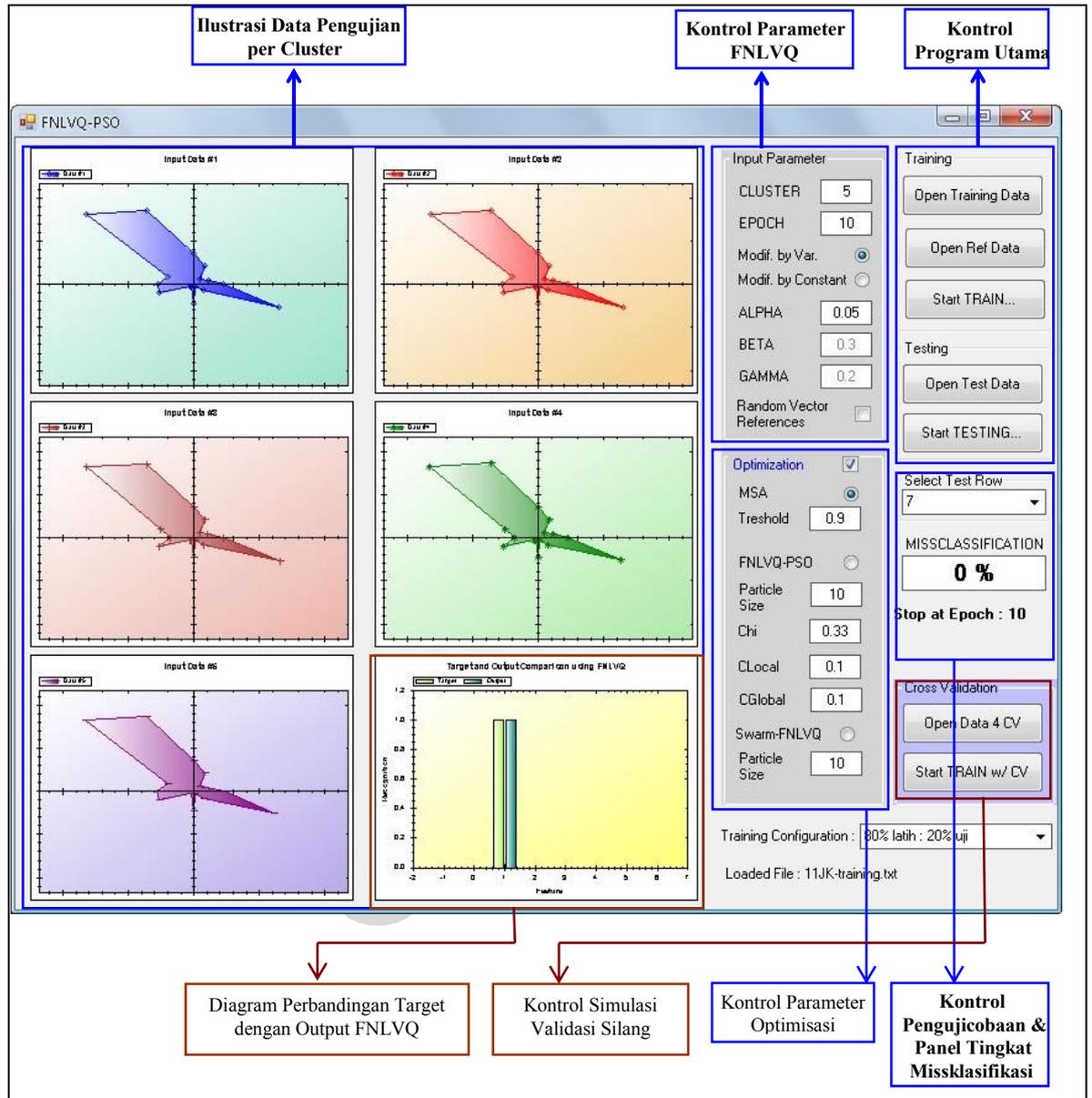
### Algoritma *Swarm-FNLVQ*

```
repeat
  for each Particle
    update fitness using equation (3.6)
    find best particle using equation (3.7)
  end for
  update FNLVQ vector reference using best particle
  continue training using FNLVQ
until stopping condition is true
```



### 3.3 Implementasi FNLVQ – PSO

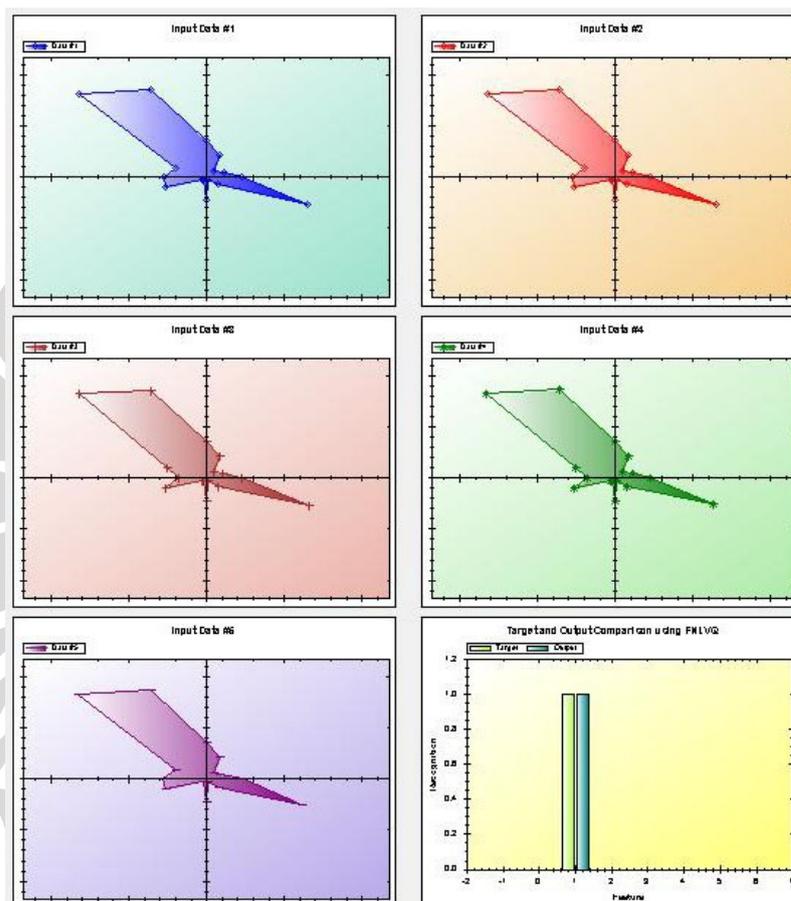
Program simulasi ini dibuat menggunakan bahasa pemrograman Visual C++ NET serta dengan menggunakan *plugin Zedgraph* untuk menampilkan diagram. Berikut tampilan program secara keseluruhan:



**Gambar 3.7 Tampilan Program Simulasi FNLVQ-PSO**

Pada gambar diatas dapat dilihat tampilan secara keseluruhan program simulasi FNLVQ, dimana terdapat kontrol : ”parameter FNLVQ”, ”pemilihan data uji coba”, ”program utama”, ”parameter simulasi validasi silang” . Kontrol ini

berguna untuk mengatur parameter-parameter dari komponen tersebut. Selain itu terdapat panel tingkat misklasifikasi yang berisi tingkat sukses pengenalan terhadap data uji coba, berdasarkan parameter-parameter yang telah dimasukkan sebelumnya. Untuk memperlihatkan bentuk input data untuk tahapan uji coba serta perbandingan antara hasil keluaran FNLVQ dengan target sesungguhnya ditampilkan pula “diagram perbandingan target dengan output FNLVQ”.



**Gambar 3.8 Visualisasi Data Aroma serta Pengenalannya dengan GUI**

Pada Gambar 3.8 dapat dilihat visualisasi data aroma serta pengenalannya yang telah diprogram menggunakan visual C++, pada gambar tersebut terlihat lima pola aroma berupa diagram radial, dimana pola aroma tersebut terdiri dari 16 titik/sensor yang nilainya berkisar antara 0 sampai dengan 1. Pada contoh gambar diatas terdapat 5 data/cluster aroma, dimana lima data aroma ini yang akan difuzzifikasikan, banyaknya cluster untuk aroma adalah bebas, khusus pada penelitian ini digunakan 5 cluster. Sedangkan pada gambar di pojok kanan bawah

adalah diagram batang pengenalan antara keluaran pada pelatihan jaringan FNLVQ dengan target sesungguhnya.

Input Parameter	
CLUSTER	5
EPOCH	10
Modif. by Var.	<input type="radio"/>
Modif. by Constant	<input checked="" type="radio"/>
ALPHA	0.05
BETA	0.3
GAMMA	0.2
Random Vector References	<input type="checkbox"/>

**Gambar 3.9 Kontrol Parameter FNLVQ**

Pada kontrol parameter FNLVQ dapat dilakukan pengaturan banyaknya cluster untuk pengelompokan data input menjadi data fuzzy, banyaknya epoh, untuk tahapan pembelajaran dapat dipilih perubahan dengan variabel maupun dengan konstan, untuk penentuan vektor perwakilan dapat dibentuk secara acak atau dengan mengambil dari file.

Optimization	
Optimization	<input checked="" type="checkbox"/>
MSA	<input checked="" type="radio"/>
Treshold	0.9
FNLVQ-PSO	<input type="radio"/>
Particle Size	10
Chi	0.33
CLocal	0.1
CGlobal	0.1
Swarm-FNLVQ	<input type="checkbox"/>
Particle Size	10

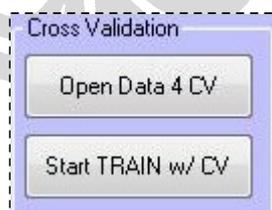
**Gambar 3.10 Kontrol Optimisasi Parameter FNLVQ**

Untuk menentukan tipe optimisasi pada FNLVQ dapat dipergunakan MSA dengan memasukkan *threshold* pemberhentiannya, maupun PSO dengan memasukkan nilai variabel: *particle size*, *chi* dan konstanta untuk *local best* dan *global best*. Selain itu juga bisa memilih menggunakan optimisasi *Swarm-FNLVQ* dengan memasukkan banyak partikel yang akan dicari oleh *swarm* untuk dicari yang terbaik.



