

# Sistem Pengenalan Bahasa Isyarat Indonesia Berbasis Sensor dengan ANMBP(Adaptive Neighborhood based Modified BackPropagation)

Endang Supriyati<sup>1,2)</sup> Handayani Tjandrasa<sup>1)</sup>

1) Jur. Teknik Informatika , Institut Teknologi Sepuluh November Surabaya(ITS), Surabaya 60111

2)Jur. Teknik Informatika ,Universitas Muria Kudus, Gondangmanis Kudus 59352

email : esupriyati@gmail.com, handatj@its.ac.id

**Abstrak** – Makalah ini membahas pengembangan sistem bahasa isyarat Indonesia dengan menggunakan sarung tangan bersensor. Sensor yang digunakan berupa flex sensor untuk mengukur tekukan jari dan accelerometer untuk mengukur gerakan pada sumbu x,y,z. Sistem yang dikembangkan mempunyai dua tahap, tahap pertama ekstraksi ciri yang menggunakan histogram yang mengukur frekuensi gerakan gerak isyarat, tahap kedua adalah proses pengenalan gerak isyarat menggunakan ANMBP (*adaptive neighborhood based modified backpropagation*). Metode ANMBP menggabungkan *error linier* dan *non linier*, *adaptive learning rate* serta *neighborhood* pada *backpropagation*. Pengujian menggunakan 200 dataset. Dataset yang digunakan untuk learning dan testing sebanyak 10 kelas learning dan 10 kelas testing. Hasil pengujian menunjukkan akurasi pengukuran mencapai 97%.

**Kata Kunci:** *bahasa isyarat Indonesia, sensor, modified backpropagation.*

## 1. PENDAHULUAN

Bahasa isyarat Indonesia merupakan alat komunikasi yang utama bagi penyandang tuna rungu dan tuna wicara di Indonesia. Tetapi komunitas di luar para penyandang tuna rungu/wicara tidak memahami bahasa isyarat. Masalah komunikasi ini memberi efek negatif bagi kehidupan dan hubungan sosial penyandang tuna rungu/wicara. Penyandang tuna rungu/wicara menggunakan penerjemah ataupun tulisan untuk berkomunikasi dengan orang normal. Akan tetapi penerjemah sering mempunyai tarif yang mahal, demikian pula resiko ketergantungan dan kehilangan privasi.

Kemajuan di bidang *pattern recognition* memberi harapan untuk otomatisasi sistem penerjemah gerak isyarat, meskipun banyak kesulitan yang dihadapi sebelum sistem tersebut menjadi kenyataan. Penelitian di bidang bahasa isyarat dapat dikelompokkan ke dalam tiga bagian yaitu (i) berbasis visi komputer[1][2][3], (ii) berbasis sensor[4][5] dan (iii) gabungan dari keduanya (visi komputer dan sensor). Visi komputer untuk pengenalan gerak isyarat berbasis pada image processing dan teknik ekstraksi ciri untuk mengambil dan mengklasifikasikan isyarat tangan dan gerakan tubuh. Untuk isyarat yang berbasis sensor menggunakan sarung tangan dan sensor gerakan untuk mendeteksi isyarat tangan dan gerakannya..

Beberapa penelitian tentang pengenalan gerak

isyarat menggunakan metode Artificial Neural Network[2][5] , Hidden Markov Model[3][4], metode statistik dan sebagainya[6]. Pada penelitian ini proses pelatihan data menggunakan metode modified backpropagation yang menggunakan *adaptive learning rate*, *random neighborhood* pada *hidden layer* serta *adaptive weight* yang merupakan penjumlahan *error linier* dan *error non linier* [7]. Untuk bahasa isyarat Indonesia, telah dilakukan penelitian sebelumnya[17],tetapi menggunakan gerak isyarat kata statis serta sensor yang digunakan adalah sensor flek.

Pada penelitian ini dilakukan pengujian terhadap gerak isyarat kata dinamis, sensor yang digunakan adalah sensor flek dan accelerometer untuk mengukur gerakan pada sumbu x,y,z.

