

**PERAMALAN BANYAK KASUS DEMAM BERDARAH DI D.I.
YOGYAKARTA DENGAN MODEL
*RADIAL BASIS FUNCTION NEURAL NETWORK***

SKRIPSI

Diajukan kepada Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Negeri Yogyakarta
untuk Memenuhi Sebagian Persyaratan guna Memperoleh Gelar Sarjana Sains



Oleh:
Fajarani Juliaristi
NIM. 10305141008

**PROGRAM STUDI MATEMATIKA
JURUSAN PENDIDIKAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI YOGYAKARTA
2014**

PERSETUJUAN

Skripsi yang berjudul “**PERAMALAN BANYAK KASUS DEMAM BERDARAH DI D.I. YOGYAKARTA DENGAN MODEL *RADIAL BASIS FUNCTION NEURAL NETWORK***” ini telah disetujui oleh dosen pembimbing untuk diujikan.



Oleh:

Fajarani Juliaristi

NIM. 10305141008

Telah disetujui pada tanggal 11 Juni 2014

untuk diujikan di hadapan Dewan Penguji Skripsi

Program Studi Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Negeri Yogyakarta

Yogyakarta, 11 Juni 2014
Dosen Pembimbing



Dr. Dhoriva Urwatul Wutsqa
NIP. 19660331 199303 2 001

HALAMAN PENGESAHAN

SKRIPSI DENGAN JUDUL :

**“PERAMALAN BANYAK KASUS DEMAM BERDARAH DI D.I.
YOGYAKARTA DENGAN MODEL
RADIAL BASIS FUNCTION NEURAL NETWORK”**

Yang disusun oleh :

Nama : Fajarani Juliaristi

NIM : 10305141008

Prodi : Matematika

Skripsi ini telah diuji di depan Dewan Penguji Skripsi pada tanggal 20 Juni 2014 dan dinyatakan **LULUS**.

Dewan Penguji

Nama	Jabatan	Tanda Tangan	Tanggal
<u>Dr. Dhoriva U. W.</u> 196603311993032001	Ketua Penguji		30-6-2014
<u>Rosita Kusuwati, M.Sc.</u> 198007072005012001	Sekretaris Penguji		30-6-2014
<u>Dr. Agus Maman A.</u> 197008281995021001	Penguji Utama		23-6-2014
<u>Elly Arliani, M.Si.</u> 196708161992032001	Penguji Pendamping		27-6-2014

Yogyakarta, 1 Juli 2014
Fakultas Matematika dan
Ilmu Pengetahuan Alam
Dekan



Dr. Hartono
NIP. 196203291987021002

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini, saya :

Nama : Fajarani Juliaristi

NIM : 10305141008

Program Studi : Matematika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

Judul Skripsi : Peramalan Banyak Kasus Demam Berdarah di D.I. Yogyakarta dengan Model *Radial Basis Function Neural Network*

Menyatakan bahwa skripsi ini merupakan hasil kerja sendiri dan sepanjang pengetahuan saya tidak berisi materi yang dipublikasikan atau dipergunakan sebagai persyaratan penyelesaian studi di perguruan tinggi kecuali pada bagian-bagian tertentu yang saya ambil sebagai acuan atau kutipan dengan mengikuti tata penulisan karya ilmiah yang telah lazim. Apabila terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya dan saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Yogyakarta, 11 Juni 2014

Yang Menyatakan,



Fajarani Juliaristi
NIM. 10305141008

MOTTO

"Sesungguhnya sesudah kesulitan itu pasti ada kemudahan. Maka apabila telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain".

(Q.S Al-Insyirah: 6-7)

"Sesuatu mungkin mendatangi mereka yang mau menunggu, namun hanya didapatkan oleh mereka yang bersemangat mengejarnya".

(Abraham Lincoln)

"...pintu kebahagiaan terbesar adalah doa kedua orang tua maka berusahalah mendapatkan doa itu dengan berbakti kepada mereka agar doa mereka menjadi benteng yang kuat untuk menjagamu dari segala hal yang tidak kita sukai..."

(DR. 'Aidh al-Qarni)

HALAMAN PERSEMBAHAN

Alhamdulillahirabbil 'alamin, segala puji bagi Allah sang penguasa langit dan bumi dan yang ada di antara keduanya yang senantiasa melindungi, memberi kesabaran dan ketekhlisan sehingga skripsi ini selesai disusun.

Skripsi ini penulis persembahkan untuk:

Bapak dan Ibu tercinta yang tak pernah lelah memberikan kasih sayang, nasihat, dukungan, dan semangat. Terimakasih untuk doa yang tak pernah putus siang dan malam dari bibir dan hati Bapak Ibu. semoga karya kecil ini dapat menjadi salah satu bentuk bakti penulis kepada Bapak Ibu tercinta.

Tika, adikku tersayang terimakasih atas kasih sayang, persaudaraan dan doa, serta semangat yang tanpa tika sadari telah banyak mendorong penulis untuk tidak mudah menyerah.

**PERAMALAN BANYAK KASUS DEMAM BERDARAH DI D.I.
YOGYAKARTA DENGAN MODEL RADIAL BASIS FUNCTION NEURAL
NETWORK**

Oleh:
Fajarani Juliaristi
NIM. 10305141008

ABSTRAK

Radial Basis Function Neural Network (RBFNN) adalah salah satu model *Neural Network (NN)* yang digunakan untuk menyelesaikan permasalahan nonlinear. Desain dari *RBFNN* adalah model *NN* yang mentransformasikan *input* secara nonlinear dengan menggunakan fungsi aktivasi Gaussian pada lapisan tersembunyi sebelum diproses secara linear di lapisan *output*. Penulisan skripsi ini bertujuan untuk menjelaskan bagaimana proses pembentukan model *Radial Basis Neural Network (RBFNN)* pada deret berkala dan bagaimana hasil peramalan *RBFNN* untuk banyak kasus demam berdarah di DIY.

Prosedur pembentukan model *RBFNN* yang pertama adalah menentukan nilai pusat dan varians dari variabel *input* pada setiap *neuron* pada lapisan tersembunyi dengan metode *K-mean cluster*. Kedua menentukan banyaknya *neuron input* dengan melihat plot fungsi autokorelasi (*ACF*). Kemudian membagi data menjadi data *training* dan data *testing*. Model terbaik diperoleh dengan menentukan banyaknya *neuron* pada lapisan tersembunyi dan *neuron input* secara *trial and error* dengan memperhatikan nilai *Means Absolute Percent Error (MAPE)* dan *Mean Squared Error (MSE)*. Banyak *neuron* ditentukan dengan banyaknya kluster atau kelompok yang terbentuk. Selanjutnya dengan menggunakan metode Kuadrat Terkecil (*Least Square*) diperoleh nilai-nilai bobot. Setelah model terbaik terbentuk, dilakukan pengecekan *white noise* pada residual hasil pembelajaran data *training*.

Aplikasi model *RBFNN* diterapkan untuk meramalkan banyak kasus demam berdarah di DIY. Data yang digunakan adalah data banyak kasus demam berdarah di DIY dari bulan Januari 2005 sampai Desember 2012. Model ini menghasilkan *MAPE* dan *MSE training* masing-masing 0,4919% dan 5.144, *MAPE* dan *MSE testing* adalah 0,7886% dan 11.348. Berdasarkan model yang telah terbentuk, diperoleh hasil peramalan untuk bulan Januari 2013 sebanyak 337 kasus. Hasil peramalan dan model ini digunakan untuk meramalkan banyak kasus demam berdarah untuk 6 bulan berikutnya.

Kata Kunci: *Radial Basis Function Neural Network (RBFNN)*, peramalan, demam berdarah

KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT atas limpahan rahmat, karunia, dan hidayah-Nya sehingga penulis mampu menyelesaikan penulisan Skripsi dengan judul **“Peramalan Banyak Kasus Demam Berdarah di D.I. Yogyakarta dengan Model *Radial Basis Function Neural Network*”**. Skripsi ini disusun untuk memenuhi persyaratan guna memperoleh gelar Sarjana Sains Program Studi Matematika di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Negeri Yogyakarta.

Penulisan skripsi ini tidak lepas dari dukungan, motivasi, kerjasama maupun bimbingan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Bapak Dr. Hartono, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah memberikan kesempatan kepada penulis untuk menyelesaikan studi.
2. Bapak Dr. Sugiman, selaku Ketua Jurusan Pendidikan Matematika yang telah memberikan kelancaran dalam pelayanan akademik.
3. Bapak Dr. Agus Maman Abadi, selaku Ketua Program Studi Matematika, Penguji utama sekaligus Penasehat Akademik yang telah memberikan bimbingan dan pengarahan dalam penulisan skripsi ini, serta dukungan akademik kepada penulis.
4. Ibu Dr. Dhoriva U. W., selaku dosen pembimbing yang telah sabar memberikan bimbingan serta masukan dalam penyusunan skripsi ini.

5. Ibu Elly Arliani, M.Si, selaku penguji pendamping dan Ibu Rosita Kusumawati, M.Sc, selaku sekretaris penguji yang telah memberikan saran dan masukan untuk perbaikan skripsi ini.
6. Seluruh Dosen Jurusan Pendidikan Matematika FMIPA UNY yang telah memberikan ilmu yang bermanfaat kepada penulis.
7. Orangtua dan keluarga yang telah memberikan doa, semangat dan dukungan yang tiada henti kepada penulis.
8. Sahabat-sahabat Noviana, Uswah, Gity, Reni, Ama, Laras, Yunin yang telah banyak membantu dan selalu memberi semangat. Terimakasih untuk persahabatan, doa, motivasi dan dukungan kalian.
9. Seluruh Mahasiswa Matematika Subsidi 2010 yang telah memberi banyak dukungan dan pelajaran berharga selama ini.
10. Semua pihak yang tidak dapat penulis sebutkan satu-persatu yang telah membantu dan memberikan dukungan dalam penulisan skripsi ini.

Penulis menyadari bahwa masih banyak kekurangan dan kesalahan dalam skripsi ini. Oleh karena itu, saran dan kritik yang membangun penulis harapkan sebagai sebuah koreksi. Demikian skripsi ini penulis susun. Semoga skripsi ini dapat memberikan manfaat bagi penulis dan pembaca.

Yogyakarta, 11 Juni 2014

Penulis,
Fajarani Juliaristi

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PERSETUJUAN	ii
HALAMAN PENGESAHAN	iii
SURAT PERNYATAAN	iv
MOTTO	v
PERSEMBAHAN	vi
ABSTRAK	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	
A. Latar Belakang Masalah	1
B. Rumusan Masalah.....	5
C. Tujuan Penelitian	5
D. Manfaat Penelitian	6
BAB II LANDASAN TEORI	
A. Demam Berdarah	7
1. Pengertian Demam Berdarah	7
2. Diagnosa Klinis	8

3. Fase Demam Berdarah	9
B. Peramalan	10
C. Analisa Deret Berkala (<i>Time series</i>)	11
D. Stasioneritas	14
E. Autokorelasi	15
F. Proses <i>White Noise</i>	20
G. Kriteria Model Terbaik	21
H. Jaringan Saraf	22
1. Jaringan Saraf Biologi.....	22
2. <i>Neural Network (NN)</i>	24
3. Kerangka <i>Neural Network</i>	25
4. Arsitektur <i>Neural Network</i>	26
5. Fungsi Aktivasi.....	28
6. Metode Pembelajaran dan Pelatihan	28

BAB III PEMBAHASAN

A. <i>Radial Basis Function Neural Network (RBFNN)</i>	30
B. <i>K-means</i> Kluster.....	32
C. Metode Kuadrat Terkecil (<i>Least Square</i>)	33
D. Metode <i>Global Ridge-Regression</i>	36
E. Pembentukan Model <i>RBFNN</i> pada Data Deret Berkala dengan Pemrograman <i>MATLAB</i>	38
F. Prosedur Pembentukan Model <i>RBFNN</i>	39
G. Aplikasi <i>RBFNN</i> untuk meramalkan Banyak Kasus Demam	

Berdarah di DIY	42
-----------------------	----

BAB IV PENUTUP

A. Kesimpulan	56
---------------------	----

B. Saran.....	58
---------------	----

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar 2.1. Contoh Pola Gerakan Horizontal	12
Gambar 2.2. Contoh Pola Gerakan <i>Trend</i>	12
Gambar 2.3. Contoh Pola Gerakan Siklis	13
Gambar 2.4. Contoh Pola Gerakan Musiman	14
Gambar 2.5. Plot Fungsi Autokorelasi dari Data Penjualan	17
Gambar 2.6. Plot Fungsi Autokorelasi Parsial dari Data Penjualan	20
Gambar 2.7. Jaringan Saraf Biologi	24
Gambar 2.8. Arsitektur Jaringan Lapisan Tunggal	26
Gambar 2.9. Arsitektur Jaringan Banyak Lapisan	27
Gambar 2.10. Arsitektur Jaringan Lapisan Kompetitif	28
Gambar 2.11. Fungsi Aktivasi: Fungsi Linear	28
Gambar 3.1. Model <i>RBFFNN</i>	31
Gambar 3.2. Prosedur Pembentukan Model <i>RBFFNN</i>	41
Gambar 3.3. Plot data banyak kasus DB di DIY dari Tahun 2005 sampai 2012	42
Gambar 3.3. Plot <i>ACF</i> data banyak kasus DB di DIY dari bulan Januari 2005 sampai bulan Desember 2012	43
Gambar 3.5. Plot <i>ACF</i> residual data <i>training</i>	47
Gambar 3.6. Plot <i>PACF</i> residual data <i>training</i>	47
Gambar 3.7. Plot Data Aktual dan Hasil Pembelajaran pada Data <i>Training</i>	48
Gambar 3.8. Plot Data Aktual dan Hasil Pembelajaran pada Data <i>Testing</i>	48
Gambar 3.9. Arsitektur <i>RBFFNN</i>	49
Gambar 3.10. Grafik Hasil Peramalan Banyak Kasus Demam Berdarah	54

DAFTAR TABEL

Tabel 3.1. Nilai <i>MAPE</i> dan <i>MSE</i> penentuan banyak <i>neuron</i> tersembunyi untuk data <i>training</i>	44
Tabel 3.2. Nilai <i>MAPE</i> dan <i>MSE</i> penentuan banyak <i>neuron</i> tersembunyi untuk data <i>testing</i>	45
Tabel 3.3. Nilai <i>MAPE</i> dan <i>MSE</i> hasil eliminasi <i>input</i> data <i>training</i>	46
Tabel 3.4. Nilai <i>MAPE</i> dan <i>MSE</i> hasil eliminasi <i>input</i> data <i>testing</i>	46
Tabel 3.5. Hasil Peramalan	54
Tabel 4.1. Nilai pusat dan varians <i>neuron</i> tersembunyi.....	80
Tabel 4.2. Perhitungan peramalan banyak kasus demam berdarah	80

DAFTAR LAMPIRAN

Lampiran 1. Data Banyak Kasus Demam Berdarah di D.I. Yogyakarta	63
Lampiran 2. Program rbfDesign	64
Lampiran 3. Program globalRidge	69
Lampiran 4. Program <i>RBFNN</i> untuk peramalan banyak kasus demam berdarah dengan 8 <i>neuron</i> dan 3 <i>input</i>	78
Lampiran 5. Perhitungan peramalan banyak kasus demam berdarah menggunakan <i>Ms. Excel</i>	80

BAB 1

PENDAHULUAN

A. Latar Belakang Masalah

Penyakit menular adalah penyakit yang menakutkan bagi sebagian besar masyarakat di berbagai belahan dunia. Salah satu penyakit menular yang menjadi perhatian adalah penyakit Demam Berdarah (DB). Demam berdarah sering menjadi penyebab terjadinya Kejadian Luar Biasa (KLB) maupun wabah di Indonesia. KLB dijelaskan sebagai timbulnya atau meningkatnya kejadian kesakitan atau kematian secara epidemiologis pada suatu daerah dalam kurun waktu tertentu. Demam berdarah adalah penyakit demam akut yang disebabkan oleh virus *dengue*, yang masuk ke peredaran darah manusia melalui gigitan nyamuk dari genus *Aedes*, misalnya *Aedes aegypty* dan *Aedes albopictus*. *Aedes aegypty* adalah nyamuk pembawa virus *dengue* yang paling banyak ditemukan menyebabkan penyakit demam berdarah (Depkes RI, 2012: 96).

Penyakit endemis demam berdarah mulai berjangkit di Indonesia pada tahun 1968 di Surabaya, dimana sebanyak 58 orang terinfeksi dan 24 orang diantaranya meninggal dunia (Indra Chahaya, 2003: 1). Demam berdarah banyak ditemukan di daerah tropis dan sub-tropis Asia Tenggara dan Wilayah Pasifik Barat. Data dari seluruh dunia menunjukkan Asia menempati urutan pertama dalam jumlah penderita demam berdarah setiap tahunnya. Terhitung sejak tahun 1968 hingga tahun 2009, *World Health*

Organization (WHO) mencatat Indonesia sebagai negara dengan kasus demam berdarah tertinggi di Asia Tenggara (Depkes RI, 2010: 1).

Daerah Istimewa Yogyakarta (DIY) adalah salah satu Provinsi yang dinyatakan sebagai daerah endemis demam berdarah. Pada tahun 2011, jumlah penderita demam berdarah sebanyak 982 kasus dengan jumlah kematian 5 orang. Sedangkan untuk tahun 2012 kasus demam berdarah mengalami penurunan menjadi 970 kasus (Dinkes Prov. DIY, 2012: 31).

Ada banyak faktor yang berkontribusi menyebabkan penyakit, begitu juga dengan penyakit demam berdarah. Faktor-faktor tersebut berasal dari individu sendiri maupun dari lingkungan. Menurut Cut I.N. Sari (2005), beberapa faktor yang terkait dalam penularan demam berdarah antara lain kepadatan penduduk, mobilitas penduduk, kualitas perumahan dan sikap hidup. Sedangkan faktor yang dapat memicu terjadinya demam berdarah adalah faktor lingkungan yang termasuk di dalamnya perubahan suhu, kelembaban udara, dan curah hujan yang mengakibatkan nyamuk lebih sering bertelur dan virus *dengue* berkembang biak dengan cepat. Parasit dan pembawa penyakit (nyamuk) sangat peka terhadap faktor iklim, khususnya suhu, curah hujan, kelembaban, permukaan air, dan angin (Marianne & Jonathan, 2001: 441-463).

Dalam rangka mengantisipasi kenaikan banyak kasus demam berdarah, Menteri Kesehatan, Ditjen PPM & PLP, Gubernur DIY, maupun

Kepala Dinas Kesehatan Provinsi DIY telah mengeluarkan berbagai peraturan dan kebijakan. Salah satu peraturannya adalah pelaksanaan Pemberantasan Sarang Nyamuk (PSN) melalui pemberdayaan masyarakat yang dikenal dengan pemberantasan 3M (Mengubur, Menutup, dan Menguras). Tetapi berbagai upaya yang telah dilakukan belum membuahkan hasil yang optimal dan kurang memuaskan sehingga penanganan kasus masih saja terlambat.

Data demam berdarah merupakan data berkala, yaitu data yang disajikan dalam kurun waktu tertentu. Data berkala erat kaitannya dengan peramalan atau prediksi. Salah satu upaya penanganan kasus demam berdarah adalah dengan memprediksi angka perkembangan penyakit pada waktu yang akan datang. Prediksi atau peramalan ini dapat membantu mengoptimalkan upaya pencegahan sejak dini agar keterlambatan penanganan tidak lagi terjadi.

