APLIKASI JARINGAN SYARAF TIRUAN DALAM PENGENALAN POLA HURUF PEGON JAWA

Hari Surrisyad [(1)], Ahmad Subhan Yazid [(2)]
Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sunan Kalijaga
Jalan Laksda Adisucipto, Yogyakarta 55281
e-mail : hari.surrisyad@gmail.com [(1)] , anfalah@outlook.com [(2)]

Abstract

Artificial Neural Network (ANN) Technology) can help humans in processing data into information with design resembling the performance of the human brain. ANN adopts 5 aspects of human capability: Memorization, Generalization, Efficiency, Accuracy, and Tolerance in its application. ANN proves to be effective in pattern recognition. Researchers developed an application implementing ANN to recognize Java Pegon Letter pattern. The research uses 160 image data, divided into 100 training data (consisting of 5 normal images for each character) and 60 test data (consisting of 1 normal data, 1 data is not complete/corrupt, and 1 data with noise) for each character. The data obtained from the processed captures, so all of data have the same dimensions and size: 100x100 pixels. All data is processed through preprocessing and extraction stages. Furthermore, the data result is used in training stage to recognize the pattern of Java Pegon by applying the Learning Vector Quantization method. The application can recognize Java pegon pattern very well. The application can recognize 100% of training data and test data. This application also has the ability to recognize abnormal data very well, such as data with noise or corrupted data.

Keywords: Artificial Neural Network, Learning Vector Quantization, feature extraction, Java Pegon.

Abstrak

Teknologi Jaringan Saraf Tiruan (JST) dapat membantu manusia dalam memproses data menjadi informasi dengan desain menyerupai kinerja otak manusia. JST mengadopsi 5 Aspek kemampuan manusia, yakni: Memorisasi, Generalisasi, Efisiensi, Akurasi, dan Toleransi dalam penerapannya. JST terbukti efektif dalam hal pengenalan pola. Peneliti mengembangkan sebuah aplikasi yang mengimplementasikan JST untuk mengenali pola Huruf Pegon Jawa. Penelitian ini menggunakan 160 data citra, terbagi menjadi 100 data latih (terdiri dari 5 citra normal untuk setiap karakter) dan 60 data uji (terdiri dari 1 data normal, 1 data tidak lengkap (corrupt), dan 1 data dengan gangguan) untuk setiap karakter. Data didapat dari hasil capture yang diolah sehingga semua data memiliki dimensi dan ukuran yang sama, yaitu 100 x 100 pixel. Semua data diproses melalui tahapan preprocessing dan ekstraksi. Selanjutnya, dilakukan proses pelatihan untuk mengenali pola Pegon Jawa dengan menerapkan metode Learning Vector Quantization. Hasilnya, aplikasi dapat mengenali pola huruf pegon Jawa dengan sangat baik. Hal ini dibuktikan dengan kemampuan aplikasi yang dapat mengenali 100% data latih dan data uji. Aplikasi ini juga memiliki kemampuan dalam mengenali data-data tidak normal dengan sangat baik, seperti data dengan noise ataupun data yang corrupt.

Kata kunci: **Jaringan Syaraf Tiruan, Learning Vector Quantization, Ekstraksi Fitur, Pegon Jawa**.

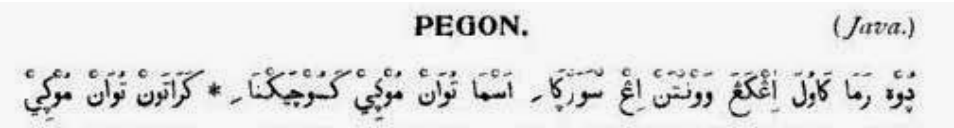
1. PENDAHULUAN

Jaringan saraf tiruan merupakan salah satu sistem pemrosesan informasi yang didesain dengan karakteristik menyerupai kinerja otak manusia (Kristanto 2004). Otak manusia memiliki 5 Aspek kemampuan, yaitu: memorisasi, generalisasi, efisiensi, akurasi, dan toleransi. Rekam data atau gangguan data di masa lalu akan dipelajari oleh jaringan saraf tiruan untuk memberikan keputusan terhadap data baru yang belum pernah dipelajari. Jaringan saraf tiruan sendiri telah banyak diaplikasikan ke dalam beberapa kasus dan terbukti efektif dalam pemecahan masalah, diantaranya adalah dalam hal pengenalan pola. Pengenalan pola

(*pattern recognition*) adalah suatu ilmu untuk mengklasifikasi atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu obyek (Putra 2010). Pengenalan pola sendiri saat ini sudah berkembang dan juga dimanfaatkan dalam banyak aspek, seperti pengenalan sidik jari yang berupa *image*, pengenalan suara, pengenalan tulisan dan lain-lain. Pada penelitian ini, jaringan saraf tiruan akan digunakan untuk mengenali pola Aksara Pegon Jawa yang memiliki keunikan dalam bentuk, dan masing-masing huruf terkadang hampir mirip satu dengan yang lainnya.

Pegon atau sering disebut Arab Pego atau Arab Jawi merupakan tulisan berabjad huruf Arab (huruf hijaiyah) yang berakulturasi dengan bahasa daerah di Indonesia, serta memiliki cara baca yang berbeda dengan bahasa Arab (Fathiyyah 2013). Sekilas tulisan Pegon akan terlihat seperti tulisan bahasa Arab pada umumnya. Namun, bila dicermati akan sangat berbeda. Dalam Pegon, abjad-abjad huruf hijaiiyah dipakai guna melafadzkan bahasa daerah di Indonesia. Huruf Pegon tidak hanya ada di Jawa dan Sunda. Di daerah Melayu, tulisan Pegon ini disebut dengan Arab Melayu karena menggunakan Bahasa Melayu atau Indonesia.

