

## PENERAPAN METODE *LEARNING VECTOR QUANTIZATION* PADA PREDIKSI INTENSITAS CURAH HUJAN DI KOTA PONTIANAK

Ilham Saputra, Setyo Wira Rizki

### INTISARI

Cuaca merupakan suatu kondisi udara di suatu wilayah atau tempat pada waktu yang relatif singkat. Keadaan cuaca yang tidak menentu tersebut merupakan hal yang mendasari perlunya untuk meramalkan cuaca. Peramalan dapat dilakukan untuk mengetahui kapan suatu peristiwa akan terjadi sehingga tindakan yang tepat dapat dilakukan. Salah satunya adalah dengan menggunakan jaringan syaraf tiruan. Jaringan syaraf tiruan adalah salah satu kecerdasan buatan yang menyerupai sistem syaraf dari otak manusia. Proses dengan menggunakan jaringan syaraf tiruan ini bertujuan untuk penerapan metode *learning vector quantization* pada prediksi intensitas curah hujan di Kota Pontianak. Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari BMKG Pontianak. Data tersebut merupakan data curah hujan, kelembapan udara, temperatur maksimum dan temperatur minimum. Proses pelatihan pada metode *learning vector quantization* ini dengan menentukan bobot awal dan parameter yang digunakan. Selanjutnya melakukan pelatihan dari epoch ke-1 pada data ke-1 sampai data ke- $n$  dan diperoleh bobot akhir. Selanjutnya mengurangi nilai alfa:  $\hat{\alpha} = \alpha - Dec\alpha * \alpha$ . Lakukan cara yang sama pada epoch ke-2 sampai max epoch dan memperoleh bobot akhir. Bobot akhir ini akan digunakan untuk melakukan proses pengujian. Hasil pengujian dengan menggunakan jaringan syaraf tiruan menggunakan metode *learning vector quantization* diperoleh untuk Maksimum epoch (MaxEpoch) = 5, learning rate ( $\alpha$ ) = 0,4, Decrease learning rate (Dec $\alpha$ ) = 0,1, dan Minimum learning rate (Min $\alpha$ ) = 0 dengan tingkat akurasi sebesar 90%.

**Kata Kunci:** Jaringan Syaraf Tiruan, Cuaca, Epoch.

### PENDAHULUAN

Cuaca merupakan suatu kondisi udara di suatu wilayah atau tempat pada waktu yang relatif singkat [1]. Keadaan cuaca yang tidak menentu tersebut merupakan hal yang mendasari perlunya untuk meramalkan cuaca. Informasi tentang peramalan cuaca yang cepat dan tepat menjadi kebutuhan utama yang tidak dapat dipisahkan dari aktifitas manusia dan mempengaruhi di berbagai bidang. Bidang-bidang yang dapat mempengaruhi cuaca pada suatu hari seperti transportasi, pertanian atau olah raga sekalipun. Misalkan kapan waktu yang baik untuk keberangkatan pesawat terbang, kapan waktu yang baik untuk bercocok tanam dan bahkan penentuan waktu untuk pertandingan sepak bola sekalipun.

Peramalan dapat dilakukan untuk mengetahui kapan suatu peristiwa akan terjadi sehingga tindakan yang tepat dapat dilakukan. Peramalan memiliki banyak metode yang tersedia dan beragam dalam hal ketepatan. Salah satunya adalah dengan menggunakan metode *learning vector quantization* (LVQ). LVQ merupakan suatu metode untuk melakukan pelatihan terhadap lapisan-lapisan kompetitif yang terawasi. Lapisan kompetitif akan belajar secara otomatis untuk melakukan klasifikasi terhadap vektor *input* yang diberikan. Apabila beberapa vektor *input* memiliki jarak yang sangat berdekatan, maka vektor-vektor tersebut akan dikelompokkan dalam kelas yang sama [2]. Oleh karena itu, tujuan pada penelitian ini adalah penerapan metode LVQ pada prediksi intensitas curah hujan di Kota Pontianak. Data yang digunakan untuk prediksi intensitas curah hujan adalah data curah hujan, kelembapan udara, temperatur maksimum dan temperatur minimum. Data yang digunakan adalah data harian dari tanggal 01 Agustus 2017 sampai dengan tanggal 29 September 2017. Data hasil peramalan terbagi menjadi 5 kelas yaitu kelas 1 hujan sangat ringan, kelas 2 hujan ringan, kelas 3 hujan normal, kelas 4 hujan lebat dan kelas 5 hujan sangat lebat. Hasil prediksi hanya berlaku untuk wilayah Kota Pontianak. Metode yang digunakan adalah *Learning Vector Quantization* (LVQ).