**Gambar 3.11 Kontrol Program Utama**

Untuk melakukan pelatihan jaringan dengan FNLVQ pada kontrol program utama; pertama-tama diambil terlebih dahulu data pelatihan dengan tombol "Open Train File", kemudian klik "Open Ref Data" lalu "Train". Untuk melakukan pengujian maka pertama-tama diambil terlebih dahulu data pengujian dengan tombol "Open Test Data", kemudian klik "Start Testing".



**Gambar 3.12 Kontrol Simulasi Validasi Silang**

Untuk melakukan pelatihan sekaligus pengujian dengan menggunakan validasi silang; pertama-tama diambil terlebih dahulu data aroma dengan tombol "Open Data 4 CV", untuk melakukan pelatihan sekaligus pengujian klik "Start TRAIN w/ CV".

## BAB 4 Analisis Pengenalan Aroma

Bab ini akan membahas tingkat pengenalan aroma menggunakan FNLVQ-PSO dan *Swarm*-FNLVQ dibandingkan dengan FNLVQ serta FNLVQ-MSA. Selain itu dibandingkan pula tingkat pengenalan metode diatas dengan menggunakan validasi silang.

### 4.1 Hasil Pengenalan Aroma Menggunakan FNLVQ

Pada penelitian ini digunakan CPU dengan kecepatan 2.4 GHz serta memory sebesar 4 GB. Data yang digunakan dalam penelitian ini adalah data dalam bentuk file TXT, yang berisikan data aroma yang telah dinormalisasi antara 0 sampai dengan 1. Data aroma yang digunakan dalam penelitian ini terdiri dari campuran dua aroma dasar serta Alkohol yang berkadar 0%, 15%, 25%, 35%, 45% dan 70% sebagai data masukan. Berikut adalah tabel jenis aroma tiga campuran yang digunakan dalam uji coba penelitian ini.

Tabel 4.1 Jenis Aroma Tiga Campuran yang Digunakan dalam Penelitian

No	Jenis Aroma	Singkatan	Banyak Kelas Keluaran
1	Jeruk-Kenanga+Alkohol 0%,15%,25%,35%,45%,75% = JKAlk	JK	6
2	Jeruk-Mawar+Alkohol 0%,15%,25%,35%,45%,75% = JMAIk	JM	6
3	Mawar-Kenanga+Alkohol 0%,15%,25%,35%,45%,75% = MKAlk	MK	6
4	JKAlk + JMAIk	JK-JM	12
5	JKAlk + MKAlk	JK-MK	12
6	JMAIk + MKAlk	JM-MK	12
7	JKAlk + JMAIk + MKAlk	JK-JM-MK	18

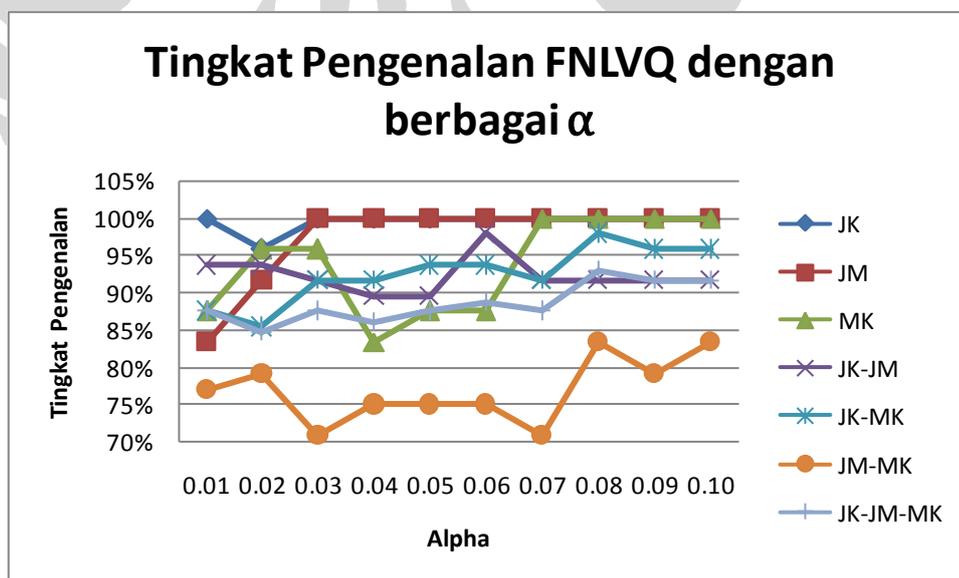
Perbandingan tingkat pengenalan FNLVQ untuk aroma tiga campuran dengan menggunakan jenis  $\alpha$  yang berbeda ditampilkan oleh Tabel 4.2. Pada percobaan ini digunakan parameter epoch = 100, inialisasi vektor masukan awal, konfigurasi data latih 80% dan data uji 20%. Hasilnya dapat dilihat, tingkat pengenalan FNLVQ rata-rata tertinggi didapat bila nilai  $\alpha = 0,08$  atau 0,1 yaitu  $\pm 95\%$ .

Oleh karena tingkat pengenalan FNLVQ didapatkan nilai yang tertinggi jika diberi nilai  $\alpha = 0.1$ , maka penelitian selanjutnya menggunakan nilai alpha 0.1 sebagai parameter pelatihannya.

**Tabel 4.2 Tingkat Pengenalan Aroma Tiga Campuran menggunakan FNLVQ dengan berbagai tingkat  $\alpha$**

$\alpha$	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK	Avg.
0.01	100%	83%	88%	94%	88%	77%	88%	88%
0.02	96%	92%	96%	94%	85%	79%	85%	89%
0.03	100%	100%	96%	92%	92%	71%	88%	91%
0.04	100%	100%	83%	90%	92%	75%	86%	89%
0.05	100%	100%	88%	90%	94%	75%	88%	90%
0.06	100%	100%	88%	98%	94%	75%	89%	92%
0.07	100%	100%	100%	92%	92%	71%	88%	92%
0.08	100%	100%	100%	92%	98%	83%	93%	95%
0.09	100%	100%	100%	92%	96%	79%	92%	94%
0.10	100%	100%	100%	92%	96%	83%	92%	95%
<b>Avg.</b>	<b>100%</b>	<b>97%</b>	<b>94%</b>	<b>92%</b>	<b>92%</b>	<b>77%</b>	<b>89%</b>	<b>92%</b>

Untuk lebih jelasnya Tabel 4.2 dapat dirubah dalam bentuk grafik, seperti terlihat dalam Gambar 4.1.



**Gambar 4.1 Tingkat Pengenalan FNLVQ untuk Aroma Tiga Campuran dengan Nilai  $\alpha$  Bervariasi**

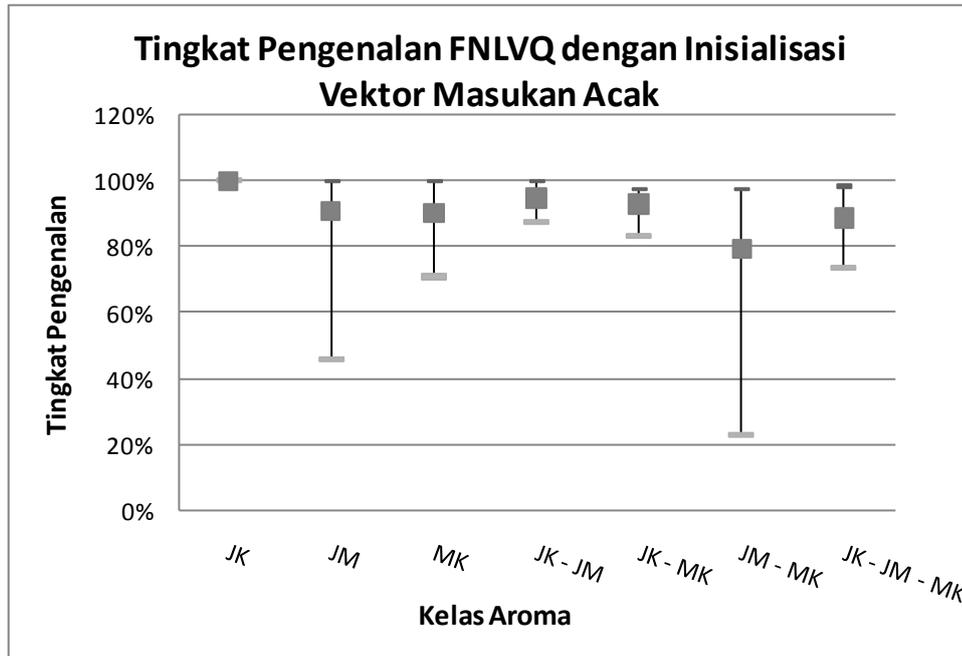
Dari hasil Gambar 4.1 diatas dapat disimpulkan bahwa Aroma campuran JMAlk – MKAlk kurang dapat dikenali (tingkat pengenalan dibawah 90%), meskipun telah dicoba dengan nilai  $\alpha$  yang berbeda-beda.

Dalam penelitian ini juga dilakukan percobaan tingkat pengenalan FNLVQ menggunakan inisialisasi vektor perwakilan acak serta inisialisasi vektor perwakilan yang telah ditentukan. Kedua inisialisasi awal percobaan tersebut mengambil vektor perwakilan dari vektor masukan. Untuk kondisi yang pertama vektor perwakilan awal diambil secara acak dari vektor masukan, sedangkan kondisi yang kedua vektor perwakilan diambil dari vektor masukan urutan pertama. Percobaan ini dilakukan untuk menentukan jenis inisialisasi apakah yang terbaik untuk FNLVQ. Berikut adalah hasil perbandingan tingkat pengenalan FNLVQ menggunakan dua macam tipe inisialisasi vektor perwakilan.