Dalam Penulisannya, Pegon yang berupa huruf vokal diwakili dengan huruf-huruf yang dalam tulisan Arab berfungsi untuk memanjangkan bacaan huruf, yakni alif (ا), wawu (و) dan ya (ي). Sedangkan huruf konsonan, tulisan Arab Pegon diwakili oleh huruf-huruf hijaiyah yang mirip bunyinya, seperti "n" dengan huruf nun, “m” dengan mim dan seterusnya. Untuk huruf yang tidak ada dalam abjad hijaiyyah seperti bunyi sengau “ng” atau dan huruf “c”, dipakai huruf tertentu dengan menambahkan titik tiga: Ng dengan ghoin (غ) titik tiga dan c dengan jim (ج) titik tiga.contoh kalimat pegon dapat dilihat pada gambar 1 berikut:



Gambar 1. Contoh penulisan huruf pegon jawa

Pada penelitian ini, akan dikembangkan suatu aplikasi Jaringan Syaraf Tiruan yang menerapkan 5 aspek kemampuan kinerja otak manusia. Metode Jaringan Syaraf Tiruan yang akan digunakan adalah Learning Vector Quantization (LVQ), Learning Vektor Quantization adalah metode untuk melakukan pembelajaran pada Lapisan kompetitif yang terbimbing (Widodo 2005). Model pembelajaran LVQ memiliki performa yang sangat baik, dalam hal kecepatan proses pelatihan, akurasi, dan klasifikasi pola.

2. METODE PENELITIAN

Pada penelitian ini, untuk mendapatkan data yang akurat dan konsisten dari setiap sampel, digunakan suatu metode ekstraksi fitur sederhana yaitu *Zoning,* dengan cara membagi area data menjadi beberapa zona kemudian menghitung jumlah pixel aktif yang terdapat pada tiap zona dari sampel.

Sedangkan metode pelatihan yang dipilih adalah Metode LVQ dengan pertimbangan, bahwa pembawaan dari input dan output tidak harus dalam bentuk diskrit atau biner, bisa dalam bentuk bilangan numerik. Tipe pelatihannya adalah pelatihan terawasi.

2.1 Perancangan

1. Perancangan Jaringan Syaraf Tiruan

   a. Tingkat Node

   Melihat karakteristik dari data yang akan dikenali adalah data berbentuk citra, Menurut (Munir 2004) citra adalah gambar pada bidang dwimatra (dua dimensi). maka tipe input data yang paling cocok adalah bertipe bilangan numerik, demikian pula dengan outputnya. Sedangkan fungsi aktifasi yang akan digunakan adalah

fungsi aktifasi Linier (Puspaningrum 2006), dengan alasan bahwa fungsi aktifasi ini adalah fungsi aktifasi yang paling umum digunakan pada metode LVQ,

Fungsi aktifasi linier didefinisikan dalam fungsu (1) berikut:

$$Y=X \qquad (1)$$

b. Tingkat Jaringan

Pada tingkat jaringan, dilakukan penentuan banyak lapisan yang akan digunakan. Dengan metode LVQ ini, lapisan input yang akan digunakan hanya 1. Secara keseluruhan, jaringan LVQ yang digunakan dalam pengenalan huruf pegon ini akan terdiri dari 2 lapisan, yaitu 1 lapisan input dan 1 lapisan output.

Jaringan Learning Vector Quantization tersebut dirancang untuk dapat menerima input bilangan biner maupun numerik, sehingga data harus diatur sebagai sekumpulan angka (Vektor). Elemen-Elemen yang dikelompokkan bersama ke dalam sebuah vektor ini menggambarkan *feature* data dan karena disusun bersama-sama maka akan mempermudah jaringan untuk mengenali *feature* tersebut.

c. Tingkat Pelatihan

Pada tingkat pelatihan, akan ditentukan parameter-parameter yang digunakan dan ketentuan ketentuan lain saat proses pelatihan dalam penelitian.

Parameter yang akan digunakan pada pelatihan:

- Lapisan input sebanyak 25 Node
- Lapisan output sebanyak 20 Node

Penentuan 25 node input ini didasarkan dari jumlah deret bilangan yang akan dihasilkan pada proses ekstraksi citra. Sedangkan penentuan 20 node output dikarenakan pada pelatihan jaringan LVQ, setiap output mewakili kelas tertentu, dimana output nilai paling minimum akan dimasukkan pada kelas yang mewakili pola tersebut. Oleh karena itu, jumlah node output disesuaikan dengan banyak karakter pada huruf pegon, yaitu 20.

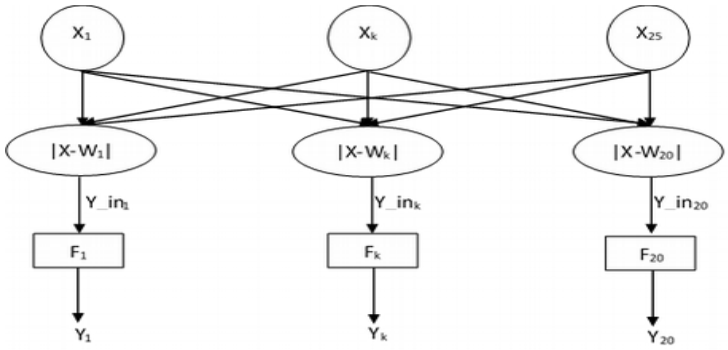
Seperti yang sudah dijelaskan sebelumnya, ketentuan kelas yang akan digunkan akan dapat dilihat pada tabel 1 dibawah ini:

Tabel *1* Pengelompokan Kelas Tiap Karakter

Karakter	Dibaca	Kelas
ه	Ha	1
ن	Na	2
چ	Ca	3
ر	Ra	4
ك	Ka	5
د	Da	6
ت	Ta	7
س	Sa	8
و	Wa	9
ل	La	10
ف	Pa	11
ذ	Dha	12
ج	Ja	13
ي	Ya	14
ڽ	Nya	15
م	Ma	16
ڬ	Ga	17
ب	Ba	18
ط	Tha	19
غ	Nga	20

2. Arsitektur Jaringan

Pembangunan Aplikasi dimulai dari perancangan arsitektur jaringannya. Gambar 2 di bawah ini adalah rancangan arsitektur Jaringan LVQ pengenalan huruf pegon jawa.



