JOMTA
Journal of Mathematics: Theory and Applications
Vol. 4, No. 1, 2022, P-ISSN 2685-9653 e-ISSN 2722-2705

K-*Means Clustering* dan *Principal Component Analysis* (PCA) Dalam *Radial Basis Function Neural Network* (RBFNN) Untuk Klasifikasi Data Multivariat

Hayqal Hazmi Qastari[1], Oni Soesanto[2], Yuana Sukmawaty[3]
[1,2,3] Program Studi Statistika, Universitas Lambung Mangkurat, Indonesia
e-mail: [1]adehayqal@gmail.com

Abstrak. Pada penelitian ini dilakukan uji simulasi data berskala besar sehingga diperlukan metode yang handal untuk permasalahan klasifikasi salah satunya adalah *Radial Basis Function Neural Network* (RBFNN). Untuk *training* data RBFNN menggunakan struktur khusus yang melibatkan dimensi tinggi pada *hidden layer*. Dengan struktur RBFNN yang khusus tersebut maka seringkali menimbulkan permasalahan karena *hidden layer*nya terlalu besar, sehingga diperlukan penambahan metode penyederhaan jaringan seperti PCA dan K-*Means Clustering*. Metode PCA digunakan untuk mereduksi dimensi *input* pada RBFNN sedangkan metode K-*Means Clustering* digunakan untuk penentuan inisialisasi *center* awal pada RBFNN. Pada hasil percobaan metode PCA dihasilkan komponen utama ke-1 dan ke-2 dengan masing-masing mewakili 55.2288% dan 27.3108% dari seluruh variabilitas, secara kumulatif kedua komponen utama menyatakan sebesar 82.5396% dan hasil percobaan perulangan iterasi di metode penelitian ini didapatkan hasil rata-rata proses akurasi PC dan Klas terbaik berada pada PC-2 Klas-3 dengan akurasi di atas 90% untuk proses *training* dan *testing* dengan akurasi kesalahan klasifikasi di bawah 10%.

Kata kunci: PCA-RBFN, K-*Means Clustering*, Data Multivariat

Abstract. In this study, a large-scale data simulation test was carried out so that a reliable method was needed for classification problems, one of which was the Radial Basis Function Neural Network (RBFNN). For training data, RBFNN uses a special structure that involves high dimensions in the hidden layer. This special RBFNN structure often causes problems because the hidden layer is too large, so it is necessary to add network simplification methods such as PCA and K-Means Clustering. The PCA method is used to reduce the input dimensions in RBFNN, while the K-Means Clustering method is used to determine the initial center initialization in RBFNN. The results of the PCA method experiment resulted in the 1st and 2nd principal components representing 55.2288% and 27.3108% of all variability cumulatively, the two principal components stated 82,5396%, and the results of iteration experiments in this research method obtained average results. The best average PC and Class accuracy processes are on PC-2 Class-3 with an accuracy above 90% for the training and testing process with an error classification accuracy below 10%.

Keywords*:* PCA-RBFN, K-*Means* Clustering, Multivariate Data

I. INTRODUCTION

Jaringan Syaraf Tiruan (*Artificial Neural Network*) merupakan salah satu metode dalam kecerdasan buatan yang berusaha meniru kecerdasan manusia dengan menerapkannya pada mesin atau komputer. Jaringan syaraf tiruan dapat berkerja selayaknya seperti manusia dalam tugas-tugas tertentu [4]. Beberapa model-model Jaringan Syaraf Tiruan telah banyak digunakan untuk aplikasi di klasifikasi dan prediksi, salah satu Jaringan Syaraf Tiruan yang banyak digunakan adalah *Radial Basis Function Neural Network* (RBFNN). RBFNN merupakan jaringan yang saat ini cukup popular digunakan karena struktur jaringan yang sederhana. *Learning system* yang sangat cepat serta kehandalannya dalam hal mengklasifikasi data untuk beberapa kasus yang dihadapi [10]. Namun pada kenyataannya pada tahap proses simulasi *data training* di metode ini seringkali menimbulkan permasalahan manakala dimensi variabel *input*nya terlalu besar tentu akan mempengaruhi kecepatan kinerja komputasinya sehingga diperlukan metode tambahan yang bertujuan untuk mereduksi dimensi variabel *input*nya, beberapa metode yang banyak digunakan adalah salah satunya *Principal Component Analysis* (PCA). Kendala berikutnya secara umum untuk pemilihan inisialisasi *center* awal pada proses metode RBFNN dilakukan secara *random* (acak). Tentu, dengan cara ini memiliki kelemahan manakala dimensi datanya yang sangat besar. Dalam hal ini, akan

membutuhkan proses kinerja komputasi yang sangat lama. Sehingga perlu metode optimasi tambahan salah satunya yaitu metode K-*Means Clustering*. Disini K-*Means Clustering* itu berfungsi untuk penentuan inisialisasi *center* awal secara optimal pada *Radial Basis Function Neural Network* (RBFNN) [14].

II. PRELIMINARIES

2.1 Jaringan Syaraf Tiruan pada Data Mining

Data *mining* adalah kegiatan menemukan pola yang menarik dari data dalam jumlah besar, data dapat disimpan dalam *database*, data *warehouse*, atau penyimpanan informasi lainnya. Data *mining* berkaitan dengan bidang ilmu-ilmu lain, seperti *database system*, data *warehousing*, *statistics*, *machine learning*, *information retrieval*, dan *high-level computing* [9].