Proses peramalan dapat dilakukan menggunakan banyak metode, antara lain metode pemulusan (*smoothing*), dekomposisi, regresi, *ARIMA Box Jenkins*, dan sebagainya. Dalam perkembangan selanjutnya berkembang metode baru menggunakan sistem Jaringan Saraf Tiruan (JST) atau *Neural Network (NN)*.

Neural Network (NN) adalah sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan saraf biologis (Fausett, 1994: 3). Model *NN* efektif digunakan untuk peramalan pada model nonlinear. Metode peramalan *NN* terus dikembangkan dari tahun 1940-an sampai

sekitar tahun 1988 dikembangkan fungsi radial basis. Salah satu metode peramalan yang tergolong dalam model *NN* adalah model *Radial Basis Function Neural Network (RBFNN)*.

Model *RBFNN* terdiri dari lapisan masukan (*input*), lapisan tersembunyi (*hidden*) dan lapisan keluaran (*output*). *RBFNN* pertama kali digunakan untuk memecahkan permasalahan interpolasi multivariabel (Yildiray & Ozan, 2013). *RBFNN* hanya memiliki bobot pada jaringan yang terhubung dari lapisan *hidden* ke lapisan *output*, terdapat fungsi aktivasi pada lapisan *hidden* dan mengeluarkan nilai berupa persamaan nonlinear, sedangkan pada lapisan *output* atau akhir proses *RBFNN* mengeluarkan nilai berupa persamaan linear (Palit dan Popavic, 2005: 86-87). *RBFNN* telah banyak digunakan dalam berbagai penelitian, misalnya Ferry Tan dkk (2012) memprediksi harga saham menggunakan jaringan *RBF* dengan metode pembelajaran *hybrid*. Pada tahun 2012, Indrabayu dkk memprediksi curah hujan menggunakan metode *RBFNN* dan Sri Utami Zuliana (2012) menggunakan *RBFNN* untuk meramalkan nilai tukar US dolar terhadap rupiah.

Pada skripsi ini, akan dilakukan peramalan banyak kasus demam berdarah di DIY menggunakan metode *RBFNN* dimana *inputnya* adalah banyak kasus demam berdarah di DIY. Penelitian untuk memprediksi banyak kasus demam berdarah di Provinsi DIY pernah dilakukan menggunakan *Backpropagation Neural Network (BPNN)* (Shinta Prajna, 2013). *BPNN* merupakan salah satu algoritma dalam *NN* yang sering

digunakan selain *Perceptron*. Ciri khas yang membedakan *RBFNN* dengan algoritma *NN* yang lain adalah *RBFNN* hanya memiliki satu lapisan tersembunyi, sedangkan algoritma yang lain dapat memiliki satu atau lebih lapisan tersembunyi. Fungsi aktivasi yang sering digunakan pada algoritma *NN* yang lain adalah fungsi *sigmoid*, sedangkan pada *RBFNN* menggunakan fungsi basis pada lapisan tersembunyi dan fungsi linear pada lapisan *output*. Ciri khas *RBFNN* ini menyebabkan proses kerja *RBFNN* lebih cepat dari algoritma *NN* yang lain.

Uraian latar belakang di atas merupakan alasan penulis tertarik membahas *RBFNN*. Untuk mengetahui teknik peramalan dengan menggunakan model *RBFNN* yang diterapkan dalam peramalan kasus demam berdarah di DIY, skripsi ini diberi judul “Peramalan Banyak Kasus Demam Berdarah di D.I. Yogyakarta dengan Model *Radial Basis Function Neural Network*”.

B. Rumusan Masalah

Berdasarkan uraian latar belakang permasalahan di atas, maka diperoleh rumusan masalah sebagai berikut:

1. Bagaimana proses pembentukan model *Radial Basis Function Neural Network (RBFNN)* pada deret berkala?
2. Bagaimana hasil peramalan banyak kasus demam berdarah di DIY dengan Model *Radial Basis Function Neural Network (RBFNN)*?

C. Tujuan Penelitian

Tujuan penelitian ini adalah:

1. Menjelaskan proses pembentukan model *Radial Basis Function Neural Network (RBFNN)* pada deret berkala.
2. Meramalkan banyak kasus demam berdarah di D.I Yogyakarta menggunakan *Radial Basis Function Neural Network (RBFNN)*.

D. Manfaat Penelitian

Manfaat penelitian ini adalah:

1. Bagi penulis
Menambah dan memperkaya pengetahuan mengenai model *Radial Basis Function Neural Network (RBFNN)* serta penerapannya pada peramalan data deret berkala.
2. Bagi Mahasiswa Matematika
 - a. Menambah pengetahuan mengenai model *Radial Basis Function Neural Network (RBFNN)*
 - b. Memberikan suatu metode alternatif untuk melakukan peramalan menggunakan model *Radial Basis Function Neural Network (RBFNN)*.
3. Bagi Dinas Kesehatan
Peramalan banyak kasus demam berdarah di DIY dapat membantu pemerintah dan dinas terkait dalam mengantisipasi dan mengambil tindakan untuk perkembangan penyakit demam berdarah.

BAB II

KAJIAN TEORI

Bab ini berisi penjelasan mengenai demam berdarah, pengertian peramalan, analisa deret berkala, stasioneritas, autokorelasi, proses *white noise*, kriteria model terbaik, jaringan saraf biologi, *neural network*, kerangka *neural network*, arsitektur *neural network*, fungsi aktivasi, dan metode pembelajaran.

A. Demam Berdarah

1. Pengertian Demam Berdarah

Demam berdarah (DB) adalah penyakit demam akut yang disebabkan oleh virus *dengue* yang masuk ke peredaran darah manusia melalui gigitan nyamuk dari genus *Aedes*, misalnya *Aedes aegypty* dan *Aedes albopictus*. *Aedes aegypty* adalah nyamuk pembawa virus *dengue* yang paling banyak ditemukan menyebabkan penyakit demam berdarah (Depkes RI, 2012: 96). Nyamuk *Aedes aegypty* berasal dari Brasil dan Ethiopia. Nyamuk ini sering menggigit manusia pada waktu pagi dan siang. Berbeda dengan nyamuk *Aedes albopictus* yang tinggal di kebun atau pekarangan rumah, nyamuk *Aedes aegypty* lebih suka tinggal di dalam rumah, di tempat gelap dan lembab seperti di bawah tempat tidur, kloset, kamar mandi, atau dapur.

Seseorang yang terkena penyakit demam berdarah ditandai dengan demam mendadak 2 sampai 7 hari tanpa penyebab yang jelas, lemah atau lesu, gelisah, nyeri ulu hati, disertai tanda pendarahan di kulit berupa

bintik pendarahan (*petechie*), lebam (*echymosis*), ruam (*purpura*), kadang-kadang mimisan, berak darah, muntah darah, kesadaran menurun atau renjatan (*shock*) (Indrawan, 2001: 102).

2. Diagnosa Klinis

Berdasarkan jenis gejala yang ditimbulkan, infeksi virus *dengue* dikelompokkan menjadi 3, yaitu Demam *Dengue* (DD), Demam Berdarah *Dengue* (DBD), dan *Dengue Shock Syndrome* (DSS) (Depkes RI, 2010: 22).

a. Demam *Dengue* (DD)

Demam *Dengue* (DD) atau *Dengue Fever* (DF) memberikan gejala infeksi yang berbeda pada golongan umur tertentu. Gejala pada bayi adalah demam yang disertai munculnya ruam. Pada orang dewasa, gejalanya adalah demam tinggi, sakit kepala berat, nyeri di belakang mata, nyeri otot dan tulang atau sendi, mual dan muntah, dan muncul ruam. Penyakit ini disertai dengan menurunnya sel darah putih (*leukosit*) dan keping darah (*trombosit*) (Anton Sitio, 2008: 11).

b. Demam Berdarah *Dengue* (DBD)

Kasus Demam Berdarah *Dengue* (DBD) atau *Dengue Hemoragic Fever* (DHF) menimbulkan gejala yang hampir sama dengan gejala Demam *Dengue* (DD). Namun pada kasus ini terjadi pendarahan hebat, pembesaran hati > 2 cm, dan kenaikan *hematokrit* dengan penurunan jumlah *trombosit* yang cepat.

c. *Dengue Shock Syndrome (DSS)*

Kasus ini terjadi jika seseorang terserang virus *dengue* untuk yang kedua kalinya. Gejalanya adalah nadi berdenyut cepat, kulit dingin dan lembab, gelisah, dan terjadi kebocoran cairan di luar pembuluh darah. *DSS* merupakan infeksi virus terparah yang dapat mengakibatkan kematian.

3. Fase Demam Berdarah (DB)

Setelah terinfeksi virus *dengue*, penderita demam berdarah akan mengalami 3 fase yaitu fase *febris*, fase kritis, dan fase pemulihan (Depkes RI, 2010: 22).

a. Fase *Febris*

Pada fase ini, demam mendadak tinggi selama 2 sampai 7 hari disertai muka kemerahan, nyeri seluruh tubuh, *eritema* kulit, dan sakit kepala. Pada beberapa kasus ditemukan nyeri tenggorok, injeksi *faring* dan *konjungtiva*, *anoreksia*, mual dan muntah.

b. Fase Kritis

Fase ini terjadi pada hari sakit ke 3 sampai 7 yang ditandai dengan penurunan suhu tubuh, kenaikan *permeabilitas kapiler* (kerusakan pada pembuluh darah) dan timbulnya kebocoran *plasma* yang biasanya berlangsung selama 24 sampai 48 jam. Pada fase ini dapat terjadi *shock*.

c. Fase Pemulihan

Setelah fase kritis terlewati, terjadi pengembalian cairan secara perlahan pada 48 sampai 72 jam setelahnya. Keadaan umum penderita membaik.

B. Peramalan

Peramalan adalah suatu teknik untuk meramalkan keadaan di masa yang akan datang melalui pengujian keadaan di masa lalu. Pada dasarnya meramalkan sama halnya dengan memprediksi atau memperkirakan suatu hal, kejadian atau peristiwa masa datang yang berdasar pada masa lalu hingga saat ini.

Berdasarkan periode waktunya, peramalan diklasifikasikan menjadi 3 bentuk (Montgomery, Jennings & Kulahci, 2008: 1):

1. Jangka Pendek (*Short Term*)

Jangka pendek meliputi kurun waktu mulai dari hari, minggu sampai bulan. Karena peramalannya sangat singkat, maka data historis terdahulu masih relevan untuk meramalkan masa datang. Contoh peramalan dalam jangka pendek adalah perkiraan penjualan produk seperti mobil atau baja.

2. Jangka Menengah (*Medium Term*)

Jangka menengah meliputi kurun waktu mulai dari satu musim (triwulan, kuartal atau yang lain) sampai dua tahun. Data historis masa lalu masih dianggap relevan untuk meramalkan masa datang. Contoh peramalan dalam jangka menengah adalah meramalkan anggaran penjualan atau produksi.

3. Jangka Panjang (*Long Term*)

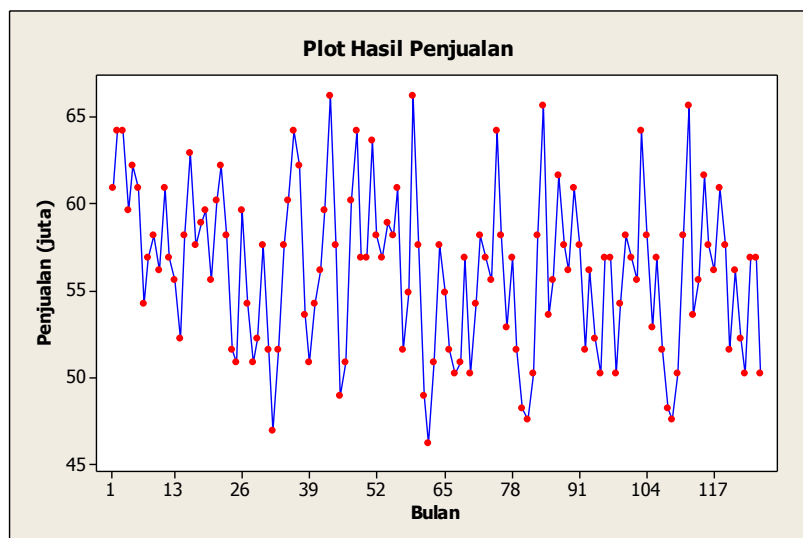
Jangka panjang meliputi kurun waktu lebih dari dua tahun. Peramalan untuk jangka panjang pada umumnya dilakukan berdasarkan intuisi dan pengalaman seseorang, tapi banyak juga perusahaan yang menggunakan data historis.

C. Analisa Deret Berkala (*Time Series*)

Deret berkala (*time series*) adalah serangkaian pengamatan terhadap suatu peristiwa, kejadian, gejala atau peubah yang diambil dari waktu ke waktu, dicatat secara teliti menurut urutan waktu terjadinya dan kemudian disusun sebagai data statistik (Hanke & Wichern, 2005: 58). Deret berkala digunakan untuk memperoleh gambaran dari keadaan atau sifat variabel di waktu yang lalu untuk peramalan dari nilai variabel itu pada periode yang akan datang. Adapun komponen-komponen deret berkala yang harus diperhatikan adalah (Hanke & Wichern, 2005: 58-59):

1. Gerakan Horizontal

Gerakan horizontal adalah suatu pergerakan data yang berfluktuasi di sekitar nilai konstan atau rata-rata yang membentuk garis horizontal. Data ini juga disebut data stasioner. Contoh gerakan horizontal dapat dilihat pada Gambar 2.1. yaitu plot suatu penjualan.

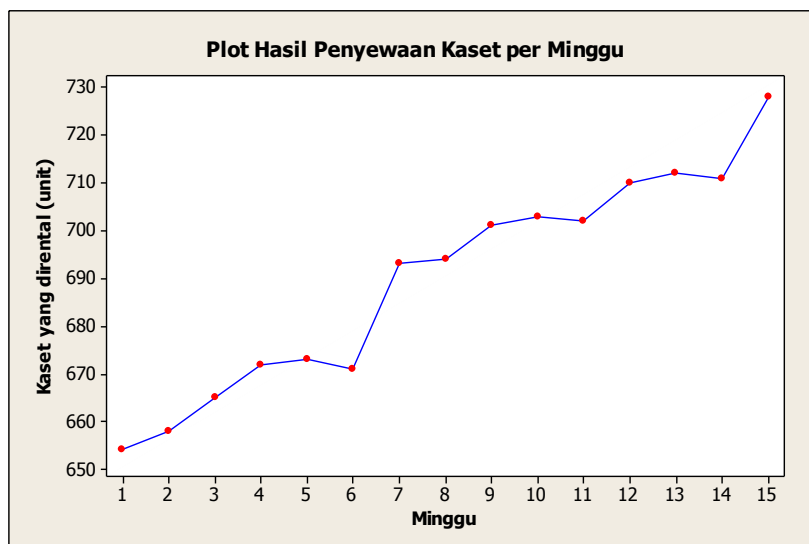


(Sumber: Hanke & Wichern, 2005: 434)

Gambar 2.1. Contoh Pola Gerakan Horizontal

2. Gerakan *Trend*

Pola gerakan ini adalah jika suatu data bergerak pada jangka waktu tertentu dan cenderung menuju ke satu arah baik naik atau turun. Contoh pola gerakan *trend* dapat dilihat pada Gambar 2.2. yang merupakan plot data hasil penyewaan kaset film sebuah perusahaan di Denver, Colorado.

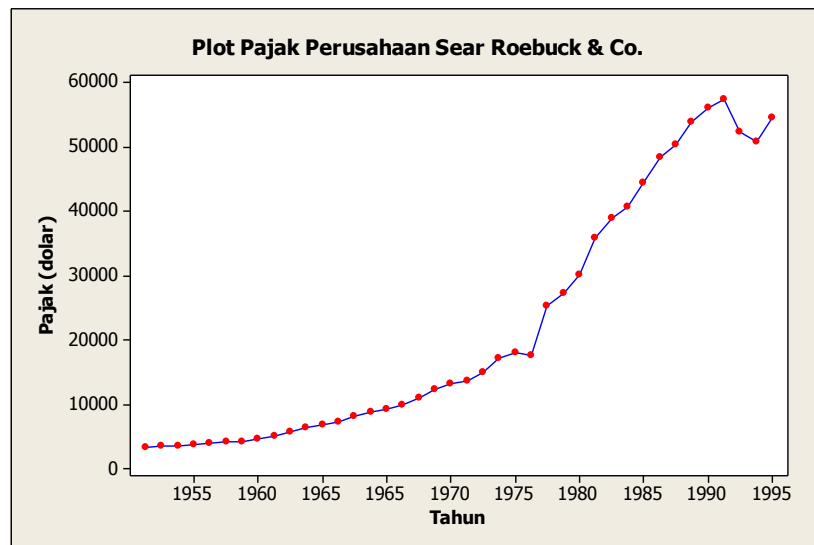


(Sumber: Hanke & Wichern, 2005: 111)

Gambar 2.2. Contoh Pola Gerakan *Trend*

3. Gerakan Siklis

Gerakan siklis adalah gerakan naik atau turun secara siklis di sekitar *tren* atau kondisi normal. Data yang sering mengalami gerakan siklis antara lain data perdagangan, industri, dan keuangan. Gambar 2.3. memperlihatkan pola data dengan gerakan siklis pada data pajak perusahaan Sears Roebuck & Co.

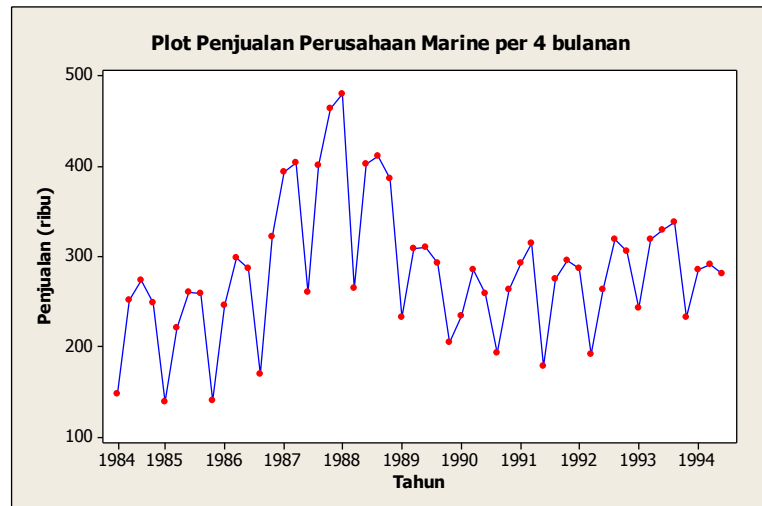


(Sumber: Hanke & Wichern, 2005: 70)

Gambar 2.3. Contoh Pola Gerakan Siklis

4. Gerakan Musiman

Gerakan musiman merupakan gerakan yang berulang-ulang secara teratur selama kurang lebih satu tahun misalnya pola yang berulang setiap minggu, bulan atau kuartalan. Pada kuartalan pengulangan terjadi setiap empat bulan. Contoh pola data musiman adalah pada data penjualan perusahaan Merine yang disajikan pada Gambar 2.4. Plot tersebut memperlihatkan bahwa pola yang berulang setiap periode 4 bulan.



