

PENERAPAN RADIAL BASIS FUNCTION NEURAL NETWORK DALAM PENGKLASIFIKASIAN DAERAH TERTINGGAL DI INDONESIA

Vira Wahyuningrum¹

¹Badan Pusat Statistik Provinsi Jawa Barat
e-mail: ¹vira.2w@bps.go.id

Abstrak

Penetapan daerah tertinggal di Indonesia merupakan kasus pengklasifikasian dengan dua kategori pada variabel respon (biner). Pengklasifikasian dengan metode klasifikasi linier yang umum digunakan yaitu regresi logistik pada tahap eksplorasi data menghasilkan *misclassification* yang relatif besar, sehingga diperlukan suatu metode alternatif. Artificial Neural Network (ANN) merupakan alternatif yang menjanjikan untuk berbagai metode klasifikasi konvensional. Radial Basis Function Neural Network (RBFNN) merupakan salah satu arsitektur ANN yang populer digunakan dalam klasifikasi. Metode RBFNN menggunakan dua pendekatan yaitu *supervised* dan *unsupervised* serta dalam beberapa penelitian menghasilkan akurasi klasifikasi yang tinggi. Penelitian ini bertujuan menerapkan metode RBFNN untuk kasus klasifikasi daerah tertinggal di Indonesia untuk melihat arsitektur RBFNN yang terbentuk dan ketepatan klasifikasi yang dihasilkan. Hasil dari penelitian ini adalah penerapan RBFNN memberikan performa yang sangat baik yaitu nilai akurasi sebesar 93,48 persen, sensitivitas 81,10 persen dan spesififikasi 97,43 persen. Nilai *F-Measure* arsitektur RBFNN yang dihasilkan mencapai 85,36 persen.

Kata kunci: Neural Network, Radial Basis Function, klasifikasi, daerah tertinggal

Abstract

Determination of underdeveloped regency in Indonesia is the case with the classification of two categories on the response variable (binary). Classification with the linear classification method that is commonly used is logistic regression at the data exploration stage resulting in relatively large misclassification, so we need an alternative method. Artificial Neural Network (ANN) is a promising alternative to various conventional classification methods. Radial Basis Function Neural Network (RBFNN) is one of the popular ANN architectures used in classification. The RBFNN method uses two approaches namely supervised and unsupervised and in several studies produces high classification accuracy. This study aims to apply the RBFNN method for the classification case of underdeveloped regency in Indonesia to see the RBFNN architecture formed and the resulting classification accuracy. The results of this study are the application of RBFNN provides an excellent performance that is an accuracy value of 93.48 percent, sensitivity 81.10 percent and 97.43 percent specifications. The F-Measure value of RBFNN architecture reached 85.36 percent.

Keywords: Neural Network, Radial Basis Function, Classification, Underdeveloped regency

PENDAHULUAN

Pembangunan Daerah Tertinggal (PDT) merupakan suatu proses, upaya, dan tindakan secara terencana untuk meningkatkan kualitas masyarakat dan wilayah yang merupakan bagian integral dari pembangunan nasional. Percepatan PDT dimulai dengan pengidentifikasian daerah tertinggal oleh Kementerian Negara Pembangunan Daerah Tertinggal (KNPDT) dengan menggunakan kriteria yang telah ditetapkan. Status daerah sebagai daerah tertinggal ditetapkan oleh Presiden melalui Peraturan Pemerintah. Berdasarkan penetapan tersebut, KNPDT kemudian bekerja sama dengan kementerian-kementerian lain dan pemerintah daerah untuk menyusun strategi percepatan PDT.

Pengklasifikasian daerah tertinggal di Indonesia menggunakan variabel yang telah digunakan sebelumnya oleh KNPDT dengan suatu metode statistik merupakan kajian yang menarik. Hasil pengklasifikasian dengan metode statistik dapat dibandingkan ketepatannya dengan hasil penetapan daerah tertinggal oleh KNPDT. Klasifikasi merupakan pengelompokan objek ke beberapa kelompok berdasarkan variabel yang diamati. Klasifikasi daerah tertinggal merupakan kasus klasifikasi dengan dua kategori pada variabel respon (biner/dikotomi). Data yang digunakan dalam pengklasifikasian daerah tertinggal melibatkan 27 variabel dalam enam kriteria yang sangat kompleks.

Metode klasifikasi yang sering digunakan untuk data dengan respon kategorik atau biner adalah metode regresi logistik. Namun, pada regresi logistik dan metode klasifikasi klasik lainnya diperlukan asumsi awal yang harus dipenuhi agar diperoleh hasil klasifikasi yang optimal. Metode regresi logistik memiliki syarat pemenuhan asumsi diantaranya tidak terjadinya multikolinieritas pada variabel prediktornya. Dalam penelitian di bidang sosial, masalah multikolinieritas seringkali tidak bisa dihindari sehingga asumsi independensi tidak terpenuhi. Berdasarkan

hasil eksplorasi data untuk mendeteksi multikolinieritas dengan nilai Variance Inflation Factor (VIF), terdapat 5 variabel prediktor yang mengandung multikolinieritas pada model regresi daerah tertinggal. Berkaitan dengan hasil eksplorasi data tersebut, metode regresi logistik memiliki keterbatasan untuk digunakan dalam pengklasifikasian daerah tertinggal dengan keseluruhan variabel yang ada. Penggunaan metode regresi logistik dengan pelanggaran asumsi akan mempengaruhi hasil klasifikasi yaitu terjadinya *misclassification* atau *error rate* yang relatif besar. Oleh karena itu, diperlukan metode alternatif yang dapat menjadi solusi dalam kasus klasifikasi ini.

Beberapa metode modern dikembangkan sebagai metode alternatif untuk membantu menyelesaikan masalah klasifikasi, salah satunya metode berbasis *machine learning* yaitu Artificial Neural Network (ANN) atau dikenal juga sebagai Jaringan Syaraf Tiruan (JST). Dalam dekade terakhir, ANN telah muncul sebagai alat menarik untuk pemodelan proses nonlinier, terutama dalam situasi di mana pengembangan fenomenologis atau model regresi konvensional menjadi tidak praktis atau rumit (Živković, 2008). ANN merupakan metode yang dapat menjadi solusi karena mampu melakukan ekstraksi hubungan antara *input* dan *output* tanpa asumsi awal dan bekerja berdasarkan informasi dari data yang tersedia. ANN memiliki kemampuan memodelkan permasalahan nonlinier kompleks yang sulit dipecahkan dengan persamaan matematis biasa (Haykin, 2008).

ANN dapat diaplikasikan untuk pengklasifikasian daerah tertinggal dengan pertimbangan banyaknya variabel yang digunakan, pola hubungan antar variabel prediktor dan variabel respon yang tidak diketahui dengan jelas dan jumlah observasi yang cukup banyak (491 kabupaten). Selain itu, penelitian-penelitian sebelumnya yang menunjukkan kelebihan metode ANN dalam menyelesaikan masalah klasifikasi sangat mendukung untuk menggunakan metode tersebut dalam penelitian ini.