Gambar 2. Rancangan Arsitektur Jaringan LVQ pengenalan pegon jawa

Dari Gambar diatas dapat dilihat bahwa:

- Unit Input : 25 Node
- Unit Output : 20 Node

Hal lain yang perlu diperhatikan dalam perancangan jaringan LVQ pada penelitian ini adalah:

- *Minimal Error(Limit)* : 0.01
- *Learning Rate(Alpha)* : 0.1
- *Reduce Alpha* : 0.1
- *Max Epoch* : 10
- Jumlah Data latih : 100

Sedangkan untuk bobot yang akan dilatih oleh jaringan LVQ pada penelitian ini adalah W_i = 25 x 20 = 500 buah

2.2 Pengumpulan Data

Pengumpulan data dilakukan dengan meng-*capture* huruf pegon jawa cetak. Untuk memperoleh 160 data berupa citra dengan 8 data untuk tiap karakter yang akan di olah, citra akan dipotong sedemikian rupa sehingga memiliki dimensi yang sama, kemudian citra akan di-*resize* menjadi 100x100 pixel. Baru kemudian data yang diperoleh tersebut akan dibagi menjadi 100 data *training* yang terdiri dari 5 citra normal untuk setiap karakter dan 60 data *testing* yang terdiri dari 1 data normal, 1 data *corrupt*, dan 1 data *noise* untuk setiap karakter. Data *training* digunakan untuk melatih jaringan, sedangkan data *testing* digunkan untuk menguji apakah jaringan yang telah di-*training* menghasilkan luaran yang diinginkan.

2.3 Alur Kerja Aplikasi

Alur kerja untuk aplikasi jaringan syaraf tiruan menggunakan metode LVQ dalam pengenalan pola huruf pegon jawa, adalah sebagai berikut :

1. Input Citra
   Sebelum Citra diekstraksi, citra akan diinputkan melalui tahapan *preprocessing*, supaya sesuai dengan kriteria fitur ekstraksi. *preprocessing* yang akan dilakukan adalah, pengabuan, binerisasi, dan *convert* citra ke *binary data*.

Pertama, citra yang diinputkan satu persatu akan diubah menjadi citra biner dengan rumus pengabuan (2):

$$X = (R+G+B)/3$$
$$Warna = RGB(X, X, X) \quad (2)$$

berikut pseudocode implementasi dari pengabuan citra:

```
Read citra;
For i=0 to length(citra)do
     For j=0 to width(citra)do
            Abu_ij = (r_ij+g_ij+b_ij)/3;
     end
end
```

Sehingga menghasilkan citra abu, sebagaimana gambar 3:

Gambar 3. Citra Abu

Citra asli yang sudah diubah menjadi citra abu inilah yang akan diubah menjadi citra biner. Cara merubahnya adalah dengan menentukan nilai keabuan yang akan digunakan sebagai nilai batas dengan fungsi (3):

$$f(x,y)' = \begin{cases} a_1, & f(x,y) < T \\ a_2, & f(x,y) \geq T \end{cases} \quad (3)$$

Pada penelitian ini, penentuan nilai ambang atau batas keabuan ditentukan dengan metode Otsu. *Pixel* dengan derajat keabuan lebih besar dari nilai batas akan diubah menjadi hitam dan sebaliknya *pixel* dengan derajat keabuan lebih kecil dari nilai batas akan diubah menjadi putih, berikut pseudocode dari binerisasi citra:

```
Read citra;
For i=0 to length(citra)do
     For j=0 to width(citra)do
            Abu_ij = (r_ij+g_ij+b_ij)/3;
            If Abu<= tet then
                  Biner_ij=0;
            Else
                  Biner_ij=255;
     end
end
```

dan hasilnya berupa citra biner yang ditampilkan dalam gambar 4:

Gambar 4. Citra Biner

2. Pengenalan Pola Citra

Setelah *preprocessing* selesai, barulah citra biner yang dihasilkan akan diubah menjadi *binary data.* Setiap *pixel* putih akan dianggap bernilai 0 dan pixel hitam akan dianggap bernilai 1. Gambar 5 merupakan *binary data* hasil dari tahapan 2:

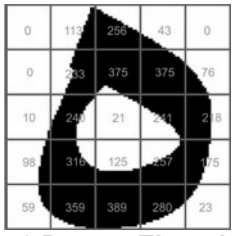
Gambar 5. *Binary Data*

Citra biner yang diubah memiliki ukuran panjang 100 *pixel* dan lebar 100 *pixel*, maka tahapan ini menghasilkan 100x100=10000 *binary data*.

Pada penelitian ini, digunakan suatu metode ekstraksi sederhana yaitu dengan menghitung jumlah *pixel* aktif yang terdapat pada bagian-bagian dari citra. Adapun langkah-langkah umum dari ekstraksi data numerik dari setiap sampel adalah sebagai berikut:

- Setiap sampel yang diamati, dibagi menjadi beberapa area, yaitu 5 kolom dan 5 baris, sehingga akan terdapat 25 area pengamatan
- Jumlah pixel yang aktif dari setiap area yang ada, dihitung secara akurat;
- Dihasilkan sejumlah 25 data numerik dengan atribut kolom dan baris yang diharapkan dapat mewakili data ciri dari sampel yang diamati.