(Sumber: Hanke & Wichern, 2005: 73)

Gambar 2.4. Contoh Pola Gerakan Musiman

D. Stasioneritas

Stasioner adalah keadaan dimana tidak ada perubahan rata-rata (*mean*) dan varians dari waktu ke waktu atau keduanya selalu konstan (tidak terjadi pertumbuhan atau penurunan) setiap waktu (Palit & Papovic, 2005: 38). Stasioner dapat juga dikatakan tidak terdapat perubahan yang drastis pada data.

Para peneliti sering mengamati pola pada plot data untuk memutuskan data yang diperoleh stasioner atau nonstasioner. Jika plot data deret berkala cenderung konstan yang tidak terdapat pertumbuhan atau penurunan maka data sudah stasioner. Plot autokorelasi juga dapat dijadikan alternatif untuk melihat kestasioneran data.

Ada beberapa cara yang dapat dilakukan untuk mengatasi ketidakstasioneran, misalnya ketidakstasioneran dalam rata-rata (*mean*) dilakukan pembedaan (*differencing*) dan untuk mengatasi ketidakstasioneran dalam varians dilakukan transformasi *Box-Cox*.

E. Autokorelasi

Autokorelasi digunakan untuk menentukan koefisien korelasi pada deret berkala dan untuk mempelajari pola data termasuk *trend* atau musiman. Fungsi autokorelasi adalah semua himpunan autokorelasi untuk semua *lag* k yang diberi simbol $\rho_k, k = 1, 2, 3, \dots$ dan $\rho_0 = 0$.

1. Autocorrelation Function (ACF)

Suatu deret $\{Y_t\}$ yaitu $Y_1, Y_2, Y_3, \dots, Y_n$ dikatakan stasioner jika *mean* $E(Y_t) = \mu$ dan varians $Var(Y_t) = E(Y_t - \mu)^2 = \sigma^2$ konstan. Kovarians antara Y_t dengan Y_{t+k} didefinisikan sebagai (Wei, 2006:10):

$$Cov(Y_t, Y_{t+k}) = E(Y_t - \mu)(Y_{t+k} - \mu) = \gamma^2$$

maka autokorelasi pada *lag* k yaitu korelasi antara Y_t dengan Y_{t+k} adalah:

$$\rho_k = \frac{Cov(Y_t, Y_{t+k})}{\sqrt{Var(Y_t)Var(Y_{t+k})}} = \frac{Cov(Y_t, Y_{t+k})}{Var(Y_t)} = \frac{\gamma_k}{\gamma_0} \quad (2.1)$$

dengan

- ρ_k : autokorelasi pada *lag* k
- μ : rata-rata
- γ_k : autokovariansi pada *lag* k
- t : waktu pengamatan, $t = 1, 2, 3, \dots$
- $Var(Y_{t+k}) = Var(Y_t) = \gamma_0$

Perkiraan ρ_k atau $\hat{\rho}_k$ dilakukan dengan menggunakan autokorelasi dari pengamatan pada waktu t sampai pengamatan pada waktu $t + k$, rumus yang digunakan adalah sebagai berikut (Tsay, 2010: 31):

$$\hat{\rho}_k = r_k = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}, k = 0, 1, 2, \dots \quad (2.2)$$

dengan

- r_k : autokorelasi pada *lag* k

\bar{Y} : rata-rata dari pengamatan $\{Y_t\}$
 Y_t : pengamatan pada waktu ke- t
 Y_{t+k} : pengamatan pada waktu $t + k, k = 1, 2, 3, \dots$

Pengujian signifikansi autokorelasi bertujuan untuk mengetahui apakah autokorelasi berbeda signifikan dari nol. Hipotesis yang digunakan adalah:

$H_0: \rho_k = 0$ (autokorelasi pada lag k tidak berbeda signifikan dari nol)

$H_1: \rho_k \neq 0$ (autokorelasi pada lag k berbeda signifikan dari nol)

Uji signifikansi ini menggunakan statistik uji:

$$t = \frac{r_k}{SE(r_k)} \text{ dengan } df = n - 1 \quad (2.3)$$

Standar *error* autokorelasi menggunakan rumus (Hanke & Wichern, 2005: 64):

$$SE(r_k) = \sqrt{\frac{1 + 2 \sum_{i=1}^{k-1} r_i^2}{n}} \quad (2.4)$$

Jika $k = 1$, maka:

$$SE(r_1) = \sqrt{\frac{1}{n}} \quad (2.5)$$

dengan

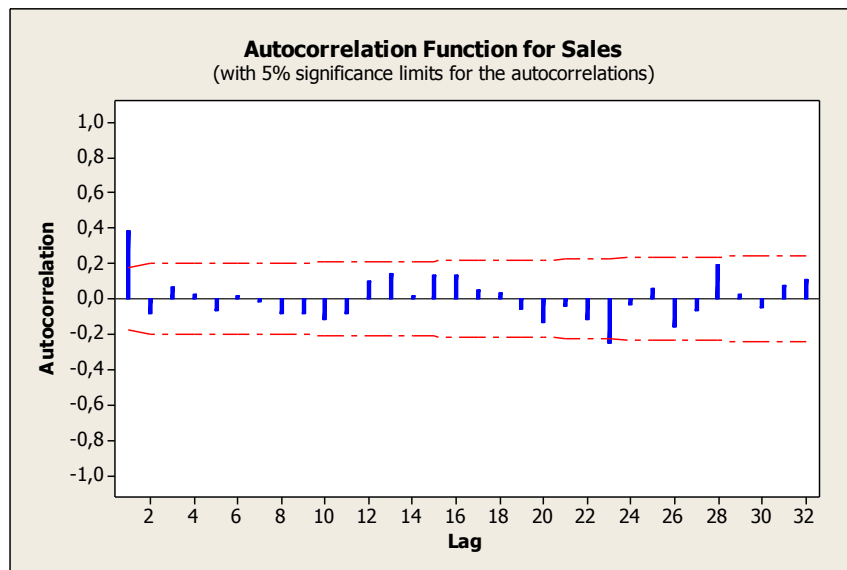
$SE(Y_t)$: standar *error* autokorelasi pada lag k
 r_k : autokorelasi pada lag k
 n : banyaknya pengamatan

Autokorelasi dikatakan berbeda signifikan dari nol jika H_0 ditolak dengan kriteria $t_{hitung} > t_{n-1}(\frac{\alpha}{2})$ atau $t_{hitung} < -t_{n-1}(\frac{\alpha}{2})$.

Signifikansi autokorelasi juga dapat ditentukan dengan melihat *correlogram*. *Correlogram* adalah plot antara lag k dengan r_k , dengan

$r_k = 0$ adalah pusat selang kepercayaan yang direpresentasikan dengan garis putus-putus berwarna merah, sedangkan *lag* data direpresentasikan dengan garis tegak berwarna biru. Selang kepercayaan tersebut dapat ditentukan dengan menggunakan rumus:

$$0 \pm t_{n-1} \left(\frac{\alpha}{2} \right) \times SE(r_k) \quad (2.6)$$



(Sumber: Hanke & Wichern, 2005: 434)

Gambar 2.5. Plot Fungsi Autokorelasi dari Data Penjualan

Gambar 2.5. memperlihatkan bahwa dari data penjualan, pada *lag* pertama dan *lag* 23 autokorelasinya berbeda signifikan dari nol karena tingginya melewati garis putus-putus merah yang merupakan selang kepercayaan dari data penjualan. Autokorelasi yang berbeda signifikan dari nol menunjukkan adanya hubungan antar pengamatan.

2. Partial Autocorrelation Function (PACF)

Partial Autocorrelation Function (PACF) atau autokorelasi parsial digunakan untuk mengukur korelasi antara Y_t dengan Y_{t+k} setelah Y_{t+1} ,

Y_{t+2}, \dots , dan Y_{t+k-1} dihilangkan. Autokorelasi parsial dinotasikan dengan ϕ_{kk} dengan rumus (Montgomery, Jennings & Kulachi, 2008: 250):

$$\phi_{kk} = \frac{|\rho_k|}{|P_k|} \quad (2.7)$$

Rumus ini didasari dari persamaan Yule-Walker,

$$P_k \phi_{kk} = \rho_k \quad (2.8)$$

dengan

$$\rho_k = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_k \end{bmatrix} \text{ dan } \phi_{kk} = \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix}$$

dan

$$P_k = \begin{bmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-1} \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-2} \\ \rho_2 & \rho_1 & 1 & \dots & \rho_{k-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & 1 \end{bmatrix}$$

untuk $k = 1, 2, 3, \dots$ diperoleh:

$$\phi_{11} = \frac{|\rho_1|}{|P_1|} = \frac{|\rho_1|}{1} = \rho_1$$

$$\phi_{22} = \frac{|\rho_2|}{|P_2|} = \frac{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & \rho_2 \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 \\ \rho_1 & 1 \end{vmatrix}} = \frac{\rho_2 - \rho_1^2}{1 - \rho_1^2}$$

dan seterusnya sampai ϕ_{kk} .

Pengujian signifikansi autokorelasi parsial menggunakan hipotesis:

$H_0: \phi_{kk} = 0$ (autokorelasi parsial pada lag k tidak berbeda signifikan dari nol)

$H_1: \phi_{kk} \neq 0$ (autokorelasi parsial pada lag k berbeda signifikan dari nol)

Uji signifikansi ini menggunakan statistik uji:

$$t = \frac{\phi_{kk}}{SE(\phi_{kk})} \text{ dengan } df = n - 1 \quad (2.9)$$

Standar *error* autokorelasi parsial menggunakan rumus (Wei, 2006: 22):

$$SE(\hat{\phi}_{kk}) = \sqrt{\frac{1}{n}} \quad (2.10)$$

dengan

$SE(\hat{\phi}_{kk})$: standar *error* autokorelasi parsial pada lag k

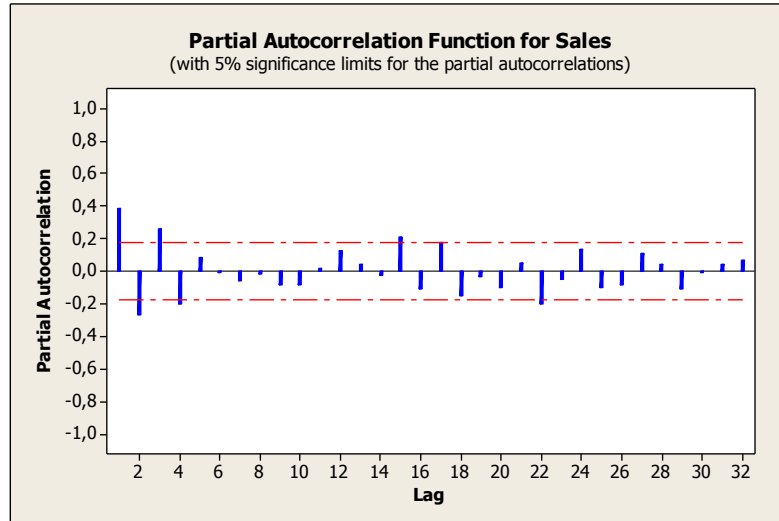
ϕ_{kk} : autokorelasi parsial pada lag k

n : banyaknya pengamatan

Autokorelasi parsial dikatakan berbeda signifikan dari nol jika H_0 ditolak

dengan kriteria $t_{hitung} > t_{n-1}(\frac{\alpha}{2})$ atau $t_{hitung} < -t_{n-1}(\frac{\alpha}{2})$.

Signifikansi autokorelasi parsial dapat juga diketahui dengan melihat *correlogram* untuk autokorelasi parsial. Berikut pada Gambar 2.6. dapat dilihat bahwa data penjualan berbeda signifikan dari nol pada lag 1, 2, 3, 4, 15, dan 23.



(Sumber: Hanke & Wichern, 2005: 434)

Gambar 2.6. Plot Fungsi Autokorelasi Parsial dari Data Penjualan

Selang kepercayaan yang berpusat di $\hat{\phi}_{kk} = 0$ dapat ditentukan dengan menggunakan rumus:

$$0 \pm t_{n-1} \left(\frac{\alpha}{2} \right) \times SE(\hat{\phi}_{kk}) \quad (2.11)$$

F. Proses *White Noise*

Deret $\{Y_t\}$ dikatakan *white noise* jika ada sebuah barisan dari variabel bebas yang tidak berkorelasi dengan rata-rata $E(Y_t) = \mu_a$, varians konstan $Var(a_t) = \sigma_a^2$, dan $\gamma_k = Cov(a_t, a_{t+k}) = 0$ untuk $k \neq 0$ (Wei, 2006: 15). Oleh karena itu, proses *white noise* dari $\{Y_t\}$ adalah stasioner dengan fungsi autokovarians:

$$\gamma_k = \begin{cases} \sigma_a^2, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (2.12)$$

fungsi autokorelasi,

$$\rho_k = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (2.13)$$

fungsi autokorelasi parsial,

$$\varphi_{kk} = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \quad (2.14)$$

Pada proses *white noise*, autokorelasi tidak berbeda signifikan dari nol. Alternatif lain pengujian *white noise* adalah melihat plot *ACF* dan *PACF*. Kriteria *error white noise* jika tidak ada *lag* yang melewati garis putus-putus merah atau selang kepercayaan.

G. Kriteria Model Terbaik

Hasil peramalan dapat juga dikatakan sebagai hasil prediksi. Nilai prediksi di sini tidak dapat dipisahkan dengan ketidakpastian karena bukan hasil yang sebenarnya sehingga pasti ada kesalahan peramalan. Kesalahan peramalan dapat diukur dengan beberapa kriteria (Hanke & Wichern, 2005: 79 - 80).

1. Means Absolute Percent Error (MAPE)

Nilai *MAPE* dapat ditentukan dengan menggunakan rumus sebagai berikut:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Y_t - \hat{Y}_t|}{Y_t} \times 100\% \quad (2.16)$$

dengan

Y_t : nilai pengamatan ke- t

\hat{Y}_t : nilai peramalan pada waktu ke- t
 n : banyaknya pengamatan

2. Mean Squared Error (MSE)

Nilai *MSE* digunakan untuk mengukur ketepatan nilai dugaan model yang dinyatakan dalam rata-rata kuadrat dari kesalahan. Berikut ini rumus untuk menghitung nilai *MSE*:

$$MSE = \frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2 \quad (2.17)$$

H. Jaringan Saraf

1. Jaringan Saraf Biologi

Otak manusia terdiri dari 10 miliar sel saraf padat yang saling berhubungan yang disebut *neuron*. Masing-masing terhubung dengan sekitar 10.000 *neuron* lain, dengan 60 triliun *sinapsis* (*synapse*) yang menghubungkannya. Dengan menggunakan beberapa *neuron* secara bersamaan, otak dapat melakukan fungsinya jauh lebih cepat daripada komputer tercepat yang ada saat ini (Yeung *et al*, 1998: 1).

Setiap sel *neuron* memiliki satu inti sel. Inti sel ini bertugas melakukan pemrosesan informasi. *Neuron* biologi merupakan sistem yang “*fault Tolerant*” dalam dua hal. Pertama, manusia dapat mengenali sinyal *input* yang agak berbeda dari yang pernah diterima sebelumnya. Kedua, otak manusia tetap mampu bekerja meskipun beberapa *neuronnya* tidak mampu bekerja dengan baik. Jika sebuah *neuron* rusak, maka *neuron* lain dapat dilatih untuk menggantikan fungsi sel yang rusak.

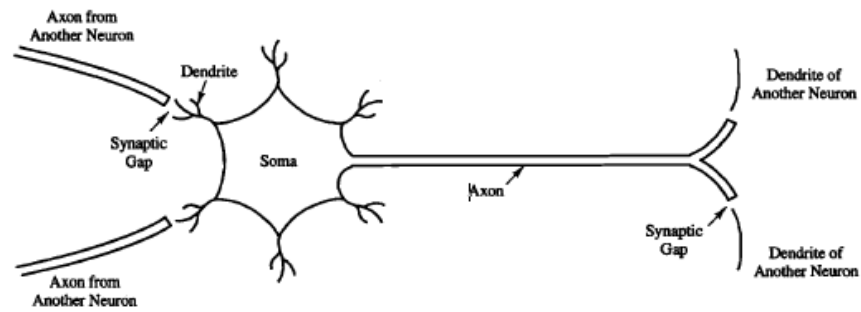
Komponen-komponen utama dari sebuah *neuron* dikelompokkan menjadi 3 bagian, yaitu (Fausett, 1994: 5):

- a. *dendrit*, bertugas menerima informasi,
- b. badan sel (*soma*), berfungsi sebagai pengolah informasi, dan
- c. *akson/axon (neurit)*, bertugas mengirim impuls-impuls ke sel saraf lainnya.

Informasi atau sinyal disampaikan oleh dendrit yang bertugas sebagai penyampai sinyal dari satu *neuron* ke *neuron* lain yang saling berhubungan. *Dendrit* merupakan serabut saraf yang bercabang-cabang pendek dan jumlahnya lebih dari 1. Sebagai keluaran, setiap *neuron* mempunyai *akson*, sedangkan bagian penerima sinyal disebut *sinapsis*. Kekuatan *sinapsis* bisa menurun atau meningkat tergantung kepada seberapa besar tingkat propagasi (penyiaran) sinyal yang diterimanya.

Secara umum, cara kerja sebuah *neuron* terhadap suatu informasi adalah berawal dari *dendrit* yang bertugas menerima informasi. Informasi ini dilewatkan melalui *sinapsis* yang pada perjalanan biologisnya terjadi proses kimiawi, pada jaringan saraf tiruan proses ini disebut pembentukan bobot. Informasi yang diterima diolah oleh *soma*. Untuk mengirim informasi ke sel lain, informasi dilewatkan melalui *akson*. Selanjutnya informasi akan melalui *sinapsis* dan disampaikan ke *soma* lain oleh *dendrit neuron* tersebut.

Berikut adalah gambar jaringan saraf secara biologi yang terdiri atas *neuron, soma, akson* dan *dendrit*.



(Sumber: Fausett, 1994: 6)

Gambar 2.7. Jaringan Saraf Biologi

2. Neural Network (NN)

Menurut Zhang (2004), *neural network (NN)* atau jaringan saraf tiruan (JST) adalah model komputasi untuk pengolahan informasi. Jaringan ini berkembang dengan pemodelan jaringan saraf biologi, khususnya otak manusia.

Neural network adalah sistem pemrosesan informasi yang memiliki karakteristik mirip jaringan saraf biologi. *NN* dibentuk sebagai generalisasi model matematika dari jaringan saraf biologi, dengan asumsi bahwa (Fausett, 1994: 3):

- a. pemrosesan informasi terjadi pada banyak elemen sederhana (*neuron*),
- b. sinyal dikirimkan diantara *neuron-neuron* melalui penghubung-penghubung,
- c. penghubung antara *neuron* memiliki bobot yang akan memperkuat atau memperlemah sinyal, dan
- d. untuk menentukan *output*, setiap *neuron* menggunakan fungsi aktivasi yang dikenakan pada jumlahan *input* yang diterima.