Penelitian mengenai klasifikasi daerah tertinggal dilakukan oleh Naibaho (2016) menggunakan metode Back Propagation Neural Network (BPNN) dan Learning Vector Quantization (LVQ) yang menghasilkan kesimpulan bahwa metode LVQ memberikan tingkat akurasi prediksi yang lebih baik daripada BPNN. Selanjutnya, penulis juga telah menerapkan metode Probabilistic Neural Network (PNN) dalam kasus klasifikasi yang sama (Wahyuningrum, 2017). Sitamahalakshmi, dkk (Sitamahalakshmi, dkk, 2011) membandingkan metode Radial Basis Function Neural Network (RBFNN) dan PNN dalam pengenalan karakter Telugu (pola tulisan bahasa Telugu di India) dan menghasilkan kesimpulan bahwa akurasi metode RBFNN secara keseluruhan lebih baik dibandingkan PNN. Metode RBFNN dipilih dalam penelitian ini karena merupakan arsitektur yang populer digunakan dalam klasifikasi. RBFNN bersifat *feed forward*, tipe jaringan (*multilayer*) serta keunggulan fungsi aktivasi yang digunakan. Berdasarkan uraian tersebut, penelitian ini bertujuan menerapkan metode RBFNN dalam pengklasifikasian daerah tertinggal di Indonesia untuk mendapatkan model/arsitektur dan melihat performa klasifikasi yang dihasilkan oleh metode tersebut.

METODOLOGI PENELITIAN

1. Tinjauan Referensi

Klasifikasi Daerah Tertinggal

Daerah tertinggal adalah daerah kabupaten yang wilayah serta masyarakatnya kurang berkembang dibandingkan dengan daerah lain dalam skala nasional. Pemerintah menetapkan 122 kabupaten di Indonesia sebagai daerah tertinggal tahun 2015-2019 dalam Peraturan Pemerintah Nomor 131 Tahun 2015. Penetapan daerah tertinggal dilakukan lima tahun sekali dan dilakukan evaluasi pada setiap tahun. Dalam Keputusan Menteri Negara PDT RI No.001/KEP/M-PDT/I/2005, suatu daerah dikategorikan sebagai daerah tertinggal

dikarenakan beberapa faktor penyebab, antara lain faktor geografis, sumber daya alam, sumber daya manusia, sarana dan prasarana, daerah rawan bencana dan konflik sosial, serta kebijakan pembangunan.

Penetapan kriteria daerah tertinggal dilakukan dengan menggunakan pendekatan berdasarkan pada 6 kriteria dasar, yaitu : (i) perekonomian masyarakat, (ii) sumber daya manusia, (iii) prasarana (infrastruktur), (iv) kemampuan keuangan daerah, (v) aksesibilitas, dan (vi) karakteristik daerah (KNPDT dalam Naibaho, 2016). Keenam kriteria tersebut dibagi menjadi beberapa variabel sebagai dasar penetapan daerah tertinggal dengan total sebanyak 27 variabel (Lampiran). Klasifikasi daerah termasuk tertinggal atau tidak tertinggal ditentukan berdasarkan hasil perhitungan indeks dari nilai 27 *standardized indicators* masing-masing daerah.

Berdasarkan Panduan Penjelasan Penetapan Daerah Tertinggal yang disusun oleh KNPDT, secara ringkas tahapan penentuan daerah tertinggal adalah sebagai berikut (Naibaho, 2016):

1. Data dasar yang digunakan adalah data yang bersumber dari Podes, Susenas dan Kemampuan Keuangan Daerah (KKD) sebanyak 27 variabel.
2. Dilakukan standarisasi variabel karena satuan masing-masing variabel tidak sama. Tujuan standarisasi agar dapat diperbandingkan antar variabel.
3. Dari hasil standarisasi variabel, selanjutnya dilakukan perkalian dengan bobot untuk masing-masing variabel.
4. Indikator dari hasil perkalian standarisasi variabel dengan masing-masing bobot variabel ditentukan arahnya berupa tanda positif atau negatif. Variabel seperti jumlah prasarana atau kemampuan keuangan daerah mempunyai arah yang negatif, dan sebaliknya. Selanjutnya dilakukan penjumlahan indeks.
5. Hasil total indeks ini yang dijadikan patokan penetapan daerah tertinggal, dimana daerah-daerah yang memiliki

total indeks di atas 0 merupakan daerah tertinggal.

Artificial Neural Network (ANN)

ANN atau dikenal juga dengan Neural Network (NN) atau JST merupakan suatu jaringan syaraf yang dibangun untuk meniru cara kerja otak manusia. ANN adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi (Fausett, 1994). ANN tercipta dari suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang berdasarkan pada asumsi berikut:

1. Pemrosesan informasi terjadi pada elemen sederhana (*neuron*).
2. Sinyal mengalir di antara *neuron* melalui suatu penghubung.
3. Setiap penghubung antara *neuron* satu dengan lainnya mempunyai bobot/*weight*.
4. Setiap *neuron* akan menerapkan suatu fungsi aktivasi (biasanya nonlinier) terhadap inputnya (hasil penjumlahan sinyal yang dibobot) untuk menentukan sinyal outputnya.

Karakteristik dari ANN antara lain meliputi bentuk/pola hubungan antar *neuron* yang disebut sebagai arsitektur, metode untuk menentukan bobot hubungan yang disebut sebagai pelatihan (*training*)/algoritma dan fungsi aktivasi. Arsitektur NN merupakan pengaturan *neuron* ke dalam lapisan, pola hubungan dalam lapisan, dan di antara lapisan (Fausset, 1994). Arsitektur NN antara lain jaringan dengan lapisan tunggal (*Single Layer Net*), jaringan dengan banyak lapisan (*Multilayer Net*) dan jaringan Kompetitif (*Competitive Layer*).

Secara umum proses pembelajaran pada NN digolongkan menjadi dua, yaitu:

1. *Supervised learning* (pembelajaran dengan pengawasan), yaitu jaringan diberikan target yang akan dicapai sebagai dasar untuk mengubah bobot pada jaringan.
2. *Unsupervised learning* (pembelajaran tanpa pengawasan), yaitu jaringan mengorganisasikan sendiri bobot dalam

jaringan berdasarkan parameter tanpa adanya target.

Beberapa metode/arsitektur yang dapat digunakan dalam pengklasifikasian menggunakan ANN diantaranya BPNN, PNN, RBFNN dan LVQ.

2. Metode Analisis

Radial Basis Function Neural Network (RBFNN)

RBFNN merupakan arsitektur dari ANN yang bersifat *feed forward* dan dapat melakukan proses klasifikasi dengan waktu singkat. RBFNN melakukan proses pembelajaran yang sangat cepat karena *neuron* disetel secara lokal (Sitamahalakshmi, dkk, 2011). RBFNN merupakan metode jaringan syaraf tiruan yang menggunakan fungsi aktivasi radial basis dan umum dipakai dalam kasus klasifikasi dan prediksi/peramalan. Dalam beberapa penelitian, metode RBFNN dimodifikasi dengan pendekatan *K-means cluster* dan fungsi aktivasi Gaussian sehingga meningkatkan keakurasian hasil klasifikasi. Waktu pelatihan (*training*) pada jaringan RBFNN sangat cepat dan memiliki kemampuan generalisasi yang baik (Sitamahalakshmi, dkk, 2011). RBFNN dapat mengatasi beberapa keterbatasan BPNN karena menggunakan *hidden layer* tunggal untuk pemodelan fungsi nonlinier, sehingga mampu melatih data lebih cepat dari BPNN (Ghaderzadeh, dkk, 2013). RBFNN adalah teknik alternatif yang menarik untuk masalah klasifikasi (Kotsiantis, dkk, 2007).