Menurut [8] bahwa jaringan syaraf tiruan diperlukan sebagai peralatan standar pada data *mining*, dan juga digunakan di berbagai kasus data *mining*, seperti klasifikasi, *analysis time series*, prediksi, dan pengelompokkan.

Kombinasi efektif *Neural Network* dan *Data Mining* merupakan teknologi yang menggunakan paket *software Analysis Neural Network* untuk menggabungkan dua kombinasi teknologi ini dengan *data warehouse* untuk meningkatkan dan mengoptimalkan teknologi *data mining* [16].

2.2 Radial Basis Function Neural Network (RBFNN)

Radial Basis Function Neural Network (RBFNN) merupakan jaringan *Multilayered Feedforward Neural Network* (MFNN) yang terdiri dari tiga lapisan, yakni *input*, *hidden layer*, dan *output*. Jaringan tersebut merupakan pemetaan fungsi *non-linear multidimensional* yang tergantung pada jarak antara *vektor input* dan *vektor center*.

Fungsi *basis radial function* (RBF), $\boldsymbol{\phi}$ dapat dikarakteristik sebagai suatu *neuron* pada ruang *input*, *center c* dan radius $\boldsymbol{r}$ adalah nilai optimum yang dicapai pada RBF pada *neuron c*. Setiap *neuron* pada *hidden layer* dihitung jarak dari *input* ke titik *center neuron* dan dimasukkan pada fungsi RBF nya, dengan persamaan:

$$\boldsymbol{h_i(x) = \phi(\|x - c_i\|^2 / r^2)} \quad (1)$$

Dimana $h_i(x)$ *output* pada *hidden layer* dan *neuron-i* untuk *input* x, $\boldsymbol{\phi}$ adalah RBF, c_i adalah *center neuron hidden i* dan r_i adalah radius. Setiap *output* dan *neuron* pada *hidden layer* dihubungkan dengan suatu bobot yang digunakan untuk menghitung nilai *output* nya, dengan persamaan sebagai berikut:

$$\boldsymbol{y_j = f(x) = \sum_{i-1}^{n-1} w_{ij} h_i(x) + w_{0j}} \quad (2)$$

Dimana y_j nilai *output* pada *neuron* ke-*j* dari *input* $\boldsymbol{x}$, $\boldsymbol{w_{ij}}$ adalah bobot yang menghubungkan *hidden neuron* ke-*i* dan *output neuron* ke-*j*, $\boldsymbol{w_{0j}}$ adalah bias untuk *neuron output*, dan *n* adalah jumlah *hidden neuron*.

RBFNN dengan *vektor input x* akan menghasilkan nilai aktual untuk *neuron output* ke$-\boldsymbol{i}$, $\boldsymbol{y_i}$ yang dinyatakan dalam persamaan berikut:

$$\boldsymbol{y_i = \sum_{k=1}^{m} w_i h_k(x)} \quad (3)$$

Atau dapat dinyatakan sebagai

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} \quad (4)$$
$$\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1r} \\ \vdots & \vdots & \ddots & \vdots \\ h_{m1} & h_{m2} & \cdots & h_{mr} \end{bmatrix}$$

dimana

$$h_k(x) = \sum_{i=1}^{n_k} \phi_i(x, c_{ki}) \sum_{i=1}^{n_k} \phi_i(\|x, c_{ki}\|2), k = 1, 2, \ldots, m. \quad (5)$$

Untuk $\boldsymbol{\phi_i}$ (•) adalah fungsi kernel *Gaussian*, diberikan sebagai berikut:

$$\boldsymbol{\phi_i(\|x, c_{ki}\|^2) = \exp\left(-\frac{\|x - c_{ki}\|_2^2}{\sigma_i^2}\right)} \quad (6)$$

dimana σ_i adalah parameter bentuk pada fungsi kernel *Gaussian*.

Struktur RBFNN yang khusus tersebut sering kali menimbulkan permasalahan karena *hidden layer* nya terlalu besar, sehingga dibutuhkan penyederhanaan jaringan [17]. Beberapa metode penyederhanaan RBFNN telah dilakukan diantaranya dengan menentukan lokasi *center* menggunakan teknik *instance reduction* [18].

2.3 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) dikenal secara luas digunakan untuk mereduksi dimensi data multivariat, kompresi data, pengenalan pola pada jaringan saraf, dan analisis statistik. Penggunaan PCA untuk *preprocessing* pada *neural network* telah banyak digunakan dalam berbagai bidang seperti klasifikasi dan peramalan. Beberapa peneliti telah menggabungkan PCA dengan model *neural network* seperti *backpropagation* (BPNN), RBFNN, dan *Self Organizing Map* (SOM). Pada permasalahan *pattern recognition*, preprocessing PCA mampu memberikan hasil yang akurat pada model *fuzzy adaptive resonance theory* (ART)-RBF [1].

Model PCA-RBFNN tidak hanya menghasilkan klasifikasi yang akurat, namun juga menyederhanakan arsitektur jaringan dan *training* yang lebih cepat dari jaringan RBFNN [7]. Algoritma PCA didesain untuk mengaproksimasi suatu pola pada ruang dimensi tinggi dengan sub ruang dimensi rendah yang merentang dengan *principal vektor eigen* dari data matriks kovarian. Dengan cara ini distribusi data dapat dinyatakan dan direkonstruksi kembali dengan *principal vektor eigen* dan nilai *eigen*nya [5].