Neural network terdiri dari unsur-unsur sederhana yang beroperasi secara paralel. Jaringan ini dapat ditentukan dengan hubungan antar unsur-unsurnya. Umumnya jaringan dapat dilatih untuk melakukan fungsi tertentu dengan menyesuaikan nilai-nilai bobot antar unsur-unsurnya untuk mencapai *output* atau target tertentu (Demuth & Beale, 1992: 18). *Neural network* dapat diaplikasikan di berbagai bidang terutama pada analisis yang lebih kompleks, masalah nonlinear atau suatu struktur paralel. Pada bidang peramalan, *neural network* digunakan untuk menyelesaikan masalah peramalan termasuk pada pembentukan model (Hu & Hwang, 2001: 240).

3. Kerangka *Neural Network*

Pada *neural network*, *neuron-neuron* akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut lapisan *neuron* (*neuron layer*). Pada umumnya, *neural network* mempunyai tiga lapisan, yaitu (Yeung *et al*, 1998: 3):

a. Lapisan *Input* (*Input Layer*)

Node-node di dalam lapisan *input* disebut *neuron-neuron input*. *Neuron-neuron input* menerima *input* berupa gambaran suatu permasalahan dari luar.

b. Lapisan Tersembunyi (*Hidden layer*)

Node-node di dalam lapisan tersembunyi disebut *neuron-neuron* tersembunyi.

c. Lapisan *Output* (*Output Layer*)

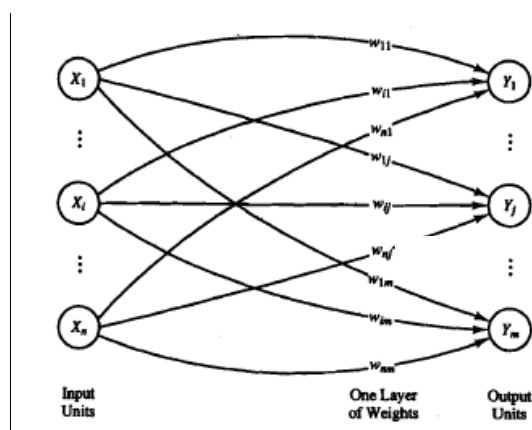
Node-node di dalam lapisan *output* disebut *neuron-neuron output*. Keluaran dari lapisan ini merupakan hasil dari *neural network* terhadap suatu permasalahan.

4. Arsitektur *Neural Network*

Pengaturan *neuron* dalam setiap lapisan dan pola hubungan antar lapisan disebut arsitektur jaringan saraf. Arsitektur *neural network* dikalsifikasikan menjadi 3, yaitu (Fausett, 1994: 12-14):

a. Jaringan Lapisan Tunggal (*Single Layer Net*)

Jaringan dengan lapisan tunggal memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan lain (lapisan tersembunyi). Dengan kata lain, ciri-ciri dari arsitektur jaringan saraf lapisan tunggal adalah hanya terdiri dari satu lapisan *input* dan satu lapisan *output*. Contoh arsitektur jaringan dengan lapisan tunggal dapat dilihat pada Gambar 2.8.

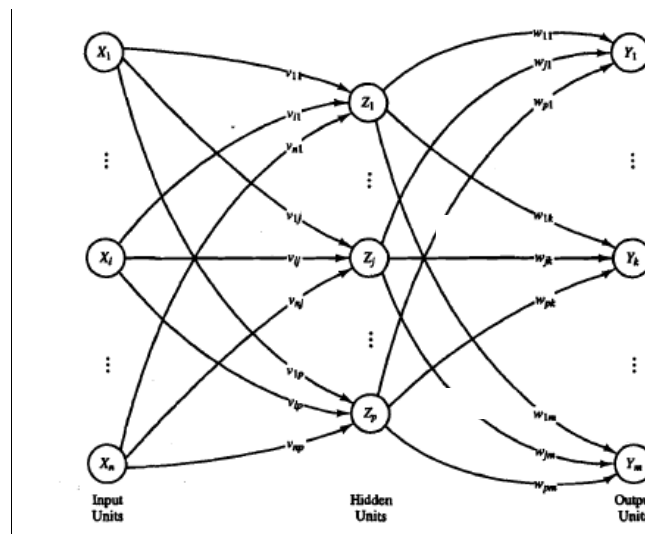


(Sumber: Fausett, 1994: 13)

Gambar 2.8. Arsitektur Jaringan Lapisan Tunggal

b. Jaringan Banyak Lapisan (*Multilayer Net*)

Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang berada diantara lapisan *input* dan lapisan *output* (terdapat satu atau lebih lapisan tersembunyi). Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit dibanding dengan jaringan dengan lapisan tunggal. Contoh arsitektur jaringan dengan banyak lapisan dapat dilihat pada Gambar 2.9.

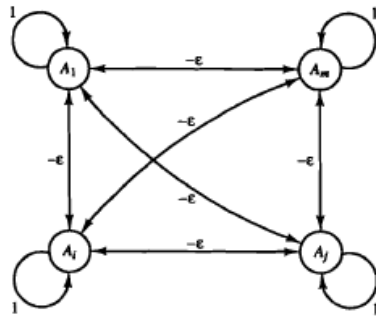


(Sumber: Fausett, 1994: 13)

Gambar 2.9. Arsitektur Jaringan Banyak Lapisan

c. Jaringan Lapisan Kompetitif (*Competitive Layer Net*)

Pada jaringan ini, antar *neuron* dapat saling dihubungkan. Contoh arsitektur jaringan dengan lapisan kompetitif dapat dilihat pada Gambar 2.10.



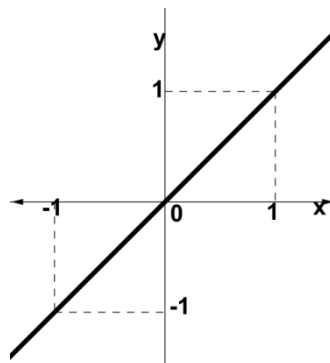
(Sumber: Fausett, 1994: 14)

Gambar 2.10. Arsitektur Jaringan Lapisan Kompetitif

5. Fungsi Aktivasi

Fungsi aktivasi adalah fungsi yang digunakan untuk menentukan keluaran suatu *neuron*. Salah satu fungsi aktivasi yang sering digunakan adalah Fungsi Linear (Identitas). Fungsi linear memiliki nilai *output* yang sama dengan nilai *inputnya*. Fungsi linear dirumuskan sebagai (Fausett, 1994: 17):

$$y = x \text{ untuk semua } x$$



(Sumber: Kusumadewi, 2004, 54)

Gambar 2.11. Fungsi Aktivasi: Fungsi Linear

6. Metode Pembelajaran dan Pelatihan

Salah satu bagian terpenting dari konsep jaringan saraf adalah terjadinya proses pembelajaran. Tujuan utama dari proses pembelajaran

adalah melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan saraf, sehingga diperoleh bobot akhir yang tepat sesuai pola data yang dilatih. Pada dasarnya terdapat 2 metode pembelajaran, yaitu metode pembelajaran terawasi (*supervised learning*) dan metode pembelajaran tidak terawasi (*unsupervised learning*) (Yeung *et al*, 1998: 5).

a. Pembelajaran terawasi (*supervised learning*)

Pada proses pembelajaran ini, satu *input* yang telah diberikan pada satu *neuron* di lapisan *input* akan dijalankan di sepanjang jaringan saraf sampai ke *neuron* pada lapisan *output*. Hasil *output* yang diperoleh kemudian dicocokkan dengan target, jika terjadi perbedaan, maka akan muncul *error*. Jika nilai *error* cukup besar, akan dilakukan pembelajaran yang lebih banyak lagi.

b. Pembelajaran tidak terawasi (*unsupervised learning*)

Pada proses pembelajaran ini, nilai bobot disusun dalam suatu interval atau range tertentu tergantung dari nilai *input* yang diberikan. Pembelajaran ini bertujuan mengelompokkan unit-unit yang hampir sama dalam satu area tertentu.

BAB III PEMBAHASAN

Bab ini berisi pembahasan mengenai *Radial Basis Function Neural Network (RBFNN)*, arsitektur *RBFNN*, *K-means Cluster*, Metode Kuadrat Terkecil (*Least Square*), Metode *Global Ridge-Regression*, pembentukan model *RBFNN* pada data deret berkala dengan pemrograman *MATLAB*, prosedur pembentukan model *RBFNN* pada deret berkala, dan aplikasi *RBFNN* untuk peramalan banyak kasus demam berdarah di DIY.

A. Radial Basis Function Neural Network (RBFNN)

Model *RBFNN* terdiri dari 3 lapisan, yaitu lapisan *input (input layer)*, lapisan tersembunyi (*hidden layer*) dan lapisan *output (output layer)*. Lapisan *input* menerima suatu vektor *input x* yang kemudian dibawa ke lapisan tersembunyi yang akan memproses data *input* secara nonlinear dengan fungsi aktivasi. *Output* dari lapisan tersembunyi selanjutnya diproses di lapisan *output* secara linear (Wei *et al*, 2011).

Model *RBFNN* menggunakan fungsi basis sebagai fungsi aktivasi untuk setiap *neuron* pada lapisan tersembunyi. Beberapa fungsi radial basis adalah sebagai berikut (Brodjol Sutijo, 2008):

1. Fungsi multikuadratik

$$\phi(x) = (x^2 + \sigma^2)^{1/2} \quad (3.1)$$

2. Fungsi invers multikuadratik

$$\phi(x) = \frac{1}{(x^2 + \sigma^2)^{1/2}} \quad (3.2)$$

3. Fungsi Gaussian

$$\phi(x) = e^{(-x^2/2\sigma^2)} \quad (3.3)$$

dengan

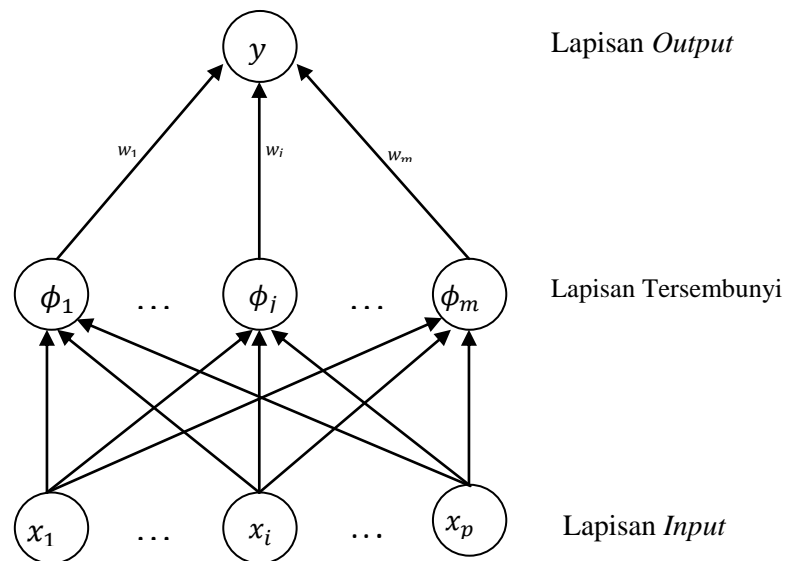
x = norm Euclidean antara vektor *input* \mathbf{x} dengan pusat *neuron* tersembunyi μ

$$x = \|\mathbf{x} - \mu\|$$

Fungsi aktivasi yang biasa digunakan dalam *RBFNN* adalah fungsi Gaussian. Hal ini dikarenakan fungsi Gaussian bersifat lokal yaitu nilai fungsi akan menuju nol ($\phi \rightarrow 0$) jika nilai x menuju tak hingga ($x \rightarrow \infty$), dan nilai fungsi menuju satu ($\phi \rightarrow 1$) jika nilai x menuju nol ($x \rightarrow 0$).

Arsitektur *RBFNN*

Arsitektur *RBFNN* dapat dilihat pada Gambar 3.1 berikut:



(Sumber: Orr, 1996: 10)

Gambar 3.1. Arsitektur *RBFNN*

Pada arsitektur di atas terdapat p komponen vektor *input* \mathbf{x} , m buah fungsi basis sebagai fungsi aktivasi *neuron* tersembunyi dan satu buah *output*. *Output* y yang dihasilkan dari model *RBFFNN* merupakan kombinasi linear dari bobot $\{w_j\}_{j=1}^m$ dengan fungsi aktivasi $\phi_j(\mathbf{x})$ dan dirumuskan sebagai berikut (Orr, 1996: 11):

$$y = \sum_{j=1}^m w_j \phi_j(\mathbf{x}) \quad (3.4)$$

dengan

- m = banyak fungsi aktivasi *neuron* tersembunyi
- w_j = bobot *output* ke- j
- $\phi_j(\mathbf{x})$ = fungsi aktivasi *neuron* tersembunyi ke- j
- \mathbf{x} = $[x_1 \ x_2 \ \dots \ x_p]$ yang merupakan vektor *input*

Berdasarkan fungsi basis Gaussian diperoleh persamaan sebagai berikut:

$$\begin{aligned} \phi_j(\mathbf{x}) &= \exp\left(-\frac{\|\mathbf{x}-\mu_j\|^2}{2\sigma_j^2}\right) \\ &= \exp\left[-\frac{1}{2}\left\{\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 + \left(\frac{x_i-\mu_j}{\sigma_j}\right)^2 + \dots + \left(\frac{x_p-\mu_m}{\sigma_m}\right)^2\right\}\right] \end{aligned} \quad (3.5)$$

dengan

- x_i = variabel *input* ke- i , $i=1,2,\dots,p$
- μ_j = pusat *neuron* tersembunyi ke- j , $j=1,2,\dots,m$
- σ_j = simpangan baku *neuron* tersembunyi ke- j , $j=1,2,\dots,m$

B. K-means Cluster

Salah satu ciri model *RBFFNN* adalah pada fungsi aktivasi yang dalam perhitungannya membutuhkan nilai pusat dan varians *neuron* tersembunyi. Metode *K-means* ini mengelompokkan data *input* menjadi beberapa kelompok

atau kluster sehingga nilai pusat dan varians setiap kluster dapat dihitung. Pusat kluster adalah rata-rata (*mean*) kluster tersebut.

Algoritma metode *K-means* adalah sebagai berikut (Johnson & Winchern, 2007: 696):

1. Tentukan k kluster dengan k nilai pusat.
2. Tempatkan setiap obyek pada kelompok yang mempunyai jarak terdekat dengan pusat, hitung kembali nilai pusat baru.
3. Ulangi langkah 2 sampai nilai pusat lama sama dengan nilai pusat baru.

C. Metode Kuadrat Terkecil (*Least Square*)

Desain *RBNN* membentuk pemetaan nonlinear dari variabel *input* ke lapisan tersembunyi dan pemetaan linear dari lapisan tersembunyi ke lapisan *output*. Oleh karena itu model *RBNN* melakukan optimasi hanya pada lapisan *output* yang dapat dilakukan dengan metode kuadrat terkecil (*least square*).

Metode kuadrat terkecil jika diterapkan pada analisis regresi bertujuan untuk memudahkan menyelesaikan masalah optimasi. Pada skripsi ini metode kuadrat terkecil digunakan untuk menentukan nilai bobot dengan nilai *error* minimum. Pada metode ini dikenal istilah *training set* yang memuat elemen-elemen pasangan nilai-nilai dari variabel *input* dan variabel *output*. Model linear yang digunakan adalah $y = \sum_{j=1}^m w_j \phi_j(\mathbf{x})$ dan *training set* $\{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^n$ maka prinsip kuadrat terkecil adalah meminimalkan jumlah kuadrat kesalahan (*sum square error (SSE)*):

$$SSE = \sum_{i=1}^n (\hat{y}_i - y)^2 \quad (3.6)$$

dengan

\hat{y}_i = nilai prediksi variabel *output* ke- i

y = nilai variabel *output* ke- i

n = banyak pengamatan

Kemudian akan ditentukan nilai optimum untuk bobot ke- j . Pertama, diturunkan fungsi *SSE* menjadi:

$$\frac{\partial s}{\partial w_j} = 2 \sum_{i=1}^n (y - \hat{y}_i) \frac{\partial y}{\partial w_j} \quad (3.7)$$

Berdasarkan persamaan (3.4) diperoleh:

$$\frac{\partial y}{\partial w_j} = \phi_j(\mathbf{x}) \quad (3.8)$$

Selanjutnya persamaan (3.8) disubstitusikan ke persamaan (3.7) dan disamadengankan nol, sehingga diperoleh:

$$2 \sum_{i=1}^n (y - \hat{y}_i) \phi_j(\mathbf{x}_i) = 0 \quad (3.9)$$

$$\sum_{i=1}^n y \phi_j(\mathbf{x}_i) = \sum_{i=1}^n \hat{y}_i \phi_j(\mathbf{x}_i) \quad (3.10)$$

Karena $j = 1, 2, \dots, m$ maka diperoleh m persamaan seperti persamaan (3.10) untuk menentukan m bobot. Untuk memperoleh penyelesaian tunggal, persamaan (3.10) ditulis dengan notasi vektor, maka menjadi:

$$\boldsymbol{\phi}_j^T \mathbf{y} = \boldsymbol{\phi}_j^T \hat{\mathbf{y}} \quad (3.11)$$

dengan

$$\boldsymbol{\phi}_j = \begin{bmatrix} \phi_j(x_1) \\ \phi_j(x_2) \\ \vdots \\ \phi_j(x_n) \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

Karena ada m persamaan untuk setiap nilai j , maka persamaan (3.11) dapat ditulis sebagai berikut:

$$\begin{bmatrix} \boldsymbol{\phi}_1^T \mathbf{y} \\ \boldsymbol{\phi}_2^T \mathbf{y} \\ \vdots \\ \boldsymbol{\phi}_m^T \mathbf{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_1^T \hat{\mathbf{y}} \\ \boldsymbol{\phi}_2^T \hat{\mathbf{y}} \\ \vdots \\ \boldsymbol{\phi}_m^T \hat{\mathbf{y}} \end{bmatrix}$$

Menggunakan hukum perkalian vektor, persamaan di atas dapat ditulis menjadi:

$$\boldsymbol{\Phi}^T \mathbf{y} = \boldsymbol{\Phi}^T \hat{\mathbf{y}} \quad (3.12)$$

dengan

$$\boldsymbol{\Phi} = [\phi_1 \ \phi_2 \ \dots \ \phi_m]$$

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_m(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_m(x_2) \\ \vdots & \vdots & \dots & \vdots \\ \phi_1(x_n) & \phi_2(x_n) & \dots & \phi_m(x_n) \end{bmatrix}$$

Matriks $\boldsymbol{\Phi}$ disebut matrik desain. Komponen ke- i dari \mathbf{y} ketika bobot pada nilai optimum adalah (Howlett & Jain, 2001: 4):

$$\mathbf{y} = \sum_{j=1}^m w_j \boldsymbol{\phi}_j(\mathbf{x}) = \bar{\boldsymbol{\phi}}_i^T \hat{\mathbf{w}} \quad (3.13)$$

dengan

$$\bar{\boldsymbol{\phi}}_i = \begin{bmatrix} \phi_1(x_i) \\ \phi_2(x_i) \\ \vdots \\ \phi_m(x_i) \end{bmatrix}$$

akibatnya $\boldsymbol{\phi}_j$ adalah salah satu kolom dari $\boldsymbol{\Phi}$ dan $\bar{\boldsymbol{\phi}}_i^T$ adalah salah satu baris dari $\boldsymbol{\Phi}$. Oleh karena itu, berdasarkan persamaan (3.13) diperoleh:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \bar{\boldsymbol{\phi}}_1^T \hat{\mathbf{w}} \\ \bar{\boldsymbol{\phi}}_2^T \hat{\mathbf{w}} \\ \vdots \\ \bar{\boldsymbol{\phi}}_n^T \hat{\mathbf{w}} \end{bmatrix} = \boldsymbol{\Phi} \hat{\mathbf{w}} \quad (3.14)$$

Persamaan (3.14) disubstitusikan ke persamaan (3.13) menjadi:

$$\begin{aligned}\Phi^T \hat{y} &= \Phi^T y \\ \Phi^T \hat{y} &= \Phi^T \Phi \hat{w}\end{aligned}\tag{3.15}$$

Jika nilai *invers* dari $\Phi^T \Phi$ dapat ditentukan, maka nilai bobot optimum dapat dicari dengan persamaan berikut:

$$\hat{w} = (\Phi^T \Phi)^{-1} \Phi^T \hat{y}\tag{3.16}$$

$$\hat{w} = A^{-1} \Phi^T \hat{y}\tag{3.17}$$

Pada beberapa kasus nilai *invers* dari $\Phi^T \Phi$ tidak dapat ditentukan karena $\Phi^T \Phi$ merupakan matrik singular. Untuk menyelesaikan masalah matrik singular ini digunakan *weigh-decay* atau sama dengan *ridge regression*. *Ridge regression* memiliki dua bentuk yaitu *global ridge* dengan parameter tunggal untuk semua fungsi aktivasi dan *local ridge* dengan m parameter untuk m fungsi aktivasi. Pada skripsi ini akan digunakan metode *global ridge* untuk menentukan parameter regulasi. Selain karena proses pengerjaannya sederhana, metode *global ridge* menghasilkan *error* yang lebih kecil dibanding metode *local ridge*.