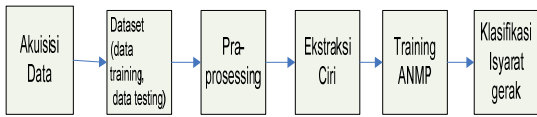
Struktur organisasi makalah ini bagian pertama adalah pendahuluan, bagian kedua tentang bahasa isyarat Indonesia, bagian ketiga sensor dan akuisisi data, bagian keempat adalah ekstraksi ciri, bagian kelima adalah ANN- ANMBP (*adaptive neighborhood based modified backpropagation*), bagian keenam adalah hasil dan pembahasan, bagian ketujuh kesimpulan

## 2. BAHASA ISYARAT INDONESIA

Bahasa isyarat isyarat Indonesia juga dikenal dengan istilah SIBI (Sistem Isyarat Bahasa Indonesia). Isyarat dalam SIBI secara garis besar dapat dibedakan menjadi dua, yaitu isyarat alfabet dan isyarat kata. Untuk isyarat alfabet, SIBI mengacu kepada ASL (*American Sign Language*), sedangkan untuk isyarat kata terdapat standar khusus kata bahasa Indonesia dan dibakukan dalam bentuk kamus sistem isyarat bahasa Indonesia. Isyarat alfabet biasanya digunakan terbatas, yaitu untuk mengeja nama atau kata yang belum tercantum dalam kosa kata kamus. Isyarat kata lebih banyak digunakan dalam prakteknya dan memiliki jumlah isyarat yang jauh lebih besar. Baik isyarat alfabet maupun isyarat kata memiliki komponen-komponen isyarat. Komponen isyarat yang utama adalah bentuk jari-jari tangan dan gerakan tangan. Pada sebagian besar isyarat kata, gerakan tangan lebih dominan dan bervariasi dibandingkan dengan bentuk jari-jari tangan. Pada penelitian Evita[17] yang hanya menggunakan sensor *flex*, akurasi pengenalannya turun drastis dari 83,18% untuk isyarat kata yang statis menjadi 49,58% untuk isyarat kata yang dinamis.

Dalam penelitian ini, pengenalan ditujukan untuk isyarat kata bahasa Indonesia, dengan penambahan jenis sensor lain. Dua informasi komponen utama isyarat kata diukur dengan penggunaan sensor *flex* dan sensor

*accelerometer* yang diintegrasikan dalam bentuk sarung tangan sebagai piranti akuisisi data.



Gambar 1: Diagram blok sistem pengenalan bahasa isyarat Indonesia menggunakan ANMBP

### 3. SENSOR DAN AKUISISI DATA

#### 3.1 Sensor

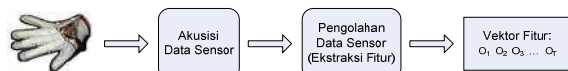
Sensor yang digunakan adalah sensor *flex* dan sensor *accelerometer*. Bentuk fisik sensor *flex* dan sensor *accelerometer* diperlihatkan pada gambar 2. Sensor *flex* adalah jenis sensor resistif yang berubah nilai resistansinya ketika ada lekukan. Pada kondisi *flat* (tidak tertekuk), sensor *flex* mempunyai nilai resistansi 10 KΩ. Nilai resistansinya akan semakin membesar pada saat terjadi tekukan sampai dengan 40 KΩ pada saat tekukan maksimal[8] Karena jari hanya mempunyai satu derajat kebebasan, maka untuk tiap-tiap jari cukup menggunakan satu sensor *flex*. Sedangkan sensor *accelerometer* yang digunakan adalah Hitachi H48C yang merupakan modul sensor buatan *Parallax* diintegrasikan dengan ADC (*Analog to Digital Converter*) dan regulator tegangan. *Accelerometer* dapat mendeteksi kemiringan (*tilt*) dan gerakan (percepatan) pada tiga sumbu x,y,z dengan jangkauan pengukuran maksimal ± 3g (1g = 9.81m/s<sup>2</sup>), non-linearity ±2% dan sensitivity 333 mV/g pada tegangan 3 Volt[9].



(a) Sensor flex (b) accelerometer  
Gambar 2. Sensor yang digunakan

#### 3.2 Akuisisi Data

Tahap ini ditujukan untuk mendapatkan data-data dari sensor untuk kemudian diolah menjadi vektor fitur (*feature vector*), sebagaimana diperlihatkan gambar 3.



Gambar 3. Blok diagram akuisisi data

Data-data yang diperoleh dari sensor adalah sebanyak 8 jenis data. Data-data tersebut sebagai berikut:

- derajat tekukan jari jempol
- derajat tekukan jari telunjuk
- derajat tekukan jari tengah
- derajat tekukan jari kelingking
- derajat tekukan jari manis
- gerakan tangan terhadap sumbu x,y,z

Untuk kelima data lekukan jari telah dinormalisasi

untuk menyamakan perbedaan nilai toleransi dari di antara sensor-sensor *flex* yang digunakan. Besaran lekukan jari dinormalisasi[18] pada range nilai 0 sampai 20 menggunakan persamaan 1 sebelum disimpan dalam *file* dataset.