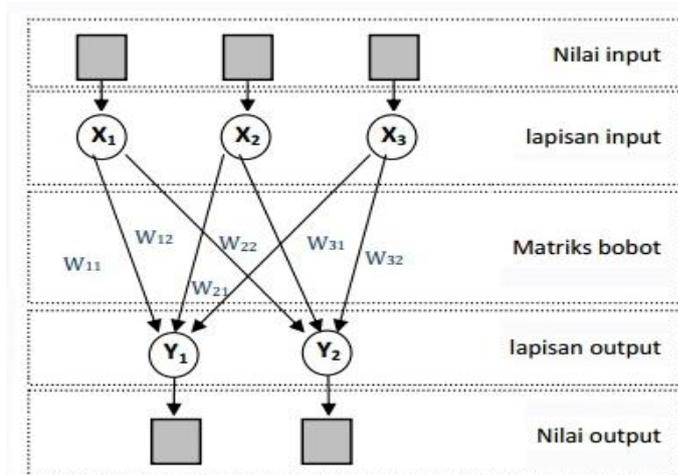
Proses pengerjaan metode LVQ ini terbagi menjadi 2 yaitu proses pelatihan dan proses pengujian. Proses pelatihan ini dilakukan dengan menentukan bobot awal dan parameter yang akan digunakan. Selanjutnya melakukan pelatihan dari *epoch* ke-1 pada data ke-1 sampai data ke-*n* dan diperoleh bobot akhir. Selanjutnya mengurangi nilai alfa:  $\alpha^* = \alpha - \text{Dec}\alpha * \alpha$ . Lakukan cara yang sama pada *epoch* ke-2 sampai max *epoch* dan memperoleh bobot akhir. Setelah memperoleh bobot akhir pada proses pelatihan, bobot akhir ini digunakan untuk melakukan proses pengujian. Hasil pengujian dengan menggunakan jaringan syaraf tiruan menggunakan metode learning vector quantization diperoleh untuk Maksimum *epoch* (MaxEpoch) = 5, learning rate ( $\alpha$ ) = 0,4, Decrease learning rate (Dec $\alpha$ ) = 0,1, dan Minimum learning rate (Min $\alpha$ ) = 0 dengan tingkat akurasi sebesar 90%.

## ARSITEKTUR JARINGAN

Umumnya neuron-neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama. Faktor terpenting dalam menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya. Pada setiap lapisan yang sama, neuron-neuron akan memiliki fungsi aktivasi yang sama. Apabila neuron-neuron dalam suatu lapisan (misalkan lapisan tersembunyi) akan berhubungan dengan neuron-neuron pada lapisan yang lain (misalkan lapisan *output*), maka setiap neuron pada lapisan tersembunyi juga berhubungan dengan setiap neuron pada lapisan *output*. Ada beberapa arsitektur jaringan syaraf, antara lain [3]:

### 1. Jaringan dengan lapisan tunggal (*single layer net*)

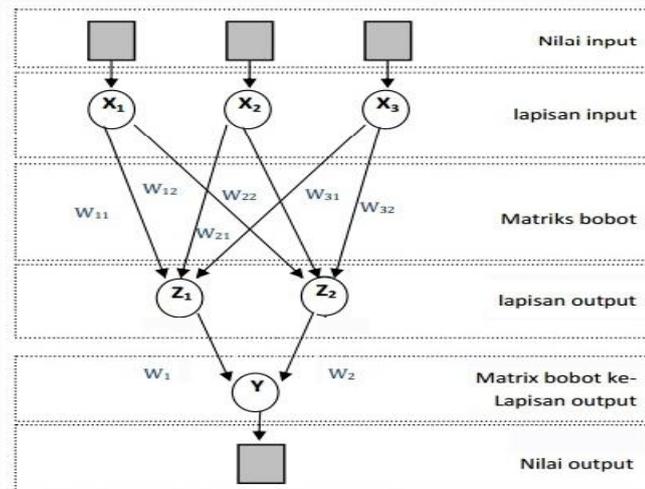
Jaringan dengan lapisan tunggal hanya memiliki satu lapisan dengan bobot-bobot yang berhubungan. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi yang dapat dilihat pada Gambar 1. Pada Gambar 1 tersebut, lapisan *input* memiliki 3 neuron yaitu  $x_1$ ,  $x_2$  dan  $x_3$ . Sedangkan pada lapisan *output* memiliki 2 neuron yaitu  $y_1$  dan  $y_2$ . Neuron-neuron pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 neuron ditentukan oleh bobot yang bersesuaian. Semua unit *input* akan dihubungkan dengan setiap unit *output* [3].



Gambar 1 Jaringan Syaraf dengan Lapisan Tunggal.

### 2. Jaringan dengan banyak lapisan (*multilayer net*)

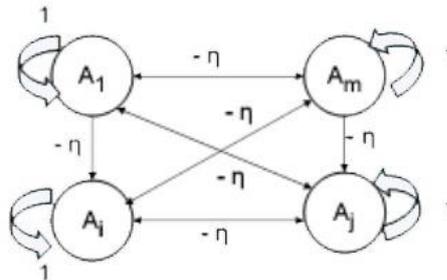
Jaringan dengan banyak lapisan memiliki 1 atau lebih lapisan tersembunyi yang terletak diantara lapisan *input* dan lapisan *output* yang dapat dilihat pada Gambar 2. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit dari pada dengan lapisan tunggal. Tentu saja dengan pembelajaran yang lebih rumit. Namun demikian, pada banyak kasus pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah [3].



Gambar 2 Jaringan Syaraf dengan Banyak Lapisan.

### 3. Jaringan dengan lapisan kompetitif (*competitive layer net*)

Hubungan antara neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot  $-\eta$  yang dapat dilihat pada Gambar 3 [3].



Gambar 3 Jaringan syaraf dengan lapisan kompetitif.

### INISIALISASI ACAK

Prosedur yang umum dilakukan adalah menginisialisasi bias dan bobot. Baik dari unit *input* ke unit tersembunyi maupun dari unit tersembunyi ke unit *output* ke dalam sebuah interval tertentu ( $-\gamma$  dan  $\gamma$ ). Misalkan antara -0,4 sampai 0,4, -0,5 sampai 0,5, dan -1 sampai 1 [4].