D. Metode Global Ridge-Regression

Metode *global ridge regression* mengestimasi bobot dengan menambahkan parameter regulasi yang bernilai positif pada *SSE* sehingga diperoleh fungsi (Orr, 1996: 21):

$$C = \sum_{i=1}^n (y - \hat{y}_i)^2 + \lambda \sum_{j=1}^m w_j^2\tag{3.18}$$

dengan

\hat{y}_i = nilai prediksi variabel *output* ke- i

- \mathbf{x}_i = vektor *input* ke- i
- y = nilai variabel *output*
- λ = parameter regulasi
- w_j = bobot ke- j
- n = banyak pengamatan

Bobot yang optimum diperoleh dengan mendifferensialkan persamaan (3.18) dengan variabel bebas yang ada kemudian ditentukan penyelesaiannya untuk differensial sama dengan nol.

$$\begin{aligned} \frac{\partial C}{\partial w_j} &= 2 \sum_{i=1}^n (y - \hat{y}_i) \frac{\partial y}{\partial w_j} + 2\lambda w_j \\ &= \sum_{i=1}^n y \frac{\partial y}{\partial w_j} - \sum_{i=1}^n \hat{y}_i \frac{\partial y}{\partial w_j} + \lambda w_j \\ \sum_{i=1}^n y \frac{\partial y}{\partial w_j} - \sum_{i=1}^n \hat{y}_i \frac{\partial y}{\partial w_j} + \lambda w_j &= 0 \\ \sum_{i=1}^n y \frac{\partial y}{\partial w_j} + \lambda w_j &= \sum_{i=1}^n \hat{y}_i \frac{\partial y}{\partial w_j} \end{aligned}$$

Berdasarkan persamaan (3.8), persamaan di atas menjadi:

$$\sum_{i=1}^n y \phi_j(\mathbf{x}_i) + \lambda w_j = \sum_{i=1}^n \hat{y}_i \phi_j(\mathbf{x}_i)$$

dan dalam notasi vektor adalah sebagai berikut:

$$\boldsymbol{\phi}_j^T \mathbf{y} + \lambda \hat{w}_j = \boldsymbol{\phi}_j^T \hat{\mathbf{y}} \quad (3.19)$$

$$\begin{bmatrix} \boldsymbol{\phi}_1^T \mathbf{y} \\ \boldsymbol{\phi}_2^T \mathbf{y} \\ \vdots \\ \boldsymbol{\phi}_m^T \mathbf{y} \end{bmatrix} + \begin{bmatrix} \lambda \hat{w}_1 \\ \lambda \hat{w}_2 \\ \vdots \\ \lambda \hat{w}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}_1^T \hat{\mathbf{y}} \\ \boldsymbol{\phi}_2^T \hat{\mathbf{y}} \\ \vdots \\ \boldsymbol{\phi}_m^T \hat{\mathbf{y}} \end{bmatrix}$$

$$\boldsymbol{\Phi}^T \mathbf{y} + \lambda \hat{\mathbf{w}}_j = \boldsymbol{\Phi}^T \hat{\mathbf{y}} \quad (3.20)$$

dengan

λ = parameter regulasi

$\hat{\mathbf{w}}$ = vektor prediksi bobot

$\hat{\mathbf{y}}$ = vektor prediksi nilai *output*

Berdasarkan definisi-definisi yang telah disebutkan di atas diperoleh persamaan sebagai berikut (Orr, 1996: 21):

$$\begin{aligned}\Phi^T \hat{\mathbf{y}} &= \Phi^T \mathbf{y} + \lambda \hat{\mathbf{w}}_j \\ &= \Phi^T \Phi \hat{\mathbf{w}} + \lambda \hat{\mathbf{w}}_j \\ &= (\Phi^T \Phi + \lambda \mathbf{I}_m) \hat{\mathbf{w}}\end{aligned}\tag{3.21}$$

dimana \mathbf{I}_m adalah matrik identitas berukuran $m \times m$. Jadi diperoleh persamaan normal untuk prediksi bobot adalah sebagai berikut:

$$\hat{\mathbf{w}} = (\Phi^T \Phi + \lambda \mathbf{I}_m)^{-1} \Phi^T \hat{\mathbf{y}}\tag{3.22}$$

E. Pembentukan Model *RBFNN* pada Data Deret Berkala dengan Pemrograman *MATLAB*

Proses pembentukan model *RBFNN* sangat penting karena akan menentukan bagaimana bentuk jaringan *RBFNN* yang optimum untuk menghasilkan *error* minimum. Pada proses pembentukan model ini digunakan pemrograman *MATLAB*.

Program-program utama yang dijalankan adalah **rbfDesign** dan **globalRidge**. Program **rbfDesign** (Lampiran 2 hal. 64) digunakan untuk membangkitkan matriks desain untuk *RBFNN* dengan variabel *input*, pusat dan nilai varians fungsi aktivasi, dan tipe fungsi. Programnya adalah sebagai berikut:

H=rbfDesign(X, C, R, options)

dengan

H = matrik desain

X = [\mathbf{x}_1 \mathbf{x}_2 ... \mathbf{x}_p] yang merupakan vektor *input*

C = [\mathbf{c}_1 \mathbf{c}_2 ... \mathbf{c}_m] adalah titik pusat *neuron* tersembunyi

R = [\mathbf{r}_1 \mathbf{r}_2 ... \mathbf{r}_m] adalah nilai varians *neuron* tersembunyi

Options = tipe *RBF*

Tipe *RBF* yang digunakan pada skripsi ini adalah fungsi Gaussian dan dituliskan '**b**' yaitu akan ditambahkan *neuron* bias pada jaringan yaitu matrik Φ akan mendapatkan satu kolom tambahan.

Program **globalRidge** (Lampiran 3, hal. 69) bertujuan untuk menentukan bobot optimum. Pemanggilan program *global ridge regression* adalah:

```
lamb = globalRidge (H, Y, l)
```

dengan

H = matrik desain

Y = vektor *input*

l = dugaan nilai parameter regulasi positif λ dengan default 0,05

Berdasarkan program **rbfDesign**, adanya penambahan *neuron* bias menyebabkan adanya bobot bias. Bobot bias akan ditambahkan pada kombinasi linear fungsi *output*.

F. Prosedur Pembentukan Model *RBFFNN*

Berdasarkan uraian di atas, dapat disusun suatu prosedur pembentukan model *RBFFNN* pada deret berkala sebagai berikut:

1. Menentukan nilai pusat dan varians

Nilai pusat dan varians ditentukan menggunakan metode *K-means*. Metode *K-means* mengelompokkan data *input* menjadi beberapa kelompok atau kluster sehingga nilai pusat dan varians setiap kluster dapat dihitung. Banyak kluster menentukan banyak *neuron* tersembunyi pada jaringan.

2. Identifikasi model

Identifikasi model bertujuan untuk menentukan banyak *neuron input* pada struktur jaringan *RBFNN*. Penentuan *input* dilakukan dengan melihat *lag-lag* yang signifikan pada plot *ACF*.

3. Pembagian data

Pada proses ini data dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Beberapa komposisi data *training* dan data *testing* yang sering digunakan masing-masing 80% dan 20%, 75% dan 25%, 60% dan 40%, dan sebagainya (Anugerah, 2007: 23).

4. Menentukan jaringan yang optimum

Menentukan jaringan yang optimum dilakukan dengan menentukan banyak *neuron* tersembunyi dan eliminasi *input* yang akan membentuk sebuah model terbaik. Model terbaik ditentukan dengan cara *trial and error* terhadap beberapa macam arsitektur yang mungkin dengan menggunakan fungsi aktivasi Gaussian. Penentuan model terbaik juga dilakukan dengan memperhatikan nilai *MAPE* dan *MSE* terkecil. Pada proses ini, dengan menggunakan metode Kuadrat Terkecil (*Least Square*) dan *Global Ridge Regression* diperoleh pula nilai-nilai bobot.

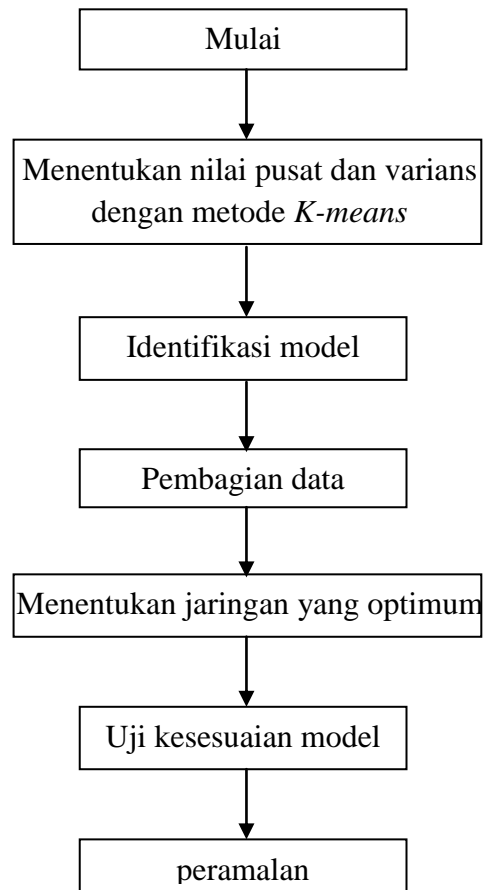
5. Uji kesesuaian model

Model dianggap baik jika residual dari hasil pembelajaran data *training* bersifat acak yang artinya proses *white noise* terpenuhi. Pengecekan ini dapat dilihat dari plot *ACF* atau *PACF* residual yang dihasilkan.

6. Peramalan

Langkah terakhir adalah peramalan dengan melakukan perhitungan menggunakan fungsi *output*.

Uraian di atas dapat dibentuk dalam bagan berikut:

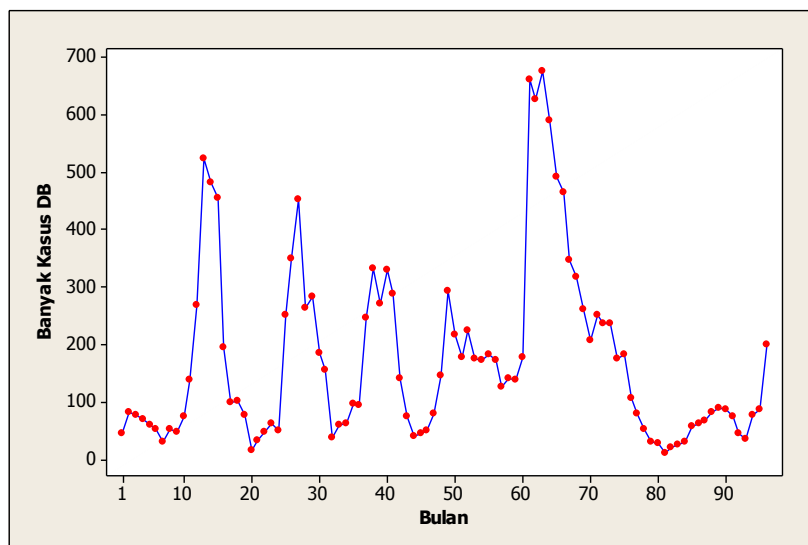


Gambar 3.2. Prosedur Pembentukan Model *RBFFNN*

G. Aplikasi *RBNN* untuk meramalkan Banyak Kasus Demam Berdarah di DIY

Model *RBNN* adalah salah satu model *NN* yang dapat digunakan untuk peramalan data deret berkala. Pada skripsi ini aplikasi model *RBNN* diterapkan untuk meramalkan banyak kasus demam berdarah di DIY.

Data banyak kasus demam berdarah di DIY merupakan data deret berkala. Data ini diperoleh dari Dinas Kesehatan Provinsi D.I. Yogyakarta yang merupakan data bulanan dari bulan Januari 2005 sampai dengan bulan Desember 2012. Banyak data ini adalah 96 data (Lampiran 1, hal. 63). Berikut adalah plot data deret berkala banyak kasus DB di DIY:



Gambar 3.3. Plot data banyak kasus DB di DIY dari Tahun 2005 sampai 2012

Berdasarkan Gambar 3.3. dan Lampiran 1 dapat dilihat bahwa data cenderung nonlinear. Data mengalami peningkatan setiap bulan Oktober sampai bulan Juni di tahun berikutnya. Lonjakan tajam terjadi pada bulan ke 61

yaitu bulan Januari 2010. Banyak kasus DB pada bulan Desember 2009 sebanyak 177 kasus, kemudian melonjak menjadi 659 kasus pada bulan Januari 2010. Selanjutnya, dapat dilihat juga setelah lonjakan tajam tersebut, terjadi penurunan data yang tidak ekstrim sampai bulan Desember 2012.

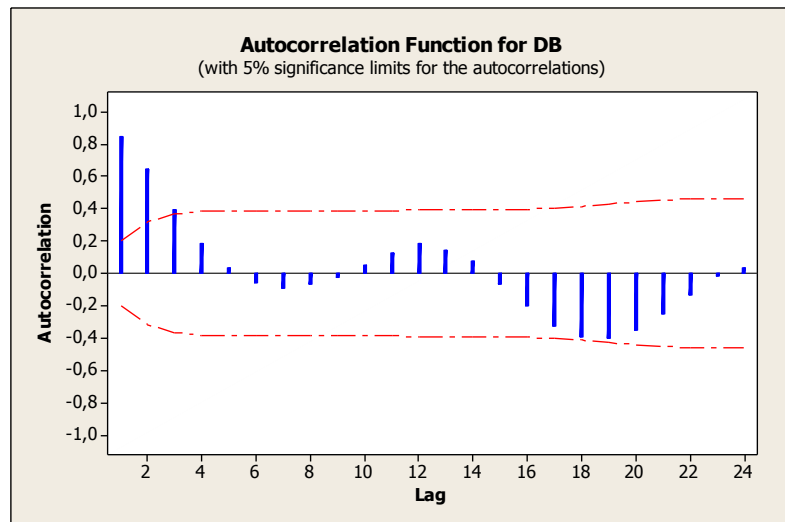
Berdasarkan prosedur pembentukan model *RBFNN* berikut pengaplikasiannya pada peramalan banyak kasus demam berdarah di DIY.

1. Menentukan Nilai Pusat dan Varians

Nilai pusat dan varians ditentukan dengan cara *trial and error* menggunakan metode *K-means* (Lampiran 5, hal. 80 pada Tabel 4.1.).

2. Identifikasi Model

Berikut adalah plot *ACF* data banyak kasus demam berdarah untuk melihat *lag-lag* yang signifikan.



Gambar 3.4. Plot *ACF* data banyak kasus DB di DIY dari bulan Januari 2005 sampai bulan Desember 2012

Berdasarkan plot *ACF* pada Gambar 3.4. diperoleh *lag* yang signifikan adalah *lag* 1, *lag* 2, dan *lag* 3. Dengan demikian diperoleh 3 variabel *input* yaitu x_{t-1} , x_{t-2} , dan x_{t-3} .

3. Pembagian Data

Data banyak kasus demam berdarah dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Pada skripsi ini berdasarkan hasil *trial and error* digunakan komposisi 60% untuk data *training* dan 40% untuk data *testing*.

4. Menentukan Jaringan yang Optimum

Pertama akan ditentukan banyak *neuron* tersembunyi. Nilai *MAPE* dan *MSE* hasil pengolahan program **rbfDesign** dan **globalRidge** dapat dilihat pada tabel berikut:

Tabel 3.1. Nilai *MAPE* dan *MSE* penentuan banyak *neuron* tersembunyi untuk data *training*

Banyak Kluster	<i>MAPE</i> (%)	<i>MSE</i>
2 kluster	0,8867	$1,2204 \times 10^4$
3 kluster	1,0033	$1,3204 \times 10^4$
4 kluster	0,5990	$0,6395 \times 10^4$
5 kluster	0,7071	$0,7791 \times 10^4$
6 kluster	0,6774	$1,2956 \times 10^4$
7 kluster	0,6454	$0,6479 \times 10^4$
8 kluster	0,4919	$0,5144 \times 10^4$
9 kluster	0,5329	$0,5734 \times 10^4$
10 kluster	0,5101	$0,5723 \times 10^4$

Tabel 3.2. Nilai *MAPE* dan *MSE* penentuan banyak *neuron* tersembunyi untuk data *testing*

Banyak Kluster	<i>MAPE</i> (%)	<i>MSE</i>
2 kluster	1,1543	1,1179x10 ⁴
3 kluster	1,3943	0,9142x10 ⁴
4 kluster	0,9141	1,0899x10 ⁴
5 kluster	1,0062	0,9504x10 ⁴
6 kluster	0,6367	0,6336x10 ⁴
7 kluster	0,9427	1,0480x10 ⁴
8 kluster	0,7886	1,1348x10 ⁴
9 kluster	0,6386	1,0225x10 ⁴
10 kluster	0,6876	1,01074x10 ⁴

Pada Tabel 3.1. dan 3.2. di atas terlihat bahwa nilai *MAPE* dan *MSE* terkecil terletak pada 8 kluster untuk data *training*. Pada data *testing* nilai *MAPE* dan *MSE* terkecil terbesar terletak pada 6 kluster. Namun berdasarkan hasil *trial and error*, hasil terbaik adalah dengan menggunakan 8 kluster. Dengan demikian model *RBFNN* yang digunakan untuk peramalan banyak kasus demam berdarah mempunyai arsitektur 3 *input* dan 8 *neuron* pada lapisan tersembunyi.

Selain terbentuknya model jaringan *RBFNN*, pada hasil pembelajaran diperoleh bobot-bobot yang akan digunakan untuk meramalkan banyak kasus demam berdarah pada periode selanjutnya.