Langkah-langkah tersebut dapat diilustrasikan dengan gambar 6 dibawah ini:



Gambar 6 Proses Ekstraksi Citra

Sehingga dari citra sampel diatas, akan diperoleh 25 data numerik sebagai berikut: [0, 113, 256, 43, 0, 0, 233, 375, 375, 76, 10, 240, 21, 241, 218, 98, 316, 125, 257, 175, 59, 359, 389, 280, 23]. Data numerik ini yang akan dijadikan data input dalam proses pelatihan.

3. Proses Pelatihan

Proses pelatihan dilakukan dengan metode LVQ, dimana pada metode ini hanya terdeapat 2 layer, yaitu layer input dan output, tanpa hidden layer. Fungsi aktifasi yang digunakan adalah fungsi linier.

Pelatihan diawali dengan menginisialisasi parameter-parameter pelatihan, seperti yang sudah ditentukan pada bab sebelumnya. parameter yang digunakan adalah: *Minimal Error*: 0.01, *Learning Rate* (α): 0.1, *Reduce Alpha* : 0.1 dan *max Epoch* : 10. Setelah itu, memuat data pelatihan yang sudah diinputkan pada langkah sebelumnya, dan yang terakhir adalah menentukan bobot awal pelatihan, pada LVQ, jumlah bobot awal dibagi sebanyak kelas yang ada, setiap kelas bobot mewakili masing masing pola huruf, dan setiap kelas bobot memiliki jumlah sebanyak node input. Penentuan bobot awal pada aplikasi ini dilakukan dengan cara mengambil secara acak satu node input pada setiap kelas huruf, sebagaimana disajikan pada tabel 2 berikut:

Tabel 2 Sampel Inisialisasi Bobot Awal

Bobot Awal	Kelas	Mewakili Pola
[0, 105, 252, 41, 0, 0, 227, 370, 374, 73, 9, 230, 15, 229, 208, 98, 318, 132, 265, 161, 58, 348, 373, 261, 15]	1	Ha
[0, 0, 195, 0, 0, 0, 8, 0, 31, 93, 48, 39, 0, 17, 161, 128, 6, 0, 26, 141, 78, 292, 286, 286, 39]	2	Na
[31, 275, 372, 304, 72, 2, 163, 78, 52, 0, 95, 17, 211, 148, 0, 112, 100, 38, 100, 0, 5, 234, 287, 247, 43]	3	Ca
[0, 0, 0, 133, 86, 0, 0, 0, 21, 197, 0, 0, 0, 0, 172, 0, 0, 0, 183, 144, 62, 170, 287, 186, 0]	4	Ra
[0, 0, 0, 118, 128, 0, 0, 105, 67, 141, 13, 0, 203, 90, 162, 111, 55, 133, 19, 167, 203, 281, 280, 280, 258]	5	Ka
[0, 0, 195, 86, 0, 0, 0, 74, 283, 3, 0, 0, 0, 116, 71, 29, 55, 37, 70, 120, 51, 384, 400, 400, 140]	6	Da
[0, 7, 77, 24, 0, 12, 64, 135, 12, 27, 90, 0, 0, 0, 179, 246, 211, 215, 220, 276, 2, 38, 40, 33, 20]	7	Ta
[0, 0, 0, 0, 0, 27, 0, 131, 21, 138, 83, 0, 60, 249, 237, 243, 186, 210, 3, 0, 13, 46, 0, 0, 0]	8	Sa
[0, 0, 10, 321, 150, 0, 0, 77, 215, 197, 0, 0, 12, 227, 219, 0, 0, 0, 177, 122, 65, 154, 270, 212, 0]	9	Wa
[0, 0, 0, 195, 6, 0, 0, 0, 129, 0, 0, 34, 0, 117, 4, 0, 84, 0, 86, 35, 0, 229, 240, 242, 4]	10	La
[0, 0, 0, 80, 35, 0, 0, 0, 105, 127, 34, 0, 0, 22, 218, 173, 34, 30, 93, 301, 86, 188, 189, 180, 171]	11	Pa
[0, 14, 174, 0, 0, 0, 7, 97, 3, 0, 0, 0, 95, 155, 0, 0, 1, 0, 115, 0, 0, 174, 300, 265, 0]	12	Dha
[55, 280, 374, 293, 29, 6, 168, 56, 0, 0, 96, 8, 129, 47, 0, 117, 86, 0, 0, 0, 8, 244, 292, 255, 43]	13	Ja
[4, 3, 0, 182, 197, 86, 2, 95, 147, 54, 144, 0, 23, 199, 156, 124, 267, 283, 188, 6, 0, 70, 203, 50, 0]	14	Ya
[0, 50, 195, 20, 0, 0, 71, 156, 53, 94, 34, 53, 0, 10, 163, 123, 12, 0, 22, 149, 63, 294, 280, 285, 43]	15	Nya
[0, 39, 270, 210, 0, 0, 107, 285, 355, 31, 0, 208, 19, 0, 0, 0, 172, 0, 0, 0, 0, 108, 11, 0, 0]	16	Ma
[0, 0, 146, 65, 49, 0, 0, 67, 105, 140, 2, 0, 182, 86, 142, 94, 41, 121, 36, 158, 172, 261, 260, 260, 224]	17	Ga
[6, 0, 0, 0, 14, 97, 0, 0, 0, 180, 255, 188, 188, 200, 262, 10, 60, 123, 54, 40, 0, 0, 88, 0, 0]	18	Ba
[0, 227, 38, 0, 0, 0, 154, 17, 0, 0, 0, 69, 75, 0, 0, 0, 44, 219, 246, 249, 193, 323, 285, 251, 299]	19	Tha
[0, 171, 23, 0, 0, 0, 122, 82, 0, 0, 0, 166, 189, 56, 0, 0, 122, 1, 0, 0, 0, 135, 176, 94, 0]	20	Nga

