

K-Means Clustering to Design Radial Basis Function Neural Network (RBFNN) Classifiers

Gellysa Urva

Sekolah Tinggi Teknologi Dumai
Jl. Utama Karya Bukit Batrem II Dumai
e-mail: gellysa.urva@gmail.com

ABSTRAK

Radial Basis Function Neural Network (RBFNN) is hybrid network. It is combination training both supervised and unsupervised. RBFNN does not use generally activation function in Neural Network. It use radial based function. Dalam proses membangun arsitektur JSTRBF membutuhkan proses *cluster*. Pada penelitian ini, penulis menggunakan algoritma *K-Means Clustering*, dimana algoritma ini menjadi algoritma yang efisien dan efektif dalam mengolah data. *K-Means Clustering* merupakan salah satu algoritma *clustering* dengan tujuan untuk membagi data menjadi beberapa kelompok. Algoritma ini akan mengelompokkan data atau objek ke dalam k buah kelompok (*cluster*) yang diinginkan. Penentuan jumlah *cluster* pada JSTRBF mempengaruhi akurasi dari JSTRBF dalam melakukan klasifikasi. Hasil penelitian ini menyatakan *desain* arsitektur JSTRBF yang memiliki akurasi terbaik pada jumlah 3 *cluster* dengan nilai akurasi sebesar 87,06%. Akan tetapi jika jumlah *cluster* pada JSTRBF di atas 4 *cluster* nilai akurasi di bawah 50%.

Kata kunci : Algoritma *K-Means Clustering*, JST, RBF

Pendahuluan

Teknologi *clustering* merupakan suatu teknik yang bertujuan untuk mengelompokkan data berdasarkan persamaan karakteristik sehingga menghasilkan informasi yang bermanfaat. Algoritma *clustering* data sudah banyak dipergunakan diberbagai bidang misalnya untuk proses pengolahan citra, data mining proses pengambilan keputusan, pengenalan pola, maupun dalam bidang bioinformatika. *Clustering* data akan mengelompokkan objek yang paling dekat dimana terdapat kesamaan dengan objek lain, serta data yang akan *dicluster* diambil secara acak atau random. [1]

Ada beberapa algoritma yang diusulkan untuk dapat melakukan proses *clustering* pada suatu dataset dalam jumlah yang banyak. Pada penelitian ini, penulis akan menggunakan algoritma *K-Means clustering* dalam menentukan jumlah *cluster* terbaik. Pada algoritma pembelajaran ini, komputer mengelompokkan sendiri data-data yang menjadi masukannya tanpa mengetahui terlebih dulu target kelasnya. Pembelajaran ini termasuk dalam *unsupervised learning* (Pelatihan tak terbimbing). Masukan yang diterima adalah data atau objek dan k buah kelompok (*cluster*) yang diinginkan. Algoritma ini akan mengelompokkan data atau objek ke dalam k buah kelompok tersebut. Pada setiap *cluster* terdapat titik pusat (*centroid*) yang merepresentasikan *cluster* tersebut. *K-Means clustering* adalah pengolahan data secara numerik, tidak terbimbing (*unsupervised*), dan merupakan salah satu metode yang interaktif. Algoritma *clustering* ini sederhana dan sangat cepat, terhadap beberapa kasus metode ini sangat efektif karena memberikan hasil *cluster* terbaik. [2]

Radial Basis Function (RBF) merupakan salah satu model dari Jaringan Syaraf Tiruan (*Neural Network*). RBF dikenal dengan jenis jaringannya *feed forward*. *Radial Basis Function Neural Network* (RBFNN) dalam pemilihan fitur lebih konsisten. Langkah-