Setelah banyak *neuron* tersembunyi ditentukan, langkah selanjutnya adalah eliminasi *input* untuk memperoleh model yang lebih sederhana. Eliminasi *input* dilakukan dengan cara mengeliminasi satu *input* dengan 8 *neuron* tersembunyi. Kriteria pemilihan yang digunakan adalah dengan

melihat nilai *MAPE* dan *MSE*. Hasil pembelajaran eliminasi *input* disajikan dalam tabel berikut:

Tabel 3.3. Nilai *MAPE* dan *MSE* hasil eliminasi *input* data *training*

Eliminasi	<i>MAPE</i> (%)	<i>MSE</i>
-	0,4919	$0,5144 \times 10^4$
x_{t-1}	0,6343	$0,7663 \times 10^4$
x_{t-2}	0,5774	$6,0625 \times 10^3$
x_{t-3}	0,6428	$0,8524 \times 10^4$

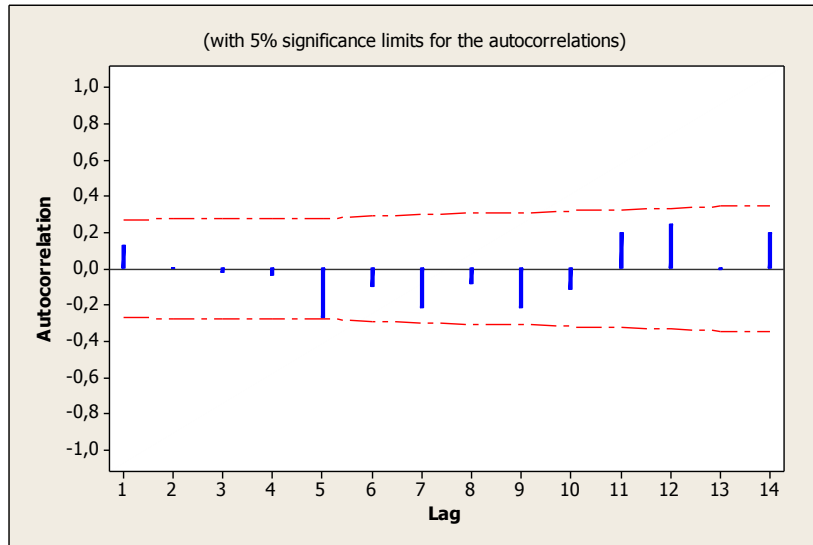
Tabel 3.4. Nilai *MAPE* dan *MSE* hasil eliminasi *input* data *testing*

Eliminasi	<i>MAPE</i> (%)	<i>MSE</i>
-	0,7886	$1,1348 \times 10^4$
x_{t-1}	0,7468	$0,8597 \times 10^4$
x_{t-2}	0,8612	$6,4337 \times 10^3$
x_{t-3}	0,8262	$0,8056 \times 10^4$

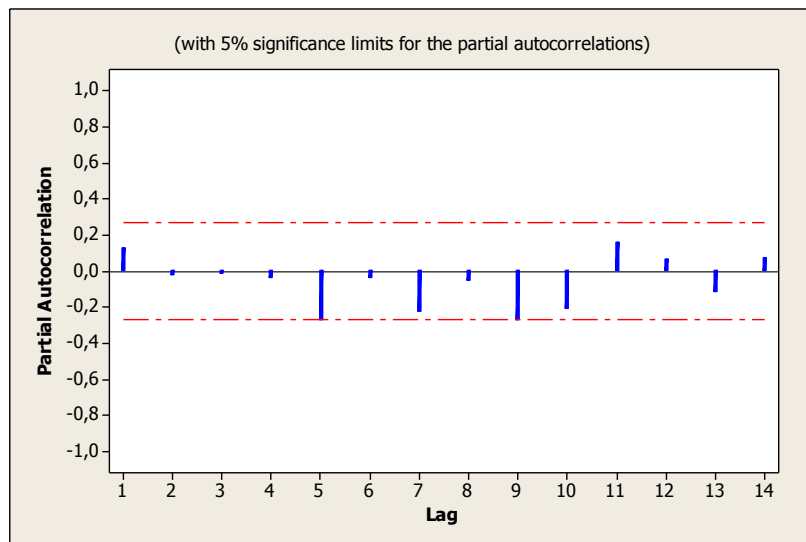
Berdasarkan nilai *MAPE* dan *MSE* pada kedua tabel di atas, penggunaan *input* untuk menghasilkan model terbaik adalah tanpa eliminasi (memiliki 3 *input*).

5. Uji Kesesuaian Model

Setelah memperoleh model terbaik dengan 3 *input* dan 8 *neuron* pada lapisan tersembunyi, langkah selanjutnya adalah pengecekan *error* pada model tersebut. Plot *ACF* dan *PACF* dari residual data *training* ditunjukkan pada Gambar 3.5. dan 3.6.berikut:



Gambar 3.5. Plot *ACF* residual data *training*

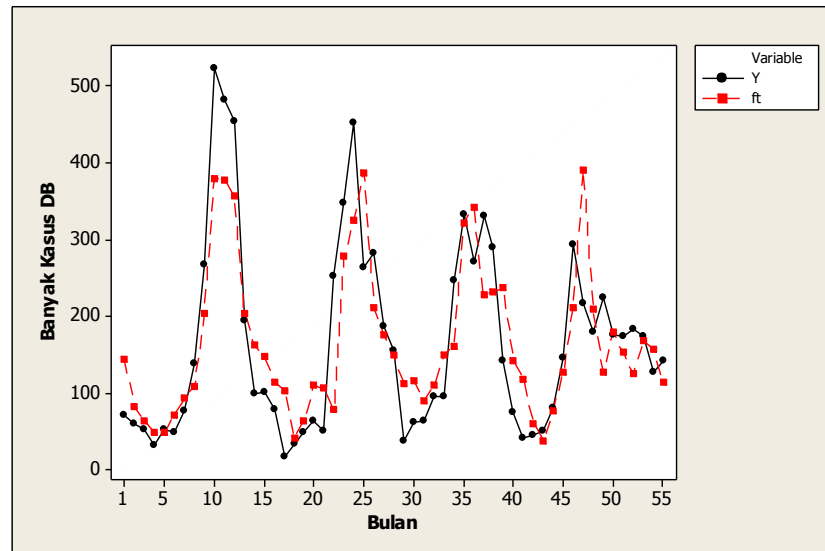


Gambar 3.6. Plot *PACF* residual data *training*

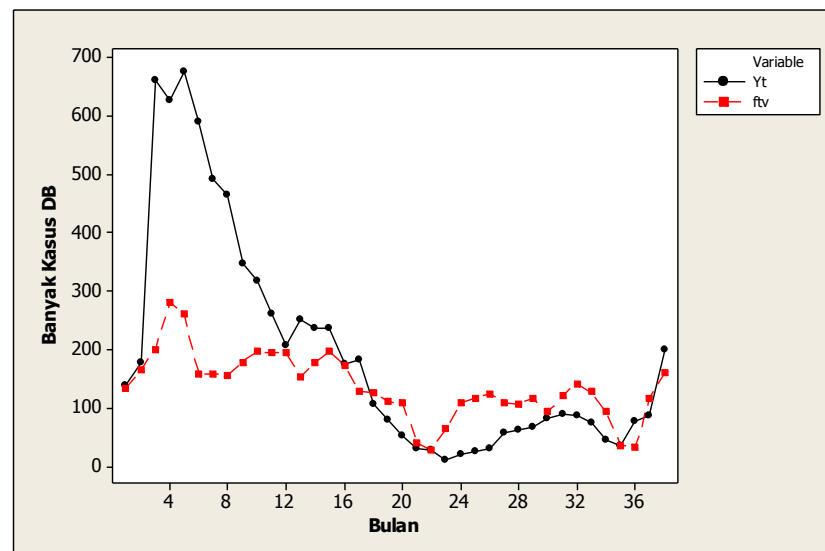
Kedua plot di atas menunjukkan bahwa tidak ada *lag* yang melebihi garis kepercayaan. Dengan kata lain tidak ada *lag* yang signifikan yang artinya *error* bersifat acak atau *white noise* terpenuhi. Oleh karena itu, model *RBNN* dengan arsitektur jaringan 8 *neuron* pada lapisan

tersembunyi dan *input* x_{t-1} , x_{t-2} , dan x_{t-3} dapat digunakan untuk meramalkan banyak kasus demam berdarah di DIY.

Berikut plot data aktual dan hasil peramalan banyak kasus demam berdarah di DIY untuk data *training* dan data *testing*.



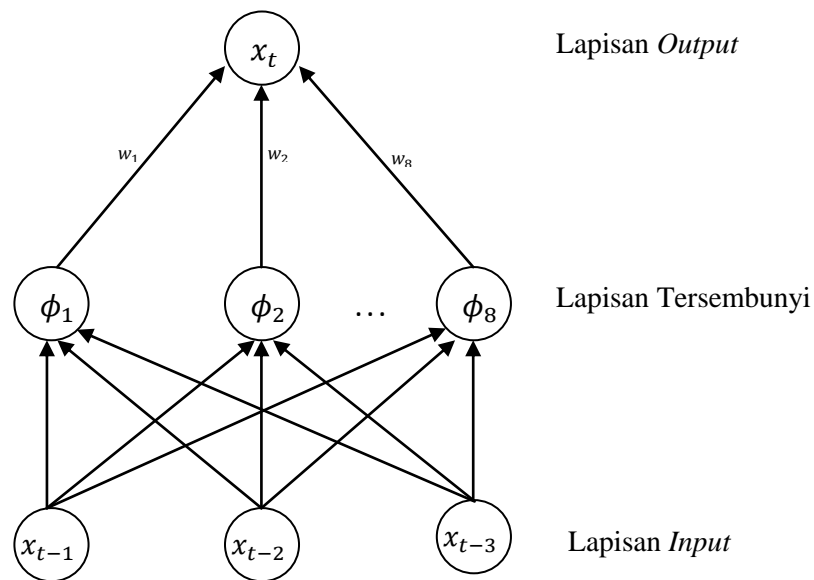
Gambar 3.7. Plot Data Aktual dan Hasil Peramalan pada Data *Training*



Gambar 3.8. Plot Data Aktual dan Hasil Peramalan pada Data *Testing*

Gambar 3.7. dan 3.8. menunjukkan bahwa data aktual dan data hasil peramalan untuk data *training* dan *testing* saling berdekatan. Berdasarkan *error* yang dihasilkan yaitu nilai *MAPE* dan *MSE* training adalah 0,4919% dan 5.144, *MAPE* dan *MSE* testing adalah 0,7886% dan 11.348. Dengan demikian dapat dikatakan bahwa model yang terbentuk sudah sesuai dan dapat digunakan untuk meramalkan banyak kasus demam berdarah di DIY.

Arsitektur *RBFNN* dengan 3 *input*, 8 *neuron* tersembunyi, dan 1 *output* dapat dilihat pada Gambar 3.9. berikut:



Gambar 3.9. Arsitektur *RBFNN* dengan 3 *input*, 8 *neuron* tersembunyi, dan 1 *output*

Berdasarkan arsitektur di atas, model *RBFNN* yang terbentuk adalah sebagai berikut:

$$x_t = \sum_{j=1}^8 w_j \phi_j(\mathbf{x})$$

6. Peramalan Banyak Kasus Demam Berdarah di DIY

Peramalan banyak kasus demam berdarah dilakukan dengan menggunakan arsitektur jaringan terbaik yaitu 3 *input*, 8 *neuron* tersembunyi dan 1 *output*. Diketahui *input* $x_{t-1} = 201$, $x_{t-2} = 87$, dan $x_{t-3} = 78$ masing-masing merupakan data pada bulan Desember 2012, November 2012 dan Oktober 2012 (Lampiran 1, hal. 63). Selanjutnya bobot yang diperoleh dari hasil pembelajaran dan digunakan untuk peramalan adalah $w_b = 291,3835$; $w_1 = -4,8856$; $w_2 = -20,6891$; $w_3 = -77,3097$; $w_4 = -141,2699$; $w_5 = -30,7144$; $w_6 = 47,4804$; $w_7 = 213,1270$; dan $w_8 = 152,8057$. Berdasarkan model *RBFNN* yang terbentuk maka perhitungan peramalan banyak kasus demam berdarah untuk bulan Januari 2013 adalah sebagai berikut:

$$\begin{aligned}
 x_t &= \sum_{j=1}^8 w_j \phi_j(\mathbf{x}) \\
 &= -4,8856\phi_1(\mathbf{x}) - 20,6891\phi_2(\mathbf{x}) - 77,3097\phi_3(\mathbf{x}) - \\
 &\quad 141,2699\phi_4(\mathbf{x}) - 30,7144\phi_5(\mathbf{x}) + 47,4804\phi_6(\mathbf{x}) + \\
 &\quad 213,1270\phi_7(\mathbf{x}) + 152,8057\phi_8(\mathbf{x}) + 291,3835 \\
 &= 291,3835 - (4,8856)(0,22153) - (20,6891)(0,00127) - \\
 &\quad (77,3097)(0,28987) - (141,2699)(0,55409) - \\
 &\quad (30,7144)(0,04731) + (47,4804)(3,0274 \times 10^{-5}) + \\
 &\quad (213,1270)(0,15057) + (152,8057)(0,76552) \\
 x_t &= 337,2037
 \end{aligned}$$

Perhitungan $\phi_j(\mathbf{x}_i)$, $j=1,2,\dots,8$, $i=t-1,t-2,t-3$ sebagai berikut:

$$\begin{aligned}
 \phi_1(\mathbf{x}_i) &= \exp\left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2}\right) \\
 &= \exp\left[-\frac{1}{2}\left\{\left(\frac{x_{t-1} - \mu_1}{\sigma_1}\right)^2 + \left(\frac{x_{t-2} - \mu_1}{\sigma_1}\right)^2 + \left(\frac{x_{t-3} - \mu_1}{\sigma_1}\right)^2\right\}\right] \\
 &= \exp\left[-\frac{1}{2}\left\{\left(\frac{201 - 548,4}{275,744}\right)^2 + \left(\frac{87 - 395}{275,744}\right)^2 + \left(\frac{78 - 194,8}{275,744}\right)^2\right\}\right] \\
 &= \exp(-1,50715) \\
 &= 0,22153
 \end{aligned}$$

$$\begin{aligned}
 \phi_2(\mathbf{x}_i) &= \exp\left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2}\right) \\
 &= \exp\left[-\frac{1}{2}\left\{\left(\frac{x_{t-1} - \mu_2}{\sigma_2}\right)^2 + \left(\frac{x_{t-2} - \mu_2}{\sigma_2}\right)^2 + \left(\frac{x_{t-3} - \mu_2}{\sigma_2}\right)^2\right\}\right] \\
 &= \exp\left[-\frac{1}{2}\left\{\left(\frac{201 - 503}{207,817}\right)^2 + \left(\frac{87 - 554,167}{207,817}\right)^2 + \left(\frac{78 - 593,67}{207,817}\right)^2\right\}\right] \\
 &= \exp(-6,66113) \\
 &= 0,00127
 \end{aligned}$$

$$\begin{aligned}
 \phi_3(\mathbf{x}_i) &= \exp\left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2}\right) \\
 &= \exp\left[-\frac{1}{2}\left\{\left(\frac{x_{t-1} - \mu_3}{\sigma_3}\right)^2 + \left(\frac{x_{t-2} - \mu_3}{\sigma_3}\right)^2 + \left(\frac{x_{t-3} - \mu_3}{\sigma_3}\right)^2\right\}\right] \\
 &= \exp\left[-\frac{1}{2}\left\{\left(\frac{201 - 233,867}{214,817}\right)^2 + \left(\frac{87 - 296,93}{214,817}\right)^2 + \left(\frac{78 - 340,8}{214,817}\right)^2\right\}\right]
 \end{aligned}$$

$$= \exp(-1,2383)$$

$$= 0,28987$$

$$\begin{aligned}\phi_4(\mathbf{x}_i) &= \exp\left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2}\right) \\ &= \exp\left[-\frac{1}{2}\left\{\left(\frac{x_{t-1} - \mu_4}{\sigma_4}\right)^2 + \left(\frac{x_{t-2} - \mu_4}{\sigma_4}\right)^2 + \left(\frac{x_{t-3} - \mu_4}{\sigma_4}\right)^2\right\}\right] \\ &= \exp\left[-\frac{1}{2}\left\{\left(\frac{201 - 151,57}{143,572}\right)^2 + \left(\frac{87 - 175,94}{143,572}\right)^2 + \left(\frac{78 - 196,26}{143,572}\right)^2\right\}\right] \\ &= \exp(-0,5904) \\ &= 0,55409\end{aligned}$$

$$\begin{aligned}\phi_5(\mathbf{x}_i) &= \exp\left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2}\right) \\ &= \exp\left[-\frac{1}{2}\left\{\left(\frac{x_{t-1} - \mu_5}{\sigma_5}\right)^2 + \left(\frac{x_{t-2} - \mu_5}{\sigma_5}\right)^2 + \left(\frac{x_{t-3} - \mu_5}{\sigma_5}\right)^2\right\}\right] \\ &= \exp\left[-\frac{1}{2}\left\{\left(\frac{201 - 43,27}{65,324}\right)^2 + \left(\frac{87 - 57,63}{65,324}\right)^2 + \left(\frac{78 - 95,27}{65,324}\right)^2\right\}\right] \\ &= \exp(-2,05098) \\ &= 0,04731\end{aligned}$$

$$\begin{aligned}\phi_6(\mathbf{x}_i) &= \exp\left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2}\right) \\ &= \exp\left[-\frac{1}{2}\left\{\left(\frac{x_{t-1} - \mu_6}{\sigma_6}\right)^2 + \left(\frac{x_{t-2} - \mu_6}{\sigma_6}\right)^2 + \left(\frac{x_{t-3} - \mu_6}{\sigma_6}\right)^2\right\}\right]\end{aligned}$$

$$\begin{aligned}
&= \exp \left[-\frac{1}{2} \left\{ \left(\frac{201 - 45,43}{36,856} \right)^2 + \left(\frac{87 - 39,43}{36,856} \right)^2 + \left(\frac{78 - 35,5}{36,856} \right)^2 \right\} \right] \\
&= \exp(-10,4052) \\
&= 3,0274 \times 10^{-5}
\end{aligned}$$

$$\begin{aligned}
\phi_7(\mathbf{x}_i) &= \exp \left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2} \right) \\
&= \exp \left[-\frac{1}{2} \left\{ \left(\frac{x_{t-1} - \mu_7}{\sigma_7} \right)^2 + \left(\frac{x_{t-2} - \mu_7}{\sigma_7} \right)^2 + \left(\frac{x_{t-3} - \mu_7}{\sigma_7} \right)^2 \right\} \right] \\
&= \exp \left[-\frac{1}{2} \left\{ \left(\frac{201 - 89,73}{57,952} \right)^2 + \left(\frac{87 - 76,33}{57,952} \right)^2 + \left(\frac{78 - 63,067}{57,952} \right)^2 \right\} \right] \\
&= \exp(-1,8933) \\
&= 0,15057
\end{aligned}$$

$$\begin{aligned}
\phi_8(\mathbf{x}_i) &= \exp \left(-\frac{\|\mathbf{x}_i - \mu_j\|^2}{2\sigma_j^2} \right) \\
&= \exp \left[-\frac{1}{2} \left\{ \left(\frac{x_{t-1} - \mu_8}{\sigma_8} \right)^2 + \left(\frac{x_{t-2} - \mu_8}{\sigma_8} \right)^2 + \left(\frac{x_{t-3} - \mu_8}{\sigma_8} \right)^2 \right\} \right] \\
&= \exp \left[-\frac{1}{2} \left\{ \left(\frac{201 - 277,42}{131,136} \right)^2 + \left(\frac{87 - 144,85}{131,136} \right)^2 + \left(\frac{78 - 77,14}{131,136} \right)^2 \right\} \right] \\
&= \exp(-0,26718) \\
&= 0,76552
\end{aligned}$$

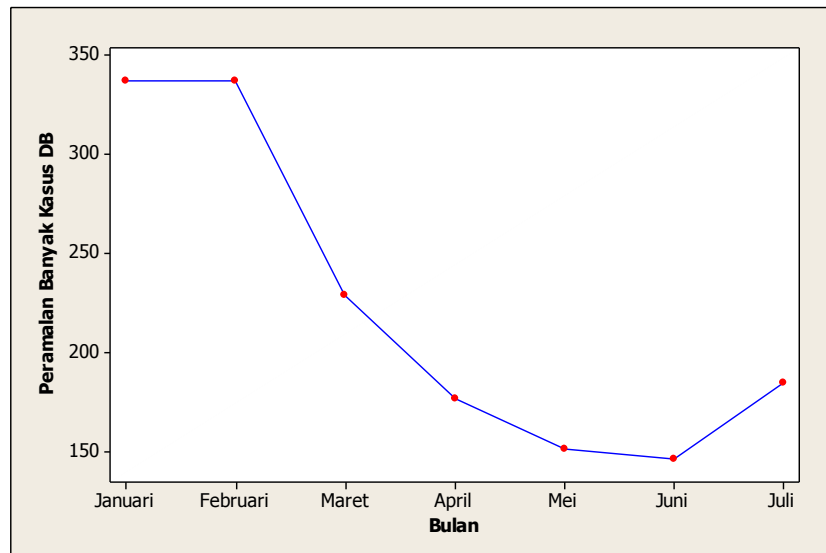
Berdasarkan hasil perhitungan di atas diperoleh hasil peramalan banyak kasus demam berdarah bulan Januari 2013 adalah sebanyak 337

kasus. Dengan cara yang sama dihitung peramalan jumlah kasus demam berdarah untuk bulan Februari sampai Juli 2013. Berikut hasil peramalan yang diperoleh:

Tabel 3.5. Hasil Peramalan

Waktu	Hasil peramalan (kasus)
Januari 2013	337
Februari 2013	337
Maret 2013	229
April 2013	177
Mei 2013	151
Juni 2013	146
Juli 2013	185

Berikut grafik hasil peramalan banyak kasus demam berdarah.