**Tabel 2.2 Tingkat Pengenalan Aroma Tiga Campuran menggunakan FNLVQ dengan Inisialisasi Acak Diujicobakan sebanyak 10 kali Percobaan**

	1	2	3	4	5	6	7	8	9	10	Avg.
<b>JK</b>	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	<b>100%</b>
<b>JM</b>	100%	100%	46%	88%	96%	100%	100%	100%	83%	100%	<b>91%</b>
<b>MK</b>	96%	100%	79%	100%	100%	79%	96%	100%	71%	83%	<b>90%</b>
<b>JK - JM</b>	98%	94%	98%	88%	90%	100%	94%	100%	96%	94%	<b>95%</b>
<b>JK - MK</b>	96%	96%	92%	90%	96%	96%	98%	83%	94%	94%	<b>93%</b>
<b>JM - MK</b>	79%	79%	90%	79%	81%	98%	23%	92%	92%	85%	<b>80%</b>
<b>JK - JM - MK</b>	99%	97%	90%	94%	82%	88%	74%	94%	94%	78%	<b>89%</b>

Pada Tabel 4.2 dapat dilihat tingkat pengenalan FNLVQ dengan inisialisasi acak sangat fluktuatif, hal ini menjadi bukti bahwa tinggi rendahnya tingkat pengenalan FNLVQ tergantung terhadap penentuan vektor perwakilan awal. Untuk lebih jelasnya Tabel 4.2 dapat dirubah dalam bentuk grafik, seperti terlihat dalam Gambar 4.2.



**Gambar 4.2 Tingkat Pengenalan FNLVQ dengan Inisialisasi Vektor Masukan Acak**

Tidak demikian halnya dengan tingkat pengenalan FNLVQ dengan inisialisasi vektor masukan awal, hasil pengenalan FNLVQ dengan inisialisasi ini konsisten. Pada tabel berikut akan diperlihatkan perbandingan tingkat pengenalan FNLVQ inisialisasi vektor masukan awal dengan acak:

**Tabel 4.3 Perbandingan Tingkat Pengenalan FNLVQ antara Inisialisasi Vektor Masukan Acak dengan Vektor Masukan Awal**

Aroma	Inisialisasi Vektor Perwakilan	
	Awal	Acak
JK	100%	100%
JM	100%	91%
MK	88%	90%
JK - JM	90%	95%
JK - MK	94%	93%
JM - MK	75%	80%
JK - JM - MK	88%	89%
<b>Rata-rata</b>	<b>90%</b>	<b>91%</b>

Dari Tabel 4.3 dapat disimpulkan bahwa tingkat pengenalan FNLVQ dengan menggunakan inisialisasi vektor perwakilan acak secara rata-rata lebih baik, namun tingkat pengenalan tersebut tidak konsisten, sehingga dalam penelitian selanjutnya penulis menggunakan inisialisasi vektor perwakilan dengan vektor masukan urutan pertama (inisialisasi awal).

Agar hasil pengenalan lebih akurat maka proporsi data pelatihan dikonfigurasi sedemikian rupa; dimana konfigurasi ini adalah perbandingan persentase data latih terhadap data uji yang digunakan untuk pelatihan FNLVQ. Hal ini bertujuan untuk memperbanyak variasi dalam percobaan, untuk kemudian dianalisis. Berikut konfigurasi pelatihan dalam penelitian ini:

**Tabel 4.4 Konfigurasi Pelatihan yang Digunakan dalam Penelitian**

No	Data Latih	Data Uji	Singkatan
1	80%	20%	80 : 20
2	70%	30%	70 : 30
3	60%	40%	60 : 40
4	50%	50%	50 : 50
5	40%	60%	40 : 60
6	30%	70%	30 : 70
7	20%	80%	20 : 80

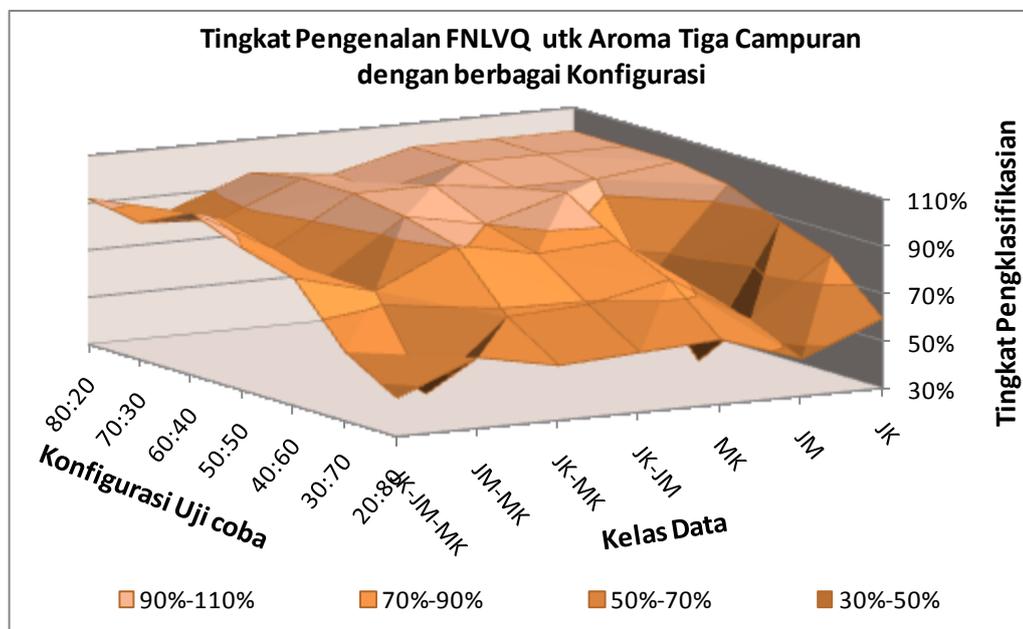
Percobaan yang dilakukan dengan konfigurasi ini menggunakan parameter  $\alpha = 0.1$  dan Epoch = 100, penelitian ini menggunakan vektor perwakilan yang diambil dari urutan pertama vektor masukan, hal ini dikarenakan tingkat pengenalan FNLVQ yang menggunakan metode inisialisasi vektor perwakilan awal secara acak tidak konsisten. Berikut hasil dari percobaan ini:

**Tabel 4.5 Tingkat Pengenalan FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan**

Aroma Konfig.	Aroma							Rata-Rata
	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK	
<b>80:20</b>	100%	100%	100%	92%	96%	83%	92%	<b>95%</b>
<b>70:30</b>	100%	100%	100%	92%	100%	86%	88%	<b>95%</b>
<b>60:40</b>	100%	96%	96%	100%	100%	80%	99%	<b>96%</b>
<b>50:50</b>	97%	60%	100%	90%	97%	73%	90%	<b>87%</b>
<b>40:60</b>	88%	32%	86%	84%	91%	75%	84%	<b>77%</b>
<b>30:70</b>	80%	56%	76%	69%	67%	38%	59%	<b>64%</b>
<b>20:80</b>	59%	46%	57%	55%	53%	60%	46%	<b>54%</b>
<b>Rata-Rata</b>	<b>89%</b>	<b>70%</b>	<b>88%</b>	<b>83%</b>	<b>86%</b>	<b>71%</b>	<b>80%</b>	<b>81%</b>

Dari Tabel 4.5 dapat dilihat rata-rata tingkat pengenalan tertinggi untuk FNLVQ adalah pada konfigurasi pelatihan 60% latih 40% uji (60:40), sedangkan yang terendah adalah pada konfigurasi pelatihan 20% latih 80% uji (80:20), tingkat pengenalan yang rendah ini dikarenakan data pelatihan yang kurang

representatif sehingga kurang mampu mengenali suatu kelas, di dunia nyata data yang didapatkan tidak selalu representatif oleh karena itu dalam penelitian ini dipergunakan semua kemungkinan konfigurasi pelatihan.



**Gambar 4.3** Tingkat Pengenalan FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pengujian

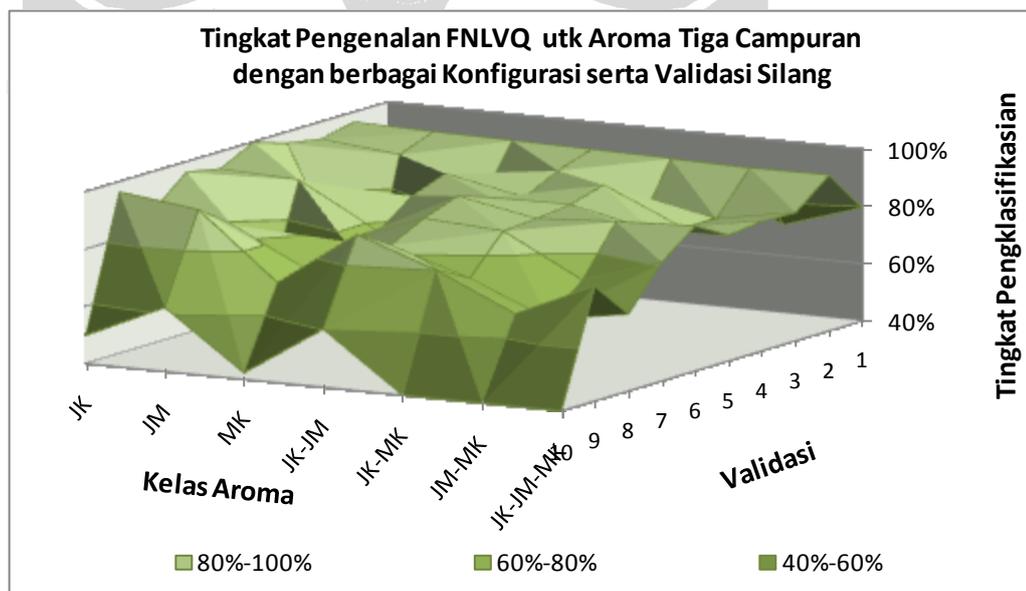
Gambar 4.3 memperlihatkan tingkat pengenalan FNLVQ terhadap data aroma tiga campuran, disini dapat dilihat jika semakin sedikit persentase data pelatihan yang dipergunakan maka tingkat pengenalan secara rata-rata akan menurun, juga dapat dilihat bahwa aroma campuran JMAIk serta JMAIk + MKAlk kurang bisa dikenali oleh FNLVQ. Untuk memperbaiki tingkat pengenalan FNLVQ terutama pada konfigurasi pelatihan 20:80, maka dikembangkan metode FNLVQ-PSO serta *Swarm-FNLVQ*. Metode ini akan memaksimalkan metode FNLVQ dengan bantuan metode *Swarm Intelligence*, lebih lanjut metode ini akan dibahas pada bab 3.

