

## **LEARNING VECTOR QUANTIZATION UNTUK PENDETEKSIAN ANAK BERBAKAT (*GIFTED CHILD*) PADA MASA PERKEMBANGAN**

Nuri Insania Andyani<sup>1</sup>, Nelly Indriani Widiastuti<sup>2</sup>

<sup>1,2</sup>Teknik Informatika – Universitas Komputer Indonesia  
Jl. Dipatiukur 112-114 Bandung  
E-mail :insania.nuri9@gmail.com<sup>1</sup>, indiwidi@gmail.com<sup>2</sup>

### **ABSTRAK**

Anak berbakat (*gifted child*) adalah anak yang memiliki potensi kecerdasan, kreatifitas dan tanggung jawab terhadap tugas diatas anak seusianya. Setiap anak berbakat memiliki gejala-gejala berdasarkan perilaku dan sosial emosi. Gejala antara satu tipe anak berbakat beririsan dengan tipe anak berbakat lainnya. Hal ini dapat menyebabkan proses deteksi menjadi sulit. Berdasarkan hal tersebut maka penelitian ini menggunakan sebuah metode yaitu metode *learning vector quantization*.

Data masukan yang digunakan adalah data gejala-gejala berdasarkan perilaku serta sosial emosi anak yang kemudian disusun menjadi 75 variabel masukan. Dari gejala-gejala tersebut maka hasil yang diperoleh yaitu tipe anak berbakat yang terdeteksi, persentasi keakuratan terhadap kelas asli dengan kelas hasil deteksi serta optimasi nilai parameter yang bervariasi dan waktu pemrosesan. Adapun tipe anak berbakat yang terdeteksi yaitu tipe *the successfuls, the challenging, the underground, the dropouts, the double labeled* dan *the autonomous learner*.

Berdasarkan hasil proses pengujian yang menggunakan *cross validation* dan *confusion matrix* dengan 5 *fold* dari jumlah data sebanyak 50 data, maka performansi metode *learning vector quantization* untuk pendeteksian anak berbakat pada masa perkembangan termasuk performansi yang baik dengan nilai parameter yang optimal pada maksimal epoch = 100, *learning rate* = 0,02 dan *error minimum* = 0,0001 dan waktu pemrosesan selama 15 detik memperoleh persentasi keakuratan yang mencapai 100%.

**Kata Kunci:** Analisis Performansi, Jaringan Syaraf Tiruan, *Learning Vector Quantization*, Anak Berbakat

### **1. PENDAHULUAN**

Jaringan Syaraf Tiruan (*Artificial Neural Network*) merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. Istilah buatan ini diimplementasikan

dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran [1]. Metode *learning vector quantization* merupakan salah satu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Metode ini hanya menggunakan alur maju saja dalam perhitungannya. Jika dua vektor mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor masukan tersebut ke dalam kelas yang sama [1]. Metode *learning vector quantization* digunakan untuk mengelompokkan pola – pola ke dalam kelas – kelas pola. Metode *learning vector quantization* banyak digunakan untuk menyelesaikan suatu masalah yang berkaitan dengan identifikasi penyakit, identifikasi suatu pola atau pengenalan suatu pola.

Berdasarkan salah satu hasil studi literatur penggunaan metode *learning vector quantization* yang pernah dilakukan sebelumnya yaitu pengenalan suatu penyakit yang memiliki gejala beririsan antara jenis yang satu dengan jenis yang lainnya. Hal ini telah dibuktikan pada jurnal “Pengenalan Jenis Penyakit THT Menggunakan Jaringan *Learning Vector Quantization*” [2]. Pada penelitian tersebut, metode *learning vector quantization* digunakan untuk mendiagnosa jenis penyakit THT pada bagian hidung berdasarkan gejala penyakit yang diolah menjadi 13 buah variabel masukan. Target keluaran berupa tiga jenis penyakit yang teridentifikasi oleh sistem, jenis penyakit yang dapat diidentifikasi oleh sistem yaitu penyakit *Rinitis Kronis, Epitaksis* atau *Sinusitis*. Jumlah data yang digunakan untuk proses pembelajaran sebanyak 64 data. Dari penelitian ini, didapatkan kesimpulan bahwa metode *learning vector quantization* dapat digunakan untuk mendiagnosis penyakit THT dengan keakuratan 94% yang diperoleh dari pelatihan 60 data dan pengujian 29 data, *learning rate* 0,00001, target *error minimum* 0,1 serta epoch tercapai pada nilai 88 [2].