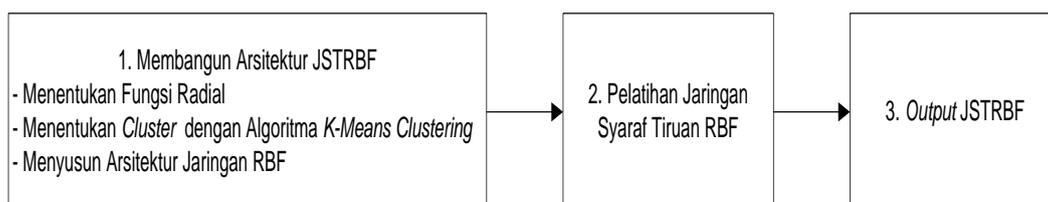
langkahnya yang konsisten ini digunakan untuk mengevaluasi keterkaitan antara fitur. JSTRBF memiliki suatu keuntungan yaitu kemampuan terbaik untuk memperkirakan struktur jaringan secara sederhana serta algoritma pembelajaran tercepat.[3] Salah satu ciri model Jaringan Saraf Tiruan RBF adalah pada fungsi aktivasi yang dalam perhitungannya membutuhkan nilai pusat dan *varians* lapisan tersembunyi. Algoritma *K-Means clustering* ini mengelompokkan data *input* menjadi beberapa kelompok atau *cluster* sehingga nilai pusat dan *varians* setiap *cluster* dapat dihitung. Pusat *cluster* adalah rata-rata (*means*) *cluster* tersebut. Algoritma *K-Means clustering* digunakan sebagai alternatif metode *cluster* untuk data dengan ukuran besar karena memiliki kecepatan yang lebih tinggi dibandingkan dengan metode hirarki.[4]

Penentuan jumlah *cluster* merupakan langkah awal dalam menentukan jumlah *hidden layer* pada JSTRBF. Proses membangun *desain* arsitektur Jaringan Syaraf Tiruan *Radial Basis Function* membutuhkan algoritma pelatihan dari *input layer* ke *hidden layer*. Algoritma ini dibutuhkan untuk mendapatkan *cluster* yang merupakan penentu jumlah dari *hidden layer* pada JSTRBF. Permasalahan yang sering terjadi pada JSTRBF yaitu penentuan jumlah *cluster* yang akan digunakan pada JSTRBF. Jumlah *cluster* ini mempengaruhi kinerja dari JSTRBF. Untuk mendapatkan kinerja yang baik, dibutuhkan pemilihan jumlah *cluster* yang tepat untuk JSTRBF. Kinerja yang akan dibahas pada penelitian ini adalah akurasi Jaringan Syaraf Tiruan RBF dalam melakukan klasifikasi.

Metode Penelitian

Teknik pengumpulan data yang digunakan dalam penelitian ini menggunakan studi literatur dan tinjauan pustaka. Data yang digunakan dalam penelitian ini adalah Data *Iris* yang diperoleh dari UCI *Machine Learning Repository* yang diperoleh dari <http://archive.ics.uci.edu/ml>. UCI *Machine Learning Repository* adalah sebuah koleksi *database*, *domain* teori, dan data *generator* yang digunakan oleh komunitas yang mempelajari mesin pembelajaran (*machine learning*), untuk keperluan analisis empiris dari algoritma *machine learning*. *Dataset* yang tersedia pada UCI *Machine Learning Repository* digunakan oleh pelajar, pendidik, dan peneliti diseluruh dunia sebagai sumber utama dari *dataset* pada *machine learning*. Jumlah *dataset* yang tersedia pada UCI *Machine Learning Repository* pada saat ini sudah berjumlah 320 *dataset* yang dapat digunakan sesuai dengan kebutuhan pada pembelajaran *machine learning*. *Data set* yang digunakan adalah *Iris Data Set* yang memiliki jumlah data sebanyak 150 dengan 4 atribut. Atribut informasi yang ada pada *Iris Data Set* adalah terdiri-dari: *Sepal Length*, *Sepal Width*, *Petal Length*, dan *Petal Width*. *Iris Data Set* memiliki 3 *class* yaitu *Iris Setosa*, *Iris Versicolor*, dan *Iris Virginica*. Masing-masing 50 data (33,3%) untuk setiap *class* yaitu *iris setosa*, *iris versicolour* dan *iris virginica*. *Iris Data set* merupakan *dataset* yang banyak digunakan di dalam permasalahan pengenalan pola dan klasifikasi.