Kemudian bila semua kebutuhan untuk pelatihan sudah siap, maka pelatihan bisa dimulai. Pelatihan diawali dengan menghitung *ecluidean distance* tiap pola terhadap tiap kelas dengan bobot awal, dengan rumus (4):

$$d_j^2 = \sum_{i=0}^{n-1} (X1(t) - W_{ij})^2 \quad (4)$$

Berikut pseudcode untuk penghitungan *Ecluidean Distance* pada pola ke-n:

```
for j=0 to length(Bobot) do
   for k=0 to length(NodeInput) do
         temp+= (DataLatih_nk-Bobot_jk)^2;
   end
   distance_j=sqrt(temp);
end
```

Penghitungan *ecluidean distance* pada pola 1 Menghasilkan:

Jarak terhadap Kelas 1 = 0
Jarak terhadap Kelas 2 = 776.627
Jarak terhadap Kelas 3 = 769.337
Jarak terhadap Kelas 4 = 842.291
Jarak terhadap Kelas 5 = 807.207
Jarak terhadap Kelas 6 = 629.883
Jarak terhadap Kelas 7 = 828.378
Jarak terhadap Kelas 8 = 904.735
Jarak terhadap Kelas 9 = 769.021
Jarak terhadap Kelas 10 = 774.6
Jarak terhadap Kelas 11 = 815.187
Jarak terhadap Kelas 12 = 782.164
Jarak terhadap Kelas 13 = 840.434
Jarak terhadap Kelas 14 = 776.769
Jarak terhadap Kelas 15 = 655.923
Jarak terhadap Kelas 16 = 747.075
Jarak terhadap Kelas 17 = 739.263
Jarak terhadap Kelas 18 = 943.312
Jarak terhadap Kelas 19 = 829.714
Jarak terhadap Kelas 20 = 814.92

Kemudian cari jarak terendah terhadap masing-masing kelas. Kemudian, bandingkan dengan kelas target dari pola tersebut, jika sesuai maka bobot yang mewakili kelas target pola n akan diupdate dengan rumus (5):

$$w_j(baru)=w_j(lama)+ \alpha \ (x-w_j(lama)) \quad (5)$$

jika tidak sesuai maka update bobot yang mewakili kelas target pola n dengan rumus (6):

$$w_j(baru)=w_j(lama)- \alpha \ (x-w_j(lama)) \quad (6)$$

berikut pseudocede dari proses update bobot:

```
If distanceMin = KelasTarget then
   for i=0 to length(Bobot)
       Bobot_baru=Bobot_lama + Alpha * (DataLatih_ni-Bobot_distanceMin i)
   end
Else
   for i=0 to length(Bobot)
       Bobot_baru=Bobot_lama + Alpha * (DataLatih_ni-Bobot_distanceMin i)
   end
```

Pada pola 1, jarak terendah adalah jarak terhadap kelas 1 dan sesuai dengan kelas target dari pola 1, maka bobot yang mewakili kelas target pola 1 adalah bobot 1, sehingga menghasilakan bobot baru sebagai berikut:

W [1][1]= 0
W [1][2]= 105.0
W [1][3]= 252.0
W [1][4]= 41.0
W [1][5]= 0
W [1][6]= 0
W [1][7]= 227.0
W [1][8]= 370.0
W [1][9]= 374.0
W [1][10]= 73.0
W [1][11]= 9.0
W [1][12]= 230.0
W [1][13]= 15.0
W [1][14]= 229.0
W [1][15]= 208.0
W [1][16]= 98.0
W [1][17]= 318.0
W [1][18]= 132.0

W [1][19]= 265.0
W [1][20]= 161.0
W [1][21]= 58.0
W [1][22]= 348.0
W [1][23]= 373.0
W [1][24]= 261.0
W [1][25]= 15.0

Proses ini akan terus dilakukan pada setiap pola input yang ada. Ketika proses penghitungan *ecluidean distance* pada setiap pola selesai, maka maka nilai Alpha akan dikurangi (*reduce*). Jika prosedur pelatihan berdasarkan max Epoch, maka pelatihan akan selesai pada perulangan ke 10. Namun, jika berdasarkan min Error, proses di atas akan terus berulang hingga nilai alpha/*learnig rate* di bawah nilai minimal error yang sudah ditentukan yaitu 0.01.

Pada penelitian ini, proses pelatihan berhenti pada epoch ke-22 dengan waktu training yang relatifi cepat. Hasilnya ditampilkan pada tabel 3:

Tabel 3 Bobot Hasil Pelatihan

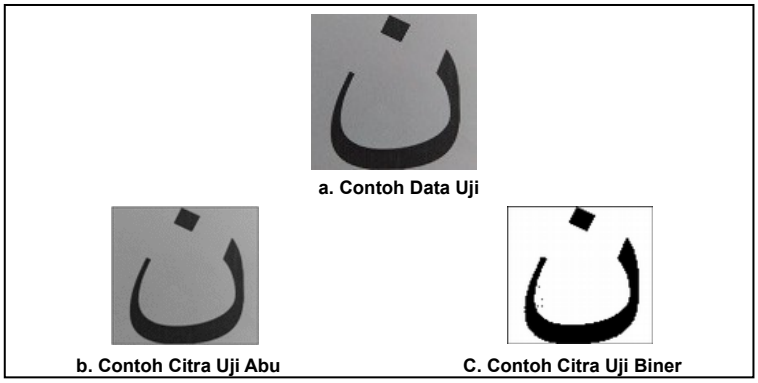
Bobot Hasil Latih	Kelas	Pola
[0.0, 111.35325739593824, 253.39802540724574, 41.785580824010665, 0.0, 0.0, 232.96200516768613, 373.8606903175981, 373.78375418334275, 73.37204034563746, 10.465002624990214, 234.0720947758441, 18.49572740877876, 237.22147612151616, 211.76984545799064, 98.99259208047555, 316.10089629854934, 126.5723274153886, 260.22135564443766, 168.80313553531553, 58.76544032787791, 354.3792859876978, 383.3450473405276, 271.93958367951456, 19.430546780297995]	1	Ha
[0.0, 0.0, 176.60753104571944, 2.5484733549107883, 0.0, 0.0, 10.416963258898548, 0.0, 22.025698088431636, 102.12239776123766, 45.026057976736936, 38.18881601622767, 0.0, 6.995006384057916, 155.06979940323635, 128.0166471620624, 6.762866149850839, 0.0, 17.932622699353306, 140.45918163266697, 76.38376920492225, 292.52370796873686, 278.6201045068523, 283.1405190825584, 47.820601293127126]	2	Na
[41.78060602384142, 271.5725835522662, 372.42063099295217, 295.42109033136313, 45.47996932886317, 6.454658299080618, 163.7288669608025, 74.15023057468443, 44.452742132927234, 0.0, 99.2459372200484, 16.61760717311419, 214.62151568671692, 126.96497389673006, 0.0, 123.00681420183663, 86.35632373519842, 46.64497439068365, 82.77006936429412, 0.0, 10.014044022856893, 244.3766649211302, 290.39249468374084, 246.7844181245173, 35.6019084738373]	3	Ca
[0.0, 0.0, 0.0, 133.2731591014508, 85.36087860918815, 0.0, 0.0, 0.0, 19.317488352766823, 188.86742771159726, 0.0, 0.0, 0.0, 0.5942455599094144, 172.6029982911101, 0.0, 0.0, 0.21056217241795236, 183.61822146768057, 145.18093804875014, 58.118145934226675, 168.13557935065032, 283.31413850799174, 185.0679034778819, 0.19572874872129675]	4	Ra
[0.0, 0.0, 0.0, 115.14200503905539, 127.7158311003402, 0.0, 0.0, 104.02550536046954, 63.73776128246515, 142.6393688734687, 12.260721830097916, 0.0, 199.10014134765947, 89.26096850603977, 161.63374653057514, 109.48140620848149, 53.29584239390435, 133.10273921698646, 20.506094813091323, 166.0520003500793, 198.43447030598566, 274.8820033821013, 272.62700114403964, 275.60482428597294, 251.89131512958642]	5	Ka
[0.0, 0.0, 214.18207068767515, 76.59576554471697, 0.0, 0.0, 0.0, 84.94256313171762, 280.4231940604508, 0.7561579561749621, 0.0, 0.0, 0.0, 136.39723299805593, 53.57079094090119, 42.97434463974693, 45.20252200218979, 27.5674765836725, 79.46381516427276, 104.79041428551889, 70.68477614711252, 386.77905964436735, 400.0, 400.0, 120.05559125346096]	6	Da
[0.0, 3.395359925576768, 63.789129626979765, 19.31892809756601, 0.0, 8.80373821505721, 65.95427940767055, 148.79502600440603, 17.239472870384166, 19.965131626832708, 91.50535855825851, 0.0, 0.0, 0.0, 180.64602938808886, 254.23455705908464, 202.40406180174634, 199.206236227209, 206.33919304989652, 265.29672379329895, 7.318501192764844, 57.09163096137169, 51.240139223261, 46.88207333673627, 44.07702494111479]	7	Ta
[0.0, 0.0, 0.0, 0.0, 0.0, 23.400799590258448, 0.0, 125.7672313213104, 19.748550179519444, 129.6385485052099, 82.80523748968048, 0.0, 61.83652094229689, 244.18852019725244, 241.05034982139242, 235.97469131815686, 166.32103643353904, 209.10259235222392,	8	Sa