Pada masalah anak berbakat (*gifted child*), setiap anak memiliki gejala - gejala berdasarkan perilaku dan sosial emosinya. Gejala – gejala yang terjadi pada anak berbakat(*gifted child*) dapat berjumlah 10 sampai 17 gejala untuk setiap tipenya. Gejala antara satu tipe anak berbakat (*gifted child*) beririsan dengan tipe anak berbakat (*gifted child*) yang lainnya. Hal ini dapat menyebabkan proses

identifikasi menjadi sulit karena ada beberapa gejala yang mirip antara satu tipe dengan tipe yang lainnya. Tipe anak berbakat (*gifted child*) dikelompokkan menjadi enam tipe yaitu tipe *the successfuls, the challenging, the underground, the dropouts, the double labeled* atau *the autonomous learner*.

Dengan gejala - gejala yang beririsan seperti itu, digunakan metode *learning vector quantization* untuk mengidentifikasi anak berbakat (*gifted child*). Metode *learning vector quantization* ini memiliki kemampuan untuk belajar mengelompokkan pola secara tipikal yaitu mengelompokkan pola - pola ke dalam kelas - kelas pola. Metode ini juga dapat meringkas keseluruhan data pelatihan yang besar menjadi vektor kode berukuran lebih kecil untuk proses pengelompokkannya serta dapat menghasilkan nilai error yang lebih kecil.

Berdasarkan masalah yang telah dipaparkan, maka pada penelitian ini dilakukan analisis performansi untuk mengetahui akurasi serta optimasi dari metode jaringan syaraf tiruan *learning vector quantization* untuk mendeteksi anak berbakat (*gifted child*) pada masa perkembangan.

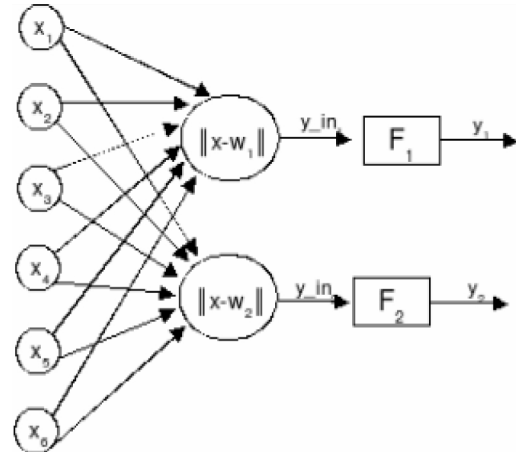
## 2. ISI PENELITIAN

### 2.1 Learning Vector Quantization

*Learning Vector Quantization* (LVQ) dalam penelitian ini menggunakan gejala-gejala *gifted child* yang dikumpulkan berdasarkan hasil wawancara dengan 50 orang tua *gifted child*. Gejala-gejala tersebut dinyatakan 0 jika tidak ditemukan gejala dan 1 jika ditemukan gejala yang dialami oleh anak yang diamati. berdasarkan hal tersebut maka data masukan untuk LVQ berjumlah 75 sesuai dengan seluruh gejala pada setiap kelas tipe *gifted child*.

*Learning vector quantization* merupakan suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengelompokkan vektor - vektor masukan. Jika dua vektor masukan mendekati sama, maka lapisan kompetitif akan meletakkan kedua vektor masukan tersebut ke dalam kelas yang sama. Untuk setiap unit keluaran akan mewakili sebuah kelas [1].

Pada gambar 1. adalah arsitektur jaringan *learning vector quantization* yang digunakan pada penelitian ini. Input layer pada arsitektur LVQ ini berjumlah 75 sesuai dengan jumlah gejala pada seluruh kelas. Setiap input layer akan dihitung jaraknya dengan dua buah bobot.