Agar penelitian ini terarah serta dapat mengetahui hasil dari penelitian, maka penulis melakukan tahapan berurut secara keseluruhan untuk penyelesaian masalah dalam penelitian. Adapun tahapan penyelesaian masalah penelitian yang penulis rancang dapat dilihat pada gambar 1.



Gambar 1. Skema Penyelesaian Masalah

Proses membangun arsitektur jaringan RBF dilakukan dengan langkah-langkah sebagai berikut :

1. Menentukan fungsi basis yaitu fungsi *Gaussian*.
2. Menentukan jumlah *cluster*. Penentuan *cluster* akan diproses dengan algoritma *K-Means clustering*. Langkah-langkah algoritma *K-Means clustering* adalah sebagai berikut :
 - Pilih K buah titik *centroid* secara acak
 - Kelompokkan data sehingga terbentuk K buah *cluster* dengan titik *centroid* dari setiap *cluster* merupakan titik *centroid* yang telah dipilih sebelumnya
 - Perbaharui nilai titik *centroid*
 - Ulangi langkah 2 dan 3 sampai nilai dari titik *centroid* tidak lagi berubah

Untuk pemahaman lebih *detail* terkait cara kerja pada algoritma *K-Means Clustering*, Gambar 3 memaparkan *pseudocode* dari algoritma *K-Means Clustering*.

```
//Tentukan k sebagai jumlah cluster yang ingin dibentuk
int k = this.k;

//Tentukan nilai centroid secara random
if (this.centroid[0] == null)
{
    Random(data);
}

//Hitung jarak data ke centroid dan pilih yang terdekat
private Func<double[ ], double[ ], double > distance ;
public int Nearest(double[ ] point)
{
    int min_cluster = 0;
    double min_distance = distance (point, centroid [0]);

    for (int i =1; i < centroids.Length; i++)
    {
        double dist = distance (point, centroids [i] );
        if (dist < min_distance)
        {
            min_distance = dist;
            min_cluster = i;
        }
    }
    return min_cluster;
}

for (int i = 0; i < data.length; i++)
{
    double [ ] point = data [ i ];
    int c = labels [ i ] = Nearest (data [ i ] );
    count [ c ]++;
    double [ ] centroid = newCentroids [ c ];
    for (int j = 0; j < centroid.Length; j++)
        centroid [ j ] += point [ j ];
}

//Ubah posisi centroid dengan cara menghitung nilai rata-rata jarak data yang memilih centroid yang sama
for (int i = 0; i < k; i++)
{
    double [ ] mean = newCentroids [ i ];
    double clusterCount = count [ i ];
    if (clusterCount !=0)
    {
        for (int j = 0; j < cols; j++)
            mean [ j ] /= clusterCount;
    }
}
if (converged(centroids, newCentroids, threshold)) break;
centroids = newCentroids;
```

Gambar 2. *Pseudocode K-Means Clustering*

3. Menyusun Arsitektur jaringan RBF

Pada penelitian ini *input* data untuk Jaringan Syaraf Tiruan RBF menggunakan empat atribut data iris, dimana :

X1 panjang sepal dalam cm
 X2 lebar sepal dalam cm
 X3 panjang petal dalam cm
 X4 lebar petal dalam cm