Proyeksi pada PCA adalah representasi himpunan data **X** ke dalam bentuk *vektor eigen orthonormal* dari matriks varian-kovarian dari **X**. Matriks kovarian merupakan korelasi antara variabel-variabel dalam himpunan data **X**. PCA merupakan proses mendapatkan *vektor eigen orthonormal* dari matriks kovarian sebagai basis untuk ditransformasi ke ruang data yang baru. *Vektor eigen* dapat dikatakan sebagai basis asli untuk multidimensi data **X**. Selanjutnya PCA akan mencari proyeksi variabel-variabel yang tidak memiliki korelasi dalam melakukan proses reduksi [13].

2.4 Matriks Kovarian

Misalkan diberikan **X** adalah data *input* dengan *zero mean* yaitu setiap data pada variabel dikurangi dengan *mean* data pada variabel tersebut. Diberikan **X** adalah data *input* $m \times n$ dimana m adalah banyaknya variabel data dan n banyaknya pengamatan. Matriks **X** dapat dinyatakan sebagai berikut:

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_m} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x_{11}} & \cdots & \boldsymbol{x_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{x_{m1}} & \cdots & \boldsymbol{x_{mn}} \end{bmatrix} \quad (7)$$

Nilai varian untuk setiap variabel ke-i dalam **X** dimana $i=1,\ldots,m$ dengan $\bar{X} = [\bar{x}_1, \ldots, \bar{x}_m]$ dapat ditentukan persamaan sebagai berikut:

$$\boldsymbol{\sigma^2_{x_1x_1}} = \frac{\sum_{i=1}^{n}\left(\mathbf{x_{i-}\bar{x}_i}\right)^2}{n-1}, \quad \ldots, \quad \boldsymbol{\sigma^2_{x_mx_m}} = \frac{\sum_{i=1}^{n}\left(\mathbf{x_{m-}\bar{x}_m}\right)^2}{n-1} \quad (8)$$

Dimana $\langle \bullet \rangle_i$ adalah selisih rata-rata atas nilai data pada variabel ke-i, sedangkan varian dari dua variabel yang berbeda, misalkan x_j dan x_k ditentukan dengan persamaan berikut:

$$\boldsymbol{\sigma^2_{x_jx_k}} = \langle \mathbf{x_jx_k} \rangle_{jk}, \text{ dimana } \boldsymbol{j, k} = \mathbf{1, \ldots, m} \quad (9)$$

Kovarian antara dua variabel yang berbeda memberikan beberapa pengertian yang bervariasi terhadap dua variabel tersebut. Jika nilai kovarian antara dua variabel positif maka variabel satu naik yang lain akan naik, jika nilai kovarian dua variabel negatif maka variabel satu naik yang lain turun. Sedangkan nilai kovarian dua variabel bernilai nol, variabel-variabel tersebut bebas terhadap yang lainnya [13].

2.5 K-Means Clustering

Ide dari K-*Means Clustering* adalah mempartisi sejumlah n elemen ke sejumlah k buah *cluster*. K-*Means Clustering* merupakan salah satu metode *cluster* non-hirarki yang berusaha untuk mempartisi objek yang ada kedalam satu atau lebih *cluster* atau kelompok objek berdasarkan karakteristiknya, sehingga objek yang mempunyai karakteristik yang sama akan dikelompokkan menjadi satu *cluster* dan objek yang mempunyai karakteristik berbeda akan dikelompokkan kedalam *cluster* yang lain [6].

Jika diberikan beberapa objek $X = (x_1, x_2, \ldots, x_n)$ maka dalam algoritma metode K-*Means Cluster* akan mempartisi **X** sebanyak k buah *cluster* dan setiap *cluster* akan memiliki *centroid* atau titik pusat dari objek dalam *cluster* masing-masing [2].

Secara umum metode K-*Means Cluster* menggunakan algoritma sebagai berikut.

1. Menentukan k sebagai jumlah *cluster* yang ingin dibentuk.
2. Membangkitkan nilai *random* untuk pusat *cluster* awal (*centroid*) sebanyak k.
3. Menghitung jarak setiap data *input* terhadap masing-masing *centroid* menggunakan rumus *Euclidean Distance*. *Euclidean Distance* adalah perhitungan jarak antar dua buah titik dalam ruang *Euclidean* (*Euclidean Space*). Pada ruang 2 dimensi yang melibatkan 2 titik (misal: titik x dan titik y), maka perhitungan *Euclidean Distance* menjadi sebagai berikut.

$$\boldsymbol{d(x,y)} = \sqrt{(\boldsymbol{x_1 - x_2})^2 + (\boldsymbol{y_1 - y_2})^2} \quad (10)$$

4. Alokasikan masing-masing objek ke dalam *centroid* yang paling terdekat.
5. Penentuan *centroid* dilakukan secara acak dari objek-objek yang tersedia sebanyak k *cluster*, kemudian menghitung *centroid cluster* ke-i berikutnya diperoleh dari rata-rata *cluster* yang bersangkutan dengan menggunakan rumus:

$$\boldsymbol{C} = \frac{\mathbf{1}}{\boldsymbol{p}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{p}} \boldsymbol{x_i}, \boldsymbol{i} = \mathbf{1, 2, 3, 4..}\boldsymbol{p} \quad (11)$$

dimana:

$\boldsymbol{C}$: *centroid* pada *cluster*

$\boldsymbol{x}$: data variabel lokasi ke$-\boldsymbol{i}$ $\boldsymbol{(i = 1,2,3, \ldots, p)}$

$\boldsymbol{p}$: banyaknya objek/ jumlah anggota yang menjadi *cluster*.