**Gambar 1.** Arsitektur Jaringan *Learning Vector Quantization* [1]

Keterangan :

- $X_1-X_{75}$  : Data masukan
- $W$  : Vektor Bobot
- $W_1$  : Vektor bobot yang menghubungkan setiap neuron pada lapisan masukan ke neuron pertama pada lapisan keluaran
- $W_2$  : Vektor bobot yang menghubungkan setiap neuron pada lapisan masukan ke neuron kedua pada lapisan keluaran
- $X-W$  : Perhitungan yang bertindak sebagai badan sel
- $F$  : Fungsi Aktivasi
- $Y$  : Data keluaran

Berikut ini algoritma pembelajaran metode *learning vector quantization*[1]:

1. Tetapkan nilai - nilai :
  - a. Bobot awal ( $W_{ij}$ );  $i$  = bobot ke- $i$ ;  $j$  = variabel masukan bobot ke- $j$ ;
  - b. Maksimum iterasi (epoh) : MaxEpoh;
  - c. Parameter *learning rate*( $\alpha$ );
  - d. *Error* minimum yang diharapkan (Eps).
2. Masukan nilai - nilai :
  - a. Masukan :  $x(m,n)$ ;  $m$  = menunjukkan data ke- $m$ ;  $n$  = menunjukkan variabel masukan ke- $n$ ;
  - b. Target :  $T(1,n)$ ;
3. Tetapkan kondisi awal :
  - a. epoh=0;
  - b. err =1.
4. Kerjakan jika : (epoh < MaxEpoh) atau ( $\alpha$  > eps)
  - a. epoh = epoh+1;
  - b. Kerjakan untuk  $i=1$  sampai  $n$ 
    - i. Tentukan  $J$  hingga  $\|X_i - W_j\|$  minimum (sebut sebagai  $C_j$ ); .....(2.1)
    - ii. Perbaiki  $W_j$  dengan ketentuan :
      - o Jika  $T = C_j$  maka :  
 $W_j(\text{baru})=W_j(\text{lama})+\alpha[X_i - W_j(\text{lama})]$ ;.....(2.2)
      - o Jika  $T \neq C_j$  maka :

$$W_j(\text{baru}) = W_j(\text{lama}) - \alpha [X_i - W_j(\text{lama})]; \dots \dots \dots (2.3)$$

iii. Kurangi nilai  $(\alpha) = \alpha - \alpha * 0,1; \dots \dots \dots (2.4)$

Keterangan rumus :

- T : Target;
- J : Jumlah selisih antara data ke-m dan bobot;
- C : Kelas selisih bobot terkecil;
- W : Bobot;
- $\alpha$  : Rasio Pembelajaran;
- x : Data.

**2.2 Anak Berbakat (Gifted Child)**

Anak berbakat (*gifted child*) menurut Joseph Renzulli adalah anak yang memiliki potensi kecerdasan atau intelegensi, kreatifitas dan tanggung jawab terhadap tugas (*task commitment*) diatas anak – anak seusianya. Joseph Renzulli (1986) yang mengemukakan sebuah konsep yang dikenal dengan istilah “*Three-Ring Conception*” menyatakan bahwa perilaku berbakat mencerminkan satu interaksi di antara tiga kelompok dasar sifat manusia [3]:

1. Kemampuan di atas rata-rata
2. Kreativitas
3. Tanggung jawab atau pengikatan diri terhadap tugas.

Tipe anak berbakat (*gifted child*) yang sampai saat ini masih sering digunakan adalah tipe anak berbakat (*gifted child*) menurut Betts & Neihart. Tipe anak berbakat (*gifted child*) dibagi menjadi enam tipe yaitu [4]:

1. *The Successfuls*
2. *The Challenging*
3. *The Underground*
4. *The Dropouts*
5. *The Double Labeled*
6. *The Autonomous Learner*

**2.3 Cross Validation**

*Cross validation* adalah salah satu cara untuk menemukan parameter terbaik dari satu model dengan cara menguji besarnya *error* pada tes. Dalam *cross validation*, data dibagi ke dalam *k* sampel dengan ukuran yang sama. Dalam penelitian ini digunakan *k-1* sampel untuk pembelajaran dan 1 sampel sisanya untuk pengujian. Ini sering disebut dengan validasi *k-fold* [5].