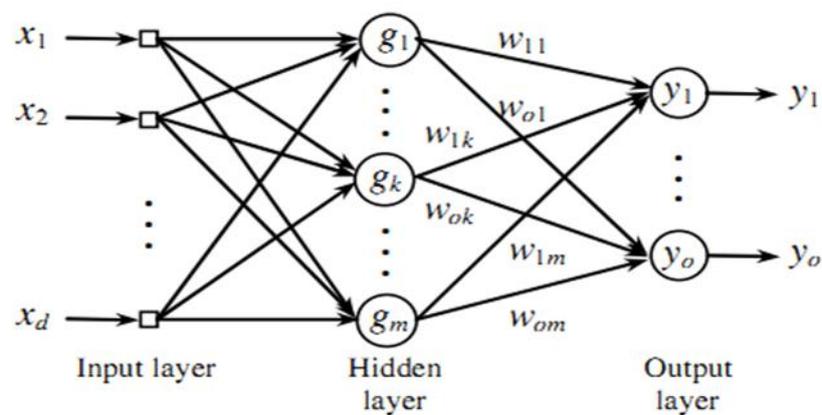
Output dari Jaringan Syaraf Tiruan RBF pada penelitian ini berupa bilangan *binner* untuk Y1 dan Y2. Output akhir JSTRBF untuk klasifikasi tumbuhan iris dengan *threshold* adalah :

Y1= 0, Y2 = 0 adalah iris *setosa*

Y1= 0, Y2 = 1 adalah iris *versicolour*

Y1= 1, Y2 = 0 adalah iris *virginica*

Gambar 2 menunjukkan *desain* Jaringan Syaraf Tiruan RBF yang terdiri dari *Input*, *Hidden* dan *Output Layer*.



Gambar 3. Arsitektur JSTRBF

Hasil dan Pembahasan

Pada penelitian ini ditampilkan hasil dari performansi untuk setiap jumlah *cluster* dengan menentukan *epoch* dan *learning rate* terlebih dahulu. Penulis membatasi *epoch* pelatihan Jaringan Syaraf Tiruan RBF antara 50 sampai 200 sedangkan untuk nilai *learning rate* pada fase pelatihan berkisar 0 sampai 1. Pemilihan jumlah *cluster* merupakan langkah awal penentuan jumlah *neuron hidden layer*, karena banyaknya jumlah *cluster* adalah banyaknya jumlah *hidden layer*, maka disini peneliti membatasi variasi jumlah *hidden layer* dimulai dari 3 sampai 10 *hidden layer*. Hasil pengujian untuk mendapatkan nilai akurasi pada penelitian ini, akan disajikan dalam bentuk tabel dan grafik.

3.1 Hasil Pengujian untuk 3 cluster

Pengujian pertama yang dilakukan menggunakan 3 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian 3 cluster

Epoch	Akurasi JSTRBF (%)
50	77.33
100	96
150	97.33
200	98.67

Hasil pengujian dengan 3 *cluster* menunjukkan akurasi JSTRBF di atas 50%, dengan nilai akurasi tertinggi 98.67% pada *epoch* 200.

3.2 Hasil Pengujian untuk 4 *cluster*

Pengujian kedua dilakukan dengan 4 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dengan 4 *cluster* dapat dilihat pada Tabel 2.

Tabel 2. Hasil Pengujian 4 *cluster*

<i>Epoch</i>	Akurasi JSTRBF (%)
50	58.67
100	62
150	78
200	82

Hasil pengujian dengan 4 *cluster* menunjukkan akurasi JSTRBF masih atas 50%, dengan nilai akurasi tertinggi 82% pada *epoch* 200.

3.3 Hasil Pengujian untuk 5 *cluster*

Pengujian selanjutnya menggunakan 5 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dengan 5 *cluster* dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian 5 *cluster*

<i>Epoch</i>	Akurasi JSTRBF (%)
50	24
100	24
150	26.67
200	28.67

Hasil pengujian dengan 5 *cluster* menunjukkan akurasi JSTRBF jauh berbeda dengan pengujian sebelumnya. Akurasi JSTRBF bernilai di bawah 30%.