RBFNN menggunakan radial basis function yaitu Gaussian. Fungsi aktivasi Gaussian merupakan fungsi yang memperhitungkan jarak atau kedekatan antara data dengan pusat data. Fungsi Gaussian mempunyai sifat lokal, yaitu bila *input* dekat dengan rata-rata (pusat), maka fungsi akan menghasilkan nilai 1, sedangkan bila *input* jauh dari rata-rata maka fungsi akan memberikan nilai nol. Selain itu fungsi Gaussian merupakan salah satu radial basis function yang memberikan hasil terbaik dalam pengenalan pola. Jaringan RBFNN termasuk dalam tipe jaringan ANN dengan banyak lapisan atau biasa disebut dengan *multilayer*. RBFNN

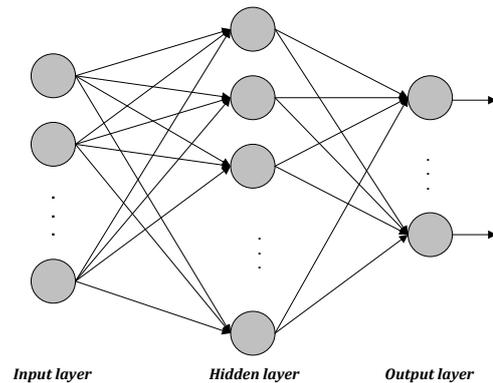
dalam proses klasifikasi menghitung bobot jarak Gaussian dengan pusat *cluster*.

RBFNN pertama kali diformulasikan oleh D.S. Broomhead dan David Lowe pada sebuah penelitian tahun 1988. RBFNN mempunyai satu lapisan tersembunyi/*hidden layer* dengan fungsi aktivasi radial basis dan lapisan *output*. Jaringan RBFNN merupakan jaringan yang unik karena menggunakan dua pendekatan dalam metodenya yaitu *supervised* dan *unsupervised*. Proses pembelajaran RBFNN hanya dilakukan satu arah dan sekali saja. Jaringan syaraf tiruan fungsi *radial basis* biasanya membutuhkan jumlah *neuron* yang lebih banyak daripada jaringan *feedforward* (Kusumadewi, 2003). Pada RBFNN, *input* akan diolah oleh suatu fungsi aktivasi dan bukan merupakan hasil penjumlahan terbobot dari data *input*, namun berupa vektor jarak antara vektor bobot dan vektor *input* yang dikalikan dengan bobot bias.

RBFNN merupakan arsitektur ANN yang dikenal sangat cepat dalam melakukan proses pembelajaran (*training*). Kelebihan jaringan RBFNN antara lain waktu pelatihan yang lebih cepat dibandingkan Multilayer Perceptron (MLP) dan interpretasi *hidden layer* pada RBFNN lebih mudah dibandingkan *hidden layer* pada MLP. Sedangkan kelemahannya, pada RBFNN generalisasi tidak begitu baik dan terlalu fleksibel, sehingga menyebabkan terjadinya *noise* pada saat pelatihan dan menyebabkan kesalahan prediksi (Devega, 2013).

RBFNN terdiri dari tiga lapisan/*layer* yaitu *input layer*, *hidden layer* dan *output layer*. Setiap *neuron/unit* pada *input layer* sesuai dengan komponen dari vektor masukan. *Hidden layer* adalah satu-satunya lapisan tersembunyi dalam RBFNN yang berlaku transformasi nonlinier dari *input layer* ke dalam *hidden layer* dengan menggunakan fungsi aktivasi nonlinier. Dengan menggunakan algoritma *K-Mean Cluster*, pola pelatihan pada *hidden layer* akan terkelompok ke dalam jumlah yang wajar. *Output layer* terdiri dari *neuron* linier terhubung ke semua *neuron* tersembunyi. Jumlah *neuron* pada *output layer* sama

dengan jumlah kelas data target. Arsitektur RBFNN digambarkan sebagai berikut (Sitamahalakshmi, dkk, 2011):



Gambar 1. Arsitektur RBFNN

Output dari setiap *neuron* pada *hidden layer* dihitung dengan menggunakan fungsi radial basis Gaussian sebagai berikut (Sitamahalakshmi, dkk, 2011):

$$G(\|x - \mu_i\|) = \exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma^2}\right). \quad (1)$$

di mana,

x : sampel pelatihan

μ_i : pusat dari *neuron* ke i pada *hidden layer*

σ : lebar *neuron* (*spread*)

Neuron output didefinisikan oleh fungsi penjumlahan berikut (Sitamahalakshmi, dkk, 2011):

$$Y(x) = \sum_i w_i G(\|x - \mu_i\|) + b. \quad (2)$$

Di mana b adalah bobot bias dan w adalah vektor bobot yang dihitung dengan mengalikan *pseudoinverse* dari matriks G dengan vektor target (d) dari data *training* dengan rumus (Sitamahalakshmi, dkk, 2011):

$$w = (G^T G)^{-1} G^T d. \quad (3)$$

Hal yang khusus pada RBFNN adalah pemrosesan sinyal dari *input layer* ke *hidden layer* bersifat nonlinier, sedangkan dari *hidden layer* ke *ouput layer* bersifat linier. Pada *hidden layer* digunakan sebuah fungsi aktivasi berbasis radial untuk membawa *input* menuju lapisan (*layer*) berikutnya. Dari beberapa fungsi aktivasi, fungsi Gaussian merupakan fungsi aktivasi yang paling sering digunakan dalam metode

RBFNN karena mempunyai sifat lokal, berdasarkan jarak/ kedekatan *input* dengan rata-rata (*pusat cluster*). *Hidden layer* pada RBFNN dapat dilihat sebagai fungsi yang memetakan pola *input* dari ruang nonlinier ke ruang linier. Pada *hidden layer*, kekuatan diskriminatif jaringan ditentukan oleh pusat *Radial Basis Function* (RBF). Ada beberapa metode yang berbeda untuk memilih pusat/*centroid*, namun yang paling sering digunakan adalah *K-means cluster*.

Tahap pembelajaran *unsupervised* pada RBFNN adalah menentukan *mean* dan standar deviasi dari variabel *input* pada setiap *node* pada unit *hidden layer*. Metode *K-Mean Cluster* adalah salah satu dari beberapa metode *unsupervised* pada pemodelan RBFNN. Metode *K-means cluster* digunakan untuk menemukan satu set pusat yang lebih akurat mencerminkan distribusi titik data. Jumlah pusat diputuskan di awal dan setiap pusat seharusnya mewakili sekelompok titik data. Pada metode *K-means cluster*, data dipartisi kedalam *K cluster*. Penentuan nilai rata-rata dari setiap *cluster* dilakukan dengan iterasi.

Dalam metode RBFNN terdapat sejumlah parameter (*weight*) yang harus ditaksir. Untuk mendapatkan model RBFNN yang sesuai, perlu menentukan kombinasi yang tepat antara jumlah variabel *input*, jumlah *node* (*cluster*) pada unit *hidden layers*, *mean* dan standar deviasi (skala atau *width*) dari variabel *input* pada setiap *node*, yang berimplikasi pada jumlah parameter yang optimal.

Performa Klasifikasi

Salah satu cara untuk mengevaluasi performa suatu metode klasifikasi/ klasifikator adalah dengan menggunakan confusion matrix. Dari confusion matrix ini

dapat dihitung beberapa parameter untuk mengukur performa klasifikasi. Menurut Han (Han dkk., 2011), confusion matrix merupakan alat yang sangat berguna untuk menganalisis seberapa baik klasifikator mengenali objek dari kelas yang berbeda. Elemen dari confusion matrix didefinisikan sebagai TP: True Positive (benar positif); FP: False Positive (salah positif); FN: False Negative (salah negatif) ;dan TN: True Negative (benar negatif).