**2.4 Analisis**

**2.4.1 Analisis Data Masukan**

Data yang digunakan dalam proses pembelajaran dan pengujian untuk mendeteksi anak berbakat (*gifted child*) adalah data gejala – gejala dari enam tipe anak berbakat (*gifted child*). Seluruh gejala dari ke enam tipe tersebut berjumlah 75. Berdasarkan

gejala - gejala tersebut disusun menjadi variabel X1 - X75 yang akan menjadi data masukan pada simulasi. Untuk nilai dari seluruh variabel dijelaskan pada tabel 1 berikut ini.

**Tabel 1.** Variabel Input Gejala-Gejala Anak Berbakat (*Gifted Child*)

Gejala
Berprestasi(X1)
Baik dalam akademis(X2)
Mencari pengakuan orang(X3)
Bukan pengambil resiko(X4)
Bisa menerima dan menyesuaikan(X5)
Tidak mandiri(X6)
Menentang Guru(X7)
Suka Menanyakan Aturan(X8)
Jujur dan terus terang(X9)
Moodnya tidak menentu(X10)
Melakukan pekerjaan yang tidak tetap(X11)
Kreatif(X12)
Tetap pada keyakinannya(X13)
Bersaing(X14)
Tidak menunjukkan prestasi yang baik(X15)
Sering terlibat konflik(X16)
Menyembunyikan kemampuan(X17)
Menjauh dari orang-orang yang mendukungnya(X18)
Menolak tantangan (X19)
Ingin memiliki hubungan sosial(X20)
Berganti - ganti teman(X21)
Jarang hadir di kelas(X22)
Tidak bisa menyelesaikan tugas(X23)
Terbujuk dengan ketertarikan luar(X24)
Suka bermimpi di kelas(X25)
Mengasingkan diri(X26)
Mengkritik diri sendiri dan orang lain(X27)
Pengganggu(X28)
Berada di tengah atau di bawah anak – anak yang lain(X29)
Tidak berprestasi(X30)
Tulisan tangannya jelek(X31)
Memiliki kemampuan bersosialisasi yang baik(X32)
Membuat tujuannya sendiri(X33)
Bekerja tanpa meminta ijin(X34)
Mengikuti area yang sangat disukainya(X35)
Pengambil resiko(X36)
Efektif dalam bekerja(X37)
Mandiri(X38)
Mempunyai jiwa kepemimpinan yang besar(X39)
Cepat bosan(X40)
Konsep diri yang positif(X41)
Cemas(X42)
Suka merasa bersalah(X43)
Motivasi ekstrinsik(X44)
Bertanggung jawab kepada orang lain(X45)
Mengurangi perasaan sendiri dan membenarkan emosi mereka(X46)
Mempunyai sifat mencela diri sendiri(X47)
Frustrasi(X48)
Kesadaran harga diri yang rendah(X49)
Tidak sabaran(X50)
Membela diri sendiri(X51)
Sangat sensitif(X52)

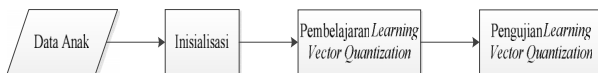
Gejala
Tidak yakin perannya dalam bergaul(X53)
Tertekan(X54)
Bingung(X55)
Merasa tidak aman(X56)
Dendam(X57)
Marah(X58)
Suka marah tiba-tiba(X59)
Konsep diri yang negatif(X60)
Tidak berdaya(X61)
Tidak menyadari kemampuannya(X62)
Merasa tidak dihargai(X63)
Merasa tidak dibantu(X64)
Merasa ditinggalkan(X65)
Percaya diri(X66)
Menerima diri sendiri(X67)
Antusias(X68)
Diterima oleh orang lain(X69)
Menerima orang lain(X70)
Keingintahuan dan belajar yang tinggi(X71)
Didukung dengan positif(X72)
Bisa menerima kegagalan(X73)
Kekuatan pribadi(X74)
Motivasi yang instrinsik(X75)

Data masukan yang harus dipilih seluruhnya yaitu sebanyak 75 buah gejala. Sedangkan minimum gejala yang terdeteksi oleh simulasi yaitu sebanyak 20 buah gejala.