Bobot Hasil Latih	Kelas	Pola
4.186648705005906, 0.0, 25.05833980743826, 63.26794772331573, 5.259497961393481, 0.0, 0.0]		
[0.0, 0.0, 10.738251434340365, 321.4973107156941, 150.40960752502752, 0.0, 0.0, 72.91091284729143, 221.05295138068547, 201.76704749145614, 0.0, 0.0, 9.97819277244481, 218.6088410267919, 227.26425220597187, 0.0, 0.0, 0.0, 166.62215051722953, 133.46878139921964, 58.29504793509653, 147.5365370631228, 262.5349861758303, 227.37516707752718, 1.007786038282572]	9	Wa
[0.0, 0.0, 0.0, 198.51364994957714, 6.0093130689122205, 0.0, 0.0, 0.0, 132.52478965915714, 0.0, 0.0, 37.541147016112824, 0.0, 122.31032540606157, 3.6261780908008596, 0.3836833874914623, 86.57859990178869, 0.0, 90.31141029649032, 34.616316612508534, 0.7673667749829246, 240.8289156992231, 240.99792455999028, 244.96854539430373, 4.005974423316758]	10	La
[0.0, 0.0, 0.0, 83.14390987389469, 29.011882098989503, 0.0, 0.0, 0.0, 105.99681799902373, 123.36428082869244, 34.906861934618945, 0.0, 0.0, 25.388325495994664, 205.51816920614834, 173.52955204420533, 34.97616459121342, 27.29278535371124, 98.18340658856518, 291.6870940140017, 88.3606679583171, 190.33781374872743, 190.3094357395262, 183.8368338749146, 176.78870692906577]	11	Pa
[0.0, 20.7321554314109, 159.7942127446944, 0.0, 0.0, 0.0, 11.666428619587874, 94.04971533135395, 2.1811970584628964, 0.0, 0.0, 0.0, 103.43077518616356, 142.13403092979235, 0.0, 0.0, 1.4003164978224925, 0.0, 113.98387429328413, 0.0, 0.0, 188.4928980497523, 292.02382155087327, 246.06170399344927, 0.0]	12	Dha
[50.219696164113095, 280.3002223997773, 376.9251741167781, 300.0664261813401, 44.8230512438955, 5.310895138522909, 167.32393059799873, 64.20183774513309, 0.0, 0.0, 95.01584254644473, 10.415086844230023, 124.98059516389563, 54.006645477460445, 0.0, 116.21418848610304, 85.44133969295079, 0.0, 0.0, 0.0, 7.918366802382985, 239.523044321489, 286.3085590243697, 251.23400052727408, 39.37615560602568]	13	Ja
[1.6685927159632201, 2.070184347702864, 0.5640695864873808, 188.2045663233821, 183.40023943378947, 81.57014184900895, 3.2243076083151925, 92.42168980068026, 146.4052471391133, 53.03384061010504, 135.10717753389412, 0.0, 25.118241107715285, 201.24406126210908, 149.66610953659458, 122.67331960520144, 265.11416391188914, 278.7840545953537, 189.53578367847888, 6.257863233097654, 0.0, 52.456329070828865, 204.1671159919097, 57.53132902142431, 0.0]	14	Ya
[0.0, 45.14408424984492, 193.9505477509208, 19.552602510862652, 0.0, 0.0, 67.33748067669676, 156.3716832756452, 54.71882285782807, 94.24607407492412, 29.242648730149025, 57.624037005681394, 0.0, 10.50721507833275, 160.26591728969439, 118.30065085856502, 16.91262811993825, 0.0, 20.987109682873086, 146.17854625503952, 59.57117488194985, 294.0356416716965, 277.9228595011649, 287.08964068022163, 45.92734764555366]	15	Nya
[0.0, 43.16756535912565, 276.728544751903, 197.22314362698714, 0.0, 0.0, 95.32851984080452, 272.208544198036, 364.3348799753901, 16.857613682090776, 0.0, 210.14237888119646, 32.32040574508574, 0.5934483511874635, 0.0, 0.0, 165.00518180885138, 1.1900283563663316, 0.0, 0.0, 0.0, 90.6735499558382, 22.446503515285862, 0.0, 0.0]	16	Ma
[0.0, 0.21056217241795236, 145.64693122777442, 70.24438751863704, 44.171650402746664, 0.0, 0.0, 74.6730909101515, 118.15853976669403, 128.88578232254488, 1.6007273868413 09, 0.0, 196.6370459714239, 93.65229981125374, 131.01362738088764, 94.70882041116596, 46.37330847364527, 128.25214792431336, 33.54500459266847, 160.73217004250938, 182.93565921976878, 260.30382005063234, 258.6571081437798, 261.2445855164776, 221.22279883406415]	17	Ga
[4.714584745508012, 0.0, 0.0, 0.0, 16.872392835746187, 89.83422938137375, 0.0, 0.0, 0.0, 180.99874023924147, 252.48822509013436, 187.5643610209349, 188.81005288646867, 193.10338257347658, 264.62499068838656, 14.00670276762954, 69.22576247188937, 119.93811356652259, 52.42520591035188, 46.767695096558235, 0.0, 0.0, 102.07388068623382, 0.0, 0.0]	18	Ba
[0.0, 230.9680738849557, 39.950296709711054, 0.0, 0.0, 0.0, 155.80242631207946, 18.71738698287566, 0.0, 0.0, 0.0, 75.99815982705498, 74.54323633585618, 0.0, 0.0, 0.0, 49.75771573312181, 220.54689306201917, 254.28978442793283, 247.15139491891827, 205.6541961875536, 325.3709467461766, 283.2300340342629, 257.1213139803252, 306.11350558226593]	19	Tha
[0.0, 153.66144913619476, 40.0074351072888, 0.0, 0.0, 0.0, 103.60659156407232, 98.50786869512152, 0.20145649609620303, 0.0, 0.0, 134.61544729162296, 195.93083258292745, 72.09871882952548, 0.0, 0.0, 111.29485379226469, 6.1582881923632655, 0.0, 0.0, 0.0, 113.68722645775601, 183.78136485603568,	20	Nga

Bobot Hasil Latih	Kelas	Pola
122.172492466508, 0.405507764148735]		

4. Proses Pengenalan Pola

Pada proses pengenalan, akan dilakukan proses yang hampir sama dengan proses sebelumnya. Pertama, dilakukan input citra. Pada proses input citra, sama dengan proses input citra yang sudah dibahas sebelumnya, yakni *preprocessing* citra, yaitu pengabuan dan binerisasi, baru kemudian dilakukan ekstraksi. Hasil ekstraksi inilah yang diproses melalui pengklasifikasian dengan menghitung *ecluidean distance* terhadap tiap kelas dengan menggunakan bobot yang sudah dilatih, contoh *preprocessing* ditampilkan oleh gambar 7 berikut:



Gambar 7. Contoh *preprocessing*

Lalu ekstraksi fitur citra menghasilkan deret angka numerik sebagai berikut:

[0, 0, 181, 0, 0, 0, 6, 0, 33, 93, 44, 37, 0, 10, 145, 122, 4, 0, 19, 136, 76, 285, 270, 281, 43]

Baru kemudian proses penghitungan *ecluidean distance* dengan rumus 4, terhadap tiap kelas, proses ini sama dengan proses saat pelatihan dan menghasilkan:

Jarak terhadap Kelas 1 = 784.464
Jarak terhadap Kelas 2 = 23.89
Jarak terhadap Kelas 3 = 603.225
Jarak terhadap Kelas 4 = 372.467
Jarak terhadap Kelas 5 = 452.948
Jarak terhadap Kelas 6 = 415.066
Jarak terhadap Kelas 7 = 598.489
Jarak terhadap Kelas 8 = 652.252
Jarak terhadap Kelas 9 = 567.237
Jarak terhadap Kelas 10 = 416.577
Jarak terhadap Kelas 11 = 360.678
Jarak terhadap Kelas 12 = 364.561
Jarak terhadap Kelas 13 = 571.571
Jarak terhadap Kelas 14 = 697.053
Jarak terhadap Kelas 15 = 181.384
Jarak terhadap Kelas 16 = 739.597
Jarak terhadap Kelas 17 = 368.048
Jarak terhadap Kelas 18 = 659.287
Jarak terhadap Kelas 19 = 597.755
Jarak terhadap Kelas 20 = 511.889

Dari penghitungan jarak di atas, dapat dilihat bahwa jarak terendah adalah jarak terhadap kelas 2 yaitu 23.89, sehingga dapat ditarik kesimpulan bahwa citra yang diuji diklasifikasikan sebagai kelas 2 yang mewakili Karakter “Na”. Perlu diingat bahwa citra yang akan diuji harus diolah terlebih dahulu sebagaimana data latih.