Parameter penting dalam pengukuran performa klasifikasi yang dapat diukur berdasarkan *confusion matrix* adalah akurasi, sensitivitas dan spesifikasi. Akurasi didefinisikan sebagai perbandingan antara jumlah diagonal utama *confusion matrix* dengan jumlah keseluruhan objek. Nilai akurasi yang semakin mendekati 100 % menunjukkan performa klasifikator semakin tinggi/ baik. Sensitivitas atau disebut juga *recall* merupakan ukuran nilai ketepatan dari suatu kejadian yang diinginkan (benar positif). Sedangkan spesifikasi adalah persentase dari suatu kejadian yang tidak diinginkan (benar negatif). Formulasi penghitungan akurasi, sensitivitas dan spesifikasi berdasarkan *confusion matrix* dituliskan dalam persamaan berikut (Ghaderzadeh, dkk, 2013) :

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (4)$$

$$Sensitivitas = \frac{TP}{TP + FN} \times 100\% \quad (5)$$

$$Spesifikasi = \frac{TN}{FP + TN} \times 100\% \quad (6)$$

Nilai *error rate* juga dipakai untuk menentukan performa hasil klasifikasi di mana nilai ini menunjukkan tingkat kesalahan klasifikasi (*misclassification*)

Tabel 1. *Confusion Matrix* dengan Respon Biner untuk Klasifikasi Daerah Tertinggal

Prediksi	Daerah Tertinggal (Y=1)	Daerah Tidak Tertinggal (Y=0)	Total
Aktual			
Daerah Tertinggal (Y=1)	TP	FN (type II error)	TP + FN
Daerah Tidak Tertinggal (Y=0)	FP (type I error)	TN	FP + TN
Total	TP + FP	FN + TN	TP+TN+FP+FN

atau kelas prediksi yang tidak sesuai dengan kelas aktualnya. Semakin semakin rendah nilai *misclassification* maka semakin baik klasifikator, dan sebaliknya.

$$missclassification (\%) = 100 - akurasi \quad (7)$$

F-Measure dapat digunakan sebagai ukuran tunggal dari uji performa klasifikasi yang merupakan ukuran rata-rata harmonik dari sensitivitas (*recall*) dan *precision*. *Recall* dan *precision* adalah dua ukuran penting untuk mengevaluasi kebenaran algoritma klasifikasi. *F-Measure* diformulasikan sebagai berikut (Sitamahalakshmi, dkk, 2011):

$$F - Measure = \frac{2 * precision * recall}{(precision + recall)} * 100\% \quad (8)$$

Penghitungan *precision* menggunakan rumus berikut:

$$precision = \frac{TP}{TP + FP} * 100\% \quad (9)$$

F-Measure menggabungkan ukuran *recall* dan *precision* dengan nilai maksimum adalah 1. Bila nilai tersebut semakin mendekati 1 atau 100 persen, maka kinerja *classifier* (algoritma klasifikasi) dianggap semakin baik.

Sumber Data dan Variabel

Penelitian ini menggunakan data sekunder meliputi 27 variabel yang

digunakan oleh Kementerian Negara Pembangunan Daerah Tertinggal dan Transmigrasi (KNPDT) dalam penetapan daerah tertinggal dan tidak tertinggal di Indonesia. Variabel *input* yang digunakan diantaranya berasal dari data Potensi Desa dan data Survei Sosial Ekonomi Nasional (Susenas) dari BPS serta data Kemampuan Keuangan Daerah (KKD) dari Kemenkeu. Penelitian ini menggunakan data sebanyak 491 kabupaten/kota di Indonesia. Variabel *output* (Y) merupakan *output* biner yaitu daerah tertinggal (Y=1) dan daerah tidak tertinggal (Y=0).

Sebelum dilakukan pengolahan data, perlu dilakukan *pre-processing* data untuk menyamakan satuan pada 27 variabel *input* melalui standarisasi. Standarisasi dilakukan dengan mengurangi setiap nilai variabel x dengan rata-rata pada variabel tersebut dan membagi dengan standar deviasinya (s). Standarisasi nilai *input* akan menghasilkan nilai pada range[-1,1].

Tahapan Penelitian

Langkah-langkah yang akan dilakukan dalam pengklasifikasian daerah tertinggal dengan metode RBFNN sebagai berikut:

1. Menyiapkan *pre-processing* data yang akan digunakan (standarisasi variabel *input*)

Tabel 2. Arsitektur RBFNN pada Klasifikasi Daerah Tertinggal

Parameter	Detail
Data Sampel	Data <i>Training</i> (90%), Data <i>Testing</i> (10%)
<i>Cross Validation</i>	k-Fold (k=10)
Standarisasi Nilai <i>Input</i>	Range [-1,1]
Pembelajaran	Kombinasi <i>Supervised</i> (terawasi) dan <i>Unsupervised</i>
Fungsi Aktivasi	Gaussian $G(\ x - \mu_i\) = \exp\left(-\frac{\ x - \mu_i\ ^2}{2\sigma^2}\right).$
Jumlah <i>Layer</i>	1 <i>input layer</i> , 1 <i>hidden layer</i> , 1 <i>output layer</i>
Jumlah <i>Neuron</i>	<i>Input layer</i> = 27 (jumlah variabel <i>input</i>) <i>Hidden Layer</i> = jumlah <i>cluster</i> (k-means <i>cluster</i>) <i>Output Layer</i> = 2 (jumlah kelas data <i>output</i>)

2. Membagi data menjadi 10 bagian ($k=10$) untuk *k-Fold cross validation*.
3. Menentukan jumlah *cluster* pada *hidden layer* dan melakukan klasifikasi menggunakan metode RBFNN pada data *training* dan data *testing* untuk masing-masing *fold*/bagian.
4. Menghitung performa klasifikasi meliputi akurasi, sensitivitas, dan spesififikasi dan *F-Measure*
5. Kesimpulan penelitian

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan metode *k-fold cross validation* yaitu membagi data ke dalam k bagian (*fold*) secara acak dengan ukuran *fold* yang mendekati sama, di mana k adalah jumlah bagian yang ditentukan. Nilai k yang sering digunakan dalam penelitian adalah 10 (Priddy, 2005). Pembagian data *training* dan data *testing* berdasarkan *10-fold cross validation* untuk 491 sampel yang digunakan dan selanjutnya dilakukan klasifikasi menggunakan RBFNN pada data *training* dan data *testing*.

