

ANALISIS ALGORITMA PREDIKSI CURAH HUJAN MENGGUNAKAN ALGORITMA COMPETITIVE NEURAL NETWORK

Muhammad Feridiansyah¹, Fhira Nhita S.T.,M.T², Dr.Adiwijaya, M.Si³

^{1,2,3}Prodi S1 Ilmu Komputasi, Fakultas Teknik Informatika, Universitas Telkom

¹mferidiansyah@gmail.com, ²farid.alchair@gmail.com, ³kang.adiwijaya@gmail.com

Abstrak

Curah hujan merupakan faktor penting dalam perubahan cuaca. Dengan iklim tropis yang dimiliki Indonesia. Penguapan air ke udara sangat besar, akibatnya intensitas curah hujan sering tidak stabil, oleh karena itu diperlukan suatu sistem untuk prediksi curah hujan. *Competitive Neural Network* merupakan bagian arsitektur *Artificial neural network* yang di mana terdiri dari lapisan *input* dan lapisan kompetitif. Hasil arsitektur yang di dapat dengan menggunakan beberapa skenario adalah berupa bobot dan akurasi yang dapat digunakan untuk memprediksi. Pada Tugas akhir ini algoritma *Competitive Neural Network* digunakan untuk memprediksi curah hujan bulanan wilayah Soreang, dengan parameter bobot. Hasil dari proses tersebut dikelompokkan menjadi beberapa *cluster* yang membentuk sebuah pola pengenalan untuk memprediksi curah hujan. Diharapkan dari hasil penelitian ini didapatkan hasil pengelompokkan data yang sesuai dengan kelompoknya masing-masing.

Bobot optimal dari *learning Competitive Neural Network* berada pada skenario dengan jumlah iterasi 500, learning rate 0.02 dan range bobot awal [-0.5 ; 0.5] dengan akurasi rata-rata *training* dan *testing* sebesar 82.915 %.

Kata kunci : *winner-takes-all, Competitive Neural Network, competitive layer.*

1. Pendahuluan

Curah hujan merupakan faktor yang berpengaruh langsung terhadap perubahan cuaca yang semakin memburuk. Curah hujan merupakan total air hujan yang terjatuh pada permukaan tanah dalam waktu tertentu yang diukur menggunakan satuan tinggi dalam milimeter (mm) pada permukaan tanah datar. Yang di mana air hujan tersebut tidak boleh menguap, meresap, maupun mengalir yang bisa menyebabkan perbedaan ketinggian air hujan ketika diukur. Menghindari perubahan cuaca yang tidak menentu dibutuhkan analisis dan prediksi yang lengkap dan akurat terhadap data-data curah hujan agar nantinya data yang telah dihasilkan berguna bagi bidang yang membutuhkan dan memperkecil risiko terkena dampak perubahan cuaca yang tidak menentu.

Dari penjelasan diatas, maka diperlukan suatu sistem untuk memprediksi curah hujan. Untuk menganalisis dan memprediksi data curah hujan penulis menggunakan arsitektur Jaringan Syaraf Tiruan yang merupakan implementasi dari fungsi sistem saraf manusia dalam pengenalan pola yang sangat akurat. Arsitektur dalam Jaringan Syaraf Tiruan sendiri terbagi dalam beberapa arsitektur, salah satunya adalah *Competitive Neural Network*.

Tujuan dari Tugas akhir ini adalah :

1. Mengimplementasikan arsitektur *Competitive Neural Network* untuk prediksi curah hujan.
2. Menemukan parameter bobot dan learning rate yang optimal untuk algoritma *Competitive Neural Network*.
3. Mengetahui hasil akurasi yang didapatkan dari algoritma *Competitive Neural Network*.

Adapun Perumusan masalah dalam Tugas akhir ini antara lain :

1. Bagaimana mengimplementasikan arsitektur Jaringan Syaraf Tiruan *Competitive Neural Network* untuk memprediksi curah hujan di wilayah Soreang Kabupaten Bandung.
2. Bagaimana menemukan parameter bobot dan learning rate yang optimal untuk algoritma *Competitive Neural Network*.
3. Bagaimana akurasi dari *Competitive Neural Network* dalam memprediksi curah hujan di wilayah Soreang Kabupaten Bandung.