Gejala – gejala untuk tipe *The Successful* mulai dari gejala X1-X6 dan X40-X47, tipe *The Challenging* mulai dari gejala X7-X16,X40 dan X49-53, tipe *The Underground* mulai dari gejala X17-X21, X42,X43,X46 dan X53-X56, tipe *The Dropouts* mulai dari gejala X11,X12, X22-X30,X48,X51,X54 dan X57-X60, tipe *The Double Labeled* mulai dari gejala X11,X23,X28-X31,X48,X49,X58,X61-X65, tipe *The Autonomous Learner* mulai dari gejala X12,X13,X32-X39,X41 dan X66-X75.

**2.2.2 Analisis Metode**

Pada tahap ini metode LVQ disimulasikan untuk mendapatkan gambaran yang jelas tentang proses yang terjadi. Adapun tahapan umum dalam metode LVQ dapat dilihat pada gambar 2 berikut:



**Gambar 2.** Block Diagram Learning Vector Quantization

**1. Inisialisasi**

Inisialisasi merupakan tahapan untuk menentukan nilai bobot awal, maksimum epoch atau maksimum iterasi(MaxEpoch), *learning rate*( $\alpha$ ), *error* minimum( $\epsilon$ ), jumlah data pembelajaran, jumlah kelas dan nilai awal epoch yang dibutuhkan

untuk melakukan proses pembelajaran *learning vector quantization*.

**2. Pembelajaran Learning Vector Quantization**

Tahapan pembelajaran ini akan menghasilkan nilai bobot akhir atau nilai bobot baru serta nilai *learning rate* ( $\alpha$ ) yang baru. Gambar 3. adalah *Flowchart* pada proses pembelajaran *learning vector quantization*.

Adapun penjelasan langkah–langkah tahapan pembelajaran *learning vector quantization* yaitu sebagai berikut :

1. Menentukan bobot awal, nilai MaxEpoch, *learning rate*( $\alpha$ ) dan *error* minimum;
2. Masukan Inputan dan Target;
3. Tentukan kondisi awal;
4. Cek kondisi epoch dan *learning rate*( $\alpha$ )

Jika epoch < MaxEpoch, maka lakukan perhitungan jarak setiap data masukan(x) terhadap setiap bobot( $w_i$ ), mencari jarak minimum, perhitungan perubahan nilai bobot dan perhitungan perubahan nilai *learning rate* ( $\alpha$ ). Jika kondisi epoch < MaxEpoch tidak terpenuhi, maka cek kondisi  $\alpha > \epsilon$ . Jika kondisi  $\alpha > \epsilon$  tidak terpenuhi, maka perulangan berhenti.

5. Menghitung jarak setiap data masukan(x) terhadap setiap data bobot

Rumus yang digunakan untuk menghitung jarak setiap data masukan(x) terhadap setiap bobot( $w_i$ ) menggunakan rumus 2.1 yaitu :

$$\text{Jarak} = \left\| x_{ij} - w_{ij} \right\|$$

contoh :

Jarak terhadap :

a. Bobot ke-1 ( $w_1$ )

$$\text{Jarak} = \left\| x_{1j} - w_{1j} \right\|$$

$$\text{Jarak} = \sqrt{12} = 3,464$$

b. Bobot ke-2 ( $w_2$ )

$$\text{Jarak} = \left\| x_{1j} - w_{2j} \right\|$$

$$\text{Jarak} = \sqrt{27} = 5,196$$

6. Menentukan posisi nilai jarak terkecil( $C_j$ )

Posisi nilai jarak terkecil didapat dari hasil perhitungan jarak setiap data masukan(x) terhadap setiap bobot( $w_i$ ) yang paling terkecil. Setelah itu, bandingkan posisi nilai jarak terkecil tersebut( $C_j$ ) dengan target kelas (T) yang ada pada tabel.