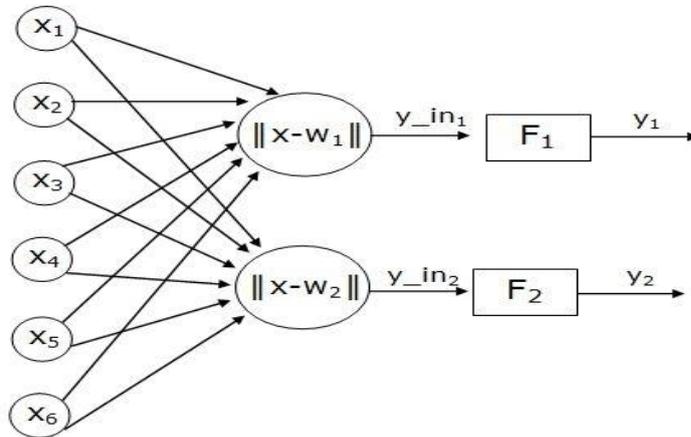
### LAMA ITERASI

Tujuan utama pada jaringan syaraf tiruan adalah mendapatkan keseimbangan antara pengenalan pola data pelatihan secara benar dan respon yang baik untuk pola data pengujian. Jaringan dapat dilatih terus menerus sehingga semua pola data pelatihan dikenali dengan benar. Akan tetapi hal itu tidak menjamin jaringan akan mampu mengenali pola data pengujian dengan tepat. Jadi tidaklah bermanfaat untuk meneruskan iterasi sehingga semua kesalahan pola data pelatihan = 0 [5].

### METODE LEARNING VECTOR QUANTIZATION (LVQ)

Metode *learning vector quantization (LVQ)* merupakan suatu metode untuk melakukan pelatihan terhadap lapisan-lapisan kompetitif yang terawasi. Lapisan kompetitif akan belajar secara otomatis untuk melakukan klasifikasi terhadap vektor *input* yang diberikan. Apabila beberapa vektor *input* memiliki jarak yang sangat berdekatan, maka vektor-vektor tersebut akan dikelompokkan dalam kelas yang sama. Pada Gambar 4 menunjukkan jaringan LVQ dengan 6 unit pada lapisan *input* dan 2 unit pada lapisan *output*. Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak antara

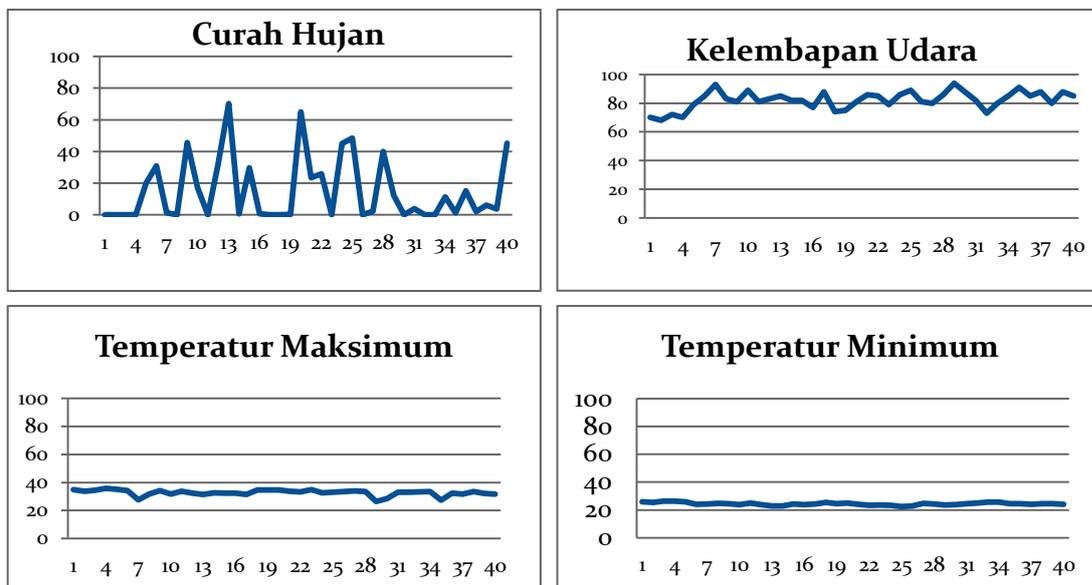
suatu vektor *input* ke bobot yang bersangkutan  $w_1$  dan  $w_2$ .  $w_1$  adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron pertama pada lapisan *output*. Sedangkan  $w_2$  adalah vector bobot yang menghubungkan setiap neuron pada lapisan *input* ke neuron kedua pada lapisan output. Fungsi aktivasi  $F_1$  akan memetakan  $y_{in1}$  ke  $y_1 = 1$  apabila  $\|x - w_1\| < \|x - w_2\|$ . Sedangkan fungsi aktivasi  $F_2$  akan memetakan  $y_{in1}$  ke  $y_1 = 1$  apabila  $\|x - w_2\| < \|x - w_1\|$  [2].