Dalam Tugas akhir ini untuk menyelesaikan masalah diatas, penulis menggunakan Enam metode antara lain :

1. Studi Literatur
Tahap pengumpulan referensi yang sesuai topik Tugas akhir dari jurnal, *paper*, internet maupun buku untuk memperkaya wawasan dan teori-teori yang dibutuhkan dalam penyusunan Tugas akhir ini.
2. Pengumpulan Data
Menggunakan data curah hujan bulanan BMKG dari tahun 2005-2014.

3. Analisis dan Perancangan Sistem
Menganalisis data dan sistem yang akan digunakan untuk penyelesaian masalah Topik Tugas akhir.
4. Implementasi Sistem
Pengimplementasian arsitektur *Competitive Neural Network* untuk data-data yang sudah diperoleh yang ke dalam rancangan sistem yang telah dibuat.
5. Pengujian dan Analisis Implementasi
Melakukan pengujian dan analisis terhadap Arsitektur *Competitive Neural Network* untuk mendapatkan tingkat akurasi yang tinggi. Kemudian menganalisis hasil yang didapat dan diprediksi agar bisa digunakan dalam kehidupan nyata.
6. Penyusunan Laporan
Tahap terakhir yang dilakukan dalam Metodologi Penelitian. Melakukan perangkuman terhadap apa yang sudah dilakukan ke dalam satu laporan tertulis.

2. Dasar Teori /Material dan Metodologi/perancangan

2.1 Competitive Neural Network

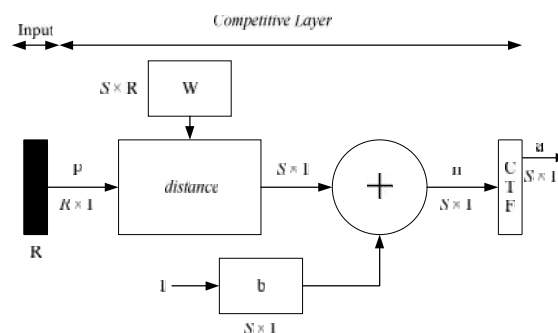
Competitive Neural Network terdiri dari dua lapisan *neuron* yaitu lapisan *distance measure* dan lapisan kompetitif. Yang dimana setiap *output neuron* langsung terhubung penuh terhadap tiap tiap simpul *input*. *Output neuron* bersaing untuk menjadi pemenang ketika ciri *input* di tampilan ke jaringan. Aktivasi dari *neuron* dengan *input* terbesar di set menjadi 1 dan aktivasi dari *neuron* yang lain di set menjadi 0. Kondisi ini dinamakan "winner takes all". Jika *neuron* memenangi kompetisi, bobot vektornya diperbarui sesuai dengan *competitive learning rule*.

Lapisan *distance measure* terdiri dari vektor m yang beris bobot $W = \{w_1, w_2, \dots, w_m\}$. Ketika contoh data uji ditampilkan, lapisan *distance measure* menghitung jarak antara vektor bobot. Jarak yang telah dihitung berada di lapisan *distance measure* menjadi *input* dari lapisan kompetitif. Lapisan kompetitif mencari vektor bobot yang paling dekat atau minimum dengan contoh data uji. *Output* dari lapisan kompetitif adalah vektor $1 \times m$. Setiap bit dari vektor *output* di beri nilai 0 atau 1. Pemenang vektor bobot w_j kemudian dijadikan vektor yang paling dekat dengan sampel data uji. Setiap kali menjadi pemenang, vektor bobot menjadi *particular sample*. Untuk vektor bobot yang memenangi kompetisi akan diperbarui. Proses ini diulangi untuk semua sampel data uji untuk beberapa proses tahapan. Akhirnya masing-masing vektor bobot akan konvergen ke *centroid cluster* [3].