3. HASIL DAN PEMBAHASAN

Dari penelitian ini dapat diketahui bagaimana cara mengembangkan aplikasi Jaringan Syaraf Tiruan Learning Vector Quantization untuk mengenali Huruf pegon jawa. Dari proses pengembangan tersebut maka didapatkan hasil kemampuan yang dimiliki aplikasi ini dalam berbagai aspek kemampuan sebagai berikut:

3.1 Aspek Memorisasi

Pada aspek memorisasi atau ingatan jaringan, didapatkan hasil bahwa aplikasi ini memiliki kemampuan ingatan jaringan yang sangan baik, dibuktikan dengan pengenalan data uji sebelumnya, aplikasi ini dapat mengenali 100% data uji, yang berjumlah 100 data dengan sangan baik.

3.2 Aspek Generalisasi

Pada aspek generalisai atau kemampuan untuk menggerelisasi hasil pelatihan, didapatkan hasil yang sangat memuaskan. Hal ini dibuktikan pada proses pengenalan data uji, aplikasi ini dapat menggeneralisasi pengetahuan yang didapat saat pelatihan dan mengenalai 100% data uji, yang berjumlah 40 data dengan sangat baik.

3.3 Aspek Efisiensi

Aspek ini berkaitan kompleksitas algoritma pembelajaran dan kecepatan pembelajaran. Hasil dari pengembangan aplikasi ini menunjukan tingkat kompleksitas algoritma pembelajaran di tingkat menengah, Sedangkan waktu pembelajaran yang dibutuhkan relatif cepat dengan hanya melakukan 22 *epoch*.

3.4 Aspek Akurasi

Aspek ini berkenaan dengan keakuratan sebuah aplikasi dalam mengenali suatu pola. Pada aplikasi ini didapatkan akurasi yang sangat baik, yaitu 100%. Hal ini dibuktikan dengan aplikasi berhasil mengenali semua data latih maupun data uji

3.5 Aspek Toleransi Noise dan Corrupt

Pada aspek ini, yang dibahas adalah seberapa mampu aplikasi menoleransi data *noise* (data yang memiliki gangguan) dan data *corrupt* (data yang tidak lengkap) pada setiap data. Hasil penelitian menunjukkan bahwa, aplikasi ini dapat menoleransi *noise* berupa bintik dan *noise* perubahan warna gelap pada citra dengan sangat baik. Sedangkan untuk data *corrupt* aplikasi juga dapat mengenali data-data tersebut. Hal ini dapat dibuktikan saat melakukan pengujian dengan data citra yang mengandung *noise* dan *corrupt*, aplikasi dapat mengenali data citra sesuai dengan apa yang diharapkan.

4. KESIMPULAN

Dari pengembangan aplikasi yang telah dilakukan pada penelitian ini, dapat ditarik kesimpulan sebagai berikut:

4.1 Fitur ekstraksi yang digunakan pada penelitian ini, dianggap sudah bisa mendapatkan pola yang mewakili pola setiap huruf. Hal ini dibuktikan pada proses penghitungan *ecluidean distance* pola terhadap kelas, pola dari karakter yang sama pasti akan memiliki jarak yang kecil.

4.2 Proses pelatihan dengan Jaringan Syaraf Tiruan Learning Vector Quantization, berjalan dengan sangat baik. Proses pelatihan berlangsung dalam jangka waktu yang relatif singkat, yakni hanya 22 Epoch dan mampu mengklasifikasikan tiap karakter dengan sangat baik.

4.3 Aplikasi ini dapat dengan baik mengadopsi kemampuan yang dimiliki otak manusia dalam beberapa aspek. Hal ini di buktikan dalam proses pengenalan pola atau pengetahuan yang dapat dikenali seluruhnya. Pada beberapa kasus citra *noise* dan *corrupt,* aplikasi ini tetap dapat dengan baik mengenali data tidak normal tersebut.

DAFTAR PUSTAKA

Fathiyyah, M. Ibrahim al. 2013. *PEGON Rahasia Sukses Belajar Tulisan Pegon.* Kediri: cv Harapan Mandiri.

Kristanto, Andi. 2004. *Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma, dan Aplikasi).* Yogyakarta: Gaya Media.

Munir, Rinaldi. 2004. *Pengolahan Citra Digital Dengan Pendekatan Algoritmik.* Bandung: Informatika.

Puspaningrum, D. 2006. *Pengantar Jaringan Syaraf Tiruan.* Yogyakarta: Andi Offset.

Putra, Darma. 2010. *Pengolahan Citra Digital.* Yogyakarta: Penerbit Andi.

Widodo, TN. 2005. *Sistem Neuro Fuzzy.* Yogyakarta: Graha Ilmu.

UCAPAN TERIMAKASIH

Ucapan terimakasih kami sampaikan kepada:
Nurochman, S.Kom, M.Kom, selaku Dosen Pembimbing dalam penelitian ini yang telah memberikan bimbingan dan pengarahan terhadap pengerjaan penelitian ini.