

**PREDIKSI TRAFIK PADA KOMUNIKASI DATA DENGAN
MENGUNAKAN JARINGAN SYARAF TIRUAN
METODE *RADIAL BASIS FUNCTION***



TUGAS AKHIR

Disusun dalam Rangka Memenuhi Salah Satu Persyaratan

Untuk Menyelesaikan Program Strata-1 Jurusan Elektro

Universitas Hasanuddin

Makassar

Disusun Oleh :

MIMIN NAPSIATUL MUTMAINNAH

NUR AINUN

D411 08 012

D411 08 279

PROGRAM STUDI TEKNIK TELEKOMUNIKASI DAN INFORMASI

JURUSAN ELEKTRO FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

MAKASSAR

2012

LEMBAR PENGESAHAN TUGAS AKHIR

**PREDIKSI TRAFIK PADA KOMUNIKASI DATA DENGAN
MENGUNAKAN JARINGAN SYARAF TIRUAN METODE**

RADIAL BASIS FUNCTION

Disusun Oleh:

MIMIN NAPSIATUL MUTMAINNAH D411 08 012

NUR AINUN D411 08 279

Disusun dalam Rangka Memenuhi Salah Satu Persyaratan untuk Menyelesaikan
Program Strata-1 pada Subprogram Teknik Telekomunikasi dan Informasi
Jurusan Elektro Fakultas Teknik Universitas Hasanuddin
Makassar, Oktober 2012

Disahkan Oleh:

Pembimbing I

Pembimbing II

(Ir. H. Syafruddin Syarif, MT.)

(Indrabayu Amirullah, ST., MT., M.Bus.Sys.)

NIP. 19611125 198802 1 001

NIP. 19750716 200212 1 004

Mengetahui,

Ketua Jurusan Elektro

Fakultas Teknik Universitas Hasanuddin

(Dr.Ir. H. Andani Achmad, MT)

NIP. 19601231 198703 1 022

ABSTRAK

Akses informasi saat ini telah semakin berkembang seiring dengan semakin meningkatnya kebutuhan akan trafik data. Salah satu bagian dari perencanaan jaringan telekomunikasi adalah bagaimana memprediksi kebutuhan trafik di masa yang akan datang dengan mengacu kepada pola data sebelumnya diluar dari faktor eksternal seperti kondisi geografis dan sosial ekonomi masyarakat. Kebutuhan trafik data erat kaitannya dengan kapasitas atau *bandwidth* yang dibutuhkan untuk jaringan telekomunikasi. Universitas Hasanuddin adalah wilayah yang dipilih untuk menganalisis dan memprediksi kebutuhan trafiknya di masa yang akan datang.

Pada penelitian ini digunakan Jaringan Syaraf Tiruan metode *Radial Basis Function*, data yang digunakan dalam penelitian ini adalah data selama 3 bulan, dari rentang tanggal 29 Juli-29 September 2012 dari pukul 00.00-23.30 yang direkam oleh software monitoring trafik, CACTI PT. Telkom Indonesia. Tbk. data yang bersifat acak dan *time series* akan diproses pada jaringan *neural network* metode *Radial Basis Function* dengan mengelompokkan data trafik berdasarkan hari yang sama seperti hari senin dikelompokkan dengan hari senin untuk minggu-minggu berikutnya dari bulan juli- September 2012, begitu pula untuk data hari selasa, rabu, kamis ,jumat dan sabtu. Setelah di kelompokkan data akan dibagi menjadi data *inputan*, data *target*, dan data uji, setelah pengelompokan data- data tersebut maka dilakukan pembelajaran/ pelatihan dengan menggunakan Jaringan Syaraf Tiruan metode *Radial Basis Function* sehingga didapatkan nilai RMSE (*Root Mean Square Error*) terkecil.

Setelah melakukan pembelajaran dengan menggunakan Jaringan Syaraf Tiruan metode *Radial Basis Function* didapatkan tingkat keakuratan data latih trafik *inbound* sebesar 87,753% dan 90,73% untuk trafik *outbound*. Serta didapatkan tingkat keakuratan data uji trafik *inbound* sebesar 82,42% dan 72,48% untuk trafik *outbound*.

Kata kunci : *Trafik komunikasi data, Jaringan Syaraf Tiruan, Radial Basic Function.*

KATA PENGANTAR

Assalamu 'alaikum Warahmatullahi Wabarakatuh.

Puji syukur penulis panjatkan kehadirat Allah SWT yang dengan limpahan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul : “**Analisis Trafik Pada Komunikasi Data Dengan Menggunakan Jaringan Syaraf Tiruan metode (*Radial Basic Function*)**”

Tugas Akhir ini disusun sebagai salah satu syarat yang harus dipenuhi untuk menyelesaikan pendidikan tahap sarjana di Jurusan Elektro, Fakultas Teknik, Universitas Hasanuddin. Berbagai masalah yang bermunculan dalam penyusunan Tugas Akhir ini bukanlah halangan yang memutuskan semangat bagi penyusun, namun semua itu merupakan tantangan yang harus dihadapi dan diatasi dengan baik dalam rangka peningkatan dan pengembangan ilmu pengetahuan dan teknologi.

penulis menyadari bahwa selama penyusunan tugas ini banyak mendapat sumbangan pikiran serta bimbingan baik moral maupun materil dari berbagai pihak. Sehingga pada kesempatan ini penulis menyampaikan ucapan terima kasih sedalam-dalamnya kepada:

1. Kedua orang tua penyusun, **Drs. A. M Agussalim dan Sry Wedari Husain S.Pd, (Alm) H. Salim Abbas** dan **San'ah** beserta seluruh keluarga besar penulis atas doa restu, dukungan, nasehat, dan motivasinya. Semoga Allah membalasnya Aamiin.

2. Saudara- saudara penulis tercinta K'Nunu, K'Cece, K'Adin, k'Saleh, k'Evi, K'Eva, Nurul, Rijal dan Dandi yang telah memberikan kesempatan kepada penulis untuk belajar menjadi adik dan kakak yang baik.
3. Bapak **Ir. H. Syafruddin Syarif, MT** selaku pembimbing I dan bapak **Indrabayu Amirullah, ST.,MT.,M.Bus.Sys** selaku pembimbing II yang dengan penuh kesabaran telah meluangkan waktu, tenaga dan pikirannya selama membimbing dan mengarahkan penulis dalam pembuatan tugas akhir ini.
4. Bapak **Dr.Ir. H. Andani Achmad, MT** selaku ketua Jurusan Teknik Elektro Fakultas Teknik Universitas Hasanuddin.
5. Bapak **Dr. Elyas Palentei ST, M. Eng** selaku ketua Konsentrasi Teknik Telekomunikasi dan Informasi Jurusan Teknik Elektro.
6. Seluruh staf dosen jurusan yang telah banyak membagikan ilmu kepada penulis selama masa perkuliahan.
7. Seluruh staf akademik Jurusan Elektro Fakultas Teknik Universitas Hasanuddin atas pelayanannya kepada penulis.
8. Teman-teman 'SPYWARE 08' atas dukungan dan semangat persahabatannya kepada penulis, kalian tidak akan terlupakan.
9. Rekan-rekan TELKOM_GENG, NEO Electrical Media, COOP PT.TELEKOMUNIKASI INDONESIA, Manegemen Daerah BEASTUDI ETOS Makassar, KAMMI Komsat UNHAS, MADZ OKJE FT_UH, KPAJ Makassar.

10. Tim_ungu yang selalu membagi Semangat dan tak henti meberikan nasehat-nasehat kepada penulis.
11. Senior dan adik-adik yang telah melengkapi hidup penulis di Jurusan Elektro Fakultas Teknik Universitas Hasanuddin.
12. Teman-teman KKN Reguler angkatan 80 kecamatan sinjai tengah desa Saotanre dan kecamatan Ujungloe desa sepang bulukumba.
13. Seluruh pihak yang tidak bisa disebutkan satu persatu terimakasih atas waktu dan tenaga dalam membantu penulis menyelesaikan tugas akhir ini.

Harapan penulis, semoga Tugas Akhir ini dapat bermanfaat bagi para penuntut ilmu, baik dalam perkuliahan maupun penelitian, guna membina generasi muda penerus bangsa yang lebih berkualitas dan berdaya saing.

Akhirnya dengan segala kerendahan hati, penulis menyadari masih terdapat kekurangan dalam penulisan tugas ini baik isi maupun cara penyajian, oleh karena itu penulis mengharapkan adanya kritik dan saran yang bersifat membangun demi kesempurnaan tugas ini.

Makassar, Oktober 2012

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xii

BAB I PENDAHULUAN

I.1 Latar Belakang	I-1
I.2 Perumusan Masalah	I-2
I.3 Maksud dan Tujuan Penelitian	I-3
I.4 Batasan Masalah	I-3
I.5 Metodologi Penelitian	I-4
I.6 Sistematika Penelitian	I-5

BAB II LANDASAN TEORI

II.1 <i>Komunikasi Data</i>	II-1
II.1.1 <i>Jaringan Komunikasi Data</i>	II-2
II.1.2 <i>Transmisi Analog dan digital</i>	II-4
II.1.3 <i>Permasalahan transmisi Komunikasi data</i>	II-5
II.2 <i>Multi Router Traffic Grapher (MRTG)</i>	II-7
II.2.1 <i>Fitur-fitur Multi Router Traffic Grapher</i>	II-9
II.2.2 <i>Topologi ME unhas</i>	II-10

II.3 <i>Artificial Neural Network</i>	II-11
II.3.1 Konsep Dasar <i>Neural Network</i>	II-15
II.3.2 Arsitektur Jaringan <i>Neural Network</i>	II-17
II.3.3 Fungsi Aktifasi <i>Neural Network</i>	II-19
II.3.4 Metode Pelatihan/Pembelajaran <i>Neural Network</i>	II-22
II.3.5 <i>Radial Basis Function Neural Network</i>	II-24

BAB III IMPLEMENTASI DAN SIMULASI JARINGAN

III.1 Pengambilan Data.....	III-2
III.2 Pembelajaran Sistem <i>Radial Basis Function Neural Network</i>	III-3
III.2.1 Pengolahan Data <i>Inputan</i>	III-4
III.2.2 Normalisasi dan Denormalisasi.....	III-5
III.2.3 Pelatihan Sistem <i>Radial Basis Function Neural Network</i>	III-6
III.2.4 Validasi data aktual trafik data.....	III-8
III.3 Prediksi Penggunaan Trafik Data dengan Sistem <i>Radial Basis Function Neural Network</i>	III-10

BAB IV PENGUJIAN DAN ANALISIS

IV.1. Analisis Pengolahan Variabel <i>Input</i> dan <i>Target</i> pada <i>Radial Basis Function Neural Network</i>	IV-2
IV.2. Analisis Hasil Output Jaringan untuk prediksi minggu terakhir bulan September Menggunakan Metode <i>RBFNN</i>	IV-7
IV.3 Analisis Hasil Prediksi Trafik Data Mingguan bulan Oktober 2012 Menggunakan Metode <i>Radial Basis Function Neural Network</i>	IV-23

BAB V PENUTUP

V.1 Simpulan V.1

V.2 Saran..... V.2

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

Gambar II.1	Sinyal Analog	II-3
Gambar II.2	Sinyal <i>digital</i>	II-3
Gambar II.3	Tampilan Grafik pada MRTG	II-9
Gambar II.4	ME CACTI PT. Telkom Indonesia. Tbk.....	II-10
Gambar II.5	Topologi BTS ME Universitas Hasanuddin	II-11
Gambar II.6	Sebuah Sel Jaringan Syaraf Tiruan	II-13
Gambar II.7	Model Tiruan Sebuah <i>Neuron</i>	II-16
Gambar II.8	Arsitektur <i>Neural Network Single Layer</i>	II-18
Gambar II.9	Arsitektur <i>Neural Network Multi Layer</i>	II-19
Gambar II.10	Arsitektur <i>Neural Network Competitive Layer</i>	II-19
Gambar II.11	Fungsi <i>Sigmoid Unipolar</i>	II-21
Gambar II.12	Fungsi <i>Sigmoid Bipolar</i>	II-22
Gambar II.13	Topologi <i>Radial Basis Function</i>	II-24
Gambar III.1	Alur Penelitian	III-1
Gambar III.2	Hasil Rekaman <i>software</i> monitoring trafik, CACTI.....	III-2
Gambar III.3	Hasil Rekaman pertahun <i>software</i> monitoring trafik, CACTI... III-2	
Gambar III.4	Flowchart Perancangan Sistem <i>Radial Basis Function</i>	III-4
Gambar. IV.1	Hasil <i>Output</i> jaringan RBFNN trafik <i>inbound</i> data latih	IV-10
Gambar IV.2	<i>Output</i> jaringan RBFNN trafik <i>outbond</i> dengan data latih.....	IV-11
Gambar IV.3	Grafik data latih aktual dan data prediksi trafik <i>inbound</i> RBFNN tanggal 23-29 September 2012.....	IV-13
Gambar IV.4	Grafik data aktual dan data prediksi trafik <i>outbound</i> RBFNN. IV-14	
Gambar IV.5	Hasil <i>output</i> jaringan RBFNN trafik <i>inbound</i> untuk data uji , Senin	