3.4 Hasil Pengujian untuk 6 *cluster*

Pengujian selanjutnya menggunakan 6 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dengan 6 *cluster* dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian 6 *cluster*

<i>Epoch</i>	Akurasi JSTRBF (%)
50	24
100	27.33
150	30
200	32

Hasil pengujian dengan 6 *cluster* menunjukkan akurasi JSTRBF mengalami peningkatan dari pengujian sebelumnya. Meskipun, naiknya nilai akurasi tidak terlalu jauh dari nilai akurasi sebelumnya. Akurasi JSTRBF tertinggi untuk pengujian 6 *cluster* adalah pada epoch 200 dengan nilai akurasinya sebesar 32%.

3.5 Hasil Pengujian untuk 7 *cluster*

Pengujian selanjutnya menggunakan 7 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dengan 7 *cluster* dapat dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian 7 *cluster*

<i>Epoch</i>	Akurasi JSTRBF (%)
50	23.33
100	26.67
150	30.67
200	34

Hasil pengujian dengan 7 *cluster* menunjukkan akurasi JSTRBF tidak stabil dari pengujian sebelumnya. Pada *epoch* 200, terlihat hasil pengujian mengalami kenaikan sebesar 2% dari pengujian sebelumnya dengan 6 *cluster*. Nilai akurasi JSTRBF tertinggi bernilai 34%.

3.6 Hasil Pengujian untuk 8 *cluster*

Pengujian selanjutnya menggunakan 8 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dengan 8 *cluster* dapat dilihat pada Tabel 6.

Tabel 6. Hasil Pengujian 8 *cluster*

<i>Epoch</i>	Akurasi JSTRBF (%)
50	27.33
100	31.33
150	34
200	37.33

Hasil pengujian dengan 8 *cluster* secara keseluruhan menunjukkan akurasi JSTRBF mengalami peningkatan dari pengujian dengan 7 *cluster*. Nilai akurasi JSTRBF tertinggi bernilai 37.33% pada *epoch* 200.

3.7 Hasil Pengujian untuk 9 *cluster*

Pengujian selanjutnya menggunakan 9 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dengan 9 *cluster* dapat dilihat pada Tabel 7.

Tabel 7. Hasil Pengujian 9 *cluster*

<i>Epoch</i>	Akurasi JSTRBF (%)
50	24.67
100	24.67

150	25.33
200	26.67

Hasil pengujian dengan 9 *cluster* secara keseluruhan menunjukkan akurasi JSTRBF mengalami penurunan dibandingkan pengujian sebelumnya dengan 8 *cluster*. Nilai akurasi JSTRBF secara keseluruhan masih dibawah 30%.

3.8 Hasil Pengujian untuk 10 *cluster*

Pengujian selanjutnya menggunakan 10 *cluster* untuk melihat akurasi JSTRBF. Dimana data diuji dengan ketentuan *epoch* mulai 50, 100, 150, 200 *epoch* serta *learning rate* berkisar 0 sampai 1. Tampilan hasil pengujian dengan 10 *cluster* dapat dilihat pada Tabel 8.

Tabel 8. Hasil Pengujian 10 *cluster*

<i>Epoch</i>	Akurasi JSTRBF (%)
50	32
100	32
150	32
200	32

Hasil pengujian dengan 10 *cluster* secara keseluruhan menunjukkan akurasi JSTRBF mengalami peningkatan dari pengujian sebelumnya dengan 9 *cluster*. Pada pengujian dengan 10 *cluster* nilai akurasi JSTRBF secara keseluruhan bernilai sama sebesar 32%.