$$X_{norm} = \frac{X_i - X_{min}}{X_{max} - X_{min}} * 20 \dots\dots\dots(1)$$

Sedangkan untuk besaran akselerasi masih berupa data *counting*, yang diambil 8-bit MSB (*Most Significant Bit*) dari 12 bit data *counter*-nya. Untuk mendapatkan nilai akselerasi yang sebenarnya diperlukan perhitungan khusus yang dijelaskan pada bagian ekstraksi ciri. Contoh data gerak isyarat kata ‘abang’ yang tersimpan pada dalam bentuk *file* dengan format teks (*ibi*) pada proses akuisisi data diperlihatkan pada tabel 1. Data yang tersimpan adalah data sekuensial, dengan panjang (baris) yang berbeda yang bergantung jenis isyarat kata dan kecepatan gerakan pada saat melakukan isyarat. Pada contoh, panjang data isyarat kata ‘abang’ adalah 13.

Tabel 1. Contoh data sekuensial isyarat kata ‘abang’

| 002_abang_001.ibi |     |     |     |     |     |     |     |     |     |  |  |  |
|-------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|--|--|--|
| 1                 | 006 | 013 | 013 | 013 | 014 | 136 | 128 | 126 | 127 |  |  |  |
| 2                 | 009 | 016 | 016 | 015 | 017 | 131 | 134 | 135 | 127 |  |  |  |
| 3                 | 009 | 015 | 015 | 015 | 017 | 136 | 133 | 157 | 127 |  |  |  |
| 4                 | 009 | 015 | 015 | 015 | 017 | 149 | 137 | 165 | 127 |  |  |  |
| 5                 | 008 | 015 | 015 | 015 | 017 | 145 | 134 | 154 | 127 |  |  |  |
| 6                 | 006 | 013 | 013 | 013 | 016 | 139 | 127 | 146 | 127 |  |  |  |
| 7                 | 003 | 002 | 004 | 004 | 004 | 140 | 120 | 152 | 127 |  |  |  |
| 8                 | 002 | 002 | 002 | 003 | 003 | 142 | 119 | 149 | 127 |  |  |  |
| 9                 | 002 | 002 | 002 | 003 | 003 | 140 | 117 | 148 | 127 |  |  |  |
| 10                | 002 | 002 | 002 | 003 | 002 | 136 | 118 | 142 | 127 |  |  |  |
| 11                | 002 | 002 | 002 | 003 | 003 | 141 | 128 | 140 | 127 |  |  |  |
| 12                | 002 | 002 | 002 | 003 | 003 | 142 | 144 | 144 | 127 |  |  |  |
| 13                | 002 | 002 | 002 | 003 | 003 | 139 | 162 | 141 | 127 |  |  |  |
| 14                |     |     |     |     |     |     |     |     |     |  |  |  |

Kolom [1...5] adalah data dari sensor *flex* untuk lekukan jari jempol sampai dengan jari kelingking, sedangkan [6...9] adalah data dari sensor *accelerometer* untuk sumbu X, Y, Z dan referensi. Baris [1..13] menunjukkan urutan data diambil/disimpan, yaitu dari awal gerakan isyarat satu kata sampai dengan selesai gerakan isyarat kata tersebut.

Gerak isyarat kata bahasa Indonesia yang dilakukan mengacu pada video gerak isyarat yang terdapat pada [15] yang merupakan visualisasi dari kamus sistem isyarat bahasa Indonesia.

### 4. EKSTRAKSI CIRI

Ekstraksi ciri dilakukan untuk mendapatkan besaran-besaran yang menunjukkan kekhususan dari data yang diolah. Ekstraksi ciri ini merupakan salah satu bagian terpenting dan berpengaruh terhadap akurasi hasil pengenalan. Untuk data testing dan data learning dilakukan ekstraksi ciri yang sama untuk mendapatkan vektor ciri. Vektor ciri terdiri dari nilai-nilai hasil olahan data *flex* dan data *accelerometer* yang membentuk baris angka (nilai).

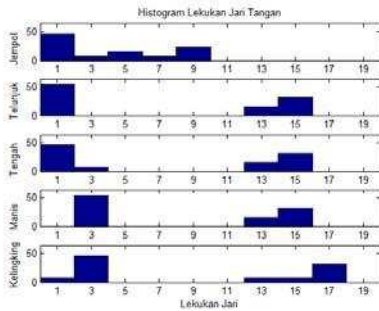
Untuk data sensor *flex* yang mewakili ciri bentuk tangan, dihitung berdasarkan frekuensi kemunculannya dengan metode histogram. Dinyatakan *d* adalah jumlah sub range, *h<sub>i</sub>* adalah kolom histogram 0 ≤ *i* ≤ *d* dan *n* adalah jumlah data.