6. Lakukan iterasi hingga anggota tiap *cluster* tidak ada yang berubah.
7. Apabila anggota tiap *cluster* tidak ada yang berubah, maka nilai rata-rata pusat *cluster* pada iterasi terakhir sudah konvergen [3].

2.5 Confusion Matrix

Confusion Matrix merupakan salah satu cara untuk melihat kinerja *classifier/supervised learning* dalam *unsupervised learning* biasa dikenal dengan istilah *matching matrix*. Setiap kolom dari matriks mewakili kelas yang diprediksi, sedangkan setiap baris mewakili kelas yang sebenarnya [15].

Jumlah prediksi dalam teknik didasarkan pada jumlah hasil pengujian dengan hasil klasifikasi secara benar atau salah yang telah diprediksi oleh model klasifikasi. Sebagaimana ditunjukkan dalam Tabel 1. *Confusion Matrix* (tabel kontingensi) dua kelas menunjukkan pola prediksi *classifier* terhadap masing-masing kelas [11].

Tabel 1. *Confusion Matrix* Dua Kelas

		Predicted	
		Yes	**No**
Actual	**Yes**	**TP**	**FN**
	No	**FP**	**TN**

Keterangan:

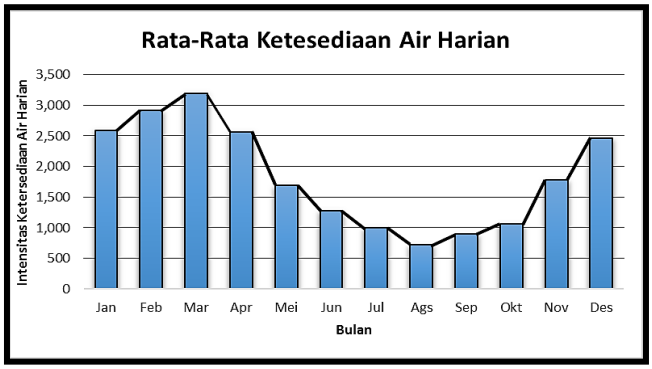
a. TP adalah *True Positive*
b. TN adalah *True Negative*
c. FN adalah *False Negative*
d. FP adalah *False Positive*

Berdasarkan nilai **TP**, **FN**, **FP** dan **TN** diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dalam mengklasifikasikan data. Nilai presisi atau *confidence* menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif, sedangkan nilai *recall* menggambarkan berapa persen dari data kategori positif yang diklasifikasikan dengan benar oleh sistem [12].

III. RESULTS AND DISCUSSION

3.1. Deskriptif Data Multivariat

Analisis secara deskriptif dilakukan untuk mengetahui pola data ketersediaan air pada 8 variabel pengamatan Bendungan Sutami. Analisis ini diolah berdasarkan catatan informasi data yang didapatkan secara *real-time* di *Station Otomatis Flood Forecasting Warning System* (FFWS) yang mewakili setiap variabel pengamatan di setiap titik lokasi di sekitar Bendungan Sutami. Karakteristik dilihat berdasarkan nilai *mean*, standar deviasi, minimum dan maksimum dari data ketersediaan air di Bendungan Sutami dan ditunjukkan pada Gambar 1.



Gambar 1. Karakteristik Deskriptif Data Variabel Parameter

Karakteristik data ketersediaan air harian selama 3 tahun dengan 8 variabel pengamatan di Bendungan Sutami dapat dilihat berdasarkan hasil dari analisis secara deskriptif. Rata-rata debit ketersediaan air tertinggi di Bendungan Sutami terjadi di musim hujan pada bulan Maret yaitu sebesar **3,189** m^3, sedangkan rata-rata ketersediaan air terendah terjadi pada bulan Agustus yaitu sebesar **723** m^3, artinya puncak ketersediaan air terendah terjadi di musim kemarau pada bulan Agustus.

3.2. Reduksi Variabel Input dengan PCA

Pada tahapan proses PCA di penelitian ini didapatkan hasil reduksi dari perhitungan matriks kovarian, nilai *eigen*, vektor *eigen*, dan kumulatif keragaman. Berikut disajikan hasil PC, nilai *eigen* dan kumulatif keragaman.

Tabel 2. Nilai Komponen utama $\boldsymbol{p}$, nilai *eigen* dan keragaman

	$\boldsymbol{p_1}$	$\boldsymbol{p_2}$	$\boldsymbol{p_3}$	$\boldsymbol{p_4}$	$\boldsymbol{p_5}$	$\boldsymbol{p_6}$	$\boldsymbol{p_7}$	$\boldsymbol{p_8}$
$\boldsymbol{x_1}$	-0.4990	-0.0143	-0.8310	-0.2372	0.0452	-0.0205	0.0368	0.0164
$\boldsymbol{x_2}$	-0.4427	-0.8464	0.2914	-0.0435	-0.0129	0.0156	-0.0141	0.0152
$\boldsymbol{x_3}$	-0.7141	**0.5315**	0.4442	-0.0934	-0.0307	-0.0150	-0.0150	-0.0096
$\boldsymbol{x_4}$	-0.0956	-0.0192	-0.0857	0.4399	-0.2997	0.3933	0.0722	-0.7348
$\boldsymbol{x_5}$	**-0.0583**	0.0075	-0.0665	0.2419	0.2267	0.0785	-0.9336	-0.0662
$\boldsymbol{x_6}$	-0.0866	0.0075	-0.0811	0.3462	-0.4463	0.8766	0.0605	-0.6652
$\boldsymbol{x_7}$	-0.1335	0.0174	-0.0386	0.6036	0.7007	0.0509	0.3347	0.1027
$\boldsymbol{x_8}$	-0.0847	0.0063	-0.0811	0.4462	-0.4463	0.3766	0.0605	0.6652
$\boldsymbol{\lambda}$	4.7031	2.3257	0.7114	0.4008	0.1553	0.1057	0.0537	0.0600
(%)	55.2288	27.3108	8.3546	4.7065	1.8233	1.2416	0.6305	0.7040