Berikut persamaan *competitive learning rule* yang akan digunakan untuk memperbarui vektor bobot [2].

$$w_j(n+1) = w_j(n) + \xi(n)(p_i - w_j) \quad (2.1)$$

Di mana ξ adalah *learning rate* dan w_j adalah vektor bobot dari *neuron* pemenang j untuk ciri *input* vektor p_i . Jaringan menginisialisasi dengan vektor bobot yang diperoleh dari *neuron* acak. *Neuron* belajar dengan memindahkan bobot mereka ke vektor *input* ketika ditampilkan ke jaringan. Vektor ciri di dalam data set ditampilkan secara acak ke jaringan beberapa kali yang nantinya akan disebut *epochs*. Setelah pelatihan, setiap *neuron output* harus merepresentasikan *cluster* dalam *input* data yang ditetapkan dengan memindahkan vektor bobot sendiri ke pusat cluster itu. Namun, kinerja CNN tergantung pada jumlah *neuron output* dan inisialisasi vektor bobot. perbedaan inisialisasi awal vektor bobot dapat menyebabkan hasil *cluster* yang berbeda karena *rule* yang diperbaharui di persamaan (2.1) di mana hanya menggerakkan vektor bobot yang menjadi pemenang menuju vektor Ciri yang terdekat. Berikut arsitektur dari *Competitive Neural Network* [3].



Gambar2-1 Arsitektur Competitive Neural Network

Vektor *input* p dan *input* bobot matriks W dimasukkan ke kotak *distance*. Dan Jumlah vektor *input* adalah R , sedangkan jumlah *neuron* dari CNN adalah S . Output dari kotak *distance* adalah vektor yang memiliki unsur S . unsur S berisi unsur negatif jarak *euclidean* antara vektor *input* p dan vektor yang terbentuk dari baris *input* bobot

matriks W . Untuk menghitung jaringan input n dari layer competitive adalah dengan menemukan jarak negatif antara vektor *input* p dan vektor bobot dan dengan menambahkan bias b . Ketika semua bias adalah 0, *input* maksimum jaringan untuk semua *neuron* adalah 0 [3].

kotak *competitive transfer function* (CTF) menerima vektor *input* jaringan sebagai *input* dari lapisan dan menghasilkan *neuron output* 0 untuk semua *neuron* kecuali pemenang yang mempunyai *output* 1. *Neuron* pemenang adalah yang mempunyai unsur paling positif dari jaringan vektor *input*. Jika semua bias adalah 0, maka *neuron* vektor bobot merupakan yang mempunyai elemen paling dekat dengan vektor *input*. *Neuron* dengan jaringan *input* nilai paling negatif, memenangkan kompetisi dan *output*nya menjadi 1 [3].

Berikut *pseudo code* algoritma *Competitive Neural Network*.

```

Start
Inisialisasi random bobot awal ;
Inisialisasi input;
Inisialisasi output;
While i <= iterasi
    For k = 1:data
        Input – bobot;
        Minimum Euclidean distance ;
        Update bobot;
        Bobot lama =bobot baru;
        Update learning rate;
    End
    Iterasi = iterasi + 1;
End
End
End

```

2.2 Centered Moving Average

Centered Moving Average yang di mana perbedaannya adalah pada *Moving average* biasa membutuhkan nilai ganjil pada observasinya dan letak pada pemilihan observasi yang digunakan berbeda. Sedangkan *Centered Moving Average* menggunakan nilai genap pada tiap observasinya, dan menggunakan rata-rata antara data sekarang yang didapat dari data sebelumnya dan data sesudahnya. Sebagai contoh kasus CMA 4 Periode. Hasil dari pusat *Centered Moving Average* pertama terletak pada titik 2.5 (berada setelah data ke 2 dan sebelum data ke 3) sedangkan yang kedua terletak pada titik 3.5 (berada setelah data ke 3 dan sebelum data ke 4). Akan tetapi seharusnya rata-rata dari dua *moving average* berpusat di titik 3 agar simetris. Masalah ini dapat diselesaikan dengan menambahkan 2 periode *moving average* dari 4 periode *centered moving average*, sehingga dapat dinotasikan menjadi 2x4 MA. Dapat dinotasikan dengan rumus sebagai berikut [16] :