Gambar 4 Arsitektur Jaringan *Learning Vector Quantization* (LVQ).

## STUDI KASUS

Data yang digunakan dalam penelitian ini adalah data sekunder yang diperoleh dari BMKG Pontianak. Data tersebut merupakan data harian curah hujan (mm), kelembapan udara (%), temperatur maksimum ( $^{\circ}\text{C}$ ), dan temperatur minimum ( $^{\circ}\text{C}$ ). Data tersebut dibagi menjadi 2 yaitu data pelatihan dan data pengujian. Data pelatihan dari tanggal 01 Agustus 2017 sampai dengan tanggal 09 September 2017. Sedangkan data pengujian dari tanggal 10 September 2017 sampai dengan tanggal 29 September 2017.



Gambar 5 Data Curah Hujan (mm), Kelembapan Udara (%), Temperatur Maksimum ( $^{\circ}\text{C}$ ) dan Temperatur Minimum ( $^{\circ}\text{C}$ ).

Grafik data pada Gambar 5 menunjukkan data harian curah hujan, kelembapan udara, temperatur maksimum dan temperatur minimum memiliki pola stasioner di mana data tersebut mengalami kenaikan atau penurunan secara konsisten.

**PROSES PELATIHAN DENGAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ)**

Variabel yang terlibat dalam meramalkan cuaca pada penelitian ini meliputi 4 variabel yaitu curah hujan, kelembapan udara, temperatur maksimum dan temperatur minimum, dimisalkan sebagai berikut:

$x_1$  = Curah Hujan (mm)

$x_2$  = Kelembapan Udara (%)

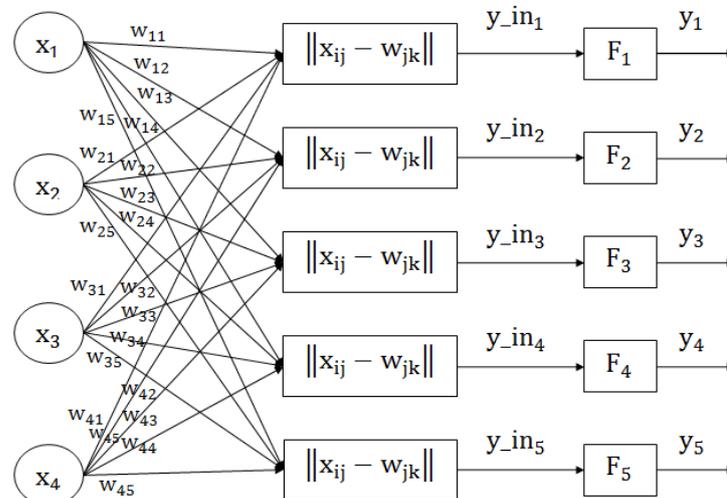
$x_3$  = Temperatur Maksimum (°C)

$x_4$  = Temperatur Minimum (°C)

T = Target dalam bentuk kelas.

Percobaan ke 1:

Langkah 1: Membuat arsitektur jaringan *Learning Vector Quantization* dengan 2 lapisan. Lapisan ke 1 yaitu lapisan *input* yang terdiri dari 4 *neuron* dan lapisan ke 2 yaitu lapisan *output* yang terdiri dari 4 *neuron*. Arsitektur jaringan *Learning Vector Quantization* disajikan pada Gambar 6 berikut:



Gambar 6 Arsitektur Jaringan *Learning Vector Quantization* (LVQ).