Gambar 3.10. Grafik Hasil Peramalan Banyak Kasus Demam Berdarah Tahun 2013

Berdasarkan hasil perhitungan dan Gambar 3.10. di atas banyak kasus demam berdarah diperkirakan akan meningkat dari bulan Desember 2012 sebanyak 201 kasus menjadi 337 pada bulan Januari 2013. Kemudian secara perlahan banyak kasus mulai menurun pada bulan Maret 2013 sampai bulan Juni 2013. Pada bulan Juli 2013 banyak kasus kembali meningkat menjadi 185 kasus. Hasil perhitungan peramalan banyak kasus demam berdarah di DIY selengkapnya dapat dilihat pada Lampiran 5 halaman 80 pada Tabel 4.2.

BAB IV

PENUTUP

A. Kesimpulan

Berdasarkan pembahasan mengenai peramalan banyak kasus demam berdarah D.I. Yogyakarta dengan menggunakan model *Radial Basis Function Neural Network (RBFNN)* dapat diperoleh kesimpulan sebagai berikut:

1. Proses pembentukan model *Radial Basis Neural Network (RBFNN)* pada deret berkala adalah sebagai berikut:

a. Menentukan nilai pusat dan varians

Nilai pusat dan varians ditentukan menggunakan metode *K-means*. Metode ini mengelompokkan data *input* menjadi beberapa kelompok atau kluster sehingga nilai pusat dan varians setiap kluster dapat dihitung.

b. Identifikasi model

Identifikasi model bertujuan untuk menentukan banyak *neuron input* pada struktur jaringan *RBFNN*. Penentuan *input* dilakukan dengan melihat *lag-lag* yang signifikan pada plot *ACF*.

c. Pembagian data

Pada proses ini data dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Komposisi data *training* dan data *testing* yang sering

digunakan masing-masing 80% dan 20%, 75% dan 25%, atau 60% dan 40%.

d. Menentukan jaringan yang optimum

Menentukan jaringan yang optimum dilakukan dengan menentukan banyak *neuron* tersembunyi dan eliminasi *input* yang akan membentuk sebuah model terbaik. Model terbaik ditentukan dengan cara *trial and error* terhadap beberapa macam arsitektur yang mungkin dengan menggunakan fungsi aktivasi Gaussian. Penentuan model terbaik juga dilakukan dengan memperhatikan nilai *MAPE* dan *MSE* terkecil. Pada proses ini, dengan menggunakan metode Kuadrat Terkecil (*Least Square*) dan *Global Ridge Regression* diperoleh pula nilai-nilai bobot.

e. Uji kesesuaian model

Model dianggap baik jika residual dari hasil pembelajaran data *training* bersifat acak yang artinya proses *white noise* terpenuhi. Pengecekan ini dapat dilihat dari plot *ACF* atau *PACF* residual yang dihasilkan.

2. Peramalan banyak kasus demam berdarah di DIY menggunakan model *RBFNN* dengan 3 variabel *input* yaitu x_{t-1} , x_{t-2} , dan x_{t-3} dan 8 *neuron* tersembunyi. Model ini menghasilkan *MAPE* dan *MSE training* masing-masing 0,4919% dan 5.144, *MAPE* dan *MSE testing* adalah 0,7886% dan 11.348. Berdasarkan model yang telah terbentuk, diperoleh hasil peramalan untuk bulan Januari 2013 sebanyak 337 kasus. Hasil peramalan ini digunakan untuk meramalkan banyak kasus demam berdarah untuk 6 bulan kedepan.

B. Saran

Dalam penulisan skripsi ini untuk mengestimasi parameter regulasi dalam menentukan banyak *neuron* pada lapisan tersembunyi digunakan model RBFNN dengan metode *Global Ridge-Regression*. Pada penulisan skripsi yang lain dapat menggunakan metode lain seperti *Local Ridge-Regression*, *Forward Select*.

DAFTAR PUSTAKA

- Anton Sitio. (2008). Hubungan Perilaku tentang Pemberantasan Sarang Nyamuk dan Kebiasaan Keluarga dengan Kejadian Demam Berdarah *Dengue* di Kecamatan Medan Perjuangan Kota Medan Tahun 2008. *Tesis*. Undip.
- Anugerah PSW. (2007). Perbandingan Jaringan Syaraf Tiruan Backpropogation dan Metode Deret Berkala Box-Jenkins (ARIMA) sebagai Metode Peramalan Curah Hujan. *Skripsi*. UNES.
- Brodjol Sutijo. (2008). Jaringan Saraf Tiruan Fungsi Radial Basis untuk Pemodelan Data Runtutn Waktu. *Disertasi*. UGM.
- Cut I.N. Sari. (2005). Pengaruh Lingkungan terhadap Perkembangan Penyakit Malaria dan Demam Berdarah Dengue. *Tesis*. IPB.
- Demuth, Howard & Beale, Mark. (1992). *Neural Network Toolbox for use with MATLAB*. Hill Drive: The Math Works.
- Depkes RI. (2010). *Buletin Jendela Epidemiologi*. Jakarta.
- Depkes RI. (2012). *Profil Kesehatan Indonesia 2012*. Jakarta.
- Dinas Kesehatan Provinsi DIY. (2012). *Profil Kesehatan Daerah Istimewa Yogyakarta*. Yogyakarta.
- Fausett, Laurence. (1994). *Fundamentals of Neural Networks (Architectures, Algorithms, and Applications)*. Upper Saddle River, New Jersey: Prentice.
- Ferry Tan., Geovani Gracianti., Susanti., Steven., Samuel Lukas. (2012). Aplikasi Prediksi Harga Saham Menggunakan Jaringan Saraf Radial Basis Function dengan Metode Pembelajaran Hybrid. *Ilmu Komputer*. Vol. 8, No. 2, Maret. Hlm. 175-181.
- Hanke, J. E. & Wichern, D. W. (2005). *Business Forecasting*. 8th Edition. Upper Saddle River, New Jersey: Prentice Hall.
- Howlett, Robert J. & Jain, Lakhmi. (2001). *Radial Basis Function Networks 2*. Heidelberg: Springer.
- Hu, Yu Hen & Hwang, Jenq-Neng. (2001). *Handbook of Neural Network Signal Processing*. London: CRCPress.

- Indra Chahaya. (2003). *Pemberantasan Vektor Demam Berdarah di Indonesia*. Medan: USU digital library.
- Indrabayu., Nadjamuddin Harun., M. Saleh Pallu., Andani Achmad., Fikha C. L.(2012). Prediksi Curah Hujan dengan Jaringan Saraf Tiruan. *Prosiding*. Vol. 6, Desember. Hlm. 1-8.
- Indrawan. (2001). *Mengenal dan Mencegah Demam Berdarah*. Bandung: Pionir Jaya.
- Johnson, R. A. & Winchern, DW. (2007). *Applied Multivariate Statistical Analysis, 6th ed*. Upper Saddle River, New Jersey: Pearson.
- Marianne, J. Hopp & Jonathan, A. Foley. (2001). Global-scale Relationship Between Climate and the Dengue Fever Vektor, *Aedes Aegypti*. *Climatic Change*. **48**, 441-463.
- Montgomery, Douglas C., Jennings, Cheryl., Kulahci, Murat. (2008). *Introduction to Time Series Analysis and Forecasting*. Hoboken, New Jersey: Willey.
- Orr, M. J. L. (1996). *Introduction to Radial Basis Function Neural Networks*. Edinburgh: University of Edinburgh.
- Palit, A., K., & Popavic, D. (2005). *Computational Intelligence in Time Series Forecasting*. Glasgow: Springer.
- Shinta Prajna. (2012). Model Backpropogation Neural Network (BPNN) pada Peramalan Kasus Demam Berdarah di D. I. Yogyakarta. *Skripsi*. UNY.
- Singgih Santoso. (2009). *Bussines Forecasting (Metode Peramalan Bisnis Masa Kini dengan Minitab dan SPSS)*. Jakarta: PT. Alex Media Komputindo.
- Sri Kusumadewi. (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB dan EXCEL LINK*. Yogyakarta: Graha Ilmu.
- Tsay, Ruey S. (2010). *Analysis of Financial Time Series*. 3rd.ed. New York: Willey.
- Uswatun Khasanah. (2008). Forward Selection untuk Penentuan Model Radial Basis Function Neural Network (RBFNN) pada Data Time Series. *Tesis*. UGM.
- Wei, Shen., Xiaopen, Guo., Chao, Wu., Desheng, Wu. (2011). Forecasting Stock Indices using Radial Basis Function Neural Networks Optimized by Artificial Fish Swarm Algorithm. *Knowledge Based System*. **24**, 378-385.

- Wei, W. W. S. (2006). *Time Series Analysis Univariate and Multivariate Method. Second Edition*. New York: Pearson Education.
- Yeung, Daniel S., Cloete, Ian., Shi, Daming., Ng, Wing W., Y. (1998). *Sensitivity Analysis for Neural Network*. London: Springer.
- Yildiray, Turhan & Ozan, Toprakci. (2013). Comparison of High-Volume Instrument and Advanced Fiber Information System Based on Prediction Performance of Yarn Properties using a Radial Basis Function Neural Network. *Textile Research*. **83**, 130-147.
- Sri Utami Zuliana. (2012). Penerapan Global *Ridge-Regression* pada Peramalan Data *Time Series* nonlinear Studi Kasus: Pemodelan Nilai Tukar US Dolar terhadap Rupiah. *Tesis*. UGM.
- Zhang, G., P. (2004). *Neural Network in Bussines Forecasting*. Hershy: Idea Group Publishing.

LAMPIRAN

Lampiran 1

Data Banyak Kasus Demam Berdarah di D.I. Yogyakarta
 Januari 2005-Desember 2012
 (Sumber: Dinas Kesehatan Provinsi D.I. Yogyakarta)

Bulan	Jumlah Kasus	Bulan	Jumlah Kasus	Bulan	Jumlah Kasus
Januari 2005	45	September 2007	61	Mei 2010	491
Februari 2005	83	Oktober 2007	64	Juni 2010	463
Maret 2005	77	November 2007	96	Juli 2010	347
April 2005	71	Desember 2007	95	Agustus 2010	318
Mei 2005	60	Januari 2008	246	September 2010	260
Juni 2005	52	Februari 2008	333	Oktober 2010	207
Juli 2005	31	Maret 2008	270	November 2010	251
Agustus 2005	53	April 2008	330	Desember 2010	236
September 2005	49	Mei 2008	289	Januari 2011	236
Oktober 2005	76	Juni 2008	141	Februari 2011	175
November 2005	138	Juli 2008	75	Maret 2011	183
Desember 2005	268	Agustus 2008	41	April 2011	107
Januari 2006	523	September 2008	45	Mei 2011	80
Februari 2006	482	Oktober 2008	50	Juni 2011	54
Maret 2006	454	November 2008	81	Juli 2011	30
April 2006	195	Desember 2008	146	Agustus 2011	28
Mei 2006	99	Januari 2009	294	September 2011	12
Juni 2006	101	Februari 2009	216	Oktober 2011	22
Juli 2006	78	Maret 2009	179	November 2011	25
Agustus 2006	17	April 2009	225	Desember 2011	30
September 2006	34	Mei 2009	175	Januari 2012	58
Oktober 2006	49	Juni 2009	174	Februari 2012	64
November 2006	64	Juli 2009	183	Maret 2012	67
Desember 2006	50	Agustus 2009	173	April 2012	82
Januari 2007	252	September 2009	127	Mei 2012	89
Februari 2007	348	Oktober 2009	141	Juni 2012	88
Maret 2007	452	November 2009	139	Juli 2012	75
April 2007	264	Desember 2009	177	Agustus 2012	45
Mei 2007	283	Januari 2010	659	September 2012	37
Juni 2007	186	Februari 2010	626	Oktober 2012	78
Juli 2007	155	Maret 2010	674	November 2012	87
Agustus 2007	38	April 2010	589	Desember 2012	201

Lampiran 2

Program **rbfDesign**

(Sumber: Brodjol Sutijo, 2008)

```
function H = rbfDesign(X, C, R, options)
% H = rbfDesign(X, C, R, options)
% By M.J.Orr
% Gets the design matrix from the input data, centre
% positions and radii factors.
% Input
%   X   Input training data (n-by-p)
%   C   List of centres (n-by-m)
%   R   Scale factors: scalar, n-vector, or n-by-n matrix
% opt   Specifying basis function type ('g' for Gaussian,
%       'c' for Cauchy) and whether bias unit is required
%       (if yes then 'b').
% Output
%   H   Design matrix (p-by-m)
% default function type 'g'=gaussian (0)
% 'c'=cauchy (1)
% 'm'=multiquadric (2)
% 'i'=inverse multiquadric (3)

type=0;

% default bias
bias=0;
% process options
if nargin > 3
    for option = options;
        if option == 'g'
            type = 0;
        elseif option == 'c'
            type = 1;
        elseif option == 'm'
            type = 2;
        elseif option == 'i'
            type = 3;
        elseif option == 'b'
            bias = 1;
        else

```

```

        error('rbfDesign: illegal option')
    end
end
end

% preliminary sizing
[n,p] = size(X);
[n1,m]=size(C);
if n~=n1
    error('rbfDesign: mismatched X, C')
end
[rr,rc] = size(R);

% determine scaling type
if rr == 1 && rc == 1;
    SCALING_TYPE = 1;           %same radius for each
    centre
elseif rr == 1
    if rc == n
        SCALING_TYPE = 2;           %same diagonal metric for
    each centre
        R = R';
    elseif rc == m
        SCALING_TYPE = 4;           %different radius for
    each centre
        R = R';
    else
        error('rbfDesign: mismatched C and row vector R')
    end
elseif rc == 1
    if rr == n
        SCALING_TYPE = 2;           %same diagonal metric for
    each centre
    elseif rr == m
        SCALING_TYPE = 4;           %different radius for
    each centre
    else
        error('rbfDesign: mismatched C and row vector R')
    end
elseif rr == n
    if rc == n
        SCALING_TYPE = 3;           %same metric for each
    centre

```

```

elseif rc == m
    SCALING_TYPE = 5;
else
    error('rbfDesign: mismatched C and matrix R')
end
elseif rc == n
    if rr == m
        SCALING_TYPE = 5;           %different diagonal
metric for each centre
        R = R';
    else
        error('rbfDesign: mismatched C and matrix R')
    end
else
    error('rbfDesign: wrong size R')
end
% start constructing H
H = zeros(p, m);
for j = 1:m
    % get p difference vectors for this centre
    D = X - dupCol(C(:,j),p);
    % do metric calculation
    if SCALING_TYPE == 1           %same radius for each
centre
        s = diagProduct(D',D)/R^2;
    elseif SCALING_TYPE == 2      %same diagonal metric for
each centre
        DR = D./dupCol(R,p);
        s = diagProduct(DR',DR);
    elseif SCALING_TYPE == 3      %same metric for each
centre
        DR = R\D;
        s = diagProduct(DR',DR);
    elseif SCALING_TYPE == 4      %different radius for
each centre
        s = diagProduct(D',D)/R(j)^2;
    else                            %different diagonal
metric for each centre
        DR = D./dupCol(R(:,j),p);
        s = diagProduct(DR',DR);
    end
    %apply basis function
    if type == 0                    %Gaussian (default)

```

```

        h = exp(-s);
    elseif type == 1      %cauchy
        h = 1./(s+1);
    elseif type == 2      %multiquadric
        h = sqrt(s+1);
    elseif type == 3      %inverse multiquadric
        h = 1./sqrt(s+1);
    end
    %insert result in H
    H(:,j) = h;
end

% add bias unit
if bias
    H = [H ones(p,1)];
end

function M = dupCol(v,n)
% M = dupCol(v,n)
% Duplicates v, a column vector, n times. Returns the
% result as matrix M with n columns, each one a copy of v
% Inputs
%   v   a column vector (m-by-1)
%   n   a positive integer
% Output
%   M   a matrix (m-by-n) matrix

[~,c] = size(v);
if c~=1
    error('dupCol: input vector must be column')
end

M = v(:, ones(1,n));

function d = diagProduct(X,Y)
% d = diagProduct(X,Y)
% Output the diagonal of the product of X and Y.
% Faster than diag(X*Y).
% Input
%   X   matrix (m-by-n)
%   Y   matrix (n-by-m)
% Output
%   d   vector (m-by-1)