Percobaan dengan menggunakan validasi silang menghasilkan tingkat pengenalan yang berbeda pada tiap sub-bagian data. Pada Tabel 4.6 dapat dilihat tingkat pengenalan data aroma tiga campuran menggunakan FNLVQ pada tiap sub-bagian berbeda satu sama lain, dengan nilai rata-rata tertinggi sebesar 95% pada sub-bagian ke-2 dan terendah 46% pada sub-bagian ke-10.

**Tabel 4.6 Tingkat Pengenalan FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan serta Validasi Silang**

Aroma	CV	1	2	3	4	5	6	7	8	9	10	Avg
	JK	89%	96%	94%	96%	100%	94%	96%	86%	96%	50%	
JM	70%	96%	91%	37%	42%	93%	53%	84%	93%	62%	<b>72%</b>	
MK	88%	95%	76%	91%	86%	78%	78%	70%	70%	42%	<b>77%</b>	
JK-JM	83%	95%	91%	85%	89%	93%	88%	83%	90%	60%	<b>86%</b>	
JK-MK	86%	95%	73%	92%	89%	88%	87%	74%	80%	34%	<b>80%</b>	
JM-MK	71%	94%	80%	80%	83%	89%	72%	62%	68%	35%	<b>73%</b>	
JK-JM-MK	80%	94%	85%	85%	84%	91%	81%	67%	80%	37%	<b>78%</b>	
<b>Rata-Rata</b>	<b>81%</b>	<b>95%</b>	<b>84%</b>	<b>81%</b>	<b>82%</b>	<b>89%</b>	<b>79%</b>	<b>75%</b>	<b>82%</b>	<b>46%</b>	<b>80%</b>	

Selanjutnya data pada Tabel 4.6 dapat dilihat lebih lanjut dengan Gambar 4.4, pada gambar ini didapatkan bahwa tingkat pengenalan FNLVQ semuanya turun pada sub-bagian data ke-10. Oleh karena dalam percobaan ini menggunakan inisialisasi awal, maka hal ini mengindikasikan bahwa data vektor perwakilan awal pada sub-bagian ini kurang baik.



**Gambar 4.4 Tingkat Pengenalan FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan serta Validasi Silang**

Dari percobaan dengan validasi silang dapat disimpulkan bahwa perbedaan tingkat pengenalan FNLVQ setidaknya tergantung dari dua hal:

1. Hal pertama yang menentukan perbedaan tingkat pengenalan FNLVQ adalah kerepresentatifan vektor perwakilan awal, artinya pada akhir pelatihan vektor perwakilan awal yang representatif mampu mengenali vektor masukan

dengan sempurna, jika vektor perwakilan awal kurang baik maka diakhir pelatihan FNLVQ kurang bisa mengenali data uji.

2. Kedua, keteraturan pola data, artinya bilamana data yang dilatihkan polanya tidak teratur, maka sulit untuk mendapatkan vektor perwakilan akhir yang representatif, sehingga pengenalan terhadap data tersebut akan kurang baik .

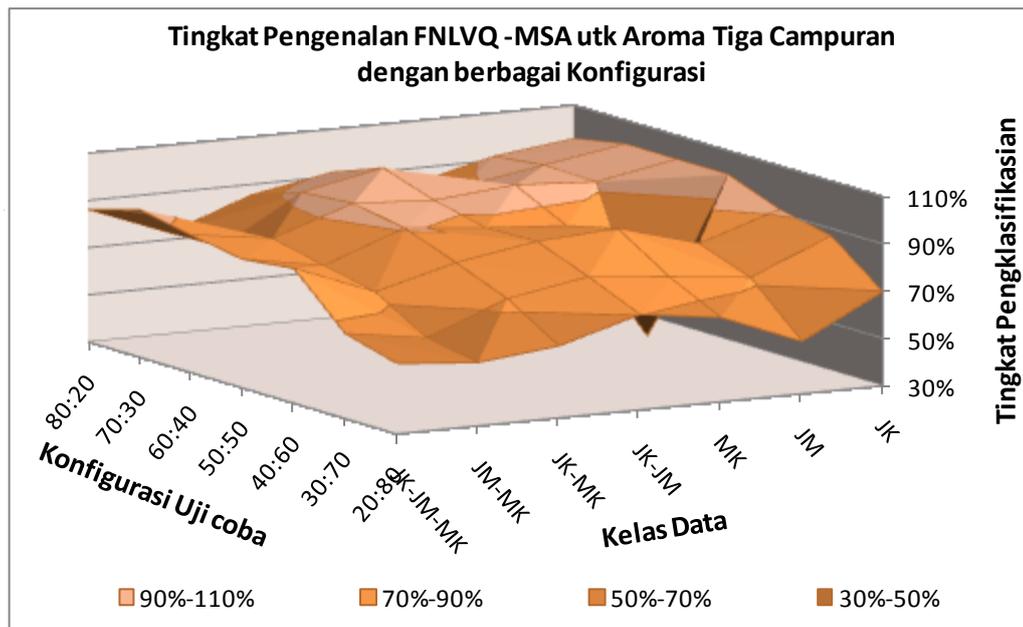
## 4.2 Hasil Pengenalan Aroma Menggunakan FNLVQ-MSA

Untuk pengujian aroma dengan menggunakan FNLVQ-MSA, parameter yang digunakan adalah :  $\alpha = 0.1$ , epoch = 100 dan MSA *threshold* = 0.9. Untuk FNLVQ juga digunakan inisialisasi vektor perwakilan urutan awal vektor masukan, alasannya sama dengan sebelumnya yaitu agar tingkat pengenalannya konsisten.

**Tabel 4.7 Tingkat Pengenalan FNLVQ-MSA untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan**

Aroma Konfig.	Aroma							Rata-Rata
	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK	
<b>80:20</b>	96%	92%	83%	92%	85%	73%	86%	<b>87%</b>
<b>70:30</b>	100%	94%	89%	100%	93%	74%	93%	<b>92%</b>
<b>60:40</b>	100%	94%	96%	93%	95%	83%	88%	<b>93%</b>
<b>50:50</b>	100%	35%	85%	88%	91%	78%	84%	<b>80%</b>
<b>40:60</b>	92%	81%	90%	88%	83%	74%	87%	<b>85%</b>
<b>30:70</b>	87%	73%	76%	80%	74%	61%	66%	<b>74%</b>
<b>20:80</b>	70%	52%	66%	70%	60%	57%	60%	<b>62%</b>
<b>Rata-Rata</b>	<b>92%</b>	<b>74%</b>	<b>84%</b>	<b>87%</b>	<b>83%</b>	<b>71%</b>	<b>81%</b>	<b>82%</b>

Dari Tabel 4.7 dapat dilihat rata-rata tingkat pengenalan tertinggi untuk FNLVQ-MSA adalah pada konfigurasi pelatihan 60:40, sedangkan yang terendah adalah pada konfigurasi pelatihan 20:80. Selanjutnya Tabel 4.7 dapat direpresentasikan dalam Gambar 4.5.



**Gambar 4.5 Tingkat Pengenalan FNLVQ-MSA untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pengujian**

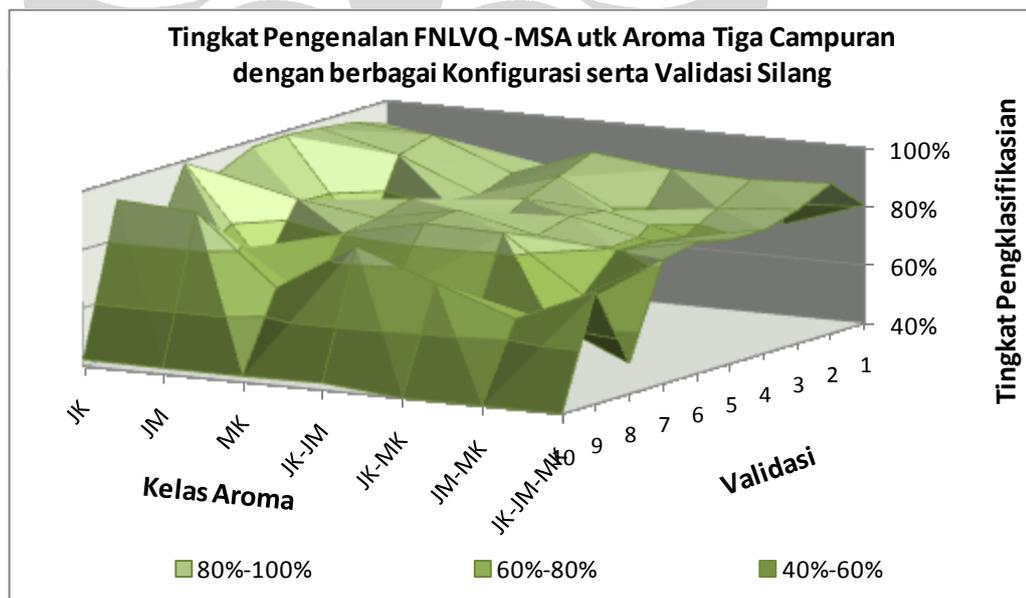
Gambar 4.5 memperlihatkan tingkat pengenalan FNLVQ-MSA terhadap data aroma tiga campuran, disini dapat dilihat jika semakin sedikit persentase data pelatihan yang dipergunakan maka tingkat pengenalan secara rata-rata akan menurun, juga dapat dilihat bahwa aroma campuran JMAlk dan JMAIk + MKAlk kurang bisa dikenali oleh FNLVQ. Kelebihan metode FNLVQ-MSA adalah bisa memberhentikan pelatihan ditengah jalan bilamana jaringan FNLVQ sudah mencapai *threshold* yang ditentukan, namun metode ini belum berhasil meningkatkan kinerja FNLVQ secara signifikan. Hal ini diketahui dengan tingkat pengenalan FNLVQ-MSA masih belum mampu mengenali dengan baik data aroma tiga campuran khususnya pada konfigurasi pelatihan 20:80.

Untuk FNLVQ-MSA hasil dari validasi silang menunjukkan hasil yang serupa dengan FNLVQ yaitu peningkatan tingkat pengenalan pada sub-bagian ke-2 dengan tingkat pengenalan sebesar 92% dan penurunan pada sub-bagian ke-10 dengan tingkat pengenalan sebesar 40%.