Langkah 2: Penentuan bobot awal lapisan *input*  $x_{ij}$  ke lapisan *output* secara acak:

$$\begin{array}{llll}
 w_{11} = 30,54 & w_{21} = 78,43 & w_{31} = 32,62 & w_{41} = 24,68 \\
 w_{12} = 34,89 & w_{22} = 80,87 & w_{32} = 30,99 & w_{42} = 24,41 \\
 w_{13} = 33,18 & w_{23} = 81,82 & w_{33} = 31,65 & w_{43} = 24,29 \\
 w_{14} = 43,70 & w_{24} = 83,39 & w_{34} = 32,12 & w_{44} = 24,04 \\
 w_{15} = 42,30 & w_{25} = 84,49 & w_{35} = 33,55 & w_{45} = 24,15
 \end{array}$$

Langkah 3: Penetapan parameter-parameter yang digunakan dalam pelatihan yaitu sebagai berikut:

Maksimum *epoch* (MaxEpoch) = 5

Parameter *learning rate* ( $\alpha$ ) = 0,4

Pengurangan *learning rate* ( $Dec\alpha$ ) = 0,1

Minimum *learning rate* ( $Min\alpha$ ) = 0

Langkah 4: Melakukan pelatihan dengan *epoch* ke 1:

Data pelatihan ke 1: (0 70 34,8 25,9)

Menghitung jarak data pelatihan ke 1 pada:

bobot kelas ke-1  $\|x_{ij} - w_{j1}\|$

$$= \sqrt{(0 - 30,54)^2 + (70 - 78,43)^2 + (34,8 - 32,62)^2 + (25,9 - 24,68)^2} = 31,780$$

$$\begin{aligned} & \text{bobot kelas ke-2 } \|x_{ij} - w_{j2}\| \\ & = \sqrt{(0 - 34,89)^2 + (70 - 80,87)^2 + (34,8 - 30,99)^2 + (25,9 - 24,41)^2} = 36,772 \\ & \text{bobot kelas ke-3 } \|x_{ij} - w_{j3}\| \\ & = \sqrt{(0 - 33,18)^2 + (70 - 81,82)^2 + (34,8 - 31,65)^2 + (25,9 - 24,29)^2} = 35,400 \\ & \text{bobot kelas ke-4 } \|x_{ij} - w_{j4}\| \\ & = \sqrt{(0 - 43,70)^2 + (70 - 83,39)^2 + (34,8 - 32,12)^2 + (25,9 - 24,04)^2} = 45,822 \\ & \text{bobot kelas ke-5 } \|x_{ij} - w_{j5}\| \\ & = \sqrt{(0 - 42,30)^2 + (70 - 84,49)^2 + (34,8 - 33,55)^2 + (25,9 - 24,15)^2} = 44,765 \end{aligned}$$

Jarak terkecil pada bobot kelas ke 1

Target data pelatihan ke-1 ( $T_i$ ) = 1

Perbaiki bobot kelas ke-1:

$$\begin{aligned} w_{11} &= w_{11} + \alpha * (x_{11} - w_{11}) = 30,54 + 0,4 * (0 - 30,54) = 18,324 \\ w_{21} &= w_{21} + \alpha * (x_{12} - w_{21}) = 78,43 + 0,4 * (70 - 78,43) = 75,058 \\ w_{31} &= w_{31} + \alpha * (x_{13} - w_{31}) = 32,62 + 0,4 * (34,8 - 32,62) = 33,492 \\ w_{41} &= w_{41} + \alpha * (x_{14} - w_{41}) = 24,68 + 0,4 * (25,9 - 24,68) = 25,168 \end{aligned}$$

Jadi bobot kelas ke 1 baru:

$$w_{j1} = (18,324 \ 75,058 \ 33,492 \ 25,168)$$

Proses tersebut diteruskan untuk data pelatihan ke 2 sampai data pelatihan ke 40 di *epoch* ke 1, maka diperoleh bobot akhir di *epoch* ke 1:

$$\begin{array}{cccc} w_{11} = -0,43 & w_{21} = 86,86 & w_{31} = 31,06 & w_{41} = 24,40 \\ w_{12} = 38,14 & w_{22} = 77,70 & w_{32} = 28,68 & w_{42} = 24,92 \\ w_{13} = 33,83 & w_{23} = 84,11 & w_{33} = 32,75 & w_{43} = 23,35 \\ w_{14} = 58,28 & w_{24} = 83,16 & w_{34} = 32,64 & w_{44} = 24,00 \\ w_{15} = 34,56 & w_{25} = 78,25 & w_{35} = 35,78 & w_{45} = 26,38 \end{array}$$