$$T_{2.5} = \frac{(Y_1 + Y_2 + Y_3 + Y_4)}{4} \quad (2.3)$$

$$T_{3.5} = \frac{(Y_2 + Y_3 + Y_4 + Y_5)}{4} \quad (2.4)$$

Sehingga rata-rata dari dua 4 MA atau 2x4 MA adalah :

$$T_3 = \frac{T_{2.5} + T_{3.5}}{2} \quad (2.5)$$

$$= \frac{1}{2} \left(\frac{Y_1 + Y_2 + Y_3 + Y_4}{4} + \frac{Y_2 + Y_3 + Y_4 + Y_5}{4} \right) \quad (2.6)$$

$$= \frac{(Y_1 + 2Y_2 + 2Y_3 + 2Y_4 + Y_5)}{8} \quad (2.7)$$

2.3 Ukuran Akurasi

Confusion matrix adalah sebuah tabel yang menyatakan jumlah data uji yang benar diklasifikasikan dan jumlah data uji yang salah diklasifikasikan, tabel ini diperlukan untuk menentukan kinerja atau akurasi suatu model klasifikasi. *Confusion matrix* ini menghasilkan *recall*, *precision*, *accuracy*, dan *error rate*. Pada tugas akhir ini akan digunakan *accuracy*.

Nilai *Accuracy* merupakan persentase jumlah record data yang diklasifikasikan secara benar oleh sebuah algoritma dapat membuat klasifikasi setelah dilakukan pengujian pada hasil klasifikasi tersebut [12]. Kebanyakan

algoritma mencari model yang mencapai akurasi paling tinggi, atau secara ekuivalen error yang paling rendah ketika diaplikasikan ke dataset test [12].

Tabel 2-1 Tabel Confusion Matrix

Confusion Matrix		Kelas Prediksi	
		1	0
Kelas Sebenarnya	1	TP	FN
	0	FP	TN

Keterangan dari tabel diatas

- *True Positive* : Jumlah data dari kelas 1 yang benar, dan diklasifikasikan sebagai kelas 1
- *True Negative* : Jumlah prediksi dari kelas 0 yang benar, dan diklasifikasikan sebagai kelas 0.
- *False Negative* : Jumlah data dari kelas 1 yang benar, dan diklasifikasikan sebagai 0.
- *False Positive* : Jumlah data dari kelas 0 yang benar dan diklasifikasikan sebagai 1

Perhitungan akurasi dinyatakan dalam persamaan berikut [14].

$$Akurasi = \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \quad (2.8)$$

3. Pembahasan

3.1 Pengujian sistem

Pada bagian ini akan dipaparkan tentang pengujian, strategi pengujian, penjelasan data dan skenario pengujian

3.1.1 Tujuan pengujian

Tujuan dari pengujian sistem pada tugas akhir ini adalah mendapatkan bobot terbaik dengan akurasi yang baik dari learning algoritma *Competitive Neural Network*.

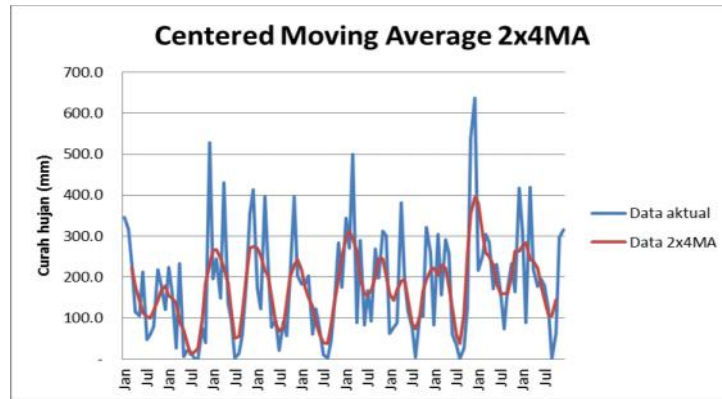
Dalam tugas akhir ini ada beberapa hal yang mempengaruhi sistem pada tugas akhir ini antara lain penentuan *learning rate*, dan pemilihan batas *range* untuk bobot awal yang ditentukan sehingga dapat digunakan memprediksi curah hujan di Soreang kabupaten bandung untuk 1 bulan ke depan.