Dari Tabel 2, untuk komponen utama ke-**1** dan ke-**2** dengan nilai *eigen* **4.7031** dan **2.3257** masing-masing mewakili **55.2288%** dan **27.3108%** dari seluruh variabilitas, secara kumulatif kedua komponen utama menyatakan **82.5396%** dari total keragaman. Hal ini berarti jika delapan variabel asli $(\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_8})$ direduksi menjadi dua variabel, maka kedua variabel pengganti tersebut dapat menjelaskan **82.5396%** dari total keragaman kedelapan variabel asli.

Selanjutnya kedua variabel pengganti (misalkan $\boldsymbol{y_i}$, $\boldsymbol{i = 1, \dots, 2}$). Didefinisikan sebagai pengganti kedelapan variabel asli $(\boldsymbol{x_j}, \ \boldsymbol{j = 1, \dots, 8})$. Berdasarkan komponen utama yang bersesuaian dengan $\boldsymbol{y_{ij}}$, sampel data pada setiap variabel asli x_i diproyeksikan ke variabel pengganti, yang dapat dinyatakan sebagai berikut:

$$\boldsymbol{y_{ij}} = \begin{bmatrix} -0.4990 & -0.4427 & -0.7141 & -0.0956 & -0.0583 & -0.0866 & -0.1335 & -0.0847 \\ -0.0143 & -0.8464 & 0.5315 & -0.0192 & 0.0075 & 0.0174 & 0.0121 & 0.0063 \end{bmatrix}$$

Hasil proyeksi (misalkan $\boldsymbol{y_{ij}}$, $\boldsymbol{i = 1, 2, j = 1, .., N}$ dengan $\boldsymbol{N =}$ jumlah sampel data) kemudian digunakan sebagai *input* pada inisialisasi *center* dan *clustering* dengan K-*Means Clustering* dan klasifikasi pada RBFNN. Dengan demikian proses PCA memberikan keuntungan yakni mereduksi variabel *input* yang secara langsung akan menyederhanakan struktur jaringan RBFNN.

3.3. Klustering dengan K-Means Clustering

Proses awal untuk RBFNN adalah penentuan *center* dan *cluster*ing untuk setiap *input* y_i. Dalam PCA-RBFNN, model K-*Means Clustering* digunakan untuk penentuan inisialisasi *center* dan *cluster*ing *input* RBFNN. Pada proses K-*Means Clustering* secara *random* urutan *center* awal dipilih dari *vektor input* y_{ij} sebagai inisialisasi *center*-nya. Dengan **8** data $\boldsymbol{p_i}$ maka terdapat **8** permutasi *random* urutan *center* awal dan 8 permutasi *random* urutan data. Dari hasil *cluster*ing dengan K-*Means Cluster*ing menghasilkan *center* dan target sebagai berikut:

$$^{c} = \begin{bmatrix} 54.0294 & 65.4574 & 38.2932 & 50.5724 & 63.6828 & 64.0579 & 47.4237 & 29.3274 \\ 12.5695 & 21.5706 & 19.9529 & 10.4443 & 9.8825 & 10.1017 & 21.0183 & -82.8771 \end{bmatrix}$$

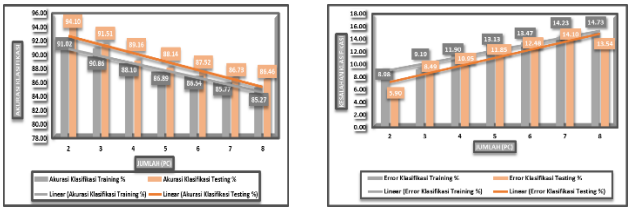
Proses K-*Means Clustering* dilakukan menggunakan 2 *cluster* dengan 20 iterasi pada *layer* kompetitifnya. Rata-rata *error* K-*Means Clustering* pada PCA-RBFNN adalah **8.81%** pada musim hujan dan **8.48%** pada musim kemarau. Hasil *clustering* dengan K-*Means Clustering* digunakan sebagai inisialisasi *center* untuk RBFNN.

3.4. Kemampuan Klasifikasi pada PCA-RBFNN

Proses untuk mengetahui performa *learning* pada PCA-RBFNN, pada penelitian ini dilakukan perhitungan

dan simulasi pada proses klasifikasi PCA-RBFNN dengan menggunakan hasil proyeksi komponen utama hasil reduksi terhadap setiap data *input*. Simulasi proses *training* dan *testing* dilakukan sebanyak **10** kali perulangan. Proses *training* pada klasifikasi digunakan data tahun 2007 - 2008 dan klasifikasi pada proses *testing* digunakan pada tahun 2009.