7. Cek nilai target (T) dengan hasil posisi nilai jarak terkecil ( $C_j$ )

Pengecekan ini dilakukan untuk melakukan perubahan bobot( $w_j$ ) terhadap bobot dengan jarak terkecil menggunakan rumus 2.2 dan rumus 2.3

Jika  $T = C_j$ , maka  $w_j(\text{baru}) = w_j(\text{lama}) + \alpha [x_i - w_j(\text{lama})]$

Jika  $T \neq C_j$  maka  $w_j(\text{baru}) = w_j(\text{lama}) - \alpha[x_i - w_j(\text{lama})]$

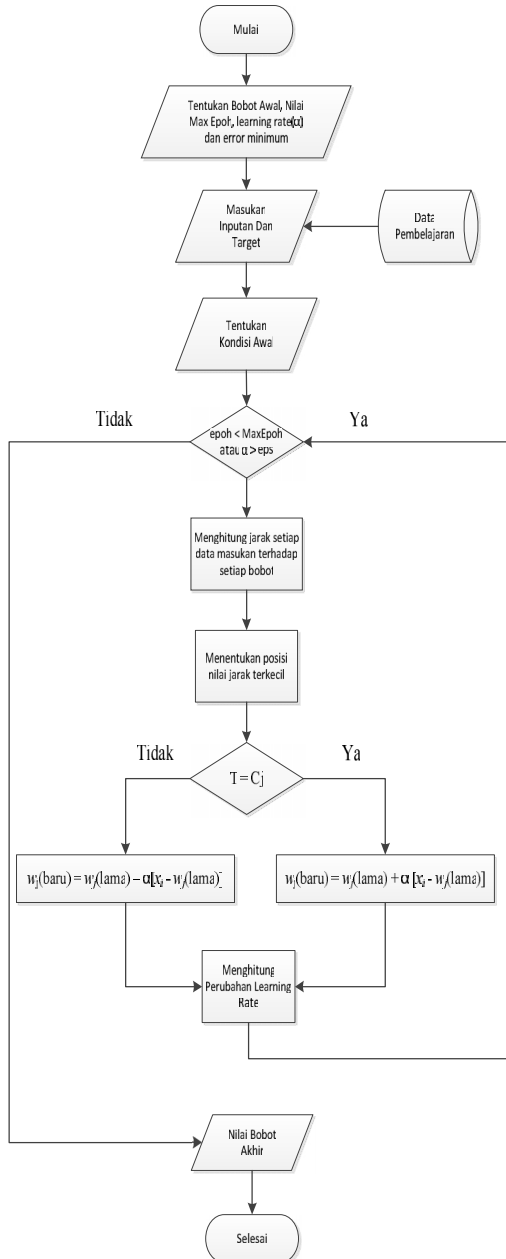
8. Menghitung perubahan *learning rate* ( $\alpha$ )

Proses ini dilakukan setelah perulangan selesai. Rumus yang digunakan untuk melakukan pengurangan *learning rate* ( $\alpha$ ) adalah rumus 2.4 sebagai berikut :

$$\alpha = \alpha - (\alpha * 0,1)$$

Contoh :

Setelah epoch ke-1, didapat nilai  $\alpha = 0,1 - (0,1 * 0,1) = 0,09$ .



Gambar 3. Flowchart Pembelajaran Learning Vector Quantization

3. Pengujian Learning Vector Quantization

Tahapan pengujian ini akan menghasilkan nilai kelas yang dikenali. Gambar 4. adalah Flowchart pada tahapan pengujian learning vector quantization.



Gambar 4. Flowchart Pengujian Learning Vector Quantization

Adapun penjelasan langkah – langkah tahapan pengujian learning vector quantization hampir sama dengan tahap pembelajaran, yaitu sebagai berikut :

1. Menghitung jarak data yang akan diuji terhadap setiap data bobot ( $w_j$ )

Data bobot yang digunakan di tahapan pengujian adalah nilai bobot akhir yang dihasilkan dari tahapan pembelajaran. Rumus yang digunakan untuk menghitung jarak data yang akan diuji terhadap setiap data bobot ( $w_j$ ) seperti pada rumus 2.1 sebagai berikut :

$$\text{Jarak} = \|x_{ij} - w_{ij}\|$$

contoh :

Jarak terhadap :

a. Bobot ke-1 ( $w_1$ )

$$\text{Jarak} = \|x_{1j} - w_{1j}\|$$

$$\text{Jarak} = \sqrt{12,298} = 3,506$$

b. Bobot ke – 2 ( $w_2$ )

$$\text{Jarak} = \|x_{1j} - w_{2j}\|$$

$$\text{Jarak} = \sqrt{26,5} = 5,147$$

2. Mencari nilai jarak terkecil

Dari hasil perhitungan data yang diuji terhadap setiap bobot( $w_j$ ), maka carilah nilai jarak terkecilnya.