Berikut adalah persamaan histogram yang digunakan :

$$h_i = \sum_{j=1}^n \eta(x_j) \cdot v_i \quad 0 \leq i \leq d \dots\dots\dots(2)$$

dimana

$$\eta(x_j) = \begin{cases} 1 & \text{Jika } x_j = i \\ 0 & \text{Jika } x_j \neq i \end{cases}$$



Gambar 4. Histogram Lekukan Jari Tangan

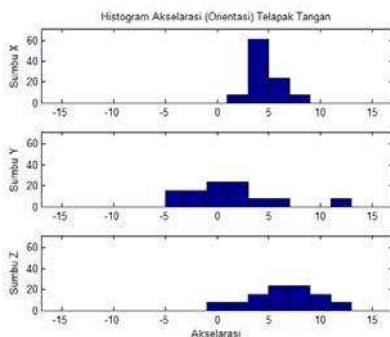
Sedangkan untuk data *accelormeter* dihitung nilai akselerasi *a* yang sebenarnya, digunakan rumus yang sebagaimana tercantum dalam datasheet [9], dimana

$$a = (c - ref) * 0.0022 * 16 \dots\dots\dots(3)$$

Nilai akselerasi *a* ini relatif kecil terhadap nilai normalisasi lekukan jari, sehingga dilakukan kuantisasi[19],[20] nilai akselerasi *a* untuk masing-masing sumbu X, Y dan Z berdasarkan tabel 2. ( $1g = 9.81m/s^2$ )

Tabel 2. Kuantisasi akselerasi

| Akselerasi (a) | Nilai Kuantisasi |
|----------------|------------------|
| $a > g$        | 16               |
| $g < a < 2g$   | 11 .... 15       |
| $0 < a < g$    | 1 ... 10         |
| $a = 0$        | 0                |
| $-g < a < 0$   | -1 ... -10       |
| $-2g < a < -g$ | -11 .... -15     |
| $a < -2g$      | -16              |



Gambar 5. Histogram Akselerasi Gerakan Tangan

**5. ANN - ANMBP**

Backpropagation merupakan algoritma pada jaringan saraf tiruan yang banyak digunakan untuk menyelesaikan

berbagai macam masalah, salah satunya adalah klasifikasi. Masalah yang terjadi pada backpropagation adalah konvergensi yang lama dan terjebak dalam lokal minimal. Banyak algoritma yang diusulkan untuk mengatasi masalah tersebut salah satunya metode ANMBP [7]. Metode ANMBP ini menggunakan algoritma modifikasi backpropagation [10], struktur *neighborhood* pada *hidden layer* [11] dan *adaptive learning rate* [12].

Pada algoritma ANMBP untuk mencari nilai *Net* digunakan :

$$u_j^s = \sum_{i=1}^n w_{ji}^s y_i^{s-1} \dots\dots\dots(4)$$

Dengan menggunakan fungsi aktivasi sigmoid biner, maka outputnya adalah :

$$f(u_j^s) = \frac{1}{(1 + e^{-u_j^s})} = y_j^s \dots\dots\dots(5)$$

Dengan *n* menunjukkan jumlah neuron dan  $w_{ij}^s$  adalah bobot dari neuron ke *i* dari layer (*s*-1) ke neuron ke *j* dari layer *s*.

Pada algoritma ANMBP digunakan  $E_p$  yang merupakan penjumlahan dari kuadrat error linier dan non linier dari output.

$$E_p = \sum_{j=1}^n \frac{1}{2} (e_{1j}^s)^2 + \sum_{j=1}^n \frac{1}{2} \lambda (e_{2j}^s)^2 \dots\dots\dots(6)$$

dimana  $\lambda$  adalah koefisien bobot , error nonlinier  $e_2$  dan error linier  $e_1$  diperoleh dari :

$$e_{1j}^s = d_j^s - y_j^s \dots\dots\dots(7)$$

$$e_{2j}^s = l d_j^s - u_j^s \dots\dots\dots(8)$$

$$l d_j^s = f^{-1}(d_j^s) \dots\dots\dots(9)$$

Dimana *d* adalah output yang diharapkan dan *y* adalah output yang dihasilkan. Sehingga perubahan bobot pada layer output adalah :

$$\Delta w_{ji}^s = -\mu \frac{\partial E}{\partial W_{ji}} \dots\dots\dots(10)$$