RBFNN merupakan jaringan dengan arsitektur yang terdiri atas 3 *layer* meliputi *input layer*, *hidden layer* dan *output layer*. Pada *input layer* terdapat sejumlah *neuron* yang merupakan jumlah variabel yang

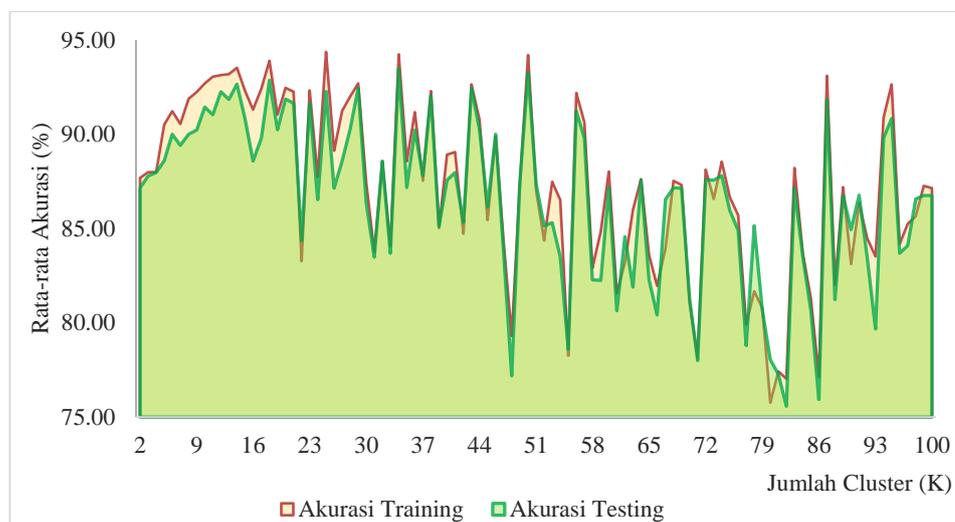
digunakan dalam pengklasifikasian yaitu sebanyak 27 variabel. Banyaknya *neuron* pada *output layer* RBFNN sama dengan jumlah kelas data yaitu 2 *neuron*. Banyak *neuron* pada *hidden layer* sebanyak *cluster* yang digunakan dalam model pengklasifikasian. *Center cluster* pada penelitian ini ditentukan dengan algoritma *K-means cluster*, dan jumlah *cluster* (K) yang diuji ditentukan pada nilai $K=2$ sampai dengan $K=100$. Rata-rata akurasi paling tinggi dihasilkan pada jumlah *cluster* $k=34$ yaitu sebesar 93,48 persen, sedangkan akurasi paling rendah pada jumlah *cluster* sebanyak 82 yaitu 75,57 persen. Oleh karena itu, jumlah *cluster* yang dipilih dalam arsitektur RBFNN ini adalah 34.

Jumlah *cluster* yang digunakan menggambarkan banyaknya *neuron* pada *hidden layer* RBFNN. Dengan demikian, arsitektur RBFNN yang dipilih dalam pengklasifikasian daerah tertinggal adalah RBFNN (27-34-2) tersusun oleh 27 *neuron* pada *input layer*, 34 *neuron* pada *hidden layer* dan 2 *neuron* pada *output layer*.

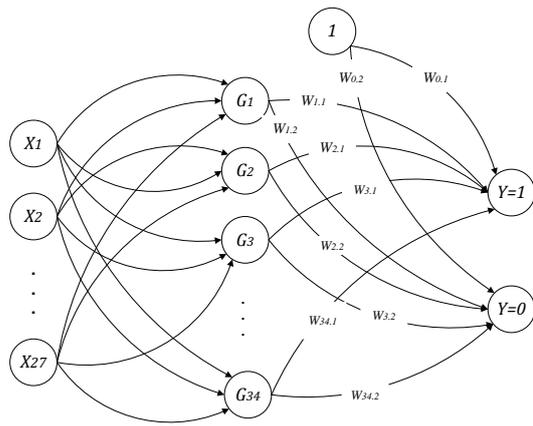
Akurasi yang dihasilkan dengan arsitektur RBFNN (27-34-2) dari data *training* melalui 10 percobaan *10-fold cross validation* berkisar antara 89,80 hingga 97,96 persen. Hasil *testing* dari 10 *fold* menunjukkan akurasi paling tinggi pada

Tabel 3. Pembagian Data Menggunakan 10-Fold cross validation

Data Sampel	1	2	3	4	5	6	7	8	9	10
Data <i>Training</i>	441	442	442	442	442	442	442	442	442	442
Data <i>Testing</i>	50	49	49	49	49	49	49	49	49	49



Gambar 2. Akurasi Klasifikasi RBFNN Menurut Jumlah *Cluster* (persen)



diklasifikasikan sesuai data aktualnya dan hanya 1 daerah yang diklasifikasikan berbeda dengan data aktualnya. *Overfitting* (akurasi hasil klasifikasi data *training* lebih tinggi daripada data *testing*) arsitektur RBFNN (27-34-2) ini terjadi pada 6 dari 10 percobaan.

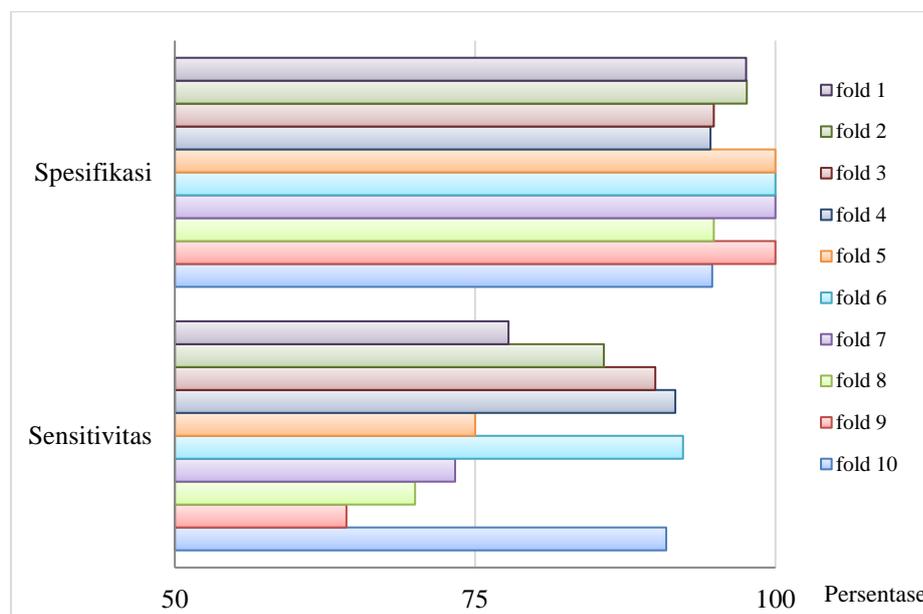
Sensitivitas arsitektur RBFNN (27-34-2) dari data *testing fold* 6 menunjukkan persentase paling tinggi yaitu sebesar 92,31

Gambar 3. Arsitektur RBFNN (27-34-2)

fold 6, artinya dengan menggunakan arsitektur RBFNN, dari 49 data *testing* dalam *fold* tersebut, 48 daerah berhasil



Gambar 4. Ketepatan Hasil Klasifikasi dengan RBFNN (27-34-2)



Gambar 5. Sensitivitas dan Spesifikasi Hasil Klasifikasi RBFNN Menurut Fold

Tabel 4. Rata-rata *Misclassification* RBFNN (27-34-2) Hasil Klasifikasi

Metode RBFNN	Rata-rata Akurasi	Rata-rata <i>Misclassification</i> (100-Akurasi)%
Data <i>Training</i>	94,23	5,77
Data <i>Testing</i>	93,48	6,52

Tabel 5. *F-Measure* Arsitektur RBFNN (27-34-2)

Metode	<i>True Positive</i>	<i>False Positive</i>	<i>Precision</i>	Sensitivitas (<i>Recall</i>)	<i>F-Measure</i>
RBFNN	91	10	90,10	81,10	85,36

persen yang berarti bahwa arsitektur RBFNN memberikan ketepatan klasifikasi sebesar 92,31 persen dalam mengklasifikasikan daerah tertinggal. Spesifikasi atau ketepatan klasifikasi daerah tertinggal dengan model RBFNN pada *fold 5*, *fold 6*, *fold 7* dan *fold 9* mencapai 100 persen, yang berarti keseluruhan daerah tidak tertinggal dalam keempat *fold* ini diklasifikasikan secara tepat sesuai dengan aktualnya. *Testing fold 10* memberikan spesifikasi terendah, yaitu 94,74 persen dari daerah tidak tertinggal yang diklasifikasikan dengan tepat sesuai status sebenarnya sebagai daerah tidak tertinggal.