**Tabel 4.8 Tingkat Pengenalan FNLVQ-MSA untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan serta Validasi Silang**

Aroma	CV	1	2	3	4	5	6	7	8	9	10	Avg
	JK		92%	96%	97%	99%	98%	92%	99%	72%	94%	
JM		74%	95%	91%	65%	61%	86%	67%	76%	92%	42%	75%
MK		84%	84%	80%	81%	86%	82%	81%	61%	69%	42%	75%
JK-JM		87%	94%	89%	85%	77%	88%	87%	66%	86%	42%	80%
JK-MK		83%	91%	76%	83%	85%	82%	86%	54%	76%	38%	75%
JM-MK		71%	90%	84%	77%	79%	79%	67%	62%	66%	38%	72%
JK-JM-MK		81%	91%	86%	82%	82%	85%	83%	50%	78%	36%	75%
Rata-Rata		82%	92%	86%	82%	81%	85%	81%	63%	80%	40%	77%

Lebih lanjut Tabel 4.8 dapat direpresentasikan dalam Gambar 4.6, dalam gambar ini dapat dilihat untuk sub-bagian ke-10 tingkat pengenalan FNLVQ-MSA untuk semua aroma hasilnya menurun, hal ini diakibatkan oleh tidak baiknya vektor perwakilan awal untuk sub-bagian ini.



**Gambar 4.6 Tingkat Pengenalan FNLVQ-MSA untuk Aroma Tiga Campuran dengan berbagai Konfigurasi serta Validasi Silang**

Untuk mengatasi rendahnya tingkat pengenalan FNLVQ pada sub-bagian ke-10 ini maka dikembangkan metode FNLVQ-PSO serta *Swarm-FNLVQ* yang akan dijelaskan pada bab selanjutnya. Tetapi sebelumnya teori dasar PSO akan dijelaskan pada sub bab berikut ini.

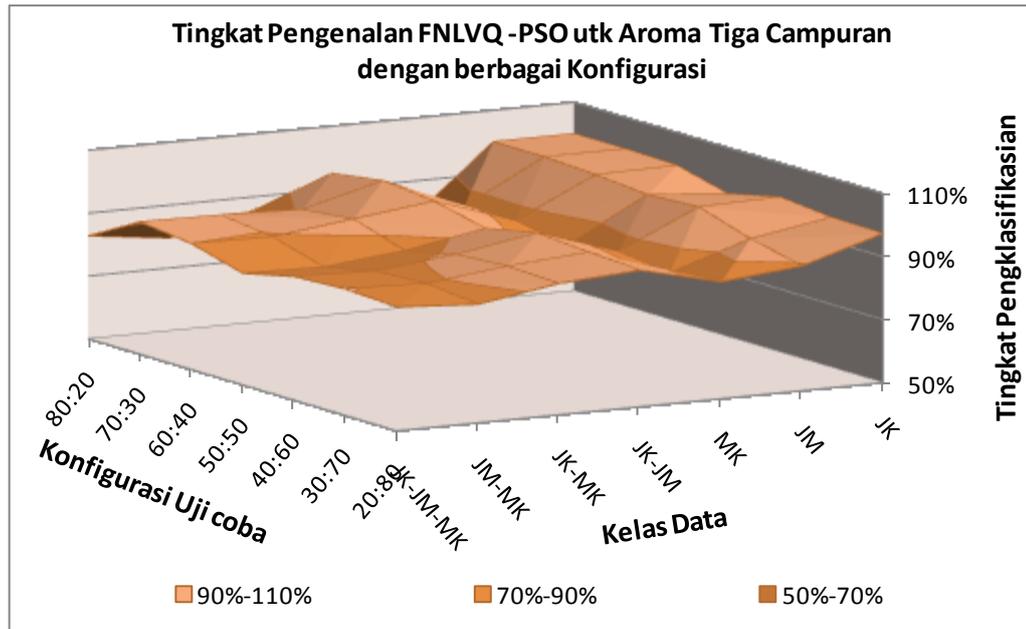
### 4.3 Hasil Pengenalan Aroma Menggunakan FNLVQ – PSO

Dengan menggunakan metode FNLVQ-PSO maka proses pembaharuan lapisan cluster dilakukan per-*neuron*/vektor perwakilan, berbeda dengan metode FNLVQ konvensional yang memperbaharui vektor perwakilan per-cluster, dengan demikian pendekatan metode ini berusaha mengoptimalkan tiap elemen. Berikut adalah hasil pengenalan aroma menggunakan metode FNLVQ-PSO:

**Tabel 4.9 Tingkat Pengenalan FNLVQ-PSO untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan**

Aroma Konfig.	Aroma							Rata-Rata
	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK	
<b>80:20</b>	100%	100%	78%	95%	85%	80%	83%	<b>89%</b>
<b>70:30</b>	100%	100%	87%	97%	91%	92%	92%	<b>94%</b>
<b>60:40</b>	100%	99%	87%	96%	93%	90%	90%	<b>93%</b>
<b>50:50</b>	95%	98%	84%	92%	88%	85%	85%	<b>90%</b>
<b>40:60</b>	99%	99%	87%	95%	91%	88%	89%	<b>93%</b>
<b>30:70</b>	98%	93%	85%	94%	92%	89%	90%	<b>92%</b>
<b>20:80</b>	97%	90%	87%	93%	92%	88%	89%	<b>91%</b>
<b>Rata-Rata</b>	<b>98%</b>	<b>97%</b>	<b>85%</b>	<b>95%</b>	<b>90%</b>	<b>87%</b>	<b>88%</b>	<b>92%</b>

Dari Tabel 4.7 dapat dilihat rata-rata tingkat pengenalan tertinggi untuk FNLVQ-PSO adalah pada konfigurasi pelatihan 70:30, sedangkan yang terendah adalah pada konfigurasi pelatihan 20:80, dari tabel ini dapat disimpulkan bahwa FNLVQ-PSO masih belum mampu mengenali dengan baik aroma MKAlk serta JMAlk + MKAlk, namun aroma lain pengenalannya sudah mencapai 90% lebih, dan kinerja FNLVQ-PSO secara rata-rata konsisten pada berbagai konfigurasi pelatihan. Selanjutnya Tabel 4.7 dapat direpresentasikan dengan Gambar 4.7.



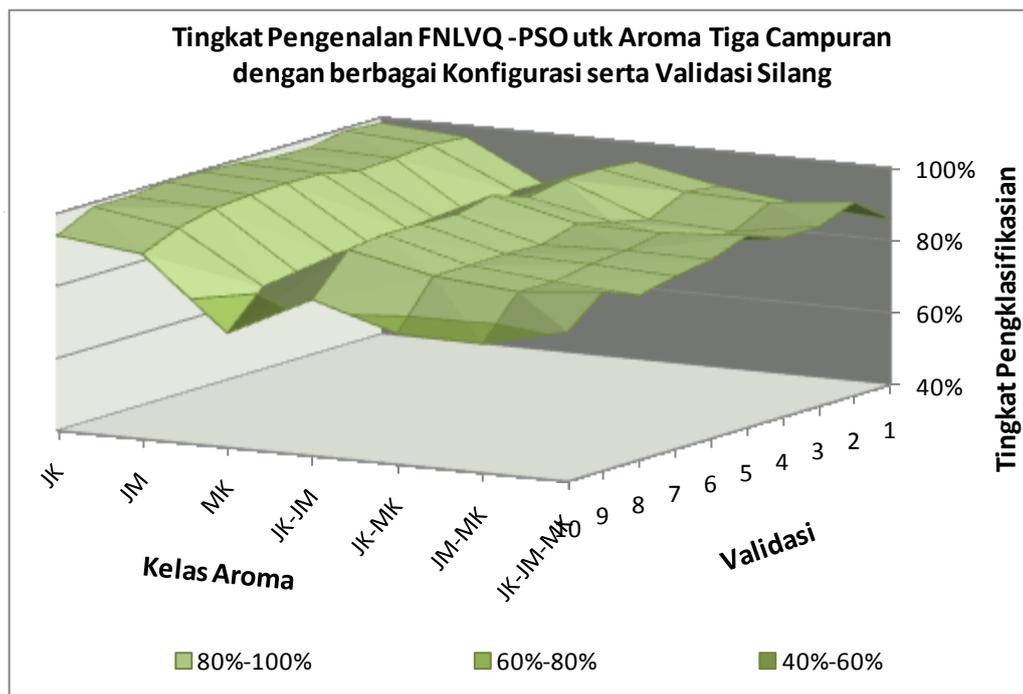
**Gambar 4.7** Tingkat Pengenalan FNLVQ-PSO untuk Aroma Tiga Campuran dengan berbagai Konfigurasi

Pada Gambar 4.7 dapat dilihat bahwa tingkat pengenalan FNLVQ-PSO secara rata-rata tidak turun terlalu jauh, terutama bila dibandingkan dengan FNLVQ maupun FNLVQ-MSA untuk konfigurasi pelatihan 20:80. Hal ini menunjukkan bahwa metode FNLVQ-PSO berhasil meningkatkan kinerja FNLVQ bilamana menggunakan jumlah data latih sedikit.

**Tabel 4.10** Tingkat Pengenalan FNLVQ-PSO untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan serta Validasi Silang

Aroma	CV	1	2	3	4	5	6	7	8	9	10	Avg
	JK	98%	99%	98%	98%	99%	100%	100%	99%	99%	94%	
JM	97%	98%	97%	96%	99%	99%	99%	98%	95%	91%	<b>97%</b>	
MK	85%	87%	82%	84%	86%	84%	84%	82%	81%	72%	<b>83%</b>	
JK-JM	95%	96%	96%	94%	97%	95%	95%	95%	94%	83%	<b>94%</b>	
JK-MK	90%	92%	87%	89%	92%	89%	89%	89%	89%	76%	<b>88%</b>	
JM-MK	87%	91%	86%	86%	91%	90%	89%	87%	87%	76%	<b>87%</b>	
JK-JM-MK	86%	94%	90%	90%	93%	89%	88%	85%	89%	81%	<b>88%</b>	

Selanjutnya untuk tingkat pengenalan FNLVQ-PSO dengan validasi silang, yang diperlihatkan pada Tabel 4.10, menunjukkan tingkat pengenalan rata-rata untuk aroma tiga campuran sudah diatas 80% . Lebih lanjut Tabel 4.10 dapat direpresentasikan dengan Gambar 4.8.