$$\begin{aligned} \Delta w_{ji}^s &= \mu e_{1j}^s \frac{\partial y_j^s}{\partial W_{ji}^s} + \mu \lambda e_{2j}^s \frac{\partial u_j^s}{\partial W_{ji}^s} \\ \Delta w_{ji}^s &= \mu e_{1j}^s \frac{\partial y_j^s}{\partial u_j^s} \frac{\partial u_j^s}{\partial W_{ji}^s} + \mu \lambda e_{2j}^s y_i^{s-1} \\ \Delta w_{ji}^s &= \mu e_{1j}^s f'(u_j^s) y_i^{s-1} + \mu \lambda e_{2j}^s y_i^{s-1} \end{aligned} \dots\dots\dots(11)$$

Error linier dan non linier pada hidden layer (*L*) adalah :

$$e_{1j}^L = \sum_{r=1}^{n_{L+1}} f'(u_r^{L+1}) e_{ir}^{L+1} w_{rj}^{L+1} \dots\dots\dots(12)$$

$$e_{2j}^L = f^1(u_j^L) \sum_{r=1}^{n_{L+1}} e_{2r}^{L+1} w_{rj}^{L+1} \dots\dots\dots(13)$$

Sehingga perubahan bobot pada hidden layer adalah

:

$$\Delta w_{ji}^L = \mu e_{1j}^L y_i^{L-1} f'(u_j^L) + \mu \lambda e_{2j}^L y_i^{L-1} \tag{14}$$

Parameter learning  $\mu$  dan  $\mu\lambda$  diganti dengan parameter adaptive[12] yaitu :

$$\eta^r = \frac{\mu \|e\|^2}{\|f' y\|^2 + \epsilon} \tag{15}$$

$$\mu' = \frac{\lambda \|e\|^2}{\|f' y\|^2 + \epsilon} \tag{16}$$

dimana  $y^r = a_j^r - y_j^r = e_{1j}^r$  (17)

dan  $f_j^r = \frac{\partial y_j^r}{\partial w_{jk}^r}$  (18)

Substitusikan (17) dan (18) ke dalam persamaan (15) dan (16) sehingga parameter adaptive learning menjadi

$$\eta^r = \frac{\mu \|e_{1j}^r\|^2}{\left\| \frac{\partial y_j^r}{\partial w_{jk}^r} e_{1j}^r \right\|^2 + \epsilon} \tag{19}$$

$$\mu' = \frac{\lambda \|e_{1j}^r\|^2}{\left\| \frac{\partial y_j^r}{\partial w_{jk}^r} e_{1j}^r \right\|^2 + \epsilon} \tag{20}$$

$$\eta^i = \frac{\mu \|e_{1j}\|^2}{\|f'(u_j) y_i^{s-1} e_{1j}\|^2 + \epsilon} \tag{21}$$

$$\mu' = \frac{\lambda \|e_{1j}\|^2}{\|f'(u_j) y_i^{s-1} e_{1j}\|^2 + \epsilon} \tag{22}$$

Dimana  $\mu$ ,  $\lambda$  adalah konstanta dengan nilai kecil positif dan  $\epsilon$  konstanta dengan nilai kecil positif untuk menjamin ketidakstabilan ketika error menuju 0. Pemilihan nilai  $\mu$ ,  $\lambda$  dan  $\epsilon$  dilakukan secara heuristik untuk mendapatkan hasil yang optimal.

Sehingga perubahan bobot pada layer output dan layer hidden menjadi :

$$\Delta w_{ji}^s = \eta^i e_{1j}^s f'(u_j^s) y_i^{s-1} + \mu' e_{2j}^s y_i^{s-1} \tag{23}$$

Dengan perubahan bobot tersebut, maka bobot baru dihitung berdasarkan persamaan:

$$w(t+1) = w(t) + \Delta w(t) \tag{24}$$

dimana  $t$  adalah iterasi.

Rata-rata kuadrat kesalahan (MSE) [21] dari jaringan dihitung dengan dari jaringan dihitung dengan rata-rata dari kuadrat error non-linier.

$$MSE = E((a_j^r - y_j^r)^2) \tag{25}$$

Pendekatan Neighborhood dalam ANMBP untuk meminimalkan waktu dan bobot selama pelatihan. Neighborhood yang berbeda dibentuk secara acak dalam hidden layer dengan jumlah neuron yang sama. Oleh karena itu menyebabkan jumlah operasi minimum dan memori yang dibutuhkan.

Pada metode artificial neural network-backpropagation, penentuan bobot awal biasanya ditentukan secara random. Untuk masalah learning bahasa isyarat indonesia, digunakan metode nguyen-widrow [13][14] untuk menentukan bobot awal.