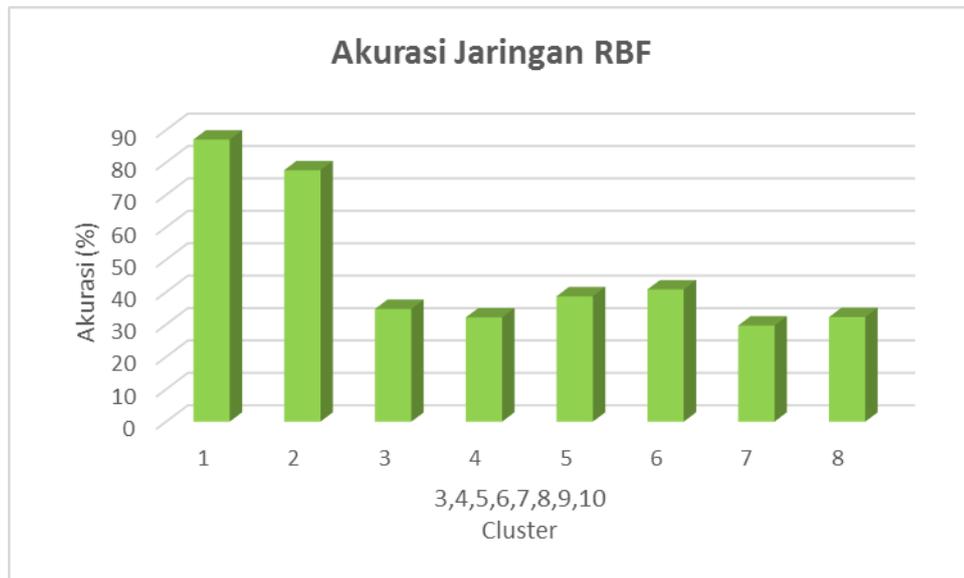
Dari pengujian masing-masing *cluster*, dapat dilihat hasil pengujian akurasi JSTRBF bervariasi. Secara keseluruhan bisa meningkat, menurun atau bahkan tidak mengalami perubahan nilai akurasi. Untuk lebih jelasnya, Tabel 9 menunjukkan hasil akurasi Jaringan Syaraf Tiruan RBF dalam melakukan klasifikasi dengan variasi jumlah *cluster* 3 sampai 10.

Tabel 9. Akurasi Jaringan RBF

Jumlah <i>Cluster</i>	Akurasi JSTRBF (%)
3	87.06
4	77.60
5	34.79
6	32.15
7	38.70
8	40.79
9	29.65
10	32.27

Hasil pengujian dari setiap *cluster* yang telah diuji dari JSTRBF menghasilkan nilai akurasi yang dapat dilihat pada tabel 9. Berikut ini penyajian perbandingan nilai

akurasi Jaringan Syaraf Tiruan RBF dalam melakukan klasifikasi yang disajikan dalam bentuk grafik yang dapat dilihat pada Gambar 4.



Gambar 4. Akurasi Jaringan RBF

Berdasarkan analisis data yang terlihat pada Tabel dan Grafik, hasil pengujian akurasi Jaringan Syaraf Tiruan RBF dalam melakukan klasifikasi dengan nilai akurasi tertinggi sebesar 87.06 % untuk 3 jumlah *cluster*. Sedangkan untuk nilai akurasi terendah yaitu saat JSTRBF dengan 9 *cluster* memiliki nilai akurasi sebesar 29.65%. Terlihat dengan jelas akurasi dari arsitektur Jaringan Syaraf Tiruan RBF dipengaruhi oleh penentuan jumlah *cluster*. Semakin banyak jumlah *cluster*, hasil akurasi semakin rendah pada kasus penelitian ini. Walaupun tidak ada kestabilan nilai akurasi pada saat jumlah *cluster* 5 sampai 10 *cluster*. Hal ini dikarenakan nilai akurasi yang tidak berubah secara konsisten baik menurun maupun semakin meningkat. Dapat dilihat dari isi tabel pada saat jumlah *cluster* JSTRBF 4,5 dan 6 nilai akurasi semakin rendah. Namun pada saat jumlah *cluster* JSTRBF 7 dan 8 nilai akurasi kembali meningkat, meskipun peningkatan nilai akurasi tidak terlalu tinggi. Pada jumlah *cluster* 9, nilai akurasi menurun kembali dan meningkat lagi pada saat *cluster* berjumlah 10. Selisih perubahan nilai akurasi dari JSTRBF berkisar 2.64% sampai 11,14%. Pada penelitian ini, algoritma *K-Means Clustering* menghasilkan *cluster* ideal pada arsitektur Jaringan Syaraf Tiruan RBF yaitu dengan jumlah *cluster* 3 sampai 4 dengan nilai akurasi diatas 50%. Sedangkan untuk arsitektur JSTRBF yang memiliki jumlah *cluster* 5, 6, 7, 8, 9, 10 memiliki nilai akurasi dibawah 50%. Dapat dilihat dari gambar 4 yang menyatakan semakin banyak jumlah *cluster* maka nilai akurasi semakin menurun. Hal ini disebabkan dengan banyaknya jumlah *cluster* maka perhitungan semakin kompleks sehingga ketelitian numerik akan semakin berkurang.