**Gambar 4.8 Tingkat Pengenalan FNLVQ-PSO untuk Aroma Tiga Campuran dengan berbagai Konfigurasi serta Validasi Silang**

Pada Gambar 4.8 dapat dilihat tingkat pengenalan FNLVQ-PSO yang telah di validasi silang untuk sub-bagian data ke-10 tidak turun terlalu jauh. Hal ini menunjukkan bahwa metode FNLVQ-PSO telah mengurangi ketergantungan inialisasi awal FNLVQ, dengan mencari vektor perwakilan yang optimal dari kumpulan vektor perwakilan awal yang dibentuk secara acak.

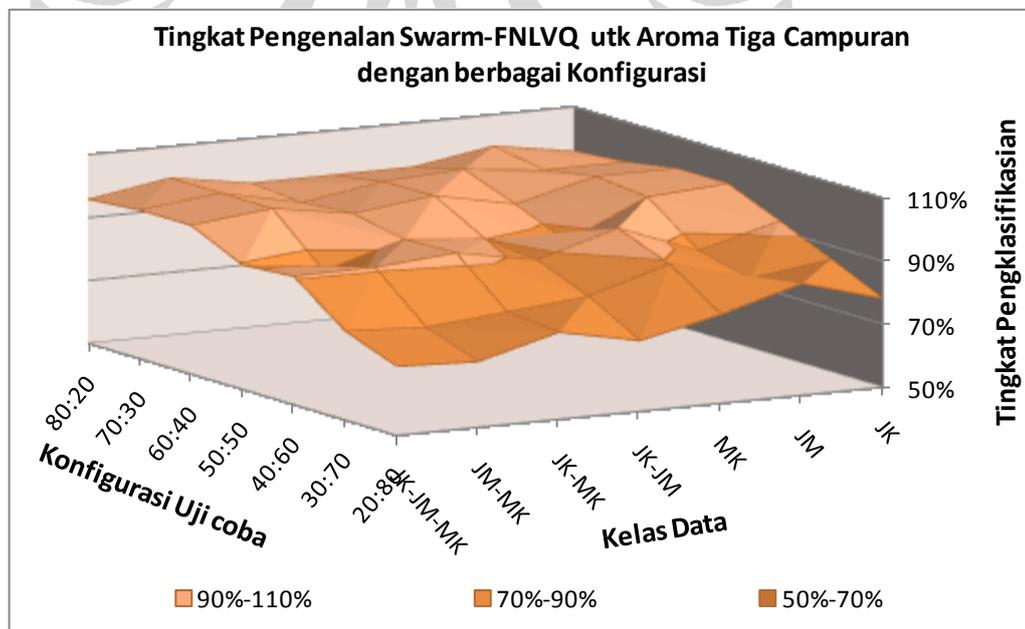
#### 4.4 Hasil Pengenalan Aroma Menggunakan Swarm-FNLVQ

Dengan menggunakan metode *Swarm-FNLVQ*, tingkat pengenalan FNLVQ dapat dimaksimalkan, hal ini karena metode ini mengambil tingkat pengenalan yang tertinggi dari tingkat pengenalan FNLVQ dengan inialisasi acak. Penentuan vektor perwakilan awal untuk FNLVQ diperoleh dengan mencari nilai *fitness* terbesar pada partikel. Dengan mendapatkan vektor perwakilan yang memiliki nilai *fitness* tinggi, maka tingkat pengenalan yang diperoleh pada FNLVQ dapat dioptimalkan.

**Tabel 4.11 Tingkat Pengenalan Swarm-FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan**

Aroma Konfig.	Aroma							Rata-Rata
	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK	
<b>80:20</b>	96%	100%	96%	96%	96%	100%	96%	<b>97%</b>
<b>70:30</b>	97%	97%	100%	99%	94%	100%	97%	<b>98%</b>
<b>60:40</b>	100%	100%	94%	99%	96%	100%	97%	<b>98%</b>
<b>50:50</b>	100%	98%	87%	90%	93%	85%	89%	<b>92%</b>
<b>40:60</b>	93%	86%	96%	88%	93%	91%	91%	<b>91%</b>
<b>30:70</b>	86%	82%	89%	82%	79%	77%	78%	<b>82%</b>
<b>20:80</b>	78%	85%	78%	72%	78%	71%	72%	<b>76%</b>
<b>Rata-Rata</b>	<b>93%</b>	<b>93%</b>	<b>91%</b>	<b>89%</b>	<b>90%</b>	<b>89%</b>	<b>89%</b>	<b>91%</b>

Tingkat pengenalan *Swarm-FNLVQ* mampu memberikan tingkat pengenalan diatas 90% untuk aroma tiga campuran dengan konfigurasi pelatihan 80:20, 70:30 dan 60:40. Namun untuk konfigurasi pelatihan 50:50, 40:60, 30:70 dan 20:80 tingkat pengenalan cenderung menurun. Tabel 4.11 lebih lanjut dapat direpresentasikan oleh Gambar 4.9. Dapat dilihat pada Gambar 4.9 tingkat pengenalan FNLVQ masih bagus untuk konfigurasi pelatihan data latih diatas 50% namun cenderung menurun untuk data latih 50% kebawah.



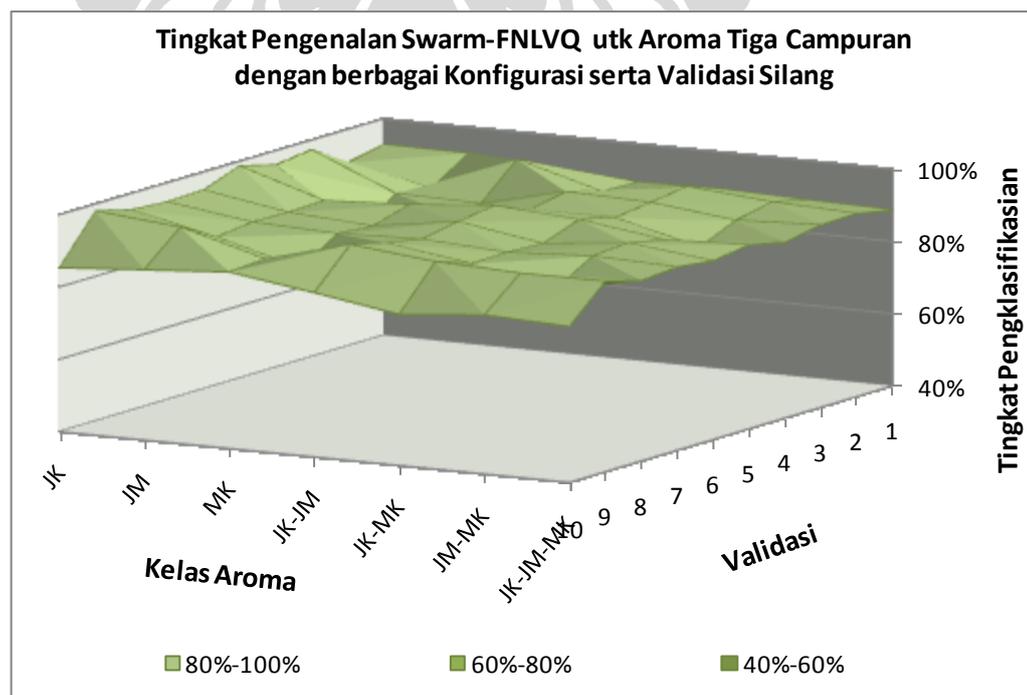
**Gambar 4.9 Tingkat Pengenalan Swarm-FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi**

Tingkat pengenalan FNLVQ untuk data aroma tiga campuran dengan menggunakan validasi silang menunjukkan tingkat pengenalan yang relatif stabil, hal ini terlihat pada Tabel 4.12, dimana tingkat pengenalan seluruh sub-bagian diatas 80%. Lebih lanjut Tabel 4.12 dapat direpresentasikan dengan Gambar 4.10.

**Tabel 4.12 Tingkat Pengenalan Swarm-FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi Pelatihan serta Validasi Silang**

Aroma	CV	1	2	3	4	5	6	7	8	9	10	Avg
	JK	93%	91%	97%	96%	99%	95%	95%	95%	98%	85%	
JM	93%	86%	87%	80%	91%	91%	91%	90%	96%	87%	<b>89%</b>	
MK	91%	96%	87%	88%	93%	90%	91%	84%	89%	89%	<b>90%</b>	
JK-JM	89%	90%	93%	91%	95%	93%	91%	93%	96%	85%	<b>92%</b>	
JK-MK	90%	93%	92%	91%	93%	89%	90%	90%	93%	82%	<b>90%</b>	
JM-MK	89%	92%	89%	85%	90%	92%	91%	85%	92%	84%	<b>89%</b>	
JK-JM-MK	89%	91%	90%	89%	91%	89%	91%	90%	93%	83%	<b>89%</b>	

Pada Gambar 4.10, dapat dilihat diagram tingkat pengenalan *Swarm-FNLVQ* yang relatif landai. Hal ini menandakan metode *Swarm-FNLVQ* telah mampu mempertahankan tingkat pengenalan walaupun dengan data latih yang berbeda-beda. Hal ini dikarenakan metode ini mampu mendeteksi keberadaan vektor perwakilan yang terbaik, dan mempergunakannya untuk dilatihkan dengan FNLVQ, sehingga tingkat pengenalan FNLVQ kian baik.