```

```

[m,n]=size(X);
[p,q]=size(Y);

if m~=q || n~=p
    error('diagProduct: bad dimensions')
end

% P - a column vector of the rows of X
P = X';
P = P(:);

% Q - a column vector of the columns of Y
Q = Y(:);

% Z - an [n,m] matrix containing the components of P.*Q
Z = zeros(n,m);
Z(:) = P.*Q;

% d - the answer is the sum of the columns of Z
d = colSum(Z)';

function s = colSum(X)
% s = colSum(X)
% Output a row vector whose elements are the sums of the
% columns of X.
% Designed to get round the feature of the standard
% routine (sum) of summing row vectors to a scalar.
% If colSum is handed a row vector, the same vector is
% given back.
% Input
% X matrix(m-by-n)
% Output
% s vector(1-by-n)

[m,~] = size(X);
if m > 1
    s = sum(X);
else
    s = X;
end
end

```

Lampiran 3

Program **globalRidge**

(Sumber: Sri Utami Zuliana, 2008)

```
function [l, e, L] = globalRidge(H, Y, l)
% [l, e, L] = globalRidge(H, Y, l)
% Calculates the best global ridge regression parameter
% (l) and
% the corresponding predicted error (e) using one of a
% number of
% prediction methods (UEV, FPE, GCV, or BIC).
% Needs a design (H), the training set outputs (Y),
% and an initial guess(l).
% The termination criterion, maximum number of
% iterations,
% verbose output and the use of a non-standard weight
% penalty
% are controlled from the options string. The non-
% standard
% metric, if used, is given in the fifth argument (U).
% L and E return the evolution of the regularisation
% parameter and error
% values from the initial to final iterations.
% If the input l is a vector (more than one guess),
% a corresponding number of
% answers will be returned, e will also be a vector and
% L and E
% will be matrices (with each row corresponding to the
% iterations
% resulting after each guess).
%
% Input
%
% H      design matrix (p-by-m)
% Y      input training data (p-by-k)
% l      initial guess(es) at lambda
%        (vector length q) (default          0.05)
% l      final estimate(s) for lambda (1-by-q)
% e      final estimate(s) for model selection score
%        (1-by-q)
% L      list(s) of running lambda values (n-by-q)
% defaults
```



```

Verbose = 0;
Flops = 0;
Model = 'g';
Threshold = 1000;
Hard = 100;
Standard = 1;

% process options
if nargin > 3

    % initialise
    i = 1;
    [arg, i] = getNextArg(options, i);

    % scan through arguments
    while ~isempty(arg)

        if strcmp(arg, '-v')

            % verbose output required
            Verbose = 1;

        elseif strcmp(arg, '-V')

            % verbose output required with compute cost
            reporting
            Verbose = 1;
            Flops = 1;

        elseif strcmp(arg, '-U')

            % non-standard penalty matrix
            Standard = 0;

        elseif strcmp(arg, '-h')

            % hard limit to specify
            [arg, i] = getNextArg(options, i);
            hl = str2double(arg);
            if ~isempty(hl)
                if hl > 1
                    Hard = round(hl);
                else

```

```

        fprintf('globalRidge: hard limit
                should be positive\n')
        error('globalRidge: bad value in -h
option')
    end
    else
        fprintf('globalRidge: value needed for
                hard limit\n')
        error('globalRidge: missing value in -h
option')
    end

elseif strcmp(arg, '-t')

    % termination criterion to specify
    [arg, i] = getNextArg(options, i);
    te = str2double(arg);
    if ~isempty(te)
        if te >= 1
            Threshold = round(te);
        elseif te > 0
            Threshold = te;
        else
            fprintf('globalRidge: threshold
                    should be positive\n')
            error('globalRidge: bad value in -t
option')
        end
    else
        fprintf('globalRidge: value needed for
                threshold\n')
        error('globalRidge: missing value in -t
option')
    end

elseif strcmpi(arg, 'uev')

    % use UEV (unbiased expected variance)
    Model = 'u';

elseif strcmpi(arg, 'fpe')

    % use FPE (final prediction error)

```

```

        Model = 'f';

elseif strcmpi(arg, 'gcv')

    % use GCV (generalised cross-validation)
    Model = 'g';

elseif strcmpi(arg, 'bic')

    % use BIC (Bayesian information criterion)
    Model = 'b';

else

    fprintf('%s\n', options)
    for k = 1:i-length(arg)-1
        fprintf(' '); end
    for k = 1:length(arg)
        fprintf('^'); end
    fprintf('\n')
    error('globalRidge: unrecognised option')

end

% get next argument
[arg, i] = getNextArg(options, i);

end

end
if nargin < 3
    l = 0.01; % default initial guess
end
if ~Standard
    if nargin < 5
        fprintf('globalRidge: specify non-standard
penalty matrix\n')
        error('globalRidge: -U option implies fifth
argument')
    end
else
    U = 1;
end
end

```

```

% initialise
[~, m] = size(H);
[p, ~] = size(Y);
[q1, q2] = size (l);
if q1 == 1
    q = q2;
elseif q2 == 1
    q = q1;
else
    error('globalRidge: list of guesses should be vector,
not matrix')
end
[u1, u2] = size(U);
if u1 == m && u2 == m
    %transform the problem - equivalent to U'*U metric
    H = H/U;
elseif u1 ~= 1 || u2 ~= 1
    estr = sprintf('%d-by-%d', m, m);
    error(['globalRidge: U should be 1-by-1 or' estr])
end
HH = H'*H;
HY = H'*Y;
e = zeros(1, q);
if nargout > 2
    L = zeros(Hard+1, q); end
if nargout > 3
    E = zeros(Hard+1, q); end
maxcount = 1;
if Verbose
    fprintf('\nglobalRidge\n')
end
if Flops
    flops(0)
end

% loop through each guess
for i = 1:q

    if Verbose
        fprintf('pass')
        fprintf(' lambda ')
        if Model == 'u'

```

```

        fprintf(' UEV ')
elseif Model == 'f'
        fprintf(' FPE ')
elseif Model == 'g'
        fprintf(' GCV ')
else
        fprintf(' BIC ')
end
fprintf(' change ')
if Flops
        fprintf(' flops\n')
else
        fprintf('\n')
end
end
notTooMany = 1;
notDone = 1;
count = 0;
A = inv(HH+l(i)*eye(m));
g = m-l(i)*trace(A);
PY = Y-H*((HH+l(i)*eye(m))\HY);
YPY = trace(PY'* PY);
if Model == 'u'
        psi = p/(p-g);
elseif Model == 'f'
        psi = (p+g)/(p-g);
elseif Model == 'g'
        psi = p^2/(p-g)^2;
else
        psi = (p+(log(p)-1)*g)/(p-g);
end
e(i) = psi*YPY/p;
if Verbose
        fprintf('%4d %9.3e %9.3e -',count, l(i), e(i))
        if Flops
                fprintf('%9d\n', flops)
        else
                fprintf('\n')
        end
end
end
if nargout > 2
        L(1,i) = l(i); end
if nargout > 3

```

```

E(1,i) = e(i); end

% re-estimate til convergence or exhaustion of
iterations
while notDone && notTooMany

    % next iteration
    count = count+1;

    % get some needed quantities
    A2 = A^2;
    A3 = A*A2;

    % re-estimate lambda
    if Model == 'u'
        eta = 1/(2*(p-g));
    elseif Model == 'f'
        eta = p/((p-g)*(p+g));
    elseif Model == 'g'
        eta = 1/(p-g);
    else
        eta = p*log(p)/(2*(p-g)*(p+(log(p)-1)*g));
    end
    nl = eta*YPY*trace(A-l(i)*A2)/trace(HY'*A3*HY);

    % store result
    if nargout > 2
        L(count+1, i) = nl; end

    % calculate new model selection score
    A = inv(HH+nl*eye(m));
    g = m-nl*trace(A);
    PY = Y-H*((HH+nl*eye(m))\HY);
    YPY = trace(PY'* PY);
    if Model == 'u'
        psi = p/(p-g);
    elseif Model == 'f'
        psi = (p+g)/(p-g);
    elseif Model == 'g'
        psi = p^2/(p-g)^2;
    else
        psi = (p+(log(p)-1)*g)/(p-g);
    end
end

```

```

ns = psi*YPY/p;

% store result
if nargout > 3
    E(count+1, i) = ns; end

% what's the change
if Threshold >= 1

    % interpret threshold as one part in many
    change = round(abs(e(i)/(e(i)-ns)));

else

    % interpret threshold as absolute difference
    change = abs(e(i)-ns);

end

% time to go home?
if count >= Hard

    notTooMany = 0;

elseif Threshold >= 1

    % interpret threshold as one part in many
    if change > Threshold
        notDone = 0;
    end
else

    % interpret threshold as absolute difference
    if change < Threshold
        notDone = 0;
    end

end

% get ready for next iteration (or end)
l(i) = nl;
e(i) = ns;

```

```

    if Verbose
        fprintf('%4d %9.3e %9.3e',count, l(i), e(i))
        if Threshold >= 1
            fprintf('%7d',change)
        else
            fprintf('%7.1e',change)
        end
        if Flops
            fprintf('%9d\n',flops)
        else
            fprintf('\n')
        end
    end
end

end

if Verbose
    if ~notTooMany
        fprintf('hard limit reached\n')
    else
        if Threshold >= 1
            fprintf('relative')
        else
            fprintf('absolute')
        end
        fprintf('threshold in')
    end
end
end
end

```


Lampiran 4

Program *RBFFN* untuk peramalan banyak kasus demam berdarah dengan 3 input 8 neuron.

(Sumber: Brodjol Sutijo, 2008)

```
% Program Model Jumlah Kasus Demam Berdarah di DIY
% Panggil data

load
D:\mArs\SKRIPSI\Data\data_DB_asli.txt;X0=data_DB_asli;
load D:\mArs\SKRIPSI\Data\kluster_8.txt;
M=kluster_8(1:3,:); SD=kluster_8(4,:);

% ambil data training n=77
Y=X0(4:58,:);X1=X0(3:57,:);X2=X0(2:56,:);X3=X0(1:55,:);
X=[X1 X2 X3];

% ambil data testing
[a,b]=size(X0);Yt=X0(59:a,:);
X1t=X0(58:a-1,:);X2t=X0(57:a-2,:);X3t=X0(56:a-3,:);
Xt=[X1t X2t X3t];

% ambil seluruh data
Yu=X0(4:a,:);X1u=X0(3:a-1,:);X2u=X0(2:a-
2,:);X3u=X0(1:a-3,:);
Xu=[X1u X2u X3u];

% Peramalan Data Training
H=rbfDesign(X',M,SD,'b');
lamb=globalRidge(H, Y, 0.05);
W=inv(H'*H+lamb*eye(9))*H'*Y;ft=H*W;e=Y-ft;Jkt=(Y-
mean(Y))'*(Y-mean(Y));
mpe=mean(abs(e./Y));sse=e'*e;MS=sse/50;R2t=1-(sse/Jkt);

% Peramalan Data Testing
Hv=rbfDesign(Xt',M,SD,'b');
ftv=Hv*W;ev=Yt-ftv;Jkv=(Yt-mean(Yt))'*(Yt-mean(Yt));
mpev=mean(abs(ev./Yt));ssev=ev'*ev;MSv=ssev/(a-
7);R2v=1-(ssev/Jkv);
```

```

% Peramalan Seluruh Data
Hu=rbfDesign(Xu',M,SD,'b');lamb=globalRidge(Hu, Yu,
0.05);
Wu=inv(Hu'*Hu+lamb*eye(9))*Hu'*Yu;ftu=Hu*Wu;eu=Yu-
ftu;Jku=(Yu-mean(Yu))'*(Yu-mean(Yu));
mpeu=mean(abs(eu./Yu));sseu=eu'*eu;MSu=sseu/(a-
8);R2u=1-(sseu/Jku);

% simpan hasil
MAPE=[mpe mpev mpeu];
SS=[sse ssev sseu];
MSt=[MS MSv MSu];
R2=[R2t R2v R2u];

%Plot
Yh=[ft;ftv];plot(Yu,'r');
Hold on;plot(Yh,'b');

```

Lampiran 5

Peramalan Banyak Basus Menggunakan *Ms. Excel*

Tabel 4.1. Nilai Pusat dan Varians *neuron* pada lapisan tersembunyi

8 kluster	Input	cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8
pusat	t-1	548,4	503	233,8667	151,5789	43,2727	45,4375	89,7333	277,4286
	t-2	395	554,1667	296,9333	175,9474	57,6364	39,4375	76,333	144,8571
	t-3	194,8	593,6667	340,8	196,2632	95,2727	35,5	63,0667	77,1429
varians		275,744	207,817	214,751	143,572	65,324	36,856	57,952	131,136

Tabel 4.2. Perhitungan peramalan banyak kasus demam berdarah

Prediksi Januari 2013											
		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
input t-1	201	-0,7936	-1,0559	-0,0117	-0,0592	-2,915	-8,9077	-1,8432	-0,1698		
input t-2	87	-0,6238	-2,5267	-0,4778	-0,1919	-0,101	-0,8327	-0,0169	-0,0973		
input t-3	78	-0,0897	-3,0785	-0,7488	-0,3393	-0,035	-0,6649	-0,0332	-2E-05		
jumlah		-1,5072	-6,6611	-1,2383	-0,5904	-3,051	-10,405	-1,8933	-0,2672		
exp		0,22154	0,00128	0,28988	0,5541	0,04731	3E-05	0,15057	0,76553		
bobot		-4,8856	-20,689	-77,31	-141,27	-30,714	47,4804	213,127	152,806	291,3835	
exp*bobot		-1,0823	-0,0265	-22,41	-78,277	-1,4532	0,00144	32,0913	116,977	291,3835	337,2037

Prediksi Februari 2013											
		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
input t	337,204	-0,2933	-0,3182	-0,1158	-0,8358	-10,123	-31,335	-9,1176	-0,1039		
input t-1	201	-0,2475	-1,444	-0,0998	-0,0152	-2,4083	-9,608	-2,3139	-0,0916		
input t-2	87	-0,0764	-2,972	-0,6984	-0,2896	-0,008	-0,9763	-0,0853	-0,0028		
jumlah		-0,6172	-4,7343	-0,9139	-1,1406	-12,539	-41,919	-11,517	-0,1984		
exp		0,53944	0,00879	0,40095	0,31962	3,6E-06	6,2E-19	1E-05	0,82007		
bobot		-4,8856	-20,689	-77,31	-141,27	-30,714	47,4804	213,127	152,806	291,384	
exp*bobot		-2,6355	-0,1818	-30,997	-45,153	-0,0001	3E-17	0,00212	125,312	291,384	337,73

Prediksi Maret 2013											
		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
input t+1	337,73	-0,2919	-0,3162	-0,117	-0,8405	-10,159	-31,448	-9,1564	-0,1057		
input t	337,204	-0,022	-0,545	-0,0176	-0,6308	-9,1579	-32,637	-10,132	-1,0757		
input t-1	201	-0,0003	-1,7851	-0,2119	-0,0005	-1,3098	-10,082	-2,8325	-0,446		
jumlah		-0,3141	-2,6463	-0,3464	-1,4718	-20,627	-74,166	-22,121	-1,6275		
exp		0,73047	0,07091	0,70721	0,2295	1,1E-09	6,2E-33	2,5E-10	0,19643		
bobot		-4,8856	-20,689	-77,31	-141,27	-30,714	47,4804	213,127	152,806	291,384	
exp*bobot		-3,5688	-1,4672	-54,674	-32,422	-3E-08	2,9E-31	5,3E-08	30,015	291,384	229,267

Prediksi April 2013											
		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
input t+2	229,267	-0,6697	-0,8675	-0,0002	-0,1464	-4,0534	-12,439	-2,8986	-0,0674		
input t+1	337,73	-0,0216	-0,5423	-0,018	-0,6349	-9,1924	-32,752	-10,173	-1,0816		
input t	337,204	-0,1334	-0,7615	-0,0001	-0,4818	-6,8582	-33,505	-11,188	-1,9664		

		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
exp		0,43839	0,11403	0,98175	0,28277	1,9E-09	6,6E-35	2,9E-11	0,04436		
bobot		-4,8856	-20,689	-77,31	-141,27	-30,714	47,4804	213,127	152,806	291,384	
exp*bobot		-2,1418	-2,3592	-75,899	-39,947	-6E-08	3,2E-33	6,2E-09	6,77816	291,384	177,815

Prediksi Mei 2013											
		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
input t+3	177,815	-0,9031	-1,2242	-0,0341	-0,0167	-2,121	-6,4503	-1,1551	-0,2885		
input t+2	229,267	-0,1806	-1,2221	-0,0496	-0,069	-3,4515	-13,264	-3,4821	-0,2072		
input t+1	337,73	-0,1343	-0,7584	-0,0001	-0,4854	-6,888	-33,622	-11,231	-1,9744		
jumlah		-1,2181	-3,2047	-0,0838	-0,5711	-12,461	-53,337	-15,869	-2,4701		
exp		0,2958	0,04057	0,91961	0,5649	3,9E-06	6,9E-24	1,3E-07	0,08458		
bobot		-4,8856	-20,689	-77,31	-141,27	-30,714	47,4804	213,127	152,806	291,384	
exp*bobot		-1,4452	-0,8394	-71,095	-79,804	-0,0001	3,3E-22	2,7E-05	12,9243	291,384	151,125

Prediksi Juni 2013											
		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
input t+4	151,125	-1,0379	-1,4335	-0,0742	-5E-06	-1,3629	-4,1115	-0,5611	-0,4638		
input t+3	177,815	-0,3102	-1,6398	-0,1538	-8E-05	-1,6923	-7,0483	-1,5332	-0,0316		
input t+2	229,267	-0,0078	-1,5373	-0,1349	-0,0264	-2,1038	-13,82	-4,1124	-0,6729		
jumlah		-1,3559	-4,6106	-0,3629	-0,0265	-5,159	-24,98	-6,2068	-1,1683		
exp		0,25773	0,00995	0,69564	0,97384	0,00575	1,4E-11	0,00202	0,3109		
bobot		-4,8856	-20,689	-77,31	-141,27	-30,714	47,4804	213,127	152,806	291,384	
exp*bobot		-1,2591	-0,2058	-53,779	-137,57	-0,1765	6,7E-10	0,42962	47,508	291,384	146,326

Prediksi Juli 2013											
		<i>neuron1</i>	<i>neuron2</i>	<i>neuron3</i>	<i>neuron4</i>	<i>neuron5</i>	<i>neuron6</i>	<i>neuron7</i>	<i>neuron8</i>	bias	Peramalan $\Sigma(\text{exp}*\text{bobot})$
input t+5	146,326	-1,0631	-1,4728	-0,0831	-0,0007	-1,2444	-3,7466	-0,4768	-0,4997		
input t+4	151,125	-0,3911	-1,8807	-0,2305	-0,0149	-1,0241	-4,5915	-0,8328	-0,0011		
input t+3	177,815	-0,0019	-2,0021	-0,288	-0,0083	-0,7983	-7,4551	-1,9603	-0,2947		
jumlah		-1,4561	-5,3556	-0,6016	-0,0239	-3,0668	-15,793	-3,2699	-0,7956		
exp		0,23315	0,00472	0,54794	0,97641	0,04657	1,4E-07	0,03801	0,45133		
bobot		-4,8856	-20,689	-77,31	-141,27	-30,714	47,4804	213,127	152,806	291,384	
exp*bobot		-1,1391	-0,0977	-42,361	-137,94	-1,4304	6,6E-06	8,10082	68,9657	291,384	185,484