24/09/2012.....	IV-18
Gambar IV.6 Hasil <i>output</i> jaringan RBFNN trafik <i>outbond</i> untuk data uji , Senin, 24 Sept 2012	1V-19
Gambar IV.7 Hasil Data Uji metode RBFNN tanggal 23-29 Sept 2012	IV-21
Gambar IV.8 Grafik Hasil simulasi data Uji trafik <i>outbond</i> , Senin-Sabtu (23-29 September 2012)	
Gambar IV.9 Hasil simulasi data Uji trafik <i>inbound</i> , 1-6 Oktober 2012.....	65
Gambar IV.10 Hasil simulasi data Uji trafik <i>inbound</i> , 1-6 Oktober 2012.....	66

DAFTAR TABEL

Tabel IV.1	Data Latih <i>input Inbound</i> hari senin	IV-3
Tabel IV.2	Data Latih <i>input Outbound</i> hari senin	IV-4
Tabel IV.3	Hasil <i>normalisasi</i> trafik data latih <i>inbound</i> , Senin	IV-5
Tabel IV.4	Data target Senin, 24 September 2012.....	IV-6
Table VI.5	Hasil <i>Output</i> jaringan RBFNN dengan data latih	IV-9
Tabel IV.6	Nilai RMSE per hari hasil simulasi sistem untuk data latih	IV-12
Tabel.IV.7	Data uji <i>input</i> trafik <i>inbound</i> , hari Senin.....	IV-16
Tabel IV.8	Hasil Prediksi Trafik <i>inbound</i> RBFNN dengan Data Uji	IV-17
Tabel IV.9	Hasil Prediksi Trafik <i>inbound</i> RBFNN dengan Data Uji	IV-20
Tabel IV.10	RMSE trafik <i>Inbound</i> dan <i>Outbond</i> untuk data Uji	IV-21
Tabel IV.11	Data Uji sebagai <i>input inbound</i> hari Senin	IV-25

BAB I

PENDAHULUAN

I.1 Latar Belakang Masalah

Komunikasi data di Indonesia saat ini telah menjadi satu kebutuhan yang pokok, terutama bagi perusahaan-perusahaan bisnis maupun institusi pendidikan. Komunikasi dapat diartikan sebagai cara untuk menyampaikan atau menyebarkan data dan informasi, sedangkan informasi berarti berita, pikiran, pendapat dalam berbagai bentuk. Berbagai inovasi teknologi dalam mengemas informasi ditawarkan seiring dengan peningkatan kualitas layanan.

Adanya tingkat kebutuhan kapasitas di setiap wilayah yang berbeda-beda maka dibutuhkan analisis jaringan yang matang. Analisis jaringan biasanya digunakan untuk tiga hal seperti : penyelesaian masalah (*troubleshooting*) pada jaringan, optimasi performa/kinerja jaringan, perencanaan dan pengujian (*planning/ testing*) jaringan.

Dalam penyelenggaraan jasa telekomunikasi diperlukan prediksi akan kebutuhan kapasitas jaringan agar kualitas dari layanan telekomunikasi bisa lebih maksimal. Prediksi atau peramalan adalah proses menentukan apa yang akan terjadi ke depannya dengan referensi akan kejadian di masa lalu. Trafik data yang bersifat acak menyulitkan kita untuk menentukan bagaimana perubahan trafik di masa yang akan datang. Dengan mengacu kepada pola data setiap harinya, jaringan syaraf tiruan dengan metode *Radial Basis Function* diharap mampu

menyelesaikan persoalan prediksi dengan cara yang singkat dan sesuai dengan hasil yang diinginkan.

Jika dalam monitoring dan prediksi trafik secara umum dilakukan dengan metode *konvensional*, pada tugas akhir kali ini akan diperlihatkan bagaimana Jaringan Syaraf Tiruan mampu mengolah data trafik dan membantu dalam menganalisis tingkat kebutuhan trafik dalam suatu jaringan. Penelitian ini akan menganalisis, merancang dan mengimplementasi untuk mendapatkan konfigurasi jaringan syaraf tiruan yang terbaik dalam penggunaannya untuk mengklasifikasi data.

I.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan beberapa permasalahan di antaranya :

1. Bagaimana membuat suatu sistem yang dapat memprediksi trafik data dengan jaringan syaraf tiruan *Radial Basis Function*.
2. Sulitnya pihak-pihak yang terkait dalam memprediksi trafik untuk masa yang akan datang.
3. Masih kurangnya sistem yang memprediksi trafik pada komunikasi data memanfaatkan aplikasi dari jaringan syaraf tiruan yang dapat membantu pihak-pihak yang terkait dalam melakukan.
4. Sejauh mana tingkat keakuratan sistem prediksi dengan metode *Radial Basis Function* dapat digunakan dalam memprediksi trafik data.

I.3 Tujuan Penelitian

1. Merancang jaringan syaraf tiruan yang dapat memberikan tingkat pengenalan terbaik dalam sistem prediksi jumlah trafik pada komunikasi data.
2. Mengetahui tingkat kebutuhan trafik data di masa yang akan datang dengan memprediksi perubahan trafik menggunakan metode *Radial Basis Function neural network*.
3. Melihat pola kebiasaan pelanggan pada waktu apa saja kepadatan trafik meningkat.

I.4 Batasan masalah

Agar masalah yang dibahas pada Tugas Akhir ini tidak terlalu meluas dan tidak menyimpang dari topik yang ada, maka penulis perlu membatasi masalah sebagai berikut :

1. Membahas mengenai trafik data pada jaringan UNHAS dengan bantuan software monitoring trafik Cacti
2. Prediksi trafik pada komunikasi data menggunakan jaringan syaraf Tiruan (*neural network*) dengan *Radial basis Function* karena kemudahan dalam penggunaannya, terutama untuk memprediksi.
3. Data yang digunakan untuk pelatihan dan prediksi adalah data trafik *Inbound* dan *Outbound* wilayah ME-UNHAS dari PT. Telkom Indonesia dari bulan juli 2012 sampai dengan September 2012.
4. Peramalan bersifat harian, tidak memperhitungkan hari besar.

5. Analisis yang dilakukan tidak memperhitungkan faktor – faktor pendukung penyebab perubahan jumlah Trafik. Analisis dilakukan hanya berdasarkan data jumlah Trafik yang didapatkan saat proses *survey*/ pengambilan data.

I.5 Metodologi Penelitian

Metode yang akan digunakan adalah:

1. Tempat dan Periode Waktu Penelitian

Penelitian dilakukan di Jurusan Elektro Fakultas Teknik Universitas Hasanuddin selama 7 bulan mulai dari bulan Mei 2012 sampai bulan Oktober 2012.

2. Studi Literatur

Bertujuan untuk mempelajari dasar teori dari literatur-literatur yang berkaitan tentang:

- Komunikasi Data
- Teori *Trafik*
- Jaringan Syaraf Tiruan (*Neural Network*)
- Pemrograman Matlab untuk Peramalan Trafik dan Jaringan Syaraf Tiruan (*Neural Network*).

3. Perancangan Sistem

Perancangan sistem Analisis Prediksi Trafik yang tersusun dari desain sistem, termasuk di dalamnya pengumpulan data, penginputan data, dan pengolahan data melalui *neural network*.

4. Pembuatan dan Pengujian Sistem

Dalam tahap ini dilakukan pembuatan dan pengujian sistem yang telah dirancang untuk melihat sejauh mana keakuratan sistem yang telah dirancang.

5. Penganalisaan terhadap hasil pengujian sistem dari proses pengimputan data trafik yang diujikan melalui metode *neural network*.

6. Membuat simpulan mengenai hasil yang dicapai.

I.6 Sistematika Penulisan

Laporan Tugas Akhir ini disusun dengan menggunakan sistematika sebagai berikut :

BAB I PENDAHULUAN

Pembahasan tentang latar belakang penulisan, perumusan masalah, maksud dan tujuan penelitian, batasan masalah, metodologi penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Pembahasan teori dasar tentang prinsip-prinsip dasar metode *neural network*, pelatihannya, khususnya pelatihan *Radial Basis Function* serta pengetahuan dasar mengenai *Komunikasi data dan Trafik*

BAB III PERANCANGAN DAN PEMBUATAN SISTEM

Pembahasan tentang deskripsi umum sistem, spesifikasi perangkat, data yang diteliti, serta alur perancangan dan pembuatan sistem yang dapat memprediksi trafik komunikasi data

BAB IV PENGUJIAN DAN ANALISIS

Pembahasan mengenai hasil dari sistem yang telah dibuat, termasuk di dalamnya pengujian sistem dan menganalisis hasil pengujian dari sistem prediksi data trafik dengan menggunakan *neural network* metode *radial basis function* yang telah dibuat.

BAB V PENUTUP

Merupakan bagian terakhir yang merangkum semua hasil yang diperoleh pada saat penyusunan Tugas Akhir dan beberapa saran yang mungkin akan diperlukan untuk penyempurnaan pembahasan dalam Tugas Akhir ini.

BAB II

LANDASAN TEORI

II.1 Komunikasi data

Teknologi komunikasi terus dikembangkan dengan tujuan memudahkan manusia dalam melakukan komunikasi. Para ahli terdorong untuk mengembangkan teknik telekomunikasi jarak jauh yang lebih *efisien* dengan metode telekomunikasi yang memanfaatkan teknologi elektronika, yang dikenal dengan istilah teknik komunikasi data.

Komunikasi data merupakan cara mengirimkan data menggunakan sistem transmisi elektronik dari satu komputer ke komputer lain atau dari satu komputer ke terminal tertentu. Sedangkan data itu sendiri merupakan sinyal elektromagnetik yang dibangkitkan oleh sumber data yang dapat ditangkap dan dikirimkan ke terminal penerima.

II.1.1 Jaringan Komunikasi Data

Beberapa klasifikasi dari jaringan Komunikasi data[1]:

1. *Wide Area Networks* (WAN)

Wide area networks umumnya mencakup area geografis yang luas sekali, melintas jalan umum, dan perlu juga menggunakan fasilitas umum. Biasanya, suatu WAN terdiri dari sejumlah node penghubung. Suatu transmisi dari suatu perangkat diarahkan melalui node-node atau persimpangan-persimpangan internal ini menuju perangkat tujuan yang dituju. Node-node ini tidak berkaitan dengan isi data, melainkan dimaksudkan untuk menyediakan fasilitas *switchin-*

yang akan memindah data dari suatu node ke node yang lain sampai mencapai tujuan.

Biasanya, WAN diimplementasikan menggunakan satu dari dua teknologi ini: *circuit switching*. Sedangkan saat ini, frame relay dan jaringan ATM juga telah memiliki peranan penting.

2. *Local Area Network (LAN)*

Sama halnya dengan WAN, LAN merupakan suatu jaringan komunikasi yang saling menghubungkan berbagai jenis perangkat dan menyediakan pertukaran data di antara perangkat-perangkat tersebut:

Terdapat beberapa perbedaan utama diantara LAN dan WAN:

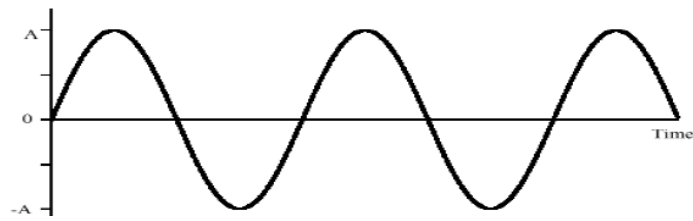
- a. Lingkup LAN kecil. Biasanya meliputi bangunan tunggal atau sekelompok gedung. Perbedaan dalam cakupan area geografis ini membawa pada solusi-solusi teknis yang berbeda pula.
- b. Merupakan hal yang umum dimana LAN dimiliki oleh suatu organisasi yang sama yang juga menguasai semua peralatan. Sedangkan untuk WAN, hanya untuk beberapa kasus tertentu saja. Suatu jaringan yang tidak dimiliki siapapun. Hal ini dua implikasi. Pertama, perawatan harus benar-benar dilakukan bila memilih LANs, sebab kemungkinan adanya investasi modal yang substansial (dibandingkan dengan *dial up* atau *leased charges* untuk WAN) dalam hal pembelian dan *maintenance*. Kedua, manajemen jaringan LAN bertanggung jawab sampai dengan tingkat *user*.