Algoritma ANMBP untuk klasifikasi multikelas gerak isyarat Indonesia adalah sebagai berikut :

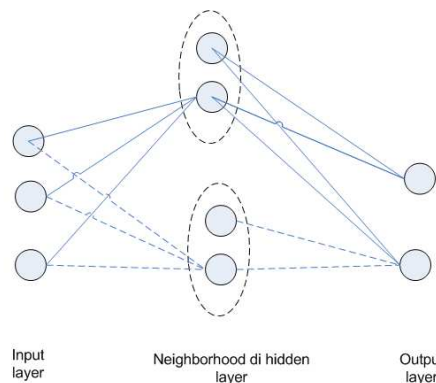
1. Inialisasi : tentukan struktur jaringan, bobot awal, definisikan neighborhood
2. Pilih neighborhood secara random
3. Tentukan pola learning

*Feedforward*

4. Untuk setiap unit/node di hidden layer, hitung :  
Net menggunakan rumus (4)  
Output menggunakan rumus (5)
5. Untuk unit/node di output layer, hitung :  
Net menggunakan rumus (4)  
Output menggunakan rumus (5)

*Backpropagation*

- Hitung linier error (7) dan error non linier(8)  
Ubah bobot dengan (21),(22),(23) untuk neighborhood yang terpilih.
6. Untuk node di hidden layer, hitung :  
Hitung linier error (12) dan error non linier(13)  
Ubah bobot dengan (21),(22),(23) untuk neighborhood yang terpilih.
7. Update bobot, update bobot pada node output layer dan node hidden layer menggunakan rumus (23) untuk neighborhood yang terpilih.
8. Ulangi langkah 3-7 untuk seluruh pola
9. Evaluasi error jaringan dengan bobot baru
10. Berhenti jika kondisi yang diinginkan tercapai, jika tidak ulangi langkah 2-10.



Gambar 6. Struktur jaringan dengan neighborhood

## 6. HASIL DAN PEMBAHASAN

Untuk melakukan pengujian, diambil data 10 kelas gerak isyarat kata yang masing-masing mempunyai 20 data sampel, sehingga terdapat total 200 dataset. Dari 20 data sampel, diambil 10 data sampel untuk masing-masing kelas sebagai data testing dan 10 sisanya sebagai

data *learning*. Kata yang diambil adalah kata yang bergerak, kata ini tidak hanya memerlukan gerakan jari tetapi juga membutuhkan gerak telapak tangan atau bagian tangan yang lain.

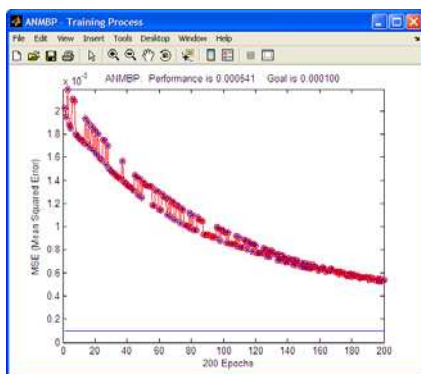
Perangkat yang digunakan untuk komputasi menggunakan *Netbook Acer Aspire 3810T* dengan spesifikasi *Intel Core2Solo 1,4GHz FSB 800MHz*, RAM 2GB DDR3 dan *Graphic Card Intel GMA 4500MHD*. Sedangkan *tool* yg digunakan adalah bahasa pemrograman *Delphi* versi 7.0 untuk proses akusisi data dan *Matlab* versi 7.0 untuk komputasi proses pengenalannya.



Gambar 7. Isyarat kata 'abang'

Jumlah input yang masuk kedalam learning ANN adalah 101 yang terdiri 5x10 dan 3x17. Dimana 5 x 10 yaitu 5 adalah jumlah flex sensor dan 10 adalah jumlah range kemunculan nilai. Sedangkan 3 x 17 yaitu 3 sumbu x,y,z yang berasal dari accelerometer dan 17 adalah jumlah range kemunculan nilai.