Proses klasifikasi dilakukan terhadap *output* **Y**, dimana **Y** merupakan skor koefisien dari data **X**. Jumlah kelas ditentukan berdasarkan jumlah komponen utama hasil reduksi dengan PCA, yaitu 2 kelas. Dari data penelitian, berdasarkan Gambar 2. Menunjukkan hasil akurasi proses *training* dan *testing* pada PC-2 dengan nilai *training* sebesar **91.02%** dan *testing* sebesar **94.10%**d an akurasi *Error* klasifikasi sebesar **8.98%** proses training dan **5.90%** proses testing.



(a) (b)

Gambar 2. Grafik Akurasi & *Error* Klasifikasi PC (2) 2 Musim untuk (a) data *training* ; (b) data *testing*

Tabel 3, menunjukkan bahwa hasil rata-rata akurasi & *error* klasifikasi tertinggi musim hujan yang terjadi dalam setiap 6 bulan (November – April) pada tahun 2007-2009 berada pada PC ke-2 dengan akurasi klasifikasi proses *training* sebesar **93.79%** dan proses *testing* sebesar **97.30%**.

Tabel 3. Akurasi Klasifikasi PC pada Musim Hujan

PC	Akurasi Klasifikasi (PC)	
	***Training* %**	***Testing* %**
2	**93.79**	**97.30**
3	92.56	95.56
4	89.14	91.11
5	87.49	89.05
6	**85.16**	**86.35**

Tabel 4, menunjukkan bahwa hasil akurasi *error* klasifikasi terendah berada pada PC-2 dengan nilai sebesar **6.21%** proses *training* dan **3.65%** proses *testing*.

Tabel 4. Akurasi *Error* Klasifikasi PC pada Musim Hujan

PC	*Error* Klasifikasi (PC)	
	***Training* %**	***Testing* %**
2	**6.21**	**3.65**
3	7.44	4.44
4	10.86	8.89
5	12.51	10.95
6	**14.84**	**13.65**

Tabel 5 menunjukkan bahwa hasil rata-rata akurasi klasifikasi tertinggi musim kemarau yang terjadi dalam setiap 6 bulan (Mei – Oktober) pada tahun 2007-2009 berada pada PC ke-2 dengan akurasi klasifikasi pada proses *training* sebesar **95.66%** dan proses *testing* sebesar **93.65%**.

Tabel 5. Akurasi Klasifikasi PC pada Musim Kemarau

PC	Akurasi Klasifikasi (PC)	
	***Training* %**	***Testing* %**
2	**95.66**	**93.65**
3	94.26	90.00
4	93.49	91.90
5	90.08	91.11
6	**85.05**	**86.67**

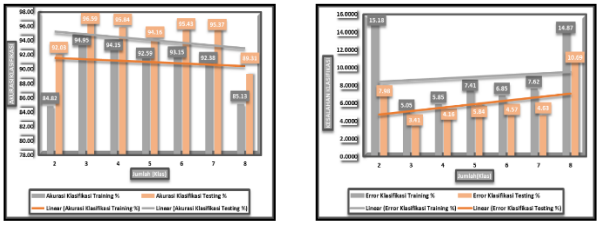
Tabel 6, menunjukkan bahwa hasil akurasi *error* klasifikasi terendah berada pada PC-2 dengan nilai sebesar **4.34%** proses *training* dan **6.35%** proses *testing*.

Tabel 6. Akurasi *Error* Klasifikasi pada Musim Kemarau

PC	*Error* Klasifikasi (PC)	
	***Training* %**	***Testing* %**
2	**4.34**	**6.35**
3	5.74	10.00
4	6.51	8.10
5	9.92	8.89
6	**14.97**	**13.33**

Setelah didapatkan akurasi PC terbaik pada 8 PC dalam 10 kali percobaan didapatkan akurasi PC terbaik berada di PC-2 dilanjutkan pada uji simulasi untuk mengetahui akurasi klasifikasi klas pada PC-2.

Gambar 3. Menunjukkan hasil akurasi percobaan perulangan terbaik 2 musim berada pada Klas-3 sebesar **94.95%** untuk proses *training* dan **96.59%** untuk proses *testing*. Dengan *error* klasifikasi sebesar **5.05%** *training* dan **3.41%** *testing*.



(a) (b)

Gambar 3. Grafik Akurasi & *Error* Klasifikasi Klas PC (2) 2 Musim untuk (a) data *training* ; (b) data *testing*

Tabel 7, menunjukkan bahwa hasil rata-rata akurasi & *error* klasifikasi Klas tertinggi PC-2 pada Musim Hujan setiap 6 bulan (November – April) pada tahun 2007-2009 dengan akurasi klasifikasi tertinggi berada pada Klas-3 dengan proses *training* sebesar **95.22%** dan proses *testing* sebesar **100.00%**.

Tabel 7. Akurasi Klasifikasi Klas PC-2 pada Musim Hujan

Kelas	Akurasi Klasifikasi (PC)	
	Training %	**Testing %**
2	93.68	100.00
3	**95.22**	**100.00**
4	94.01	98.89
5	94.09	100.00
6	93.95	96.67

7	93.29	94.44
8	**92.31**	**91.11**

Tabel 8. menunjukkan bahwa akurasi *Error* klasifikasi terendah berada pada Klas-3 dengan nilai sebesar **4.78%** proses *training* dan **0.00%** proses *testing*.