- c. Tingkat kecepatan data *internal* LAN biasanya lebih besar daripada tingkat kecepatan data *internal* WAN.

II.1.2 Transmisi Analog dan Digital

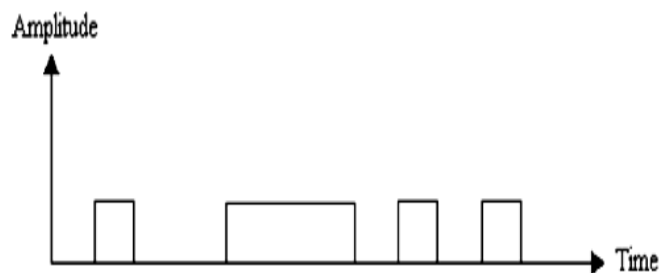
Transmisi data dibagi menjadi dua, yaitu transmisi *analog* dan *digital*.

1. Sinyal analog juga disebut dengan *broadband*, merupakan gelombang-gelombang elektronik yang bervariasi dan secara *kontinu* di transmisikan melalui beragam media tergantung frekuensinya. Sinyal analog bisa diubah ke bentuk sinyal digital dengan dimodulasi terlebih dahulu.



Gambar II.1 Sinyal Analog[2].

2. Sinyal Digital juga disebut dengan *baseband*, memuat denyut *voltase* yang ditransmisikan melalui media kawat.



Gambar II.2 Sinyal Digital[2].

Perbedaan antara dua tipe sinyal ini di antaranya :

Analog :

- a. Dirancang untuk suara (*voice*)

- b. Tidak *efisien* untuk data.
- c. Banyak terdapat *noise* dan rentan kesalahan (*error*).
- d. Kecepatannya *relative* rendah.
- e. *Overhead* tinggi
- f. Setiap sinyal analog dapat dikonversi ke bentuk digital.

Digital :

- a. Dirancang untuk data dan suara.
- b. Informasi *discrete-level*.
- c. Kecepatan tinggi.
- d. *Overhead* rendah.
- e. Setiap sinyal digital dapat dikonversi ke analog.

II.1.3 Permasalahan Transmisi Komunikasi Data

Pada sistem komunikasi apapun, sinyal yang diterima akan selalu berbeda dengan sinyal yang dikirim. Pada sinyal analog, hal ini berarti dihasilkan variasi pada modifikasi *random* yang berakibat pada penurunan kualitas sinyal.

Pada sinyal digital, terjadi kesalahan bit. Artinya, biner '1' akan menjadi biner '0' dan sebaliknya.

Kelemahan-kelemahan tersebut secara umum adalah[2]:

1. Atenuasi dan distorsi oleh atenuasi
2. Distorsi oleh penundaan
3. Noise.

II.1.4 Media Transmisi Komunikasi data

Media transmisi pada komunikasi data merupakan hal yang sangat penting mengingat data atau informasi yang dikirimkan harus mempunyai media untuk menyampaikan ke si penerima. Media transmisi data pada komunikasi data dapat dibagi menjadi dua bagian, yaitu[3]:

1. Media transmisi *guided*: merupakan media kasat mata yang mentransmisikan sekaligus memandu gelombang untuk menuju pada tujuan.

Contoh media transmisi *guided* :

➤ Media transmisi kabel

Kabel merupakan media transmisi yang sudah sejak dulu digunakan dalam sistem komunikasi data. Kabel merupakan jenis media transmisi *guided* yang mentransmisikan sekaligus memandu arah pengiriman data. Komunikasi berbasis kabel mungkin dilakukan jika jarak antara pengirim dan penerima tidak terlalu jauh dan berada dalam area local. Media kabel sering digunakan dalam jaringan telepon dan jaringan computer local. Secara umum terdapat tiga jenis kabel yang digunakan sebagai media transmisi data, yaitu:

1. Kabel *twisted pair*
 2. Kabel *koaksial*
 3. Serat optik
2. Media transmisi *un-guided*: berfungsi untuk mentransmisikan data tetapi tidak bertugas sekaligus sebagai pemandu yang mengarahkan ke tujuan transmisi.

Beberapa contoh media transmisi yang termaksud *Un-guided*, yaitu:

➤ Media Transmisi tanpa kabel (*Wireless*)

Tidak semua transmisi dilakukan menggunakan kabel. Ada transmisi yang dilakukan tanpa menggunakan kabel, yaitu yang dikenal sebagai *wireless*. Banyak contoh penerapan teknologi *wireless* dalam kehidupan sehari-hari, seperti LAN *wireless* yang menggunakan frekuensi radio, inframerah untuk berkomunikasi antar perangkat, *bluetooth* dan lain sebagainya. Gelombang radio untuk komunikasi ini terdiri dari berbagai frekuensi seperti:

1. HF (*High Frequency*)
2. VHF (*Very High Frequency*)
3. UHF (*Ultra High frequency*)

Pada media *wireless*, transmisi dan penangkapan dilakukan melalui sebuah alat yang disebut antenna. Untuk *transmisi*, antenna menyebarkan energi elektromagnetik ke dalam media (biasanya udara). Sedangkan untuk penerimaan sinyal, antena menangkap gelombang elektromagnetik dari media. Transmisi jenis ini juga disebut transmisi *wireless*, yaitu searah dan ke segala arah. Untuk konfigurasi segala arah, sinyal yang ditransmisikan menyebar ke segala penjuru dan diterima oleh banyak antena.

Ada tiga jangkauan frekuensi dalam transmisi *wireless*. Frekuensi dengan jangkauan sebesar 2 GHz sampai 40GHz, disebut sebagai frekuensi gelombang mikro. Pada frekuensi ini dimungkinkan untuk

menghasilkan sinar searah yang sangat tinggi. Gelombang mikro sesuai untuk transmisi titik ke titik.

Gelombang mikro juga digunakan untuk komunikasi satelit, memiliki jangkauan sebesar 30MHz sampai 1GHz. Frekuensi ini sesuai untuk alokasi segala arah. Jangkauannya biasa disebut sebagai siaran radio. Jangkauan lainnya adalah 300 GHz sampai 200THz (terra hertz). Frekuensi ini sesuai untuk aplikasi lokal, yaitu inframerah. Inframerah berguna untuk aplikasi multitik dari titik ke titik lokal di daerah yang terbatas, misalnya ruangan tunggal.

II. 2 *Multi Router Traffic Grapher*

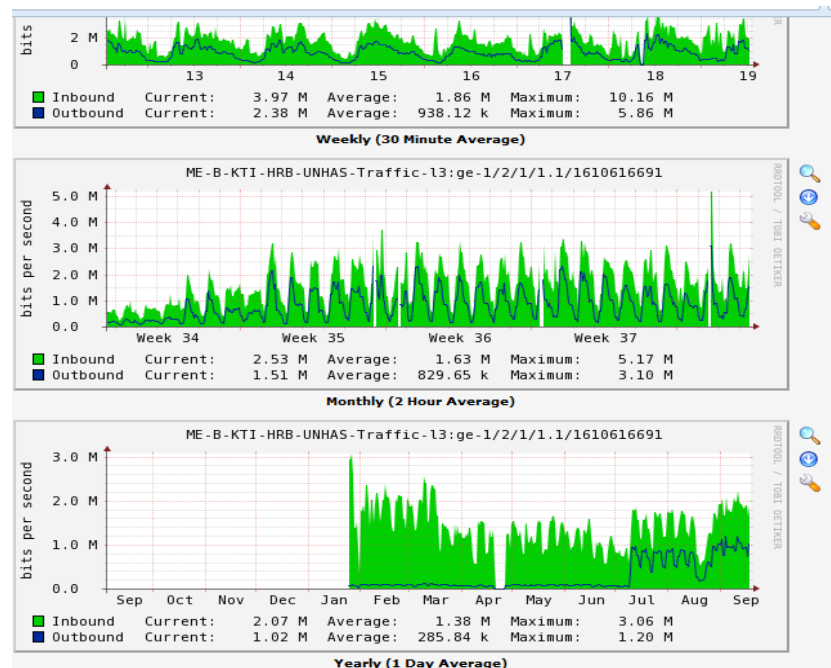
Multi router Traffic Grapher (MRTG) merupakan aplikasi yang dapat digunakan untuk memonitoring *traffic load* dalam suatu jaringan MRTG dapat digunakan oleh seorang *user* untuk melihat *traffic load* yang terdapat pada jaringan pada kurun waktu tertentu dalam bentuk tampilan grafik[4].

MRTG dibuat dengan menggunakan bahasa *perl*, dan dapat berjalan pada beberapa sistem operasi seperti Unix/Linux, Windows, dan Netware. Pada awalnya, MRTG dibuat oleh Tobias oetiker dan dave Rand untuk memonitor *trafik router*. Selanjutnya, aplikasi ini dikembangkan sebagai *tool* yang dapat menghasilkan grafik dan statistic dari trafik jaringan . dalam *Multi router Traffic Grapher* menggunakan SNMP dan MIB, SNMP (*Simple Network management protokol*) adalah *Internet protocol Suite* yang dibuat oleh *Internet Engineering Task Force* (IETF) pada sekitar tahun 1988. Tujuan awal diciptakannya protocol

SNMP ini adalah untuk mengatur berbagai device yang semakin banyak seiring dengan berkembangnya jaringan internet. SNMP merupakan protocol dari lapis aplikasi yang digunakan untuk *network management system* untuk memonitor perangkat jaringan sehingga dapat memberikan informasi yang dibutuhkan bagi pengelolanya. Dan yang dimaksud dengan MIB (*Management Informasi Base*) merupakan sekumpulan informasi yang teratur tentang keberadaan seluruh peralatan jaringan . informasi-informasi tersebut akan diambil oleh *agen* dan diberikan kepada manajer SNMP berdasarkan permintaan.

MRTG menggunakan SNMP untuk mengirimkan dua buah objek *identifiers* (OIDs) ke sebuah perangkat. Perangkat ini, harus dapat mendukung SNMP, dan kemudian MIB akan mencari OIDs yang telah dispesifikasi. Selanjutnya, MRTG akan menghasilkan HTML dari log yang dihasilkan, yang didalamnya berisi daftar trafik dalam bentuk detail grafik untuk perangkat yang dimaksud.

MRTG mengoleksi informasi dari statistic trafik jaringan dan menghasilkan grafik dalam bentuk *web pages* sehingga mudah dan menarik untuk diamati. Gambar II.3 merupakan gambar penampilan grafik yang direkam oleh software monitoring trafik, CACTI.



Gambar II.3 Tampilan Grafik pada MRTG[4].

II.2 Fitur-fitur *Multi router Traffic Grapher*

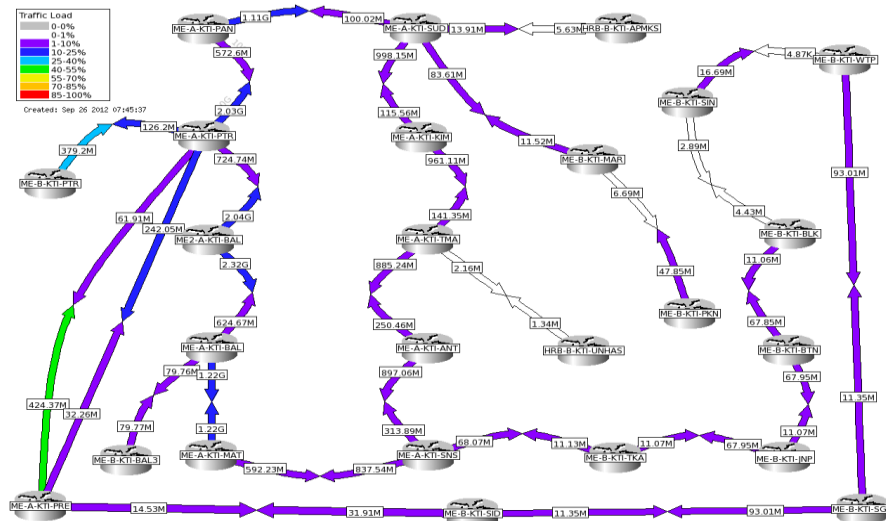
Sebagai aplikasi monitoring yang banyak digunakan para system administrator saat ini, MRTG sudah memiliki cukup banyak fitur. Beberapa fitur yang dimiliki antara lain[5]:

1. Dapat mengukur dua buah nilai (1 *input*, 0 untuk *output*).
2. Mengambil data dengan menggunakan SNMP agen, atau melalui *output* yang dihasilkan dari *command line*.
3. Dapat menghasilkan laporan data setiap lima menit sekali.
4. Menghasilkan sebuah HTML per target, yang menghasilkan tampilan gambar dalam bentuk grafik
5. Menghasilkan laporan berdasarkan periode waktu tertentu (hari, minggu, bulan, tahun).