3. Menentukan nilai kelas atau target keluaran

Nilai kelas atau target keluaran didapat dari posisi nilai jarak terkecil.

**2.5 Implementasi Dan Pengujian**

**2.5.1 Implementasi Sistem**

Pada penelitian ini metode LVQ diimplementasikan dalam sebuah simulator berbentuk prototype dengan menggunakan bahasa c# berbasis desktop dan sistem operasi window 7. Data latih dan data uji disimpan dalam database menggunakan MySQL sebanyak 3 tabel yaitu tabel *tbl\_anak*, *tbl\_gejala* dan *tipe\_gifted*.

**2.5.2 Pengujian**

Proses pengujian dilakukan dengan 2 tahap yaitu tahap pertama memasukkan nilai parameter maksimum epoch, *learning rate*, dan *error minimum* yang bervariasi dengan menggunakan kelompok data yang sama yaitu D1,D2,D3,D5 sebagai data pembelajaran dan D4 sebagai data pengujian. Setelah itu, dilakukan tahap kedua dengan cara merubah kombinasi *fold cross validation* dengan menggunakan masukan kombinasi parameter yang optimal.

Pengujian dilakukan dengan jumlah data keseluruhan sebanyak 50 data dan *k-fold cross validation* dengan nilai *k* = 5. Nilai 5 ini menunjukkan jumlah *fold* data. Masing – masing *fold* terdiri dari 10 data. Kombinasi *foldcross validation* yang digunakan dapat dilihat pada tabel 2. dibawah ini.

Pengujian	Data	Parameter
Pengujian 6	<i>fold cross validation2</i>	Max epoch = 500 Learning rate = 0,1 Error minimum = 0,00001
Pengujian 7	<i>fold cross validation2</i>	Max epoch = 1000 Learning rate = 0,02 Error minimum = 0,00001
Pengujian 8	<i>fold cross validation2</i>	Max epoch = 1000 Learning rate = 0,05 Error minimum = 0,00001
Pengujian 9	<i>fold cross validation2</i>	Max epoch = 1000 Learning rate = 0,09 Error minimum = 0,00001
Pengujian 10	<i>fold cross validation2</i>	Max epoch = 1000 Learning rate = 0,1 Error minimum = 0,00001
Pengujian 11	<i>fold cross validation1</i>	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001
Pengujian 12	<i>fold cross validation3</i>	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001
Pengujian 13	<i>fold cross validation4</i>	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001
Pengujian 14	<i>fold cross validation5</i>	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001

**Tabel 2.** Kombinasi *Fold Cross Validation*

Iterasi	Data Pembelajaran	Data Pengujian
1	D1,D2,D3,D4	D5
2	D1,D2,D3,D5	D4
3	D1,D2,D4,D5	D3
4	D1,D3,D4,D5	D2
5	D2,D3,D4,D5	D1

Rencana proses pengujian yang akan dilakukan dapat dilihat pada tabel 3. dibawah ini.

**Tabel 3.** Rencana Proses Pengujian

Pengujian	Data	Parameter
Pengujian 1	<i>fold cross validation2</i>	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001
Pengujian 2	<i>fold cross validation2</i>	Max epoch = 100 Learning rate = 0,05 Error minimum = 0,0001
Pengujian 3	<i>fold cross validation2</i>	Max epoch = 100 Learning rate = 0,1 Error minimum = 0,001
Pengujian 4	<i>fold cross validation2</i>	Max epoch = 500 Learning rate = 0,02 Error minimum = 0,001
Pengujian 5	<i>fold cross validation2</i>	Max epoch = 500 Learning rate = 0,09 Error minimum = 0,0001

Tabel 4 berikut ini adalah hasil pengujian yang dilakukan pada tabel 3.