Tabel 8. Akurasi *Error* Klasifikasi Klas PC-2 pada Musim Hujan

Kelas	*Error* Klasifikasi	
	Training %	**Testing %**
2	6.32	0.00
3	**4.78**	**0.00**
4	5.98	7.78
5	5.91	0.00
6	6.05	5.56
7	**6.71**	**8.89**
8	**6.69**	**3.33**

Tabel 9, menunjukkan bahwa hasil rata-rata akurasi & *error* klasifikasi Klas tertinggi PC-2 pada Musim Kemarau setiap 6 bulan (Mei – Oktober) pada tahun 2007-2009 dengan akurasi klasifikasi tertinggi berada pada Klas-3 dengan proses *training* sebesar **96.91%** dan proses *testing* sebesar **100.00%.**

Tabel 9. Akurasi Klasifikasi Klas PC-2 pada Musim Kemarau

Jumlah Klas	Akurasi Klasifikasi	
	Training %	**Testing %**
2	96.24	100.00
3	**96.91**	**100.00**
4	95.51	91.11
5	**95.03**	**97.78**
6	95.47	88.89
7	**95.30**	**81.11**
8	**94.96**	**80.00**

Tabel 10, menunjukkan bahwa akurasi *Error* klasifikasi terendah berada pada Klas-3 dengan nilai sebesar **3.09%** proses *training* dan **0.00%** proses *testing*.

Tabel 10. Akurasi *Error* Klasifikasi Klas PC-2 pada Musim Kemarau

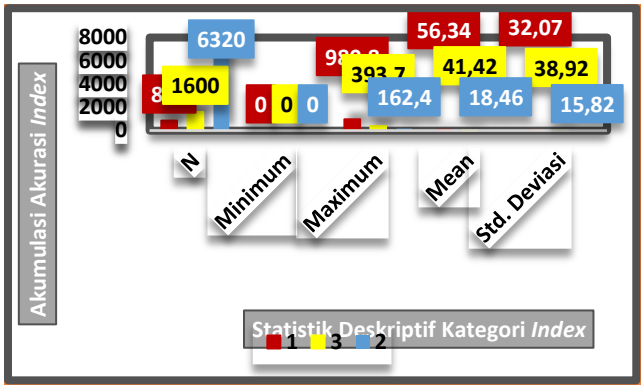
Klas	*Error* Klasifikasi	
	Training %	**Testing %**
2	3.76	0.00
3	**3.09**	**0.00**
4	4.49	8.89
5	4.97	2.22
6	4.53	11.11
7	4.70	18.89
8	**5.04**	**20.00**

Dengan didapatkan nilai hasil akurasi klas terbaik PC-2, berada pada klas-3 maka nilai *center* awal tersebut digunakan pada metode RBFNN untuk mengklasifikasi data pada penelitian ini. Dari uji simulasi di penelitian berikut ini masing-masing memberikan penjelasan klasifikasi variabel pengamatan terhadap intensitas pengaruh pada data ketersediaan air harian di Bendungan, Sutami melalui Klasterisasi *Index* pada *input* RBFNN.

Tabel 11. Deskriptif Data Keseluruhan *Index*

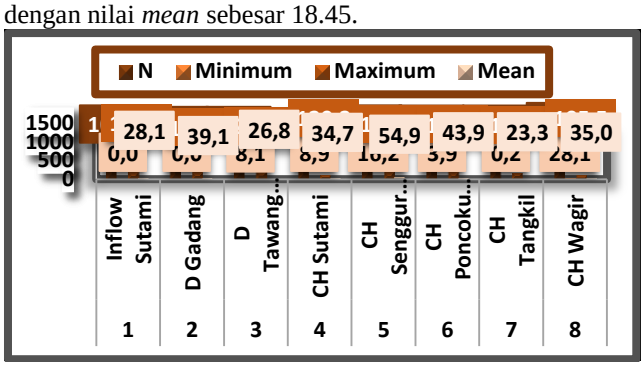
Kategori Index	N	Min	Max	Mean	Std. Deviasi	Kategori
1	848	0.00	980.8	56.34	32.07	Intensitas Tinggi
3	1600	0.00	393.7	41.41	38.92	Intensitas Sedang
2	6320	0.00	162.4	18.45	15.82	Intensitas Rendah

Tabel 11, menunjukkan bahwa hasil klasterisasi rata-rata **'*Index* tertinggi'** berada pada kategori ***index*-1**, di lanjutkan pada Kategori ***Index*-3** dengan Status **'Intensitas Sedang'** dan Kategori ***Index*-2** dengan Status **'Intensitas Rendah'** pada data ketersediaan air harian di Bendungan, Sutami.



Gambar 4. Analisis Grafik pada Kategori *Index*

Gambar 4. Menunjukkan bahwa kategori *index* tertinggi berapa pada *index*-1, dengan nilai *mean* sebesar 56.34 dilanjutkan dengan kategori *index* intensitas sedang berada pada kategori *index*-3 dengan nilai *mean* sebesar 41.41 dan kategori *index*-2 memiliki nilai *index* terendah dengan nilai *mean* sebesar 18.45.



Gambar 5. Deskriptif Data Variabel Pengamatan Kategori Seluruh *Index*

Gambar 5, menunjukkan bahwa *index* kumulatif tertinggi dapat ditunjukkan pada Curah Hujan Sengguruh, dengan memiliki 1096 nilai titik data dengan *mean* tertinggi sebesar 54.9 dilanjutkan dengan *index* kumulatif tertinggi yang kedua Curah Hujan Poncokusumo dengan memiliki 1096 nilai titik data dengan *mean* sebesar 43.9. Sedangkan variabel pengamatan dengan *index* terendah pada Curah Hujan Tangkil memiliki 1096 nilai titik data dengan nilai *mean* sebesar 23.3.