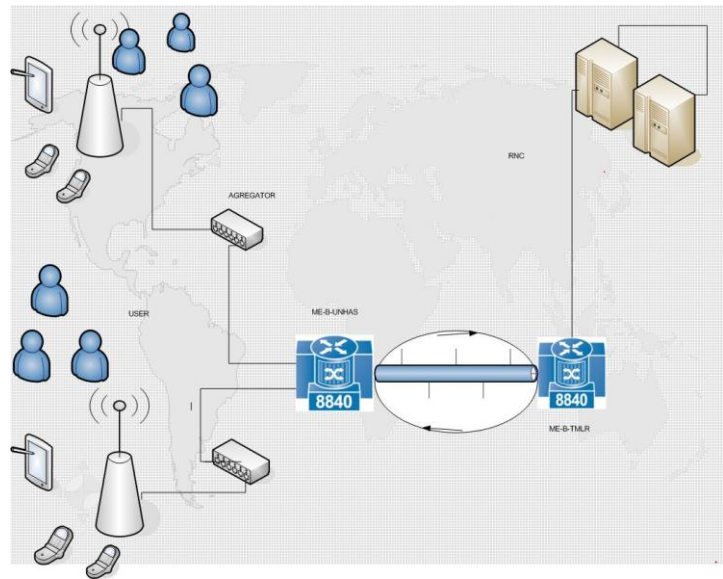
6. Secara otomatis dapat menghasilkan skala Y axis dari grafik, untuk menampilkan grafik secara *detail*.
7. Dapat mengirimkan pesan warning via *e_mail*, jika terdapat suatu ancaman dalam jaringan yang tidak diinginkan.

II. 3 Topologi ME Universitas Hasanuddin

Gambar II.4 dan Gambar II.5 merupakan topologi dari ME Universitas Hasanuddin yang didapatkan dari PT. Telkom Indonesia. Tbk.



Gambar II.4 ME Cacti PT. Telkom Indonesia. Tbk



Gambar II.5 Topologi BTS ME Unhas

II.3 Artificial Neural Network

Artificial Neural Network atau Jaringan Syaraf Tiruan dibuat pertama kali oleh McCulloch dan Pitts pada tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa *neuron* sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang dipakai adalah fungsi *threshold*. Selanjutnya pada tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan baru yang terdiri dari beberapa lapisan yang disebut *Perceptron*. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasinya. Widrow dan Hoff pada tahun 1960 mengembangkan *Perceptron* dengan memperkenalkan aturan pelatihan jaringan, yang dikenal sebagai aturan *delta* (atau sering disebut kuadrat rata-rata terkecil). Aturan ini akan mengubah bobot *perceptron* apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan. Apa yang dilakukan

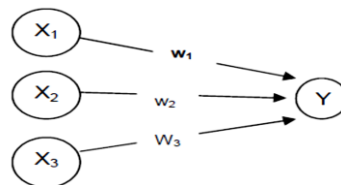
peneliti terdahulu hanya menggunakan jaringan dengan layer tunggal (*single layer*). Rumelhart bersama McClelland pada tahun 1986 mengembangkan *Perceptron* menjadi *backpropagation*, yang memungkinkan jaringan diproses melalui beberapa layer. Beberapa model jaringan syaraf tiruan juga dikembangkan oleh Kohonen pada tahun 1972 yang mengembangkan arsitektur jaringan syaraf tiruan model Kohonen dan Hopfield yang mengembangkan arsitektur jaringan syaraf tiruan model Hopfield pada tahun 1982. Pengembangan yang ramai dibicarakan sejak tahun 1990-an adalah aplikasi model-model jaringan syaraf tiruan yang digunakan untuk menyelesaikan berbagai masalah di dunia nyata[6].

Jaringan Syaraf Tiruan (*Neural Network*) merupakan paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem syaraf biologi, sama seperti otak yang memproses suatu informasi dimana terdiri dari unit-unit pemroses terkecil yang disebut *neuron*. Jaringan syaraf tiruan dibentuk untuk memecahkan suatu masalah seperti pengenalan pola, klasifikasi, atau deteksi karena proses pembelajaran.

Jaringan syaraf tiruan, seperti manusia, belajar dari suatu contoh karena mempunyai karakteristik yang adaptif, yaitu dapat belajar dari data-data sebelumnya dan mengenal pola data yang selalu berubah. Selain itu, *neural network* merupakan sistem tak terprogram, artinya semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran/pelatihan.

Neuron dalam jaringan syaraf tiruan sering diganti dengan istilah simpul. Setiap simpul tersebut berfungsi untuk menerima atau mengirim sinyal dari atau

kesimpul-simpul lainnya. Pengiriman sinyal disampaikan melalui penghubung. Kekuatan hubungan yang terjadi antara setiap simpul yang saling terhubung dikenal dengan nama bobot. Arsitektur jaringan dan algoritma pelatihan sangat menentukan model-model jaringan syaraf tiruan. Arsitektur tersebut gunanya untuk menjelaskan arah perjalanan sinyal atau data di dalam jaringan. Sedangkan algoritma belajar menjelaskan bagaimana bobot koneksi harus diubah agar pasangan masukan-keluaran yang diinginkan dapat tercapai. Dalam setiap perubahan harga bobot koneksi dapat dilakukan dengan berbagai cara, tergantung pada jenis algoritma pelatihan yang digunakan. Dengan mengatur besarnya nilai bobot ini diharapkan bahwa kinerja jaringan dalam mempelajari berbagai macam pola yang dinyatakan oleh setiap pasangan masukan-keluaran akan meningkat. Sebagai contoh, perhatikan *neuron* Y pada gambar berikut :



Gambar II.6 Sebuah Sel Jaringan Syaraf Tiruan[6].

Y menerima *input* dari *neuron* x_1 , x_2 , dan x_3 dengan bobot hubungan masing-masing adalah w_1 , w_2 dan w_3 . Ketiga impuls *neuron* yang ada dijumlahkan $net = x_1w_1 + x_2w_2 + x_3w_3$. Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(net)$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

Jaringan Syaraf Tiruan ditentukan oleh 3 hal yaitu[7]:

1. Pola hubungan antar *neuron* (disebut arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung (metode *training/learning*).
3. Fungsi aktivasi, yaitu fungsi yang digunakan untuk menentukan keluaran suatu *neuron*.

Beberapa kelebihan *Neural Network* :

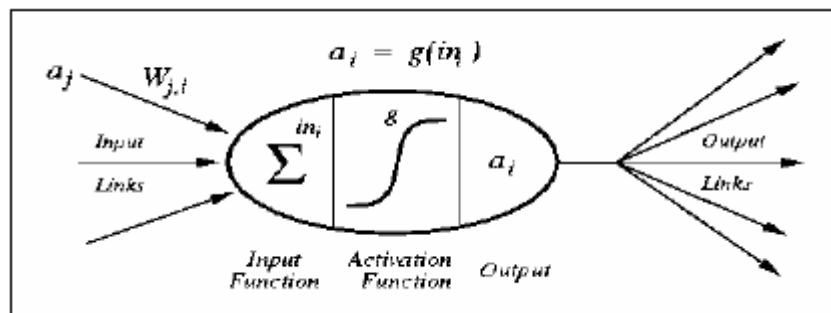
1. *Adaptif Learning* : Suatu kemampuan untuk melakukan sesuatu kegiatan yang didasarkan atas data yang diberikan pada saat pembelajaran / pelatihan atau dari pengalaman sebelumnya.
2. *Self Organization* : Dapat membuat organisasi sendiri atau merepresentasi yang didapat pada saat pembelajaran / pelatihan.
3. *Real Time Operation*, dapat melakukan perhitungan paralel dan dengan *device hardware* yang khusus dibuat akan memberikan keuntungan dengan adanya kemampuan tersebut.
4. *Fault Tolerance* melalui *Redundant Information Coding*: Kerusakan pada bagian tertentu pada jaringan akan mengakibatkan penurunan kemampuan, tetapi beberapa jaringan mempunyai kemampuan untuk menahan akibat dari kerusakan besar dari jaringan.
5. Kelebihan *Neural Network* terletak pada kemampuan belajar yang dimilikinya. Dengan kemampuan tersebut pengguna tidak perlu merumuskan kaidah atau fungsinya. *Neural Network* akan belajar mencari sendiri kaidah atau fungsi tersebut. Dengan demikian Jaringan Syaraf Tiruan (*Neural Network*) mampu digunakan untuk menyelesaikan masalah

yang rumit dan atau masalah yang terdapat pada kaidah atau fungsi yang tidak diketahui.

6. Dapat memecahkan problema non-linear yang umum di jumpai di aplikasi
7. Kemampuan memberikan jawaban terhadap *pattern* yang belum pernah dipelajari (*generalization*).
8. Dapat secara otomatis mempelajari data numerik yang diajarkan pada jaringan tersebut.
9. Kemampuan Jaringan Syaraf Tiruan dalam menyelesaikan masalah yang rumit telah dibuktikan dalam berbagai macam penelitian.

II.3.1 Konsep Dasar Neural Network

Tiruan *neuron* dalam struktur jaringan saraf tiruan adalah sebagai elemen pemroses berfungsi seperti halnya sebuah *neuron*. Sejumlah sinyal *input* dikalikan dengan masing-masing penimbang yang bersesuaian w . Kemudian dilakukan penjumlahan dari seluruh hasil perkalian tersebut dan *output* yang dihasilkan dilakukan kedalam fungsi pengaktif untuk mendapatkan tingkatan derajat sinyal *outputnya* $F(a,w)$. Walaupun masih jauh dari sempurna, namun kinerja dari tiruan *neuron* ini identik dengan kinerja dari sel biologi yang kita kenal saat ini[8].



Gambar II.7 Model Tiruan Sebuah *Neuron*[8].

Dimana:

- a_j : Nilai aktivasi dari unit j
- $w_{j,i}$: Bobot dari unit j ke unit i
- in_i : Penjumlahan bobot dan *input* ke unit i
- g : Fungsi aktivasi
- a_i : Nilai aktivasi dari unit i

Misalkan ada n buah sinyal *input* dan n buah penimbang, fungsi *output* dari *neuron* adalah seperti persamaan berikut,

$$in_i = \sum_j W_{j,i} \cdot a_j \quad \dots(2-10)$$

Kumpulan dari *neuron* dibuat menjadi sebuah jaringan yang akan berfungsi sebagai alat komputasi. Jumlah *neuron* dan struktur jaringan untuk setiap problema yang akan diselesaikan adalah berbeda.

Setiap pola-pola informasi *input* dan *output* yang diberikan kedalam *neural network* diproses dalam *neuron*. *Neuron-neuron* tersebut terkumpul di dalam lapisan-lapisan yang disebut *neuron layers*.

Lapisan tersebut dapat dibagi [7]:

1. Lapisan *input*

Unit-unit di dalam lapisan input disebut unit-unit *input*. Unit-unit *input* tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan.

2. Lapisan tersembunyi

Unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi.

Dimana outputnya tidak dapat secara langsung diamati.

3. Lapisan *output*

Unit-unit di dalam lapisan *output* disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi *neural network* terhadap suatu permasalahan.

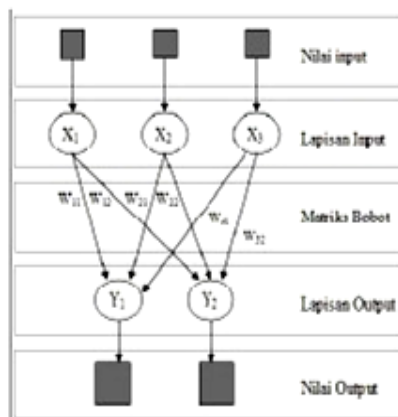
II.3.2 Arsitektur Jaringan Neural Network

Neural Network memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur *Neural Network* tersebut, antara lain[7]:

1. Jaringan layar tunggal (*single layer network*)

Jaringan dengan lapisan tunggal terdiri dari 1 layer *input* dan 1 layer *output*. Setiap *neuron/unit* yang terdapat di dalam lapisan/layer *input* selalu terhubung dengan setiap *neuron* yang terdapat pada layer *output*. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi.