Pengurangan nilai alfa ( $\alpha$ ) :

$$\begin{aligned} \hat{\alpha} &= \alpha - \alpha * \text{Dec } \alpha \\ &= 0,02 - 0,02 * 0,1 = 0,018 \end{aligned}$$

Lakukan proses yang sama untuk *epoch* ke 2 sampai *max epoch* yaitu *epoch* ke 5, untuk setiap data dengan menggunakan cara yang sama. Setelah di *epoch* yang ke 5 maka diperoleh bobot akhir:

$$\begin{array}{cccc} w_{11} = 0,87 & w_{21} = 85,85 & w_{31} = 31,30 & w_{41} = 24,48 \\ w_{12} = 14,49 & w_{22} = 83,43 & w_{32} = 30,84 & w_{42} = 25,46 \\ w_{13} = 42,08 & w_{23} = 85,41 & w_{33} = 32,68 & w_{43} = 23,75 \\ w_{14} = 67,34 & w_{24} = 82,41 & w_{34} = 33,24 & w_{44} = 24,10 \\ w_{15} = 36,90 & w_{25} = 73,61 & w_{35} = 37,71 & w_{45} = 28,42 \end{array}$$

### PROSES PENGUJIAN DENGAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ)

Setelah diperoleh bobot akhir pada proses pelatihan dengan metode LVQ, kemudian kita akan melakukan simulasi terhadap data pengujian. Apakah hasilnya sesuai dengan target yang sebenarnya.

Data pengujian ke 1: (25,6 97 29 24,2)

Menghitung jarak data pengujian ke 1 pada:

Bobot kelas ke 1  $\|x_{ij} - w_{j1}\|$

$$= \sqrt{(25,6 - 0,87)^2 + (97 - 85,85)^2 + (29 - 31,30)^2 + (24,2 - 24,48)^2} = 27,182$$

bobot kelas ke 2  $\|x_{ij} - w_{j2}\|$

$$= \sqrt{(25,6 - 14,49)^2 + (97 - 83,43)^2 + (29 - 30,84)^2 + (24,2 - 25,46)^2} = 17,585$$

bobot kelas ke 3  $\|x_{ij} - w_{j3}\|$

$$= \sqrt{(25,6 - 42,08)^2 + (97 - 83,41)^2 + (29 - 32,68)^2 + (24,2 - 23,75)^2} = 20,417$$

bobot kelas ke 4  $\|x_{ij} - w_{j4}\|$

$$= \sqrt{(25,6 - 67,34)^2 + (97 - 82,41)^2 + (29 - 33,24)^2 + (24,2 - 24,10)^2} = 44,378$$

bobot kelas ke 5  $\|x_{ij} - w_{j5}\|$

$$= \sqrt{(25,6 - 36,90)^2 + (97 - 73,61)^2 + (29 - 37,71)^2 + (24,2 - 28,42)^2} = 27,620$$

Jarak terkecil pada bobot kelas ke-2

Proses tersebut diteruskan untuk data pengujian ke 2 sampai data pengujian ke 20.