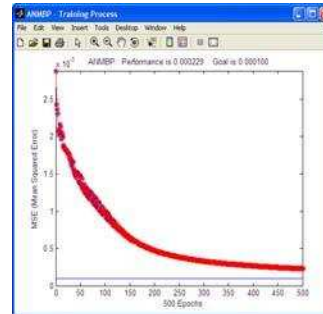
Struktur jaringan adalah *input layer* 101 unit, *hidden layer* ke-1 : 150, yang dibagi menjadi dua yaitu 75-75, hidden layer ke-2 : 80, yang juga dibagi dua yaitu 40-40, sedangkan output layer sebanyak 10 unit. Fungsi aktivasi yang digunakan adalah sigmoid biner. Bias ditambahkan pada input layer dan hidden layer. Untuk mencapai hasil yang optimal perlu diperhatikan pemilihan nilai konstanta  $\mu$ ,  $\lambda$  dan  $\epsilon$  pada *adaptive learning rate*. Pemilihan nilai ini dilakukan secara heuristik. Pada penelitian ini nilai konstanta  $\mu = 0.2$ ,  $\lambda = 0.00001$ ,  $\epsilon = 0.1$  serta bias = 0.5.



Gambar 8. proses learning untuk 10 kelas,200 epoch

Pada gambar 8 menunjukkan hasil learning untuk klasifikasi 10 kelas. Epoch dibatasi sampai 200. Disini dapat dilihat pada awal epoch, terjadi error yang naik turun dikarenakan pemilihan random neighborhoo pada hidden layer, tetapi menuju arah konvergen, error semakin kecil demikian error pada masing-masing

neighborhood berangsur-angsur sama. Waktu yang diperlukan untuk proses learning adalah 506.45 second. Akurasi yang diperoleh adalah 97% dengan menggunakan data testing.



Gambar 9. proses learning untuk 10 kelas,500 epoch

Pada gambar 9 ,dengan epoch 500,gerakan naik-turun neighborhoo tidak begitu kelihatan. Waktu yang diperlukan untuk proses learning ini adalah 1279.83 second. Akurasi yang diperoleh adalah 97% dengan menggunakan data testing

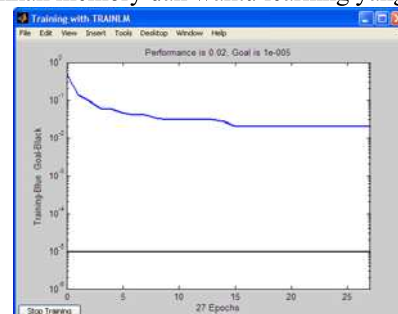
Akurasi yang diperoleh menggunakan rumus confusion matrik[16].

|              |         | Predicted Class |       |       |       |       |       |         |        |        |        |
|--------------|---------|-----------------|-------|-------|-------|-------|-------|---------|--------|--------|--------|
|              |         | Aku             | Abang | Agama | Bagus | Bapak | Bebek | Bendera | Cacing | Cahaya | Cantik |
| Actual Class | Aku     | 10              | 0     | 0     | 0     | 0     | 0     | 0       | 0      | 0      | 0      |
|              | Abang   | 0               | 9     | 0     | 0     | 0     | 0     | 0       | 2      | 0      | 0      |
|              | Agama   | 0               | 0     | 10    | 0     | 0     | 0     | 0       | 0      | 0      | 0      |
|              | Bagus   | 0               | 0     | 0     | 10    | 0     | 0     | 0       | 0      | 0      | 0      |
|              | Bapak   | 0               | 0     | 0     | 0     | 10    | 0     | 0       | 0      | 0      | 0      |
|              | Bebek   | 0               | 0     | 0     | 0     | 0     | 10    | 0       | 0      | 0      | 0      |
|              | Bendera | 0               | 0     | 0     | 0     | 0     | 0     | 10      | 0      | 0      | 0      |
|              | Cacing  | 0               | 1     | 0     | 0     | 0     | 0     | 0       | 8      | 0      | 0      |
|              | Cahaya  | 0               | 0     | 0     | 0     | 0     | 0     | 0       | 0      | 10     | 0      |
|              | Cantik  | 0               | 0     | 0     | 0     | 0     | 0     | 0       | 0      | 0      | 10     |

Gambar 10. Confusion matrix untuk akurasi

Dari 10 kelas, isyarat yang kurang sempurna dikenali adalah isyarat 'abang' dan 'cacing'. Kata 'abang' dikenali sebagai kata 'abang' sebanyak 9 buah dan dikenali sebagai kata 'cacing' sebanyak 2 buah. Kata 'cacing' dikenali sebagai kata 'cacing' sebanyak 8 buah dan dikenali sebagai kata 'abang' sebanyak 1 buah.

Gerak isyarat ini juga diujicoba dengan algoritma modifikasi backpropagation Lavenberg-Marquardt (TRAINLM). Algoritma Lavenberg-Marquardt membutuhkan memory dan waktu learning yang besar.