Simpulan

Penentuan jumlah *cluster* yang diproses melalui *Algoritma K-Means Clustering* mempengaruhi akurasi pada Jaringan Syaraf Tiruan *Radial Basis Function* (JSTRBF) dalam melakukan klasifikasi. Semakin banyak jumlah *cluster* maka semakin menurun nilai akurasinya. Hal ini dibuktikan dari hasil pengujian pada penelitian ini yang menyatakan arsitektur jaringan dengan jumlah *cluster* lebih dari empat, nilai akurasi yang dihasilkan rata-rata dibawah 50%. Pengujian yang dilakukan pada penelitian ini dengan variasi jumlah *cluster*, menghasilkan akurasi terbaik JSTRBF dalam melakukan

klasifikasi dengan jumlah *input* 4 *neuron* dan *output* 2 *neuron* yaitu menggunakan arsitektur *cluster* sebanyak 3. Akurasi yang dihasilkan oleh JSTRBF yang terbaik sebesar 87.06%. Penelitian selanjutnya dapat menggantikan algoritma *K-Means clustering* yang digunakan pada penelitian ini dengan algoritma *clustering* yang lainnya untuk menentukan pusat *cluster*. Hal ini dilakukan untuk melihat pengaruh algoritma *clustering* yang dipilih terhadap hasil akurasi JSTRBF.

Daftar Pustaka

- [1] Debatty, Thibault., et.Al., *Determining thek in k-means with MapReduce*. Proceedings of the EDBT/ICDT (ISSN 1613-0073). 2014:19-28.
- [2] N. Singh, D. Singh, Performance Evaluation of K-Means and Heirarichal Clustering in Terms of Accuracy and Running Time. *International Journal of Computer Science and Information Technologies (IJCSIT)*. 2012;3:4119-4121.
- [3] Thandar, A. M., & Khaing, M. K. Radial Basis Function (RBF) Neural Network Classification based on Consistency Evaluation Measure. *International Journal of Computer Applications*. 2012; **54**(15): 0975-8887.
- [4] Loochach, Richa dan Garg, Kanwal. Effect of Distance Functions on *k-Means* Clustering Algorithm. *International Journal of Computer Application*. 2012; **49**(6): 7-9.
- [5] William, Graham. *Data Mining Cluster*. ANU Data Mining Group. 2015.
- [6] Berkhin, Pavel. *Survey on Clustering Data Mining Techniques*. Department of Electrical and Computer Engineering University of California. 2015
- [7] Shovon, Md. Hedayatul Islam, Haque, Mahfuza. An Approach of Improving Student's Academic Performance by Using *k-Means* Clustering Algorithm and Decision Tree. *International Journal of Advanced Computer Science and Applications*. 2012;**3**(.8): 146-149.
- [8] Dileep K, Yadav, Dutta, M. *A comparative study of clustering Algorithms for wheat data*. ACEIT Conference Proceeding. India. 2016 : 238-241.