**Tabel Hasil Pengujian.**

Data ke-	Data (X)				T <sub>i</sub>	Jarak Data Pengujian ke-					Jarak Terkecil	Masuk Kelas ke-
	x1	x2	x3	x4		1	2	3	4	5		
1	25,6	97	29,0	24,2	3	27,182	17,585	20,417	44,378	27,620	17,585	2*
2	5,7	79	33,1	24,6	2	8,446	10,066	36,924	61,727	32,251	8,446	1*
3	0,1	90	32,0	24,4	1	4,359	15,929	42,250	67,687	40,919	4,359	1
4	0	80	32,9	24,6	1	6,484	15,146	42,484	67,399	37,889	6,484	1
5	0	75	36,1	26,0	1	11,888	17,523	43,514	67,820	37,045	11,888	1
6	0	76	36,2	26,8	1	11,083	17,101	43,323	67,741	37,059	11,083	1
7	10	87	30,8	24,4	2	9,204	5,741	32,174	57,563	31,030	5,741	2
8	8,8	89	30,0	23,4	2	8,775	8,397	33,603	59,023	33,424	8,397	2
9	1	85	33,0	24,0	1	1,983	13,823	41,084	66,388	38,201	1,983	1
10	0,3	83	32,5	24,6	1	3,400	14,329	41,875	67,045	38,267	3,400	1
11	0	73	35,1	25,1	1	13,656	18,487	44,023	68,058	37,151	13,656	1
12	0	71	33,6	25,8	1	15,031	19,242	44,510	68,308	37,307	15,031	1
13	0	72	35,0	26,0	1	14,619	19,039	44,340	68,217	37,122	14,619	1
14	0	76	33,0	23,3	1	9,822	16,442	43,061	67,622	37,646	9,822	1
15	0	75	34,1	23,6	1	11,512	17,303	43,432	67,781	37,406	11,512	1
16	0	74	34,6	25,6	1	12,560	17,793	43,730	67,916	37,139	12,560	1
17	0,4	74	35,0	25,4	1	12,392	17,422	43,290	67,493	36,728	12,392	1
18	0	84	30,6	25,2	1	1,920	14,530	42,167	67,431	39,247	1,920	1
19	18,3	85	32,1	24,1	2	17,501	4,401	23,803	49,102	22,745	4,401	2
20	0	80	34,6	24,6	1	6,425	15,296	42,428	67,384	37,843	6,425	1

Keterangan \* : pada data ke 1 diperoleh jarak terkecil dari data pengujian ke 1 pada bobot kelas ke 2 yaitu sebesar 17,585 yang masuk ke kelas 2 tidak sesuai dengan target aslinya (T<sub>i</sub>) yang seharusnya masuk dalam kelas ke 3.

## PENUTUP

Penulis telah melakukan beberapa kali percobaan dengan mengimplementasikan jaringan syaraf tiruan metode *learning vector quantization* (LVQ) untuk peramalan cuaca diperoleh hasil untuk maksimum *epoch* (MaxEpoch) = 5, *learning rate* ( $\alpha$ ) = 0,4, pengurangan *learning rate* ( $Dec\alpha$ ) = 0,1, dan minimum *learning rate* ( $Min\alpha$ ) = 0 dengan tingkat keakurasi sebesar 90%.

**DAFTAR PUSTAKA**

- [1]. Suchayono, D. S. & Ribudiyanto, K. (2013). Cuaca dan Iklim Ekstrim di Indonesia. Jakarta: Pusat Penelitian dan Pengembangan, BMKG.
- [2]. Kusumadewi, S. (2003). Artificial Intelligence (Teknik dan Aplikasinya). Yogyakarta: Graha Ilmu.
- [3]. Kusumadewi, S. (2004). Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK. Yogyakarta: Graha Ilmu.
- [4]. Puspitaningrum, D. (2006). Pengantar Jaringan Syaraf Tiruan. Yogyakarta: ANDI OFFSET.
- [5]. Siang, J. J. (2005). Jaringan Syaraf Tiruan dan Pemrograman Menggunakan Matlab. Yogyakarta: ANDI OFFSET.

ILHAM SAPUTRA : Jurusan Matematika FMIPA Untan, Pontianak,  
ilham.saputra.mtk12@gmail.com  
SATYO WIRA RIZKI : Jurusan Matematika FMIPA Untan, Pontianak,  
setyo.wirarizki@math.untan.ac.id

---