Gambar 11. proses learning 10 kelas

Waktu yang dibutuhkan untuk proses learning 10 kelas adalah 5001.23400 seconds. Perhitungan akurasi hanya mencapai 75% dengan testing menggunakan data testing.

## 7. KESIMPULAN

Penelitian ini mengembangkan sistem pengenalan bahasa isyarat Indonesia berbasis sensor *accelerometer* dan sensor *flex*. Pada tahap penelitian ini, dilakukan pengujian untuk mengenali 10 isyarat kata (kelas) SIBI dengan menggunakan metode ANN-ANMBP (*Adaptive Neighborhood based Modified BackPropagation*) Hasil pengujian menunjukkan bahwa metode ANN-ANMBP yang digunakan mampu mengenali isyarat kata dengan akurasi sampai dengan 97%.

Sampai saat ini penelitian masih dikembangkan untuk penggunaan *dataset* yang lebih besar. Penelitian ini mendapat bantuan dana dari Direktorat Jenderal Perguruan Tinggi melalui Beasiswa Program Pasca Sarjana (BPPS) untuk periode tahun akademik 2009/2010 sampai dengan 2010/2011.

## DAFTAR REFERENSI

- [1] Qutaishat et al (2007), *American sign language (ASL) recognition based on Hough transform and neural networks*, Expert Systems with Applications 32, pp. 24-37
- [2] Y.-H.Lee, C.-Y.Tsai, (2009), *Taiwan sign language (TSL) recognition based on 3D data and neural networks*, Expert Systems with Applications 36, pp. 1123–1128
- [3] M.AL-Rousan et al., (2009), *Video-based signer-independent Arabic sign language recognition using hidden Markov models*, Applied Soft Computing 9, pp. 990–999.
- [4] W.Gaoetal, (2004), *A Chinese sign language recognition sistem based on SOFM/SRN/HMM*, Pattern Recognition 37, pp. 2389–2402
- [5] C.Oz, M.C.Leu, (2007), *Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker*, Neuro computing 70, pp. 2891–2901
- [6] M.H. Yang, N. Ahuja(1999), *Recognizing hand gestures using motion trajectories*, IEEE 1,pp 466–472.
- [7] T.Kathirvalakumar et al,(2009),*Neighborhood based modified backpropagation algorithm using adaptive learning parameter for learning feedforward neural*, Neuro computing 72, pp. 3915-3921.
- [8] spectrasymbol.com, *Flex Sensor*.
- [9] Hitachi Metals, Ltd., *H48C ±3g 3→ Axis Accelerometer with IC*.
- [10] S. Abid, F. Fnaiech, M. Najim(2001), *A fast feedforward learning algorithm using a modified form of the sgerakrd backpropagation algorithm*, IEEE Trans. Neural Networks 12 ,pp 424–430.
- [11] A. Toledo, M. Pinzolas, J.J. Ibarrola, G. Lera(2005), *Improvement of the neighborhood based Levenberg–Marquardt algorithm by local adaptation of the learning coefficient*, IEEE Trans. Neural Networks 16 , pp 988–992.
- [12] L. Behera, S. Kumar, A. Patnaik(2006), *On adaptive learning rate that guarantees convergence in feedforward networks*, IEEE Trans. Neural Networks 17 ,pp 1116–1125.
- [13] Nguyen-Widrow(1999), *Improving the Learning Speed of 2-Layer Neural Networks by Choosing Initial Values of the Adaptive Weights*, Neural Networks, IJCNN International Joint Conference on
- [14] Jong Jek siang(2005), “*Jaringan Syaraf Tiruan & pemrogramannya menggunakan Matlab*”, Penerbit Andi.
- [15] <http://www.i-chat.web.id> diakses 4/6/2010
- [16][http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion\\_matrix/confusion\\_matrix.html](http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html) (1 of 3) diakses 2/5/2008.
- [17] Evita Tunjung Sekar (2001), *Perancangan dan Implementasi Prototipe Sistem Pengenalan Bahasa Isyarat*. Tesis Magister ITB. Bandung.
- [18] Mauridhi HP, Agus K(2006), *Supervised Neural Network dan Aplikasinya*, Penerbit Graha Ilmu
- [19] Ahmad Akl(2010), *A Novel Accelerometer-based Gesture Recognition System*, thesis, Department of Electrical and Computer Engineering, University of Toronto.
- [20] Jiayang Liu et al (2009), *uWave: Accelerometer-based personalized gesture recognition and its applications*, Pervasive and Mobile Computing 5, pp 657-675
- [21]<http://lmi.bwh.harvard.edu/papers/pdfs/2004/martin-fernandezCOURSE04b.pdf>, diakses 2/5/2008