3.1.2 Strategi pengujian sistem

- Menentukan data yang digunakan dalam sistem prediksi, data yang digunakan pada sistem tugas akhir ini adalah curah hujan bulanan di Kabupaten Soreang Bandung
- Melakukan tahap *preprocessing* data dengan salah satu metode peramalan, yaitu *Moving Average* dengan menggunakan 2x4 CMA.
- Melakukan pembagian data sesuai *cluster* yang sudah ditentukan batasan-batasannya oleh BMKG.
- Percobaan pencarian batas nilai *random* awal untuk bobot.
- Menentukan learning gain yang optimal untuk pelatihan CNN.
- Mendapatkan bobot CNN terbaik yang akan digunakan untuk pengujian data testing.

3.1.3 Data Curah Hujan Kabupaten Soreang Bandung

Pada data curah hujan Kabupaten Soreang Bandung yang penulis dapatkan dari BMKG dilakukan *Centered Moving Average* 2x4 MA dengan 5 periode Didapatkan hasil bahwa metode *Centered Moving Average* 2x4 MA dengan 5 periode memiliki data yang lebih *smooth* daripada data aktual. Kemudian data tersebut dilakukan pembagian data sesuai *cluster* yang sudah ditentukan batasan-batasannya oleh BMKG. Selanjutnya data akan dibagi menjadi *training* dan *testing* di dalam sistem



Gambar 3-1 Perbandingan data aktual dengan data Centered Moving Average 2x4MA

3.1.4 Skenario pengujian

Pada proses pengujian akan dilakukan beberapa observasi terhadap parameter parameter. Diantaranya adalah terhadap besar *learning rate*, dan percobaan batas nilai *random awal* bobot.

Tabel 4-1 Skenario Pengujian

Jumlah Iterasi	Range Bobot Awal	Learning Rate
500	[Xmin ; Xmax]	0.01
		0.02
		0.03
		0.04
		0.05
500	[-0.5 ; 0.5]	0.01
		0.02
		0.03
		0.04
		0.05

Yang dimaksud dengan *Range* bobot pada skenario pertama [Xmin ; Xmax] yaitu Xmin dan Xmax adalah data minimum dan maximum [2] setelah dilakukan metode *Centered Moving Average 2x4 MA*. Untuk skenario kedua digunakan *Range* bobot [-0.5 ; 0.5] dengan nilai paling minimum -0.5 dan maximum 0.5 [6]. Dengan observasi diatas akan diperoleh bobot yang akan dipilih menjadi calon bobot optimal dalam pelatihan berdasarkan akurasi yang paling tinggi. Setelah mendapat bobot yang optimal, bobot tersebut akan digunakan untuk memprediksi curah hujan 1 bulan kedepannya.

3.2 Analisis Hasil Pengujian

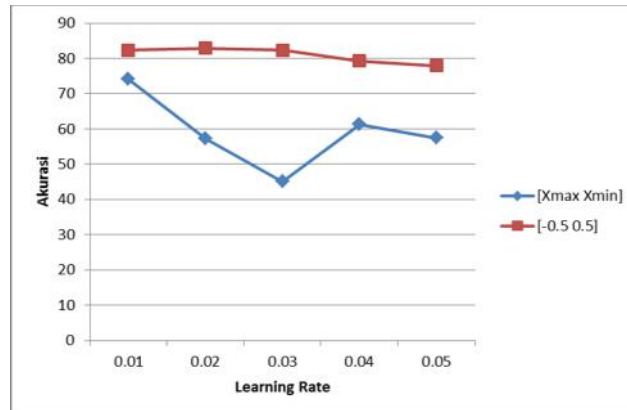
Berdasarkan skenario yang telah ditentukan diatas dengan metode *Moving Average 2x4 MA* berupa 5 interval curah hujan bulanan yang akan menghasilkan *output* berupa bobot dan akurasi. Pada tiap-tiap skenario akan dijalankan dengan menggunakan data *training* dan *testing* yang sudah ditentukan. Berikut ini merupakan hasil dari keseluruhan skenario yang sudah ditentukan.