Rata-rata akurasi yang dihasilkan berdasarkan 10-*fold cross validation* pada arsitektur RBFNN (27-34-2) mencapai 93,48 persen. Rata-rata sensitivitas yang menggambarkan kemampuan jaringan alam mengklasifikasikan daerah tertinggal sesuai data aktualnya memberikan hasil sebesar 81,10 persen. Untuk spesifikasi, RBFNN mampu menghasilkan ketepatan klasifikasi daerah tidak tertinggal sebesar 97,43 persen.

Secara umum, berdasarkan evaluasi performa klasifikasi dengan menggunakan arsitektur RBFNN (27-34-2) diperoleh hasil performa klasifikasi yang baik. RBFNN memiliki kelemahan pada sensitivitas namun unggul dalam spesifikasi. Rata-rata *misclassification* dari model RBFNN sebesar 6,52 persen dengan hasil *misclassification* sebanyak 32 daerah. *Misclassification* pada RBFNN meliputi 22 daerah tertinggal yang diklasifikasikan sebagai daerah tidak tertinggal dan 10

daerah tidak tertinggal yang diklasifikasikan sebagai daerah tertinggal. Daftar daerah yang mengalami *misclassification* dari total 10 *fold* data *testing* dapat dilihat pada Lampiran.

Misclassification merupakan besarnya kesalahan klasifikasi pada hasil prediksi dibandingkan data aktual. Rata-rata *Misclassification* yang dihasilkan dari data *training* sebesar 5,77 persen, sedangkan pada data *testing* sedikit meningkat menjadi sebesar 6,52 persen.

F-Measure merupakan ukuran tunggal dari uji performa klasifikasi berdasarkan sensitivitas (*recall*) dan *precision*. Nilai *F-Measure* dari RBFNN sebesar 85,36 persen, artinya kinerja algoritma RBFNN dianggap baik (mendekati 100 persen semakin baik). Dalam sudut pandang metode RBFNN sebagai metode yang relatif akurat untuk pengklasifikasian daerah tertinggal di Indonesia, data *misclassification* yang dihasilkan dapat menjadi perhatian untuk dikaji lebih lanjut ataupun menjadi pertimbangan dalam penentuan kebijakan bagi *stakeholder* dan pihak yang terkait. Data hasil klasifikasi dengan metode RBFNN ini dapat menjadi pembanding untuk penentuan status daerah tertinggal pada periode mendatang.

KESIMPULAN DAN SARAN

Pada pengklasifikasian dengan RBFNN, dengan algoritma *k-means cluster* didapatkan jumlah *cluster* sebanyak 34 *cluster*, sehingga arsitektur yang terbentuk adalah RBFNN (27-34-2). Arsitektur ini tersusun oleh 27 *neuron* pada *input layer*,

34 neuron pada *hiddenn layer*, dan 2 neuron pada *output layer*. Dari segi ukuran arsitektur model, RBFNN cukup efisien dengan lebih sedikit neuron pada *hidden layer* (sesuai jumlah *cluster*). Rata-rata performa klasifikasi berdasarkan 10-fold *cross validation* yang diperoleh dengan RBFNN (27-34-2) yaitu akurasi 93,48 persen, sensitivitas 81,10 persen, spesifisitas 97,43 persen. Berdasarkan evaluasi performa klasifikasi dari arsitektur RBFNN dapat disimpulkan bahwa metode RBFNN memiliki keunggulan pada spesifisitas. Untuk performa klasifikasi secara keseluruhan, RBFNN menghasilkan nilai *F-Measure* sebesar 85,36 persen.

Untuk pengembangan penelitian, dapat dilakukan pengklasifikasian dengan menggunakan penggabungan (*hybrid*) model, misalnya model Radial Basis Probabilistic Neural Network (RBPNN) yang merupakan kombinasi dari arsitektur jaringan RBFNN dengan PNN. RBPNN memadukan keunggulan dari kedua model yang digabungkan sehingga diharapkan dapat meningkatkan performa klasifikasi yang dihasilkan. Metode RBFNN ini juga dapat diterapkan pada pemodelan klasifikasi dengan lebih banyak variabel dan cakupan data.

DAFTAR PUSTAKA

- Devega, Mariza. 2013. *Diagnosa Kerusakan Bantalan Gelinding Menggunakan Metode RBFNN*. Tesis. Semarang: Universitas Diponegoro.
- Fausett, L. 1994. *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*. Prentice-Hall, New Jersey, USA.
- Ghaderzadeh, M., Fein, R., Standing, A. 2013. *Early Detection of Cancer from Benign Hyperplasia of Prostate*. *Applied Medical Informatics* Vol. 33, No. 3, pp: 45-54
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining Concepts and Techniques Third Edition*. Waltham: Elsevier Inc.
- Haykin, S. 2008. *Neural Networks and Learning Machines*. New Jersey: Prentice Hall.
- Kotsiantis, S., Zaharakis, I., Pintelas, P. 2007. *Supervised machine learning: A review of classification techniques*. *Frontiers in Artificial Intelligence and Applications*.
- Kusumadewi, S. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta (ID): Graha Ilmu.
- Naibaho, E. 2016. *Perbandingan Back Propagation Neural Network dan Learning Vector Quantization*. Tesis. Bandung: Universitas Padjajaran
- Priddy, K.L., Keller, P.E. 2005. *Artificial Neural Network : An Introduction*. Washington: Spie Press
- Sitamahalakshmi, T., et al. 2011. *Performance Comparison of Radial Basis Function Networks and Probabilistic Neural Networks for Telugu Character Recognition*. *Global Journal of Computer Science and Technology Volume 11 Issue 4*
- Wahyuningrum, V. 2017. *Classification of Underdeveloped Regency using Probabilistic Neural Networks*. *Proceeding International Conference (ICAS), Vol.2 No 1 (2017)*
- Živković, Ž., Mihajlović, I., Nikolić, D. 2008. *Artificial Neural Network Method Applied on The Nonlinear Multivariate Problems*. University of Belgrade, Serbia

LAMPIRAN

Daftar Variabel *Output* (Y) dan Variabel *Input* (X)