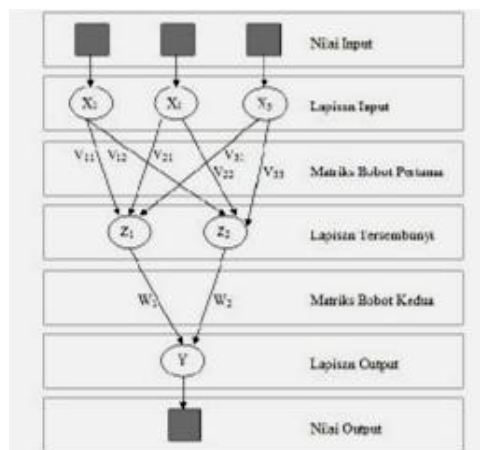
Contoh algoritma JST yang menggunakan metode ini yaitu : *ADALINE*, *Hopfield*, *Perceptron*.



Gambar II.8 Arsitektur *Neural Network Single Layer*[7].

2. Jaringan layar jamak (*multi layer network*)

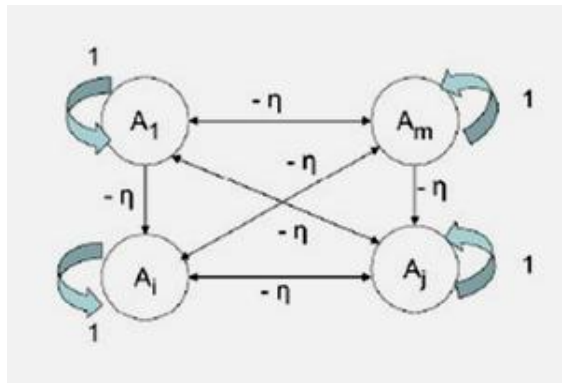
Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis layer yakni layer *input*, layer *output*, dan juga layer tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma JST yang menggunakan metode ini yaitu : *MADALINE*, *backpropagation*, *Neocognitron*.



Gambar II.9 Arsitektur *Neural Network Multi Layer* [7].

3. Jaringan dengan lapisan kompetitif (*competitive layer network*)

Pada jaringan ini sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Contoh algoritma yang menggunakan metode ini adalah LVQ.



Gambar II.10 Arsitektur *Neural Network Competitive Layer* [7].

II.3.3 Fungsi Aktivasi Neural Network

Dalam *Neural Network*, fungsi aktivasi digunakan untuk menentukan keluaran suatu *neuron*. Banyak fungsi yang dipakai sebagai fungsi aktivasi, seperti fungsi-fungsi goniometri dan hiperboliknya, fungsi unit step, impuls, sigmoid, tetapi yang paling lazim digunakan adalah fungsi sigmoid.

Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya). Jika $net = \sum xiwi$, maka fungsi aktivasinya adalah $f(net) = f(\sum xiwi)$.

Beberapa fungsi aktivasi yang digunakan adalah :

a) Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1 \dots x \geq a & \text{jika } x \geq a \\ 0 \dots x \leq a & \text{jika } x \leq a \end{cases} \quad \dots(2-11)$$

Untuk beberapa kasus, fungsi *threshold* yang dibuat tidak berharga 0 atau 1, tapi berharga -1 atau 1 (sering disebut *threshold bipolar*). Jadi

$$f(x) = \begin{cases} 1 \dots x \geq a \\ -1 \dots x \leq a \end{cases} \quad \dots(2-11)$$

Adakalanya dalam JST ditambahkan suatu unit masukan yang nilainya selalu 1. Unit tersebut dikenal dengan bias. Bias dapat dipandang sebagai sebuah *input* yang nilainya selalu 1. Bias berfungsi untuk mengubah *threshold* menjadi = 0.

b) Fungsi *sigmoid*

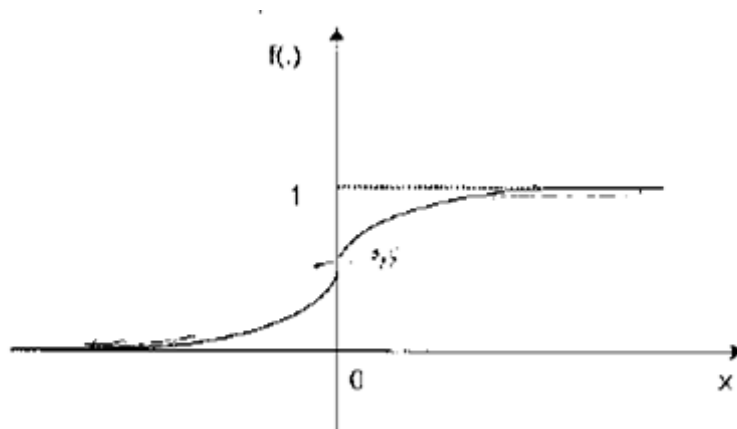
$$f(x) = \frac{1}{1+e^{-x}} \quad (2-12)$$

Fungsi ini sering digunakan karena dianggap lebih mendekati kinerja sinyal pada otak manusia dan nilai fungsinya yang sangat mudah untuk didiferensiasikan seperti :

$$f'(x) = f(x)(1 - f(x)) \quad (2-13)$$

Ada dua jenis fungsi sigmoid, yaitu *unipolar* dan *bipolar*. Fungsi sigmoid unipolar dituliskan pada persamaan (2-14) dan dinyatakan pada gambar II.8

$$f(x) = \left(\frac{1}{1+e^{-f(x)}}\right) \quad \dots(2-14)$$

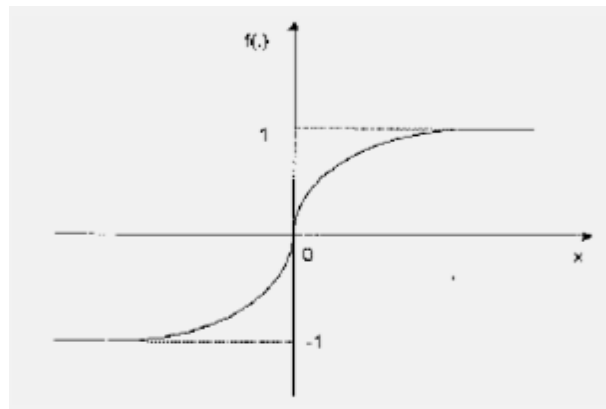


Gambar II.11 Fungsi *Sigmoid Unipolar*

Sedangkan fungsi aktivasi sigmoid bipolar adalah persamaan 2-15 atau 2-16. Persamaan 2-16 disebut juga persamaan tangen hiperbolik, dan bentuk fungsinya seperti pada gambar II.12.

$$y = \left[\frac{1 - e^{-f(x)}}{1 + e^{-f(x)}} \right] \quad \dots(2-15)$$

$$y = \left[\frac{e(x) - e(-x)}{e(x) + e(-x)} \right] \quad y = \left[\frac{1 - e(-2x)}{1 + e(-2x)} \right] \quad \dots(2-16)$$



Gambar II.12 Fungsi Sigmoid Bipolar

c) Fungsi identitas

$$f(x) = x \quad \dots(2.17)$$

Fungsi identitas sering digunakan jika diinginkan keluaran jaringan berupa bilangan riil acak (bukan hanya pada range 0 dan 1 atau 1 dan -1).

II.3.4 Metode Pelatihan/Pembelajaran Neural Network

Cara berlangsungnya pembelajaran atau pelatihan JST dikelompokkan menjadi 3 yaitu[8] :

a) *Supervised learning* (Pembelajaran Terawasi)

Pada metode ini, setiap pola yang diberikan kedalam JST telah diketahui outputnya. Selisih antara pola *output* aktual (*output* yang dihasilkan) dengan pola *output* yang dikehendaki (*output* target) yang disebut *error* digunakan untuk mengoreksi bobot JST sehingga JST mampu menghasilkan *output* sedekat mungkin dengan pola target yang telah diketahui oleh JST. Contoh algoritma JST yang menggunakan metode ini adalah : *Hebbian, Perceptron, ADALINE, Boltzman, Hopfield, Backpropagation*.

b) *Unsupervised learning* (Pembelajaran Tak Terawasi)

Pada metode ini, tidak memerlukan target *output*. Pada metode ini tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi pola. Contoh algoritma JST yang menggunakan metode ini adalah : *Competitive, Hebbian, Kohonen, LVQ (Learning Vector Quantization), Neocognitron*.

c) *Hybrid Learning* (Pembelajaran Hibrida)

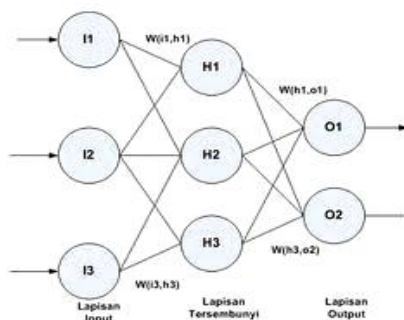
Merupakan kombinasi dari metode pembelajaran *supervised learning* dan *unsupervised learning*. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma JST yang menggunakan metode ini yaitu algoritma *RBF*.

II.3.5 Radial Basis Function Neural Network

Radial Basis Function adalah model *neural network* yang mentransformasi input secara nonlinear dengan menggunakan fungsi aktivasi Gaussian pada lapisan unit *hidden* sebelum diproses secara linear pada lapisan output.

Dalam penerapannya, model *Radial Basis Function* mengandung sejumlah parameter (*weight*) yang harus ditaksir. Untuk mendapatkan *model Radial Basis Function* yang sesuai, perlu menentukan kombinasi yang tepat antara jumlah variabel *input*, jumlah node (*cluster*) pada unit *hidden layers*, nilai tengah dan standar deviasi (skala atau *width*) dari variabel input pada setiap *node*, yang berimplikasi pada jumlah parameter yang optimal.

Seperti halnya jaringan saraf tiruan yang lain, *Radial Basis Function* juga memiliki topologi jaringan. Topologi milik *Radial Basis Function* terdiri atas unit lapisan masukan (*input*), unit lapisan tersembunyi (*hidden*), dan unit lapisan keluaran (*output*).



Gambar II.13 Topologi *Radial Basis Function*[8].

Pada jaringan radial basis function, pemrosesan sinyal dari *input layer* ke *hidden layer* bersifat nonlinier, sedangkan dari *hidden layer* ke *ouput layer* sifatnya *linear*. Pada *hidden layer* digunakan sebuah fungsi aktivasi yang berbasis *radial*, misalnya fungsi Gaussian. Fungsi Gaussian bisa dituliskan sebagai berikut,

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad \sigma > 0 \quad (2-18)$$

di mana σ adalah nilai *spread*

$$\sigma = \frac{\text{jarak maksimum antara 2 pusat}}{\sqrt{\text{banyaknya pusat}}} = \frac{d \max}{\sqrt{m1}}$$

Nilai *spread* menentukan bagaimana data tersebar. Jika nilai *spread* makin besar sensitivitas antardata semakin berkurang.

Radial Basis Function bersifat *feed-forward* dimana jaringan *feed-forward* adalah jaringan yang sinyalnya bergerak dari *input* kemudian melewati lapisan tersembunyi dan akhirnya mencapai unit *output* (mempunyai struktur perilaku yang stabil).