**Tabel 4.** Hasil Pengujian

Pengujian	Data	Keterangan
Pengujian 1	<i>fold cross validation</i> 2	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001 Akurasi = 80% Waktu = 16 detik
Pengujian 2	<i>fold cross validation</i> 2	Max epoch = 100 Learning rate = 0,05 Error minimum = 0,0001 Akurasi = 70% Waktu = 15 detik
Pengujian 3	<i>fold cross validation</i> 2	Max epoch = 100 Learning rate = 0,1 Error minimum = 0,001 Akurasi = 60% Waktu = 15 detik
Pengujian 4	<i>fold cross validation</i> 2	Max epoch = 500 Learning rate = 0,02 Error minimum = 0,001 Akurasi = 80% Waktu = 1 menit 26 detik
Pengujian 5	<i>fold cross validation</i> 2	Max epoch = 500 Learning rate = 0,09 Error minimum = 0,0001 Akurasi = 50% Waktu = 1 menit 19 detik
Pengujian 6	<i>fold cross validation</i> 2	Max epoch = 500 Learning rate = 0,1 Error minimum = 0,00001 Akurasi = 60% Waktu = 1 menit 19 detik
Pengujian 7	<i>fold cross validation</i> 2	Max epoch = 1000 Learning rate = 0,02 Error minimum = 0,00001 Akurasi = 80% Waktu = 2 menit 37 detik
Pengujian 8	<i>fold cross validation</i> 2	Max epoch = 1000 Learning rate = 0,05 Error minimum = 0,00001 Akurasi = 70% Waktu = 2 menit 37 detik
Pengujian 9	<i>fold cross validation</i> 2	Max epoch = 1000 Learning rate = 0,09 Error minimum = 0,00001 Akurasi = 60% Waktu = 2 menit 38 detik
Pengujian 10	<i>fold cross validation</i> 2	Max epoch = 1000 Learning rate = 0,1 Error minimum = 0,00001 Akurasi = 60% Waktu = 2 menit 39 detik
Pengujian 11	<i>fold cross validation</i> 1	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001 Akurasi = 100% Waktu = 15 detik
Pengujian 12	<i>fold cross validation</i> 3	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001 Akurasi = 100% Waktu = 15 detik

Pengujian 13	<i>fold cross validation</i> 4	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001 Akurasi = 80% Waktu = 16 detik
Pengujian 14	<i>fold cross validation</i> 5	Max epoch = 100 Learning rate = 0,02 Error minimum = 0,0001 Akurasi = 100% Waktu = 16 detik

### 3. PENUTUP

#### 3.1 Kesimpulan

Berdasarkan hasil penelitian ini dan tujuan dari penelitian ini, maka diperoleh kesimpulan bahwa performansi metode *learning vector quantization* untuk pendeteksian anak berbakat (*gifted child*) pada masa perkembangan termasuk performansi yang cukup baik dengan persentasi tingkat akurasinya mencapai 50% sampai 100% dengan nilai parameter yang optimal yang berada pada maksimal epoch = 100, learning rate = 0,02 dan error minimum = 0,0001 dan waktu lamanya proses selama 15 detik.

#### 3.2 Saran

Berdasarkan hasil dari keseluruhan pengujian, maka dapat diberikan saran yaitu perlu dilakukan lebih banyak lagi pembelajaran dengan nilai parameter yang bervariasi agar hasil dari pengujian dapat mencapai tingkat akurasi yang lebih baik lagi serta penambahan dataset yang akan digunakan

### DAFTAR PUSTAKA

- [1] S. Kusumadewi, Artificial Intelligence (Teknik dan Aplikasinya), Yogyakarta: Graha Ilmu, 2003.
- [2] E. I. Sela and S. Hartati, "Pengenalan Jenis Penyakit THT Menggunakan Jaringan Learning Vector Quantization".
- [3] J. M. V. Tiel, Anaku Terlambat Bicara, Anak Berbakat Dengan Disinkronitas Perkembangan : Memahami dan Mengasuhnya, Membedakannya dengan Autisme, ADHD, dan Permasalahan Gangguan Belajar, Jakarta: Prenada Media Group, 2008.
- [4] J. M. V. Tiel, Pendidikan Anaku Terlambat Bicara, Prenada Media Group: Jakarta, 2011.
- [5] B. Santoso, Data Mining : Teknik Pemanfaatan Data Untuk Keperluan Bisnis, Yogyakarta: Graha Ilmu, 2007.