Tabel 4-2 Hasil uji skenario *learning rate* dan *range bobot awal*

Iterasi	Learning Rate	Range Bobot Awal	Akurasi Maximum (%)		Akurasi Rata-rata (%)
			Training	Testing	
500	0.01	[Xmin ; Xmax]	68.831 %	79.31 %	74.070 %
		[-0.5 ; 0.5]	71.429 %	93.103 %	82.266 %
	0.02	[Xmin ; Xmax]	45.454 %	68.965 %	57.21 %
		[-0.5 ; 0.5]	72.727 %	93.103 %	82.915 %
	0.03	[Xmin ; Xmax]	35.065 %	55.172 %	45.118 %
		[-0.5 ; 0.5]	71.429 %	93.103 %	82.266 %
	0.04	[Xmin ; Xmax]	46.753 %	75.862 %	61.307 %
		[-0.5 ; 0.5]	68.831 %	89.655 %	79.243 %
	0.05	[Xmin ; Xmax]	38.961 %	75.862 %	57.411 %
		[-0.5 ; 0.5]	66.234 %	89.655 %	77.945 %

Berdasarkan Tabel 4-2, skenario terbaik untuk *range bobot* [Xmin ; Xmax] berada pada *Learning rate* 0.01 dengan akurasi 74.070. Sedangkan skenario terbaik untuk *range bobot* [-0.5 ; 0.5] berada pada *Learning rate* 0.02

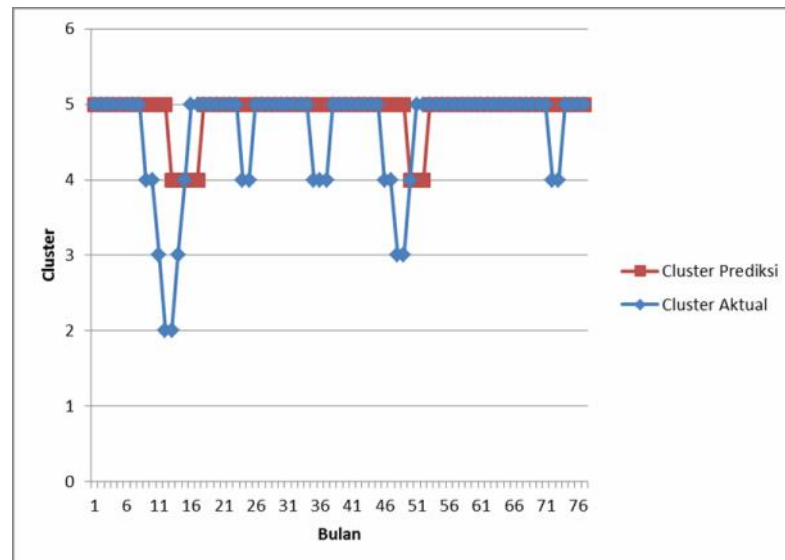
dengan akurasi 82.915, sekaligus menjadikan skenario dengan *range* bobot $[-0.5 ; 0.5]$ menjadi yang terbaik diantar skenario yang lain.



Gambar 4-2 Perbandingan akurasi range bobot awal

Berdasarkan Gambar 4-2, *range* bobot yang minimum memberikan akurasi yang baik dibandingkan dengan *range* bobot $[Xmin ; Xmax]$. Karena dengan *range* awal bobot minimum rentang bobot awal menjadi kecil dan ruang solusi tidak terlalu luas dan tidak mudah menyimpang dari titik pusat *cluster*.

Kemudian, didapatkan akurasi terbaik pada *learning rate* 0.02. Setiap *learning rate* menghasilkan akurasi yang berbeda-beda setelah dilakukan percobaan. Karena *learning rate* merupakan parameter kecepatan perubahan pada tiap update bobot.