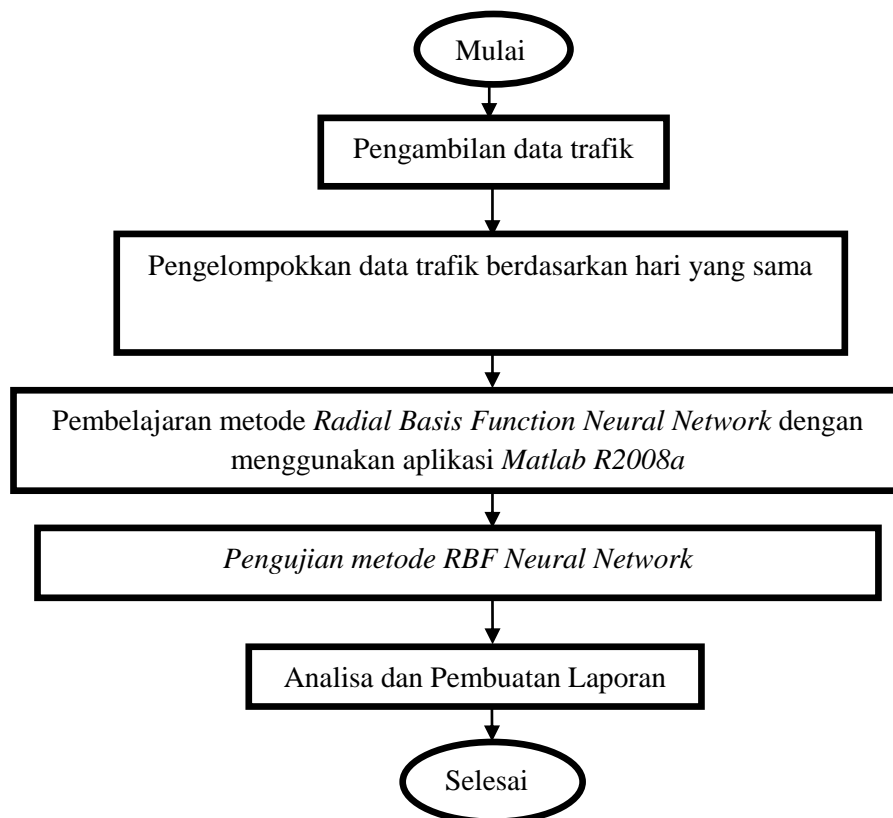
Tipe jaringan *feed-forward* mempunyai sel syaraf yang tersusun dari beberapa lapisan. Lapisan *input* bukan merupakan sel syaraf. Lapisan ini hanya memberi pelayanan dengan mengenalkan suatu nilai dari suatu variabel. Lapisan tersembunyi dan lapisan *output* sel syaraf terhubung satu sama lain dengan lapisan sebelumnya. Kemungkinan yang timbul adalah adanya hubungan dengan beberapa unit dari lapisan sebelumnya atau terhubung semuanya.

Berguna atau setidaknya suatu jaringan saraf tiruan ditentukan dari hasil pelatihannya yang berupa bobot neuronnya. *Radial Basis Function* memiliki algoritma pelatihan yang agak unik karena terdiri atas metode *supervised* dan *unsupervised* sekaligus, dimana metode *supervised* merupakan metode yang setiap pola yang diberikan ke dalam jaringannya telah diketahui outputnya. Selisih antara pola *output* aktual (*output* yang dihasilkan) dengan pola *output* yang dikehendaki (*output target*) yang disebut *error* digunakan untuk mengoreksi bobot jaringan. Sedangkan metode *unsupervised* adalah metode yang tidak membutuhkan target *output*. Pada metode ini tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Kombinasi antara 2 jenis metode ini pada jaringan *Radial Basis Function* menghasilkan suatu sistem yang handal dalam mengatasi ketidaklinearan sistem itu sendiri.

BAB III

IMPLEMENTASI DAN SIMULASI JARINGAN

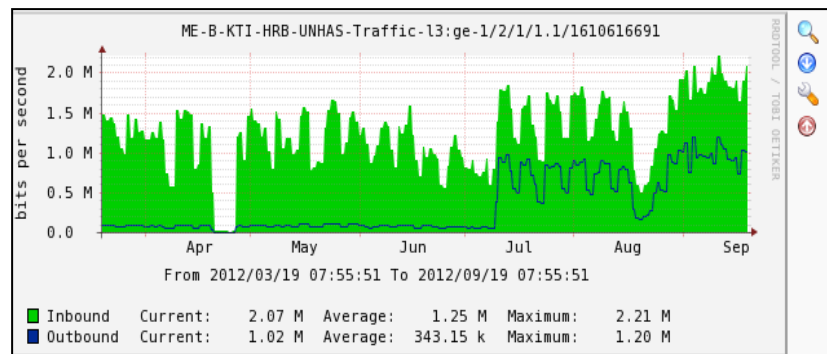
Pada bab ini akan dibahas tentang prosedur implementasi dan pembuatan metode prediksi trafik data *forecasting* dengan metode *Radial Basis Function Neural Network* yang nantinya akan digunakan untuk memprediksi besarnya pemakaian trafik data pada jaringan di wilayah Universitas Hasanuddin. Trafik data ini diambil dari PT. Telkom Indonesia, Tbk. Adapun alur penelitian dapat dilihat dibawah ini :



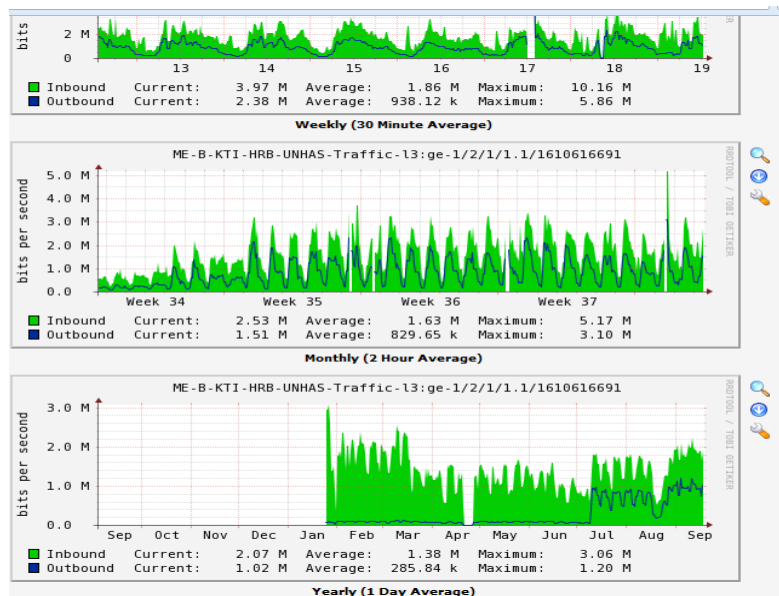
Gambar III.1 Alur Penelitian

III.1 Pengambilan Data

Tahap pertama dalam proses prediksi trafik adalah pengambilan data yang diperoleh dari NETRE KTI. Data adalah data pemakaian trafik dengan rentang tanggal 29 Juli-29 September 2012 dari pukul 00.00-23.30 yang direkam oleh software monitoring trafik, CACTI.



Gambar III.2 hasil Rekaman software monitoring trafik, CACTI



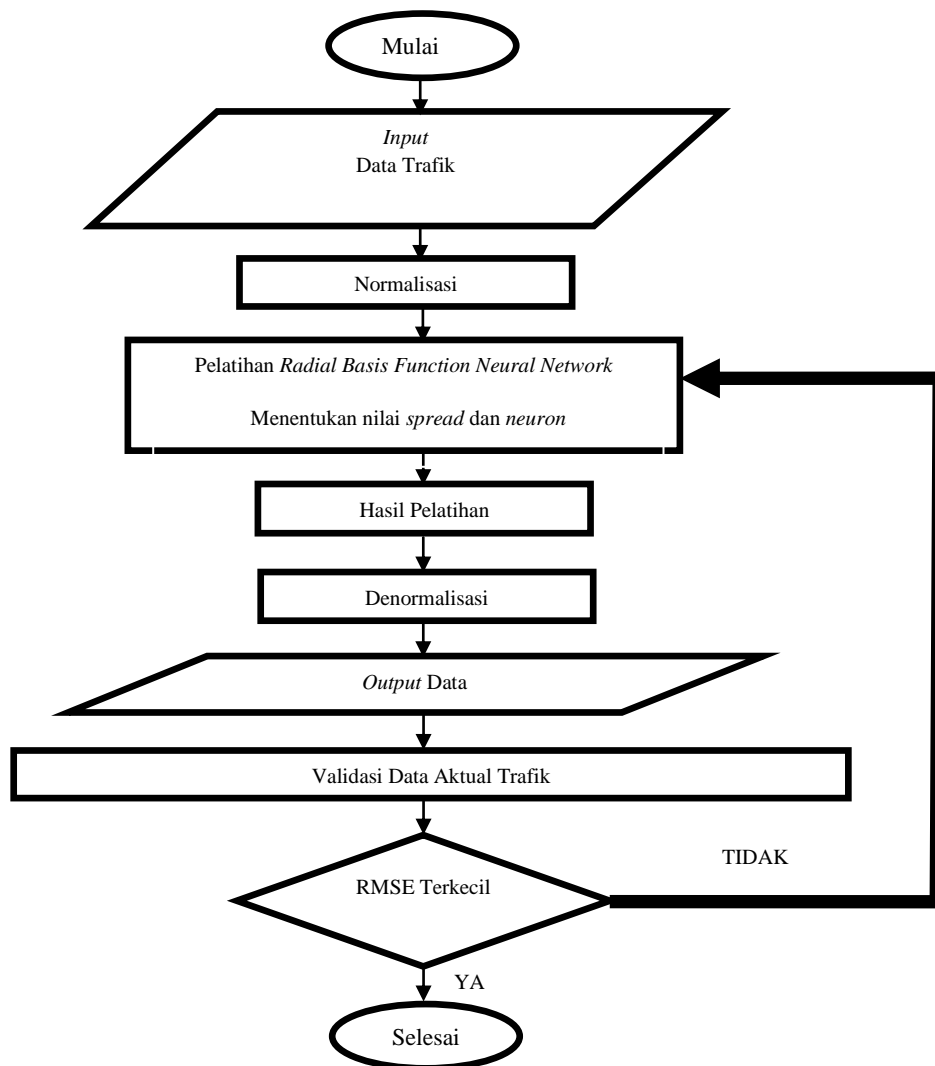
Gambar III.3 hasil Rekaman pertahun software monitoring trafik, CACTI

Data trafik ini mencakup pemakaian trafik *inbound* dan *outbound* untuk wilayah jaringan Unhas. Selanjutnya, implementasi prediksi trafik dilanjutkan dengan cara mengelompokkan data trafik *inbound* dan *outbound* pada hari yang sama. Selanjutnya sistem *forecasting* akan dilakukan menggunakan metode *Radial Basis Function Neural Network* dengan bantuan *software* Matlab R2008a.

III.2 Pembelajaran metode *Radial Basis Function Neural Network*

Pemilihan metode penelitian metode *Radial Basis Function Neural Network* dilakukan dengan pertimbangan sifatnya yang mampu menghasilkan *output* dengan tingkat akurasi yang tinggi pada saat data pelatihan dan waktu pelatihan yang relatif lebih singkat dibandingkan dengan metode lainnya. Jaringan syaraf tiruan metode *Radial basis function* menghasilkan prediksi data trafik dalam bentuk nilai kuantitatif yaitu hasil prediksi berupa numerik. Setelah pengambilan data, kemudian dilakukan pembelajaran metode *Radial Basis Function Neural Network* menggunakan aplikasi *Matlab* R2008a. implementasi metode *Radial Basis Function Neural Network* terdiri dari 2 tahap sebelum memprediksi, yaitu tahap pelatihan dan tahap validasi.

Adapun langkah-langkah metode RBFNN sebagai berikut :



Gambar III.4 Flowchart prediksi trafik data metode RBFNN

III.2.1 Pengolahan Data *Inputan*

Data variabel *input* terdiri dari data trafik pada hari t , dimana inputannya adalah $t-7$, $t-14$ dan seterusnya. Data variabel *input* terlebih dahulu digolongkan sesuai dengan hari yang sama. Hal ini dilakukan karena pola trafik data tiap hari berbeda. Sebagai contoh, untuk prediksi harian parameter trafik data untuk hari

senin dikelompokkan dari minggu terakhir bulan Juli 2012 yaitu tanggal 28 juli 2012 hingga minggu akhir bulan September 2012 tanggal 24 september 2012. Setelah di kelompokkan data akan dibagi menjadi data *inputan*, data *target*, dan data uji. Data dari minggu terakhir bulan Juli hingga minggu ke-4 bulan September, digunakan sebagai data *inputan*. Data inputan terdiri dari data uji dan data latih yang dipisahkan berdasarkan minggu ganji dan genap dari keseluruhan data yang diperoleh. Sedangkan untuk data *target* adalah data pemakaian trafik pada minggu terakhir bulan September 2012.