Variabel	Kode	Nama Variabel
<i>Output</i> (Y)	Y	Status Daerah Y=1 Daerah Tertinggal Y=0 Daerah Tidak Tertinggal
<i>Input</i> (X)		<p>Kriteria Perekonomian Masyarakat</p> <p>X₁ Persentase penduduk miskin (BPS,Susenas) X₂ Pengeluaran konsumsi per kapita (BPS,Susenas)</p> <p>Kriteria Sumber Daya Manusia</p> <p>X₃ Angka harapan hidup (BPS,Susenas) X₄ Rata-rata lama sekolah (BPS,Susenas) X₅ Angka melek huruf (BPS,Susenas)</p> <p>Kriteria Infrastruktur</p> <p>X₆ Jumlah desa dengan jenis permukaan jalan terluas aspal/beton (BPS,Podes) X₇ Jumlah desa dengan jenis permukaan jalan terluas diperkeras (BPS,Podes) X₈ Jumlah desa dengan jenis permukaan jalan terluas tanah (BPS,Podes) X₉ Jumlah desa dengan jenis permukaan jalan terluas lainnya (BPS,Podes) X₁₀ Persentase rumah tangga pengguna listrik (BPS,Podes) X₁₁ Persentase rumah tangga pengguna telepon (BPS,Podes) X₁₂ Persentase rumah tangga pengguna air bersih (BPS,Podes) X₁₃ Jumlah desa yang memiliki pasar tanpa bangunan permanen (BPS,Podes) X₁₄ Jumlah prasarana kesehatan per 1000 penduduk (BPS,Susenas) X₁₅ Jumlah dokter per 1000 penduduk (BPS,Susenas) X₁₆ Jumlah SD dan SMP per 1000 penduduk (BPS,Susenas)</p> <p>Kriteria Kemampuan Keuangan Daerah</p> <p>X₁₇ Kemampuan keuangan daerah (Kemenkeu)</p> <p>Kriteria Aksesibilitas</p> <p>X₁₈ Rata-rata jarak dari kantor desa/kelurahan ke kantor kabupaten yang membawahi (BPS,Podes) X₁₉ Jumlah desa dengan akses ke pelayanan kesehatan >5 km (BPS,Podes) X₂₀ Jarak desa ke pelayanan pendidikan dasar (BPS,Podes)</p> <p>Kriteria Karakteristik Daerah</p> <p>X₂₁ Persentase desa gempa bumi (BPS,Podes) X₂₂ Persentase desa tanah longsor (BPS,Podes) X₂₃ Persentase desa banjir (BPS,Podes) X₂₄ Persentase desa bencana lainnya (BPS,Podes) X₂₅ Persentase desa di kawasan hutan lindung (BPS,Podes) X₂₆ Persentase desa berlahan kritis (BPS,Podes) X₂₇ Persentase desa konflik satu tahun terakhir (BPS,Podes)</p>

Hasil Eksplorasi Data: Uji Multikolinieritas dan *Output* Regresi Logistik

UJI MULTIKOLINIERITAS

VIF >5 artinya variabel mengandung masalah multikolinieritas

```
> data<-read.csv("dataNN.csv")
>
reg.log<-
glm(Y_2015~X1+X2+X3+X4+X5+X6+X7+X8+X9+X10+X11+X12+X13+X14+X15+X16+X17+
X18+X19+X20+X21+X22+X23+X24+X25+X26+X27,data=data)
> vif(reg.log)
X1      X2      X3      X4      X5      X6      X7      X8
2.831022 2.253291 1.562333 5.540024 5.655349 3.148160 1.517814 3.439646
X9      X10     X11     X12     X13     X14     X15     X16
2.274809 4.922362 2.752009 2.334973 1.463256 7.959401 3.319661 6.485056
X17     X18     X19     X20     X21     X22     X23     X24
1.577984 2.019335 5.429571 4.280363 1.194668 1.327243 1.241577 1.219430
X25     X26     X27
1.858747 1.489184 1.243659
```

REGRESI LOGISTIK

```
data<-read.csv("dataNN.csv")
kab <- data[,1]
data[,1] <- NULL
train2<-sample(nrow(data),round(0.9*nrow(data)))
data_train2<-data$Y_2015[train2]
data_test<-data$Y_2015[-train2]
fit<-glm(Y_2015~.,data=data[train2,],family="binomial")
logistics.train2<-fit$fitted.value ##prediction
mat.logis.train2<-table(round(data$Y_2015[train2]),round(logistics.train2,0)) #confusion matrix
acc.logis.train2<-sum(diag(mat.logis.train2))/sum(mat.logis.train2) #accuracy
logistics.test<-predict(fit,newdata=data[-train2,],type="response") ##prediction
mat.logis.test<-table(round(data$Y_2015[-train2]),round(logistics.test,0)) #confusion matrix
acc.logis.test<-sum(diag(mat.logis.test))/sum(mat.logis.test) #accuracy
result.acc.training2<-c(acc.logis.train2) ##akurasi training
result.acc.testing2<-c(acc.logis.test)
result<-rbind(result.acc.training2,result.acc.testing2)
colnames(result)<-c("Logistics Regression")
rownames(result)<-c("Training", "Testing")
result
```

```
> result
Logistics Regression
Training      1.0000000
Testing       0.8571429

> result
Logistics Regression
Training      0.9841629
Testing       0.8571429
```

Ringkasan Data Penelitian

No. Urut	Nama Kabupaten	Y	X ₁	X ₂	...	X ₂₇
1	SIMEULUE	0	20,57	628,09	...	0,00
2	ACEH SINGKIL	1	18,73	620,40	...	3,33
3	ACEH SELATAN	0	13,44	616,71	...	0,38
4	ACEH TENGGARA	0	14,39	609,76	...	4,16
5	ACEH TIMUR	0	16,59	599,27	...	0,58
.
.

490	DEIYAI	1	47,52	593,06	...	0,00
491	JAYAPURA	0	16,19	650,99	...	5,13

Sumber: BPS dan Kemenkeu, diolah

Ringkasan Data Hasil Standarisasi Variabel Input (X)

No. Urut	Nama Kabupaten	X ₁	X ₂	...	X ₂₇
1	SIMEULUE	0,41119	0,81892	...	0,00000
2	ACEH SINGKIL	0,37099	0,78362	...	0,08772
3	ACEH SELATAN	0,25541	0,76669	...	0,01012
4	ACEH TENGGARA	0,27616	0,73478	...	0,10936
5	ACEH TIMUR	0,32423	0,68663	...	0,01536
·	·	·	·	·	·
·	·	·	·	·	·
·	·	·	·	·	·
490	DEIYAI	1,00000	0,65813	...	0,00000
491	JAYAPURA	0,31549	0,92403	...	0,13495

Sumber: Hasil Pengolahan

Pembagian Data Berdasarkan 10-Fold cross validation

Sampel \ Fold	Sampel									
	1	2	3	4	5	6	7	8	9	10
Fold 1	50	50	50	50	50	50	50	50	50	50
Fold 2	49	49	49	49	49	49	49	49	49	49
Fold 3	49	49	49	49	49	49	49	49	49	49
Fold 4	49	49	49	49	49	49	49	49	49	49
Fold 5	49	49	49	49	49	49	49	49	49	49
Fold 6	49	49	49	49	49	49	49	49	49	49
Fold 7	49	49	49	49	49	49	49	49	49	49
Fold 8	49	49	49	49	49	49	49	49	49	49
Fold 9	49	49	49	49	49	49	49	49	49	49
Fold 10	49	49	49	49	49	49	49	49	49	49

Data Sampel	Data Training	Data Testing
1	441	50
2	442	49
3	442	49
4	442	49
5	442	49
6	442	49
7	442	49
8	442	49
9	442	49
10	442	49

116.