Gambar 4-3 Cluster aktual dengan prediksi (Training)

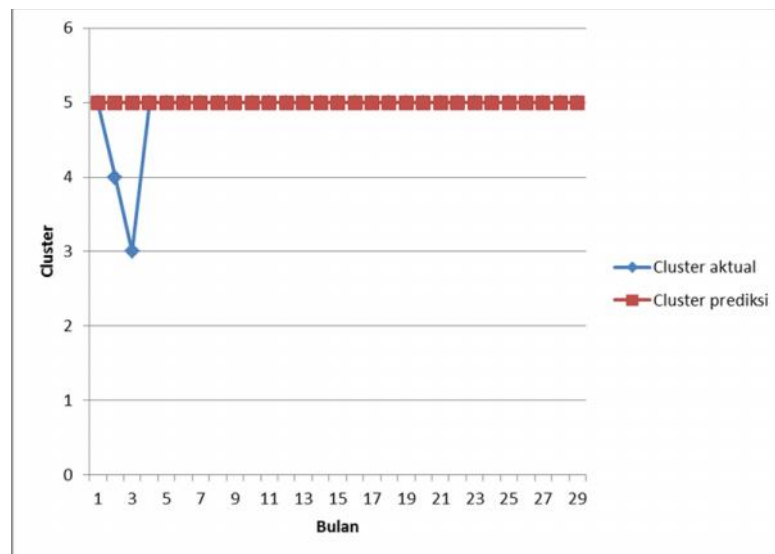
Berdasarkan hasil pengelompokkan pada gambar 4-3, *training* algoritma *Competitive Neural Network* lebih cenderung berada pada cluster 5 karena pada data training juga cenderung berada pada cluster 5. Didapatkan juga bobot terbaik dari *training* algoritma *Competitive Neural Network*, bobot yang terbaik akan digunakan sebagai bobot *testing* untuk prediksi curah hujan

Tabel 4-3 Bobot terbaik

0.056	-0.128	0.106	0.102	-0.446
0.150	-0.048	0.151	-0.065	0.189
-0.431	-0.401	-0.038	0.019	-0.175
18.647	11.856	8.786	9.736	16.571
167.094	167.614	167.959	167.635	165.597

Berdasarkan skenario *learning rate* 0.02 dan *range* bobot awal [-0.5 ; 0.5] dengan akurasi terbaik maka didapatkan bobot terbaik dari pelatihan algoritma *Competitive Neural Network*. Kemudian bobot tersebut digunakan untuk prediksi untuk 1 bulan kedepannya, berikut hasil prediksi menggunakan data *testing* yang sudah ditentukan sebelumnya.

Sesuai dengan Gambar 4-4, dengan bobot yang didapatkan pada *learning* algoritma *Competitive Neural Network*, dapat dihasilkan prediksi dengan data testing pada gambar diatas berada tetap di *cluster* 5 yang merupakan hasil prediksi pengelompokkan curah hujan dengan metode *Centered Moving Average 2x4 MA*.



Gambar 4-4 hasil prediksi data testing

Sesuai dengan Gambar 4-4, dengan bobot yang didapatkan pada *learning* algoritma *Competitive Neural Network*, dapat dihasilkan prediksi dengan data testing pada gambar diatas berada tetap di *cluster* 5 yang merupakan hasil prediksi pengelompokkan curah hujan dengan metode *Centered Moving Average 2x4 MA*.

4. Kesimpulan

Berdasarkan hasil eksperimen yang telah dilakukan, maka dapat disimpulkan bahwa :