**Gambar 4.10 Tingkat Pengenalan Swarm-FNLVQ untuk Aroma Tiga Campuran dengan berbagai Konfigurasi serta Validasi Silang**

#### 4.5 Perbandingan Tingkat Pengenalan Aroma tanpa Validasi Silang

Sub-bab ini akan melakukan perbandingan metode FNLVQ, FNLVQ-MSA, FNLVQ-PSO dan *Swarm-FNLVQ* dengan menggunakan data aroma tiga campuran, selanjutnya pada percobaan ini menggunakan parameter sebagai berikut:

<b>FNLVQ</b>	Alpha = 0.1 Epo = 100
<b>FNLVQ-MSA</b>	Alpha = 0.1 Epo = 100 Threshold = 0.9
<b>FNLVQ-PSO</b>	Particle Size = 10 Chi = 0.33 Clocal = 0.1 Cglobal = 0.1 Epo = 100
<b>Swarm-FNLVQ</b>	Alpha = 0.1 Epo = 100 Particle Size = 16

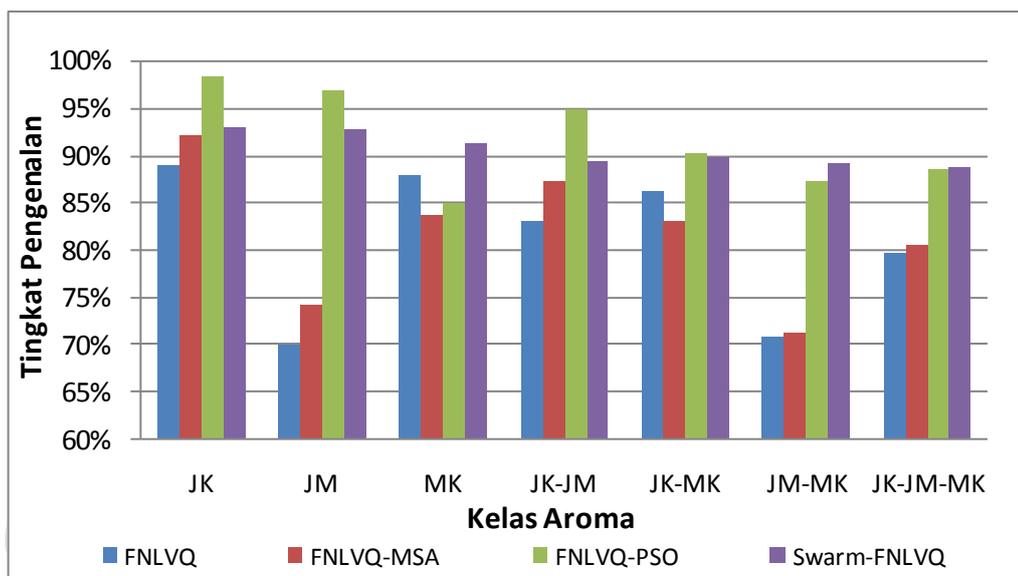
**Gambar 4.11 Parameter yang digunakan dalam Percobaan**

Berikut adalah tingkat pengenalan empat metode FNLVQ tersebut :

**Tabel 4.13 Perbandingan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran per-Metode Pengklasifikasian**

	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK	Rata-Rata
<b>FNLVQ</b>	89%	70%	88%	83%	86%	71%	80%	<b>81%</b>
<b>FNLVQ-MSA</b>	92%	74%	84%	87%	83%	71%	81%	<b>82%</b>
<b>FNLVQ-PSO</b>	98%	97%	85%	95%	90%	87%	88%	<b>92%</b>
<b>Swarm-FNLVQ</b>	93%	93%	91%	89%	90%	89%	89%	<b>91%</b>
<b>Rata-Rata</b>	<b>93%</b>	<b>83%</b>	<b>87%</b>	<b>89%</b>	<b>87%</b>	<b>80%</b>	<b>84%</b>	

Pada Tabel 4.13 dapat dilihat tingkat pengenalan FNLVQ-PSO secara rata-rata mencapai nilai tertinggi yaitu 92%, sedangkan dibawahnya Swarm-FNLVQ sebesar 91%, disusul FNLVQ-MSA sebesar 82% dan FNLVQ konvensional sebesar 81%. Sedangkan kategori aroma yang kurang bisa dikenali adalah JMAIk + MKAlk. Agar lebih mudah dimengerti, Tabel 4.13 dapat direpresentasikan menjadi



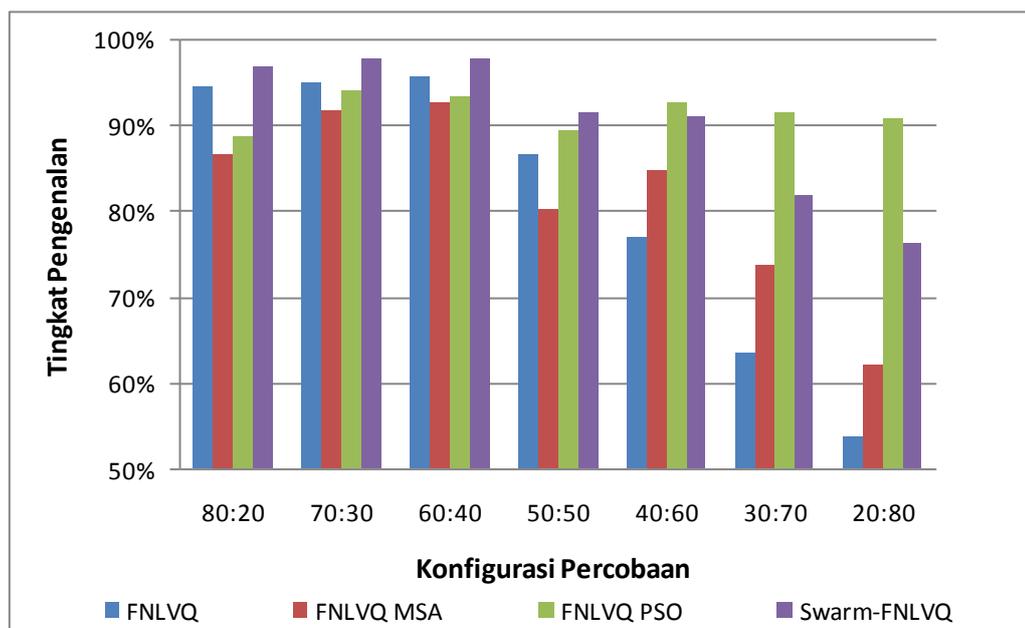
**Gambar 4.12 Perbandingan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran per-Metode Pengklasifikasian**

Pada Gambar 4.12 dapat dilihat bahwa tingkat pengenalan FNLVQ-PSO secara rata-rata lebih baik daripada FNLVQ maupun FNLVQ-MSA. Demikian juga tingkat pengenalan *Swarm-FNLVQ* secara keseluruhan lebih baik daripada FNLVQ dan FNLVQ-MSA.

**Tabel 4.14 Perbandingan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran per-Konfigurasi Pelatihan**

	FNLVQ	FNLVQ-MSA	FNLVQ-PSO	Swarm-FNLVQ
<b>80:20</b>	95%	87%	89%	97%
<b>70:30</b>	95%	92%	94%	98%
<b>60:40</b>	96%	93%	93%	98%
<b>50:50</b>	87%	80%	90%	92%
<b>40:60</b>	77%	85%	93%	91%
<b>30:70</b>	64%	74%	92%	82%
<b>20:80</b>	54%	62%	91%	76%
<b>Rata-Rata</b>	<b>81%</b>	<b>82%</b>	<b>92%</b>	<b>91%</b>

Pada Tabel 4.14 dapat dilihat bahwa tingkat pengenalan FNLVQ-PSO untuk aroma tiga campuran dengan validasi silang lebih konsisten dari sisi konfigurasi pelatihan daripada FNLVQ dan FNLVQ-MSA. Sedangkan *Swarm-FNLVQ*, meskipun tingkat pengenalannya hampir setinggi FNLVQ-PSO namun pada konfigurasi pelatihan 20:80 turun menjadi 76%. Selanjutnya Tabel 4.14 dapat direpresentasikan menjadi Gambar 4.13.



**Gambar 4.13 Perbedaan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran per-Konfigurasi Pelatihan**

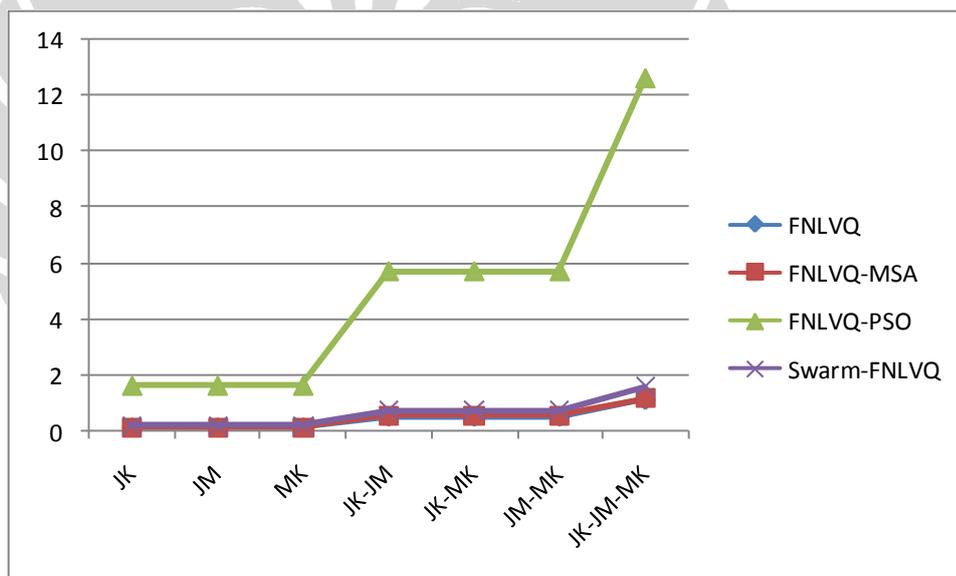
Pada Gambar 4.13 dapat dilihat bahwa tingkat pengenalan *Swarm-FNLVQ* lebih superior daripada FNLVQ dan FNLVQ-PSO namun ia cenderung menurun tingkat pengenalannya pada konfigurasi pelatihan 20:80, sedangkan FNLVQ-PSO meskipun tingkat pengenalan diawal konfigurasi pelatihan kalah dibandingkan FNLVQ dan *Swarm-FNLVQ*, namun pada segmen konfigurasi pelatihan akhir metode ini lebih baik.

Penelitian ini selain membandingkan tingkat pengenalan data aroma juga dibandingkan waktu pelatihan tiap metode pengklasifikasian. Pada Tabel 4.13 dapat dilihat perbandingan tingkat kecepatan antara tiap metode tersebut.