4. Data Training	5. Data Testing
------------------	-----------------

Rata-rata Akurasi Training dan Testing RBFNN Menurut Jumlah Cluster (k-means cluster)

Cluster (k)	Akurasi Training	Akurasi Testing	Cluster (k)	Akurasi Training	Akurasi Testing	Cluster (k)	Akurasi Training	Akurasi Testing
2	0,8769	0,8716	35	0,8857	0,8718	68	0,8753	0,8716
3	0,8798	0,8778	36	0,9118	0,9022	69	0,8731	0,8715
4	0,8798	0,8798	37	0,8753	0,8781	70	0,8106	0,8126
5	0,9052	0,8859	38	0,9228	0,9206	71	0,7812	0,7800
6	0,9122	0,9001	39	0,8502	0,8512	72	0,8812	0,8759
7	0,9054	0,8940	40	0,8891	0,8756	73	0,8656	0,8756
8	0,9188	0,9001	41	0,8905	0,8798	74	0,8855	0,8780
9	0,9224	0,9022	42	0,8473	0,8531	75	0,8667	0,8596
10	0,9269	0,9144	43	0,9265	0,9246	76	0,8570	0,8491

11	0,9305	0,9103
12	0,9314	0,9226
13	0,9319	0,9185
14	0,9353	0,9266
15	0,9231	0,9083
16	0,9131	0,8859
17	0,9242	0,8981
18	0,9389	0,9287
19	0,9104	0,9025
20	0,9246	0,9187
21	0,9226	0,9165
22	0,8327	0,8435
23	0,9233	0,9164
24	0,8774	0,8654
25	0,9437	0,9227
26	0,8912	0,8715
27	0,9124	0,8858
28	0,9201	0,9028
29	0,9269	0,9246
30	0,8744	0,8636
31	0,8375	0,8348
32	0,8853	0,8858
33	0,8405	0,8369
34	0,9423	0,9348

44	0,9076	0,9024
45	0,8545	0,8614
46	0,8989	0,9001
47	0,8430	0,8369
48	0,7930	0,7718
49	0,8724	0,8736
50	0,9421	0,9328
51	0,8701	0,8737
53	0,8436	0,8514
53	0,8749	0,8532
54	0,8651	0,8350
55	0,7826	0,7859
56	0,9219	0,9123
57	0,9063	0,8981
58	0,8291	0,8228
59	0,8484	0,8226
60	0,8803	0,8718
61	0,8156	0,8065
62	0,8321	0,8456
63	0,8599	0,8189
64	0,8762	0,8756
65	0,8359	0,8228
66	0,8197	0,8042
67	0,8394	0,8654

77	0,7991	0,7879
78	0,8167	0,8514
79	0,8078	0,8070
80	0,7576	0,7804
81	0,7743	0,7727
82	0,7703	0,7557
83	0,8821	0,8716
84	0,8368	0,8351
85	0,8131	0,8065
86	0,7713	0,7593
87	0,9310	0,9185
88	0,8201	0,8123
89	0,8719	0,8675
90	0,8312	0,8493
91	0,8635	0,8678
92	0,8448	0,8348
93	0,8352	0,7968
94	0,9088	0,8981
95	0,9265	0,9083
96	0,8418	0,8369
97	0,8525	0,8409
98	0,8565	0,8657
99	0,8726	0,8675
100	0,8715	0,8676

Sumber: Hasil Pengolahan

Confision matrix Hasil Prediksi RBFNN (27-34-2) Menurut 10-fold cross validation

Aktual		Prediksi			
		RBFNN (27-34-2)			
		Training		Testing	
		1	0	1	0
Fold 1	1	85	19	7	2
	0	9	328	1	40
Fold 2	1	90	16	6	1
	0	8	328	1	41
Fold 3	1	83	20	9	1
	0	7	332	2	37
Fold 4	1	85	16	11	1
	0	6	335	2	35
Fold 5	1	79	22	9	3
	0	7	334	0	37
Fold 6	1	79	21	12	1
	0	9	333	0	36
Fold 7	1	82	16	11	4
	0	8	336	0	34
Fold 8	1	87	16	7	3
	0	6	333	2	37
Fold 9	1	82	17	9	5
	0	8	335	0	35
Fold 10	1	84	18	10	1
	0	6	334	2	36
Total	1			91	22
	0			10	368

Sumber: Hasil Pengolahan

Performa Klasifikasi Data *Training* dan *Testing* dengan Arsitektur RBFNN (27-34-2)

Percobaan	Training					Testing			
	M	AK	SV	SP	Lama	M	AK	SV	SP
1	6,35	93,65	81,73	97,33	4,989	6,00	94,00	77,78	97,56
2	5,43	94,57	84,91	97,62	4,924	4,08	95,92	85,71	97,62
3	6,11	93,89	80,58	97,94	4,973	6,12	93,88	90,00	94,87
4	4,98	95,02	84,16	98,24	4,950	6,12	93,88	91,67	94,59
5	6,56	93,44	78,22	97,95	5,320	6,12	93,88	75,00	100,00
6	6,79	93,21	79,00	97,37	5,414	2,04	97,96	92,31	100,00
7	5,43	94,57	83,67	97,67	23,150	8,16	91,84	73,33	100,00
8	4,98	95,02	84,47	98,23	5,008	10,20	89,80	70,00	94,87
9	5,66	94,34	82,83	97,67	4,930	10,20	89,80	64,29	100,00
10	5,43	94,57	82,35	98,24	5,008	6,12	93,88	90,91	94,74
Rata-rata	5,77	94,23	82,19	97,82	6,867	6,52	93,48	81,10	97,43

Sumber: Hasil Pengolahan

Keterangan: M (%*Misclassification*); AK (%Akurasi); SV (%Sensitivitas); SP (%Spesifikasi); Lama (Waktu Komputasi dalam detik)

Daftar Daerah dengan Prediksi *Misclassification* per *Testing Fold* Menggunakan RBFNN (27-34-2)

No.	Testing Fold	Nama Daerah	Aktual	Prediksi
1	1	KEPULAUAN MERANTI	Tidak Tertinggal	Tertinggal
2	1	MUSI RAWAS	Tertinggal	Tidak Tertinggal
3	1	BOALEMO	Tertinggal	Tidak Tertinggal
4	2	SIMEULUE	Tidak Tertinggal	Tertinggal
5	2	LEBAK	Tertinggal	Tidak Tertinggal
6	3	HALMAHERA UTARA	Tidak Tertinggal	Tertinggal
7	3	BIAK NUMFOR	Tertinggal	Tidak Tertinggal
8	3	KATINGAN	Tidak Tertinggal	Tertinggal
9	4	MANOKWARI	Tidak Tertinggal	Tertinggal
10	4	KAPUAS HULU	Tertinggal	Tidak Tertinggal
11	4	MALUKU TENGGARA	Tidak Tertinggal	Tertinggal
12	5	TOLI-TOLI	Tertinggal	Tidak Tertinggal
13	5	BONDOWOSO	Tertinggal	Tidak Tertinggal
14	5	KEEROM	Tertinggal	Tidak Tertinggal
15	6	NIAS	Tertinggal	Tidak Tertinggal
16	7	SITUBONDO	Tertinggal	Tidak Tertinggal
17	7	HULU SUNGAI UTARA	Tertinggal	Tidak Tertinggal
18	7	BENGKAYANG	Tertinggal	Tidak Tertinggal
19	7	POHUWATO	Tertinggal	Tidak Tertinggal
20	8	NIAS BARAT	Tertinggal	Tidak Tertinggal
21	8	POSO	Tidak Tertinggal	Tertinggal
22	8	NUNUKAN	Tertinggal	Tidak Tertinggal
23	8	SELUMA	Tertinggal	Tidak Tertinggal
24	8	HALMAHERA TENGAH	Tidak Tertinggal	Tertinggal
25	9	PANDEGLANG	Tertinggal	Tidak Tertinggal
26	9	KONAWA	Tertinggal	Tidak Tertinggal
27	9	LAMPUNG BARAT	Tertinggal	Tidak Tertinggal
28	9	BOMBANA	Tertinggal	Tidak Tertinggal
29	9	PASAMAN BARAT	Tertinggal	Tidak Tertinggal

30	10	KOLAKA UTARA	Tidak Tertinggal	Tertinggal
31	10	PROBOLINGGO	Tidak Tertinggal	Tertinggal
32	10	NABIRE	Tertinggal	Tidak Tertinggal

Sumber: Hasil Pengolahan