III.2.2 Normalisasi dan Denormalisasi

Sebelum melakukan pelatihan dalam metode *Radial Basis Function Neural Network* data *input* dan data uji akan di normalisasi. Normalisasi ini bertujuan untuk mendapatkan data dengan ukuran yang lebih kecil yang mewakili data yang asli tanpa kehilangan karakteristik sendirinya. Rumus dari normalisasi yaitu:

$$Normalisasi = \frac{X - min}{max - min} \quad (3.1)$$

Dimana:

X = data

Min = data minimum

Max = data maksimum

Sedangkan denormalisasi adalah mengembalikan ukuran data yang telah di normalisasi sebelumnya untuk mendapatkan data yang asli. Denormalisasi dilakukan pada hasil keluaran dari pelatihan berupa prediksi trafik data. Adapun rumus dari denormalisasi yaitu sebagai berikut :

$$\text{Denormalisasi} = Y (\text{max} - \text{min}) + \text{min} \quad (3.2)$$

Dimana :

Y = hasil keluaran dari pelatihan

Min = data minimum

Max = data maksimum

III.2.3 Pelatihan metode *Radial Basis Function Neural Network*

Sebelum *Radial Basis Function Neural Network* melakukan proses prediksi trafik data, terlebih dahulu dilakukan proses pelatihan. Proses pelatihan pada metode ini ada tiga tahapan, yaitu pemasukan data variabel *input* pada saat proses pelatihan, pengaturan nilai *spread* , *neuron/epoch* dan perhitungan nilai *error*. Proses pengaturan nilai *spread* dan *epoch* bertujuan untuk meminimalisir nilai *error* yang sesuai dengan data yang aktual dari aslinya.

Langkah - langkah dalam proses pelatihan *Radial Basis Function Neural Network* adalah sebagai berikut :

1. Me-load data *input* pemakaian trafik data minggu terakhir bulan Juli sampai minggu ke-3 bulan Agustus 2012. Sebagai data targetnya adalah pemakaian trafik data pada minggu terakhir bulan Agustus 2012.

2. Melakukan proses normalisasi data untuk mendapatkan interval data dari 0 sampai 1 agar dapat diproses pada jaringan.
3. Membuat inisialisasi jaringan yang akan dilatih untuk memprediksi data yang akan datang dengan fungsi di *Matlab*, *NEWRB*
4. Proses pelatihan jaringan dengan menggunakan fungsi *train* pada aplikasi *Matlab*. Proses pelatihan ini dilakukan agar sistem ini dapat mempelajari pola data dari minggu pertama hingga minggu terakhir sampai mendapatkan performansi dan persentasi pencapaian target yang terbaik.

Pada struktur jaringan *Radial Basis Function Neural Network* yang digunakan dalam penelitian ini menggunakan 3 lapisan yaitu lapisan masukan (*input layer*) sebanyak 50 neuron, lapisan tersembunyi (*hidden layer*) dan lapisan keluaran (*output layer*) sebanyak 1 neuron. Penentuan jumlah neuron pada lapisan tersembunyi dilakukan dengan cara mencoba-coba (*trial and error*), hingga mendapatkan nilai *error* yang kecil dan waktu pelatihan yang cepat. Jumlah *neuron* lapisan tersembunyi sama dengan nilai *epoch* pelatihan. Jumlah maksimal *neuron* sama dengan jumlah data dalam sehari.

Pada metode *Radial Basis Function Neural Network* hasil *train* sedikit berbeda dengan metode lain. Dimana hasil *trainnya* 100% mirip dengan *target* data. Sehingga dalam metode ini langsung dilakukan validasi hasil prediksi dengan data *aktual*.

III.2.4 Validasi data aktual trafik data

Validasi data digunakan untuk melihat kehandalan sistem ini dalam memprediksi dengan mencocokkan hasil keluaran dengan data aktual trafik data. Apabila hasil keluaran sistem mendapatkan nilai RMSE yang terkecil, maka hasil pelatihan digunakan untuk meramalkan nilai trafik data ke depannya. Apabila RMSE yang diperoleh masih besar maka perlu peninjauan kembali terhadap proses sebelumnya.

Nilai RMSE digunakan untuk menjadi salah satu kriteria bagi sistem pembelajaran untuk berhenti belajar. Selain nilai RMSE, RMSE merupakan tingkat kesalahan nilai keluaran dari sistem pembelajaran terhadap nilai pasangan keluaran data latih atau dengan kata lain RMSE merupakan perbedaan yang terjadi antara keluaran yang diinginkan dengan keluaran yang sebenarnya.

Langkah-langkah dalam proses validasi dengan *Radial Basis Function Neural Network* adalah sebagai berikut:

1. Me-load data uji dari minggu pertama bulan September hingga minggu terakhir bulan September 2012.
2. Melakukan proses normalisasi data untuk mendapatkan interval data 0 sampai 1.
3. Memasukkan *net* hasil pelatihan *Radial Basis Function Neural Network* sebelumnya.
4. Melakukan proses prediksi trafik data untuk awal bulan Oktober 2012.

5. Melakukan proses denormalisasi nilai trafik data untuk mengembalikan ke nilai sebenarnya.
6. Menghitung nilai RMSE (*Root Mean Square Error*) dari data aktual trafik data dengan hasil prediksi. Jika RMSE masih besar maka akan di latih ulang dengan mencoba-coba (*trial and error*) untuk mendapatkan *net* baru, hingga menghasilkan RMSE terkecil.

RMSE (*Root Mean Square Error*), Dalam statistika, *Mean Square Error* (MSE) digunakan untuk mengukur kesalahan (*error*). Kesalahan yang dimaksud adalah perbedaan nilai dua buah obyek.

Adapun rumus RMSE (Root Mean Square Error) sebagai berikut :

$$RMSE = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^n (P - a)^2}}{P_{max} - P_{min}} \quad (3.3)$$

Dimana:

P = data aktual

a = hasil prediksi

N = jumlah data

P_{max} = Nilai tertinggi dari data *actual*

P_{min} = Nilai terendah dari data *actual*

III.3 Prediksi Penggunaan Trafik Data dengan Metode *Radial Basis Function Neural Network*

Proses prediksi bertujuan untuk menentukan hasil akhir berupa nilai trafik data pada jaringan Universitas Hasanuddin dan nantinya akan dibandingkan dengan data aktual dari network monitoring sistem. Proses tersebut dinilai baik apabila memiliki nilai *error* yang kecil dan dilanjutkan untuk memprediksi pada periode selanjutnya.

BAB IV

ANALISA HASIL SIMULASI

Pada bab ini akan dibahas mengenai hasil uji coba sistem pada proses prediksi trafik data dengan metode Radial Basis Function Neural Network. Pengujian sistem terdiri dari dua tahap, yaitu tahap pelatihan sistem dan tahap uji. Pada bab ini juga akan dibandingkan tingkat keberhasilan sistem berdasarkan nilai error terkecil setelah tahap validasi dilakukan.

Pada dasarnya, prediksi trafik data menggunakan Radial Basis Function adalah sebuah proses untuk menghasilkan output atau hasil simulasi sistem berdasarkan target yang diinginkan. Karena data trafik bersifat time series dan stochastic, maka pengelompokan data input untuk sistem prediksi dilakukan berdasarkan hari yang sama dan tren datanya tidak jauh berbeda. Data yang diperoleh berupa data trafik setiap harinya dari pukul 00:00-23.30 WITA selama 9 minggu terhitung sejak akhir bulan Juli hingga akhir September 2012. Data input dikelompokkan menjadi empat variable sebagai data latih, dan empat variable lainnya sebagai data uji. Berikut adalah pembahasan mengenai hasil uji sistem.

IV.1. Analisis Pengolahan Variabel *Input* dan *Target* pada *Radial Basis Function Neural Network*

Pada perancangan sistem prediksi trafik data, record trafik data selama 9 minggu dibagi menjadi dua, yaitu data yang dipakai sebagai pelatihan dan data yang dipakai sebagai pengujian. Misalkan untuk sistem prediksi hari Senin, diambil trafik data hari Senin pada minggu ganjil (pola-1,3,5,... dalam table 1.1).

Selanjutnya, data sisanya digunakan sebagai data pengujian sistem. Berikut adalah contoh trafik data untuk hari Senin sebagai input sistem.

Tabel IV.1 data Input inbound hari senin Agustus – September 2012

TIME	TRAFIK DATA INBOUND (bits per second) , SENIN			
	30/07/2012	13/08/2012	27/08/2012	10/08/2012
0:00	800000	979718	865881	850885
0:30	1134122	890687	717118	856865
1:00	1234545	821671	812950	2556879
1:30	1617320	719343	674131	2597302
2:00	1523108	629214	469278	2654217
2:30	709126	851613	487190	3746067
3:00	923448	788546	472292	2575261
3:30	850111	637269	969130	2562505
4:00	1234538	790257	579285	866215
4:30	630273	665761	982861	1680198
5:00	910213	778013	1142893	1757765
5:30	986214	676187	1357925	902557
6:00	623495	567325	1601893	875824
6:30	693126	692394	2441984	1590200
7:00	933893	806570	2709595	2526694
7:30	1323464	1051381	2246722	2996100
8:00	1212354	1295294	2533538	2992304
8:30	1251235	1817427	2150876	2806878
9:00	1812475	2505405	3143173	3385319
9:30	3161235	2257367	2548438	3315284
10:00	2531646	2516848	3023903	3375248
10:30	3554216	2739328	2986237	3415405
11:00	2472134	2893430	2713148	3487956
11:30	2812314	1995295	2836855	3704951
12:00	1600684	1883472	3128995	3853440
12:30	3125534	2529059	3346312	3783523
13:00	3360860	2743029	3336800	1687995
13:30	3101745	2540546	2924803	3231093
14:00	2123818	3004205	2503428	3464670
14:30	2370053	2568477	1896931	2879422
15:00	2143252	2103411	1832180	1748594
15:30	1719344	2084688	3269334	2177406
16:00	1701124	2122205	2997370	3133553
16:30	1274500	1452605	2943394	2776225
17:00	1491111	1548163	1430844	2333587
17:30	1366157	1212566	1964246	1800501
18:00	2169525	1316828	1303452	1818907
18:30	1404593	1381102	1279875	1623999
19:00	1423810	1223858	1262619	2127338
19:30	1380908	1107869	1149692	1848340
20:00	1131761	1248116	1299906	1545430
20:30	1125893	1195009	1418204	1615721
21:00	1274280	1222521	1352557	1948992
21:30	1240267	1306178	1062033	2013776
22:00	1623620	1419844	887869	1573760
22:30	1544284	1241380	1026425	1986274
23:00	1596926	1003697	770946	2989223
23:30	1185227	856339	723897	1141136

Sambungan data Input inbound hari senin Agustus- September 2012

TIME	TRAFIK DATA OUTBOND (bits per second) , SENIN			
	30/07/2012	13/08/2012	27/08/2012	10/08/2012
0:00	262231	298447	274344	352576
0:30	277411	267039	185132	409551
1:00	219000	316924	171235	363352
1:30	186381	337915	144818	244655
2:00	115680	359404	144379	278539
2:30	96974	300158	124467	305751
3:00	90923	262409	81847	216690
3:30	87688	304255	91899	347109
4:00	85750	316091	76213	263984
4:30	57432	264631	106940	236534
5:00	71135	256232	149020	241997
5:30	64023	247130	214848	300130
6:00	77579	239661	397618	380401
6:30	88305	312530	890405	556899
7:00	83476	393923	967254	1233807
7:30	96717	560834	1003653	1385167
8:00	96746	667308	1207745	943146
8:30	112783	851896	1547870	1416138
9:00	96764	1079233	1681738	1582311
9:30	187800	1335898	1896062	1593543
10:00	88635	1542434	1817759	1847946
10:30	166596	1655247	1941896	1738156
11:00	185219	1838700	1795851	1893169
11:30	187633	1818504	1419682	1865197
12:00	1793276	1773704	1106429	1146119
12:30	1986959	1793629	1613996	1893169
13:00	1790237	1926662	1787056	1865197
13:30	2222410	1866181	906869	1146119
14:00	2057171	1734107	993010	5859177
14:30	1837254	1512981	1049014	1768612
15:00	1419611	1241607	938324	2128541
15:30	1205877	1145880	907940	1661621
16:00	1076394	1077742	966576	1503627
16:30	899456	694615	790406	1388353
17:00	847594	615467	695598	1139754
17:30	817735	554786	739022	918406
18:00	864015	553645	802105	715489
18:30	710065	620994	892605	878373
19:00	636424	574843	758240	1017111
19:30	766726	540906	889574	876749
20:00	775890	644939	956380	876829
20:30	775036	654231	821668	847456
21:00	750134	665393	807310	983264
21:30	634423	676486	923819	867380
22:00	600798	707734	908786	749345
22:30	613270	645425	845039	569736
23:00	536172	602682	650333	476857
23:30	453762	449661	448012	414768

Data input kemudian ditransformasikan ke dalam range fungsi sigmoid [0,1] sebelum menjadi input untuk sistem pada jaringan Radial basis Function.