**Tabel 4.15 Perbandingan Tingkat Kecepatan FNLVQ, FNLVQ-MSA, FNLVQ-PSO dan Swarm-FNLVQ untuk Epoch 100 (dalam detik)**

	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK
<b>FNLVQ</b>	0.141	0.141	0.141	0.531	0.531	0.531	1.141
<b>FNLVQ-MSA</b>	0.141	0.141	0.141	0.547	0.547	0.547	1.172
<b>FNLVQ-PSO</b>	1.64	1.64	1.64	5.734	5.734	5.734	12.625
<b>Swarm-FNLVQ</b>	0.203	0.203	0.203	0.734	0.734	0.734	1.578
<b>Rata-rata</b>	<b>0.5313</b>	<b>0.5313</b>	<b>0.531</b>	<b>1.8865</b>	<b>1.8865</b>	<b>1.8865</b>	<b>4.129</b>

Pada tabel ini diketahui waktu pelatihan tercepat adalah ditempati oleh FNLVQ, kedua oleh FNLVQ-MSA dan ketiga oleh *Swarm-FNLVQ*, sedangkan urutan waktu pelatihan terlama ditempati oleh FNLVQ-PSO. Hal ini dikarenakan metode FNLVQ-PSO, pada tahapan pelatihannya turut menghitung kumpulan lapisan cluster sehingga proses pelatihan menjadi lambat. Untuk lebih jelasnya Tabel 4.13 dapat direpresentasikan dengan Gambar 4.14.



**Gambar 4.14 Perbandingan Kecepatan FNLVQ, FNLVQ-MSA, FNLVQ-PSO dan Swarm-FNLVQ pada Pelatihan dengan epoch = 100 (dalam detik)**

Pada Gambar 4.14 dapat dilihat bahwa kecepatan pelatihan FNLVQ-PSO terpaut sangat jauh daripada ketiga metode lainnya. Metode *Swarm-FNLVQ* terpaut sedikit dari metode FNLVQ, hal ini dikarenakan proses pencarian vektor perwakilan awal dilakukan pada saat epoch pertama saja, selanjutnya pelatihannya dilakukan seperti FNLVQ konvensional.

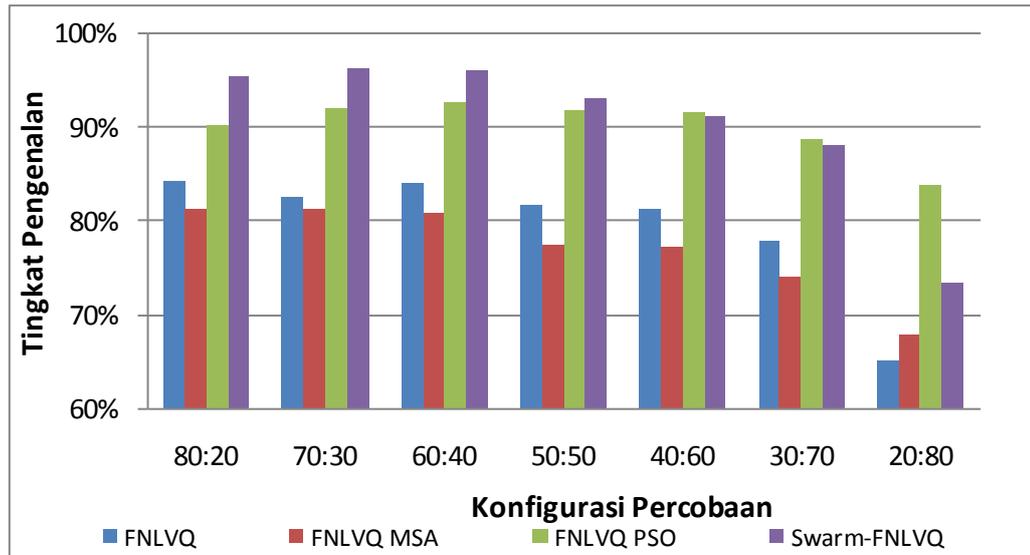
#### 4.6 Perbandingan Tingkat Pengenalan Aroma dengan Validasi Silang

Untuk tingkat pengenalan dengan validasi silang, tingkat pengenalan aroma tiga-campuran dengan menggunakan algoritma FNLVQ-PSO maupun *Swarm*-FNLVQ rata-rata masih diatas tingkat pengenalan FNLVQ dan FNLVQ-MSA bila dilihat dari sisi rata-rata konfigurasi pelatihan. Pada Tabel 4.16 ditampilkan tingkat pengenalan untuk aroma tiga-campuran dengan validasi silang.

**Tabel 4.16 Perbandingan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran dengan Validasi Silang per-Konfigurasi Pelatihan**

	FNLVQ	FNLVQ MSA	FNLVQ PSO	Swarm- FNLVQ
<b>80:20</b>	84%	81%	90%	96%
<b>70:30</b>	83%	81%	92%	96%
<b>60:40</b>	84%	81%	93%	96%
<b>50:50</b>	82%	77%	92%	93%
<b>40:60</b>	81%	77%	92%	91%
<b>30:70</b>	78%	74%	89%	88%
<b>20:80</b>	65%	68%	84%	73%
<b>Rata-Rata</b>	<b>80%</b>	<b>77%</b>	<b>90%</b>	<b>91%</b>

Dapat dilihat pada Tabel 4.16 semua tingkat pengenalan pada metode yang diujicobakan cenderung menurun, namun metode FNLVQ-PSO cenderung lebih konsisten. Dari tabel ini dapat dilihat pula tingkat pengenalan *Swarm*-FNLVQ pada konfigurasi pelatihan 80:20, 70:30, 60:40 dan 50:50 lebih baik daripada metode yang lain, setelah itu metode FNLVQ-PSO yang lebih baik. Selanjutnya Tabel 4.16 dapat direpresentasikan lebih lanjut dengan Gambar 4.15.



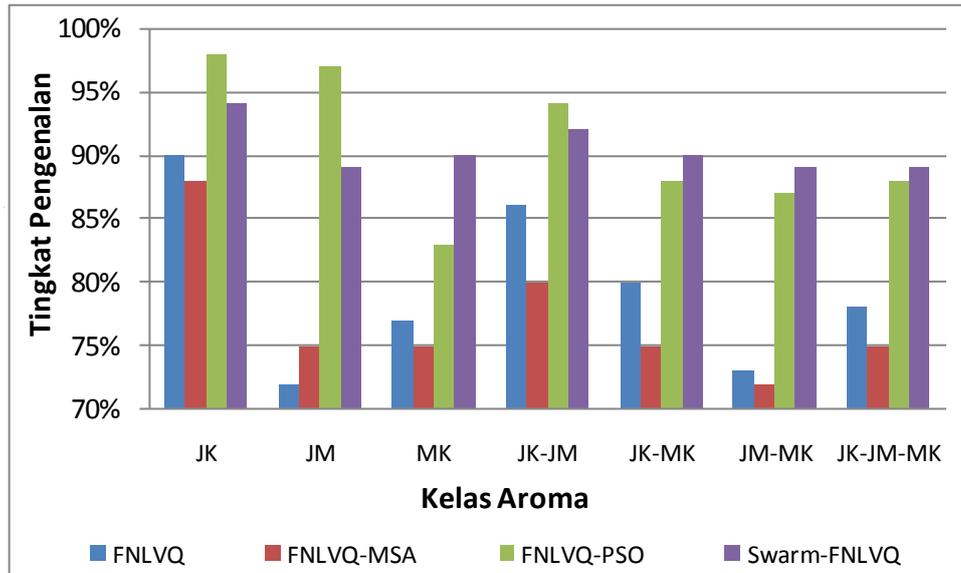
**Gambar 4.15** Perbandingan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran dengan Validasi Silang per-Konfigurasi Pelatihan

Pada Gambar 4.15 dapat dilihat untuk tingkat pengenalan yang telah divalidasi silang, kedua metode FNLVQ-PSO dan Swarm-FNLVQ lebih superior dibandingkan dengan FNLVQ dan FNLVQ-MSA bila dilihat dari sisi konfigurasi pelatihan.

**Tabel 4.17** Perbandingan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran dengan Validasi Silang per-Metode Pengklasifikasian

	JK	JM	MK	JK-JM	JK-MK	JM-MK	JK-JM-MK	Rata-Rata
<b>FNLVQ</b>	90%	72%	77%	86%	80%	73%	78%	<b>79%</b>
<b>FNLVQ-MSA</b>	88%	75%	75%	80%	75%	72%	75%	<b>77%</b>
<b>FNLVQ-PSO</b>	98%	97%	83%	94%	88%	87%	88%	<b>91%</b>
<b>Swarm-FNLVQ</b>	94%	89%	90%	92%	90%	89%	89%	<b>90%</b>
<b>Rata-Rata</b>	<b>93%</b>	<b>83%</b>	<b>81%</b>	<b>88%</b>	<b>83%</b>	<b>80%</b>	<b>83%</b>	

Dapat dilihat pada Tabel 4.17 tingkat pengenalan rata-rata aroma tiga campuran dengan validasi silang dimenangi oleh FNLVQ-PSO, namun tingkat pengenalan jenis aroma MKAlk masih rendah yaitu sekitar 83%. Jika dibandingkan dengan tingkat pengenalan *Swarm-FNLVQ* maka tingkat pengenalan terendahnya adalah 89%. Hal ini mengindikasikan meskipun tingkat pengenalan *Swarm-FNLVQ* lebih rendah namun ia lebih stabil tingkat pengenalannya. Selanjutnya Tabel 4.17 dapat direpresentasikan dengan Gambar 4.16.



**Gambar 4.16 Perbedaan Tingkat Pengenalan Metode yang diujicobakan untuk Aroma Tiga Campuran dengan Validasi Silang per-Metode Pengklasifikasian**

Pada Gambar 4.16 dapat dilihat tingkat pengenalan yang telah divalidasi silang baik FNLVQ-PSO maupun *Swarm-FNLVQ* lebih superior daripada FNLVQ dan FNLVQ-MSA. Dan pada akhirnya dapat disimpulkan bahwa dua metode yang telah dikembangkan yaitu FNLVQ-PSO dan *Swarm-FNLVQ* lebih superior bila dibandingkan dengan dua metode sebelumnya jika digunakan validasi silang.