- Algoritma *Competitive Neural Network* dan Metode *Centered Moving Average 2x4 MA* dapat digunakan untuk mengembangkan sistem prediksi curah hujan bulanan.
- Penentuan Parameter yang tepat akan menjadi faktor penting yang mempengaruhi hasil prediksi. Parameter parameter yang dimaksud antara lain: *Range* bobot awal, dan *learning rate*.
- Algoritma *Competitive Neural Network* menghasilkan akurasi 82.915% pada *range* bobot awal [-0.5 ; 0.5] dan *learning rate* 0.02 dengan menggunakan metode *Centered Moving Average 2x4 MA*.
- *Learning rate* yang kecil belum tentu menghasilkan akurasi yang baik, maka dari itu dilakukan observasi 0.01 – 0.05 terhadap *learning rate* agar menghasilkan akurasi yang baik. Perbandingan hasil akurasi *range* bobot awal [Xmin ; Xmax] dengan [-0.5 ; 0.5] dapat disimpulkan bahwa *range* bobot yang minimum menjadi yang terbaik berdasarkan akurasi yang dihasilkan oleh masing-masing range bobot awal.
- Hasil prediksi menggunakan algoritma *Competitive Neural Network* dapat memberikan performansi diatas 75%.

Daftar Pustaka:

- [1]. Abas Ahmed R. 2014. "Adaptive competitive learning neural networks". Egyptian Informatics Journal, Cairo University.
- [2]. Abbas Syed Rahad, Arif Muhammad. 2005. "Competitive Neural Network Based Algorithm for Long Range Time Series Forecasting Study : Electric Load Forecasting". 9th International Multitopic Conference, IEEE INMIC 2005, Indigenous Phd Program of Higher Education Comission (HEC) of Pakistan.
- [3]. Andriani Anik. 2013. "SISTEM PENDUKUNG KEPUTUSAN BERBASIS DECISION TREE DALAM PEMBERIAN BEASISWA STUDI KASUS : AMIK "BSI YOGYAKARTA". Seminar Nasional Teknologi informasi dan Komunikasi 2013 (SENTIKA 2013), Yogyakarta.
- [4]. Du K.-L. 2010. "Clustering : A neural network approach". Department of Electrical and Computer Engineering, Concordia University, 1455 de Maisonneuve West, Montreal, Canada, H3G 1M8. Volume 23, Issue 1.
- [5]. Indriani Aida. 2014. "Klasifikasi Data Forum dengan menggunakan metode Naive Bayes Classifier", Seminar asional Aplikasi Teknologi Informasi (SNATI). Yogyakarta.
- [6]. Jones M. Tim. 2008. "Artificial Intelligence Asystem Approach". Infinity Science Press LLC. Massachusetts.
- [7]. Lei John Zhong, Ghorbani Ali A. 2012. "Improved competitive learning neural networks for network intrusion and fraud detection". International Conference on Hybrid Artificial Intelligence Systems (HAIS 2010). Volume 75, Issue 1.
- [8]. Lestari Wiji, Purnomo Singgih. 2013. "Aplikasi Algoritma Competitive Network untuk Clustering Minat Mahasiswa Terhadap Topik-Topik Penelitian". Laporan Tugas akhir. STMIK Duta Bangsa, Surakarta.
- [9]. Nhita Fhira, Adiwijaya, & U.N Wisesty, "A Forecasting of Weather Data in Indonesia Using Evolving Neural Network on Genetic Algorithm". The Second International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE 2014), pp.78-82, *March* 2014.
- [10]. Nurcahyo Septian, Adiwijaya, Nhita Fhira. 2014. "Rainfall Prediction in Kemayoran Jakarta Using Hybrid Genetic Algorithm (GA) and Partially Connected Feedforward Neural Network (PCFNN)". The Second International Conference on Information and Communication Technology (ICOICT 2014).
- [11]. Park Dong-Chul. 2000. "Centroid Neural Network for Unsupervised Competitive learning". IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 11, NO. 2.
- [12]. Rokach lior & Oded Maimon. "CLUSTERING METHODS". Tel-Aviv University.
- [13]. Spyros Makridakis, Steven C. Wheelwright dan Rob J. Hyndman (Wiley, 1998, 3rd edition). "Forecasting : Methods and Applications".
- [14]. Suyanto. 2008. "Soft computing". Informatika, Bandung.
- [15]. Yegnanarayana B. 2005. "Artificial Neural Networks". Departemt of Computer Science and Engineering Indian Institute of Technology Madras Chennai.
- [16]. "Modul Matakuliah Semester 4 daming", [Online]. Available : biomaterial.lipi.go.id [Diakses 1 agustus 2015].