IV. CONCLUSION

Berdasarkan hasil penelitian pada pembahasan dapat disimpulkan bahwa proses mereduksi dimensi variabel *input* dari data multivariat pada jaringan RBFNN. menggunakan metode PCA dihasilkan komponen utama ke-1 dan ke-2 dengan nilai *eigen* 4.7031 dan 2.3257

masing-masing mewakili 55.2288% dan 27.3108% dari seluruh variabilitas, secara kumulatif kedua komponen utama menyatakan sebesar 82.5396%. Dari proses tersebut, dihasilkan akurasi *Principal Component* (PC) tertinggi berada pada PC-2 di 2 musim dengan nilai *training* sebesar 91.02% dan *testing* sebesar 94.10%, berdasarkan kondisi geografis akurasi klasifikasi dibagi 2 menjadi musim hujan dan kemarau, pada akurasi musim hujan dihasilkan 93.79% *training* dan 97.30% *testing* dan musim kemarau 95.66% *training* dan 93.65% *testing*. Dihasilkan akurasi *error* klasifikasi pada PC-2 di 2 musim dengan nilai *training* sebesar 8.98% dan *error* sebesar 5.90%, dan berdasarkan kondisi geografis akurasi *error* klasifikasi dibagi 2 menjadi musim hujan dan kemarau, pada akurasi musim hujan dihasilkan 6.21% *training* dan 3.65% *testing* dan musim kemarau 4.34% *training* dan 6.35% *testing*. Pada proses penentuan inisialisasi *center* awal pada PC terbaik PC-2 dengan menggunakan K-*Means Clustering*. Pada hasil percobaan perulangan pada penelitian ini didapatkan hasil rata-rata proses akurasi model Klas terbaik di 2 musim, musim hujan dan kemarau berada pada Klas-3 dengan akurasi klasifikasi di atas 90% dan akurasi kesalahan klasifikasi di bawah 10%.

REFEENCE

[1] Cho, B. Y. 2008. Nonlinear Support Vector Machine Visualization for Risk Factor Analysis Using Nomograms and Localized Radial Basis Function Kernels. *IEEE Transactions On Information Technology In Biomedicine*, 12, 247-256.

[2] Ediyanto, M. N. 2013. Pengklasifikasian Karakteristik Dengan Metode K-Means Cluster Analysis. *BIMASTER*, 2(02).

[3] Hasanah, N. U. 2017. Sistem Pengelompokan Curah Hujan Menggunakan Metode K-Means Di Wilayah Kalimantan Timur. In *Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi* , Vol (Vol. 2, No. 2).

[4] Haykin, S. 1994. Neural networks-A comprehensive foundation. New York: IEEE Press.

[5] Huang, D. Y. 2008. A new local PCA-SOM algorithm. *Neurocomputing*, 71, 3544-3552.

[6] Jain. A.K. 2009. Data Clustering: 50 Years Beyond K-Means. *Pattern Recognition Letters*, 2009.

[7] Lu, W. W. 2004. Potensial Assesment of A Neural Network Model with PCA/RBF Approach Forecasting Pollutant Trends in Mong Kok Urban Air, Hong Kong. *Environmental Research*, 96, 79–87.

[8] Maimon, O. d. 2010. Introduction to knowledge discovery and data mining. Boston: *In Data mining and knowledge discovery handbook*, pp. 1-15. Springer.

[9] Meilani, B. D. 2014. Aplikasi Data mining Untuk Menghasilkan Pola Kelulusan Siswa Dengan Metode Naive Bayes. *Jurnal LINK*, 21 (2), 1-6.

[10] Munnoli, A. 2013. Clustering Algorithms for Radial Basis Function Neural Network. *ITSI Transactions on Electrical and Electronics Engineering (ITSI-TEEE)*, Volume -1, Issue.

[11] Omary, Z. 2010. Machine Learning Approach to Identifying the Dataset Threshold for the Performance Estimators in Supervised Learning . *International Journal for Infonomics (IJI)*, 3(3) : 314-325.

[12] Rosandy, T. 2016. Perbandingan Metode Naive Bayes Classifier Dengan Metode Decision Tree (C4.5) Untuk Menganalisa Kelancaran Pembiayaan (Study Kasus: KSPPS/BMT Al-Fadhila). *Jurnal Teknologi Informasi Magister Darmajaya*, 2(01), 52-62.

[13] Shlens, J. 2009. *A Tutorial on Principal Component Analysis*. New York University: Version 3.01. Center.

[14] Soesanto, O. 2010. PCA-RBPNN untuk Klasifikasi Data Multivariat dengan Orthogonal Least Square (OLS). *Jurnal Matematika Murni Dan Terapan*, 4(2) : 51-60.

[15] Susanto, E. 2016. Evaluasi Hasil Klaster Pada Dataset Iris, Soybean-small, Wine Menggunakan Algoritma Fuzzy C-Means dan K-Means++. *Surya Informatika*, 2(1) : 6-13.

[16] Tewary, G. 2015. Effective Data mining For Proper Mining Classification Using Neural Network. *International Journal Of Data mining & Knowledge Management Process* (IJDKP), Volume 5:2.

[17] Wang, X. L. 2008. A Definition of Partial Derivative of Random Functions and It's Application to RBFNN Sensitivity Analysis. *Neurocomputing*, 71, 1551-1526..

[18] Yousef, R. D. 2006. Locating Center Points for Radial Basis Function Networks Using Instance Reduction Techniques. World Acad*emy of Science, Engineering and Technology*, 4, 213-216.