Jika P adalah range data, a adalah data minimum dari P dan b adalah data maksimum dari P, normalisasi data menggunakan rumus sebagai berikut :

$$\text{Hasil normalisasi (p)} = [(P_n - a) / (b - a)]$$

Berikut data hasil normalisasi Trafik data Inbound Tabel IV.3. Sementara untuk data target adalah data acuan dari sistem prediksi Radial Basis Function untuk memprediksi unit masukan. Data ini diambil dari trafik data hari Senin, minggu terakhir bulan September (24/09/2012) dan hasilnya pada Tabel IV.4 berikut.

Tabel VI.3 Hasil Normalisasi Trafik data Inbound

TIME	HASIL NORMALISASI TRAFIK DATA INBOUND, SENIN			
	30/07/2012	13/08/2012	27/08/2012	10/08/2012
0:00	0.097726	0.150832	0.117194	0.112763
0:30	0.196457	0.124524	0.073235	0.11453
1:00	0.226132	0.10413	0.101553	0.616874
1:30	0.33924	0.073893	0.060533	0.628819
2:00	0.311401	0.04726	0	0.645637
2:30	0.070874	0.112978	0.005293	0.968272
3:00	0.134205	0.094342	0.000891	0.622306
3:30	0.112534	0.04964	0.147703	0.618536
4:00	0.22613	0.094847	0.032506	0.117293
4:30	0.047573	0.05806	0.151761	0.35782
5:00	0.130294	0.091229	0.199049	0.38074
5:30	0.152752	0.06114	0.26259	0.128031
6:00	0.04557	0.028972	0.334681	0.120132
6:30	0.066146	0.065929	0.582923	0.331226
7:00	0.137291	0.099668	0.662001	0.607954
7:30	0.252407	0.172008	0.525224	0.746661
8:00	0.219575	0.244083	0.609977	0.745539
8:30	0.231064	0.39837	0.496902	0.690747
9:00	0.396907	0.601664	0.79012	0.861673
9:30	0.795457	0.52837	0.61438	0.840978
10:00	0.609418	0.605045	0.754877	0.858697
10:30	0.911581	0.670786	0.743747	0.870563
11:00	0.591832	0.716323	0.66305	0.892002
11:30	0.692353	0.450929	0.699605	0.956122
12:00	0.334324	0.417886	0.785931	1
12:30	0.784908	0.608653	0.850147	0.97934
13:00	0.854446	0.67188	0.847336	0.360124
13:30	0.777879	0.612048	0.725593	0.8161
14:00	0.488907	0.749056	0.601079	0.885121
14:30	0.561668	0.620301	0.421863	0.712183
15:00	0.494649	0.482877	0.40273	0.37803
15:30	0.369387	0.477344	0.8274	0.504742
16:00	0.364003	0.48843	0.747036	0.787278
16:30	0.237938	0.290567	0.731087	0.681689
17:00	0.301946	0.318804	0.284137	0.550892
17:30	0.265022	0.219637	0.441754	0.393369
18:00	0.502413	0.250446	0.246494	0.398807
18:30	0.27638	0.269439	0.239527	0.341213
19:00	0.282059	0.222974	0.234428	0.489947
19:30	0.269381	0.1887	0.201058	0.407505
20:00	0.19576	0.230142	0.245446	0.317997
20:30	0.194026	0.214449	0.280402	0.338767
21:00	0.237873	0.222579	0.261004	0.437247
21:30	0.227823	0.247299	0.175156	0.45639
22:00	0.341101	0.280887	0.123691	0.326368
22:30	0.317658	0.228152	0.164634	0.448263
23:00	0.333213	0.157918	0.089141	0.744629
23:30	0.211559	0.114374	0.075238	0.19853

Tabel VI.3 data target senin 24 september 2012

TIME	DATA TARGET , SENIN	
	DATA AKTUAL	NORMALISASI
0:00	862251	0.116121
0:30	708037	0.070552
1:00	669890	0.05928
1:30	513198	0.012978
2:00	463407	-0.00173
2:30	374227	-0.02809
3:00	1548654	0.318949
3:30	848428	0.112037
4:00	328994	-0.04145
4:30	405042	-0.01898
5:00	500190	0.009134
5:30	751138	0.083288
6:00	838941	0.109233
6:30	1140532	0.198352
7:00	2654942	0.645851
7:30	2364103	0.55991
8:00	2499977	0.60006
8:30	2004949	0.453782
9:00	2816762	0.693668
9:30	3153643	0.793214
10:00	2995633	0.746523
10:30	3282260	0.83122
11:00	3808674	0.986772
11:30	3346489	0.850199
12:00	1563868	0.323445
12:30	2235505	0.52191
13:00	3651391	0.940296
13:30	1770026	0.384363
14:00	2887242	0.714494
14:30	56546	-0.12196
15:00	778288	0.091311
15:30	1881880	0.417416
16:00	1396250	0.273915
16:30	2494087	0.598319
17:00	2462011	0.588841
17:30	1935424	0.433238
18:00	2369208	0.561418
18:30	1727152	0.371694
19:00	1926563	0.430619
19:30	1777986	0.386716
20:00	1822299	0.39981
20:30	1800436	0.393349
21:00	1891501	0.420259
21:30	1911083	0.426045
22:00	1882151	0.417496
22:30	2297425	0.540207
23:00	2689705	0.656123
23:30	2147092	0.495784

Setelah data *input* dan *target* dinormalisasi, maka data tersebut digunakan dalam jaringan *Radial Basis Function Neural Network* untuk pelatihan. Fungsi

jaringan yang digunakan adalah newrb dengan pelatihan supervised dan unsupervised. Output jaringan dari data pelatihan adalah nilai trafik data yang hasilnya tidak jauh berbeda dari target yang telah ditentukan sebelumnya. Tahap terakhir adalah denormalisasi, yakni mengembalikan angka-angka dalam range 0-1 menjadi nilai bits per second yang sebenarnya lalu kemudian divalidasi sejauh mana error kesalahan dari hasil pelatihan data tersebut.

IV.2. Analisis Hasil Output Jaringan untuk prediksi minggu terakhir bulan September Menggunakan Metode *Radial Basis Function Neural Network*

Berguna atau tidaknya suatu jaringan saraf tiruan ditentukan dari hasil pelatihannya yang berupa bobot neuronnya. RBFN memiliki algoritma pelatihan yang agak unik karena terdiri atas cara *supervised* dan *unsupervised* sekaligus. Sebagai sistem peramalan, RBFNN mampu mengidentifikasi input jaringan dan menyesuaikan bobot untuk setiap lapis neuron sehingga hasilnya tidak jauh berbeda dengan data actual atau target jaringan. Berikut akan dijelaskan proses prediksi trafik bulan September dengan metode RBFNN.

Seperti yang dijelaskan di awal pembahasan bab ini, data yang diperoleh adalah data trafik selama 9 minggu terhitung akhir bulan Juli 2012 sampai akhir September 2012. Untuk memprediksi trafik dalam satu minggu di akhir September, sebagai inputan adalah data trafik 4 minggu sebelumnya terhitung mulai akhir bulan July 2012.

Sebagai contoh, hasil dari output sistem prediksi trafik untuk hari Senin, 24 September 2012 adalah sebagai berikut.

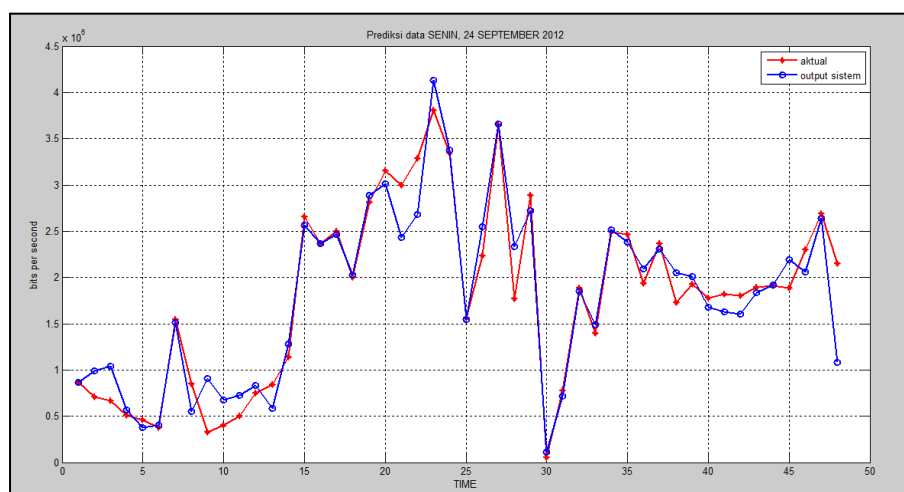
Tabel VI.4 contoh hasil dari output prediksi senin 24 september

TIME	TRAFIK DATA SENIN, 24 SEPTEMBER 2012	OUTPUT RBFNN (BITS PER SECOND)	ERROR (A-B)/A
0:00	862251	869268	0.008072
0:30	708037	989784	0.284655
1:00	669890	1038871	0.355175
1:30	513198	571766	0.102434
2:00	463407	379834	-0.22003
2:30	374227	402497	0.070237
3:00	1548654	1508539	-0.02659
3:30	848428	549754	-0.54329
4:00	328994	906339	0.637008
4:30	405042	676349	0.401135
5:00	500190	728670	0.313558
5:30	751138	835694	0.101181
6:00	838941	582742	-0.43964
6:30	1140532	1280824	0.109533
7:00	2654942	2564233	-0.03537
7:30	2364103	2364032	-3E-05
8:00	2499977	2468756	-0.01265
8:30	2004949	2023976	0.009401
9:00	2816762	2885825	0.023932
9:30	3153643	3010826	-0.04743
10:00	2995633	2432540	-0.23148
10:30	3282260	2677176	-0.22602
11:00	3808674	4128671	0.077506
11:30	3346489	3380272	0.009994
12:00	1563868	1544216	-0.01273
12:30	2235505	2552253	0.124105
13:00	3651391	3662018	0.002902
13:30	1770026	2330688	0.240556
14:00	2887242	2726174	-0.05908
14:30	56546	109936	0.485646
15:00	778288	717490	-0.08474
15:30	1881880	1853428	-0.01535
16:00	1396250	1483603	0.058879
16:30	2494087	2517422	0.009269
17:00	2462011	2379443	-0.0347
17:30	1935424	2091187	0.074485
18:00	2369208	2303948	-0.02833
18:30	1727152	2050038	0.157502
19:00	1926563	2009927	0.041476
19:30	1777986	1680140	-0.05824

20:00	1822299	1628578	-0.11895
20:30	1800436	1603070	-0.12312
21:00	1891501	1835668	-0.03042
21:30	1911083	1922412	0.005893
22:00	1882151	2195755	0.142823
22:30	2297425	2062396	-0.11396
23:00	2689705	2636609	-0.02014
23:30	2147092	1081847	-0.98465

Dari table di atas, terlihat bahwa error terkecil ada pada data ke-27 dengan nilai error sebesar 0.002. Pada jaringan RBFNN yang paling berpengaruh dalam besar kecilnya nilai error untuk setiap data adalah nilai spread dan neuron maksimal yang digunakan. Apabila neuron mendekati angka maksimal dari banyaknya jumlah data, maka hasil output sistem akan jauh lebih baik dan errornya jauh lebih kecil.

Berikut hasil output sistem untuk prediksi hari Senin, 24 September 2012 dan weekly trafik minggu terakhir bulan September 2012 sebagai hasil simulasi dengan metode RBFNN.



Gambar VI.1. gambar Hasil *output* sistem prediksi hari senin 24/09/ 2012

Dari grafik output sistem, *red line* adalah data trafik actual sedangkan *blue line* adalah data trafik hasil jaringan neural network newrb. Maksimum trafik berdasarkan nilai actual dan prediksi tidak jauh berbeda yakni sekitar 4,2 Mbps. Dan kepadatan trafik dimulai dari pukul 07:00 s.d 14:00 WITA.

Nilai RMSE nya dapat dihitung menggunakan rumus sebagai berikut :

$$RMSE = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^n (P - a)^2}}{P_{max} - P_{min}}$$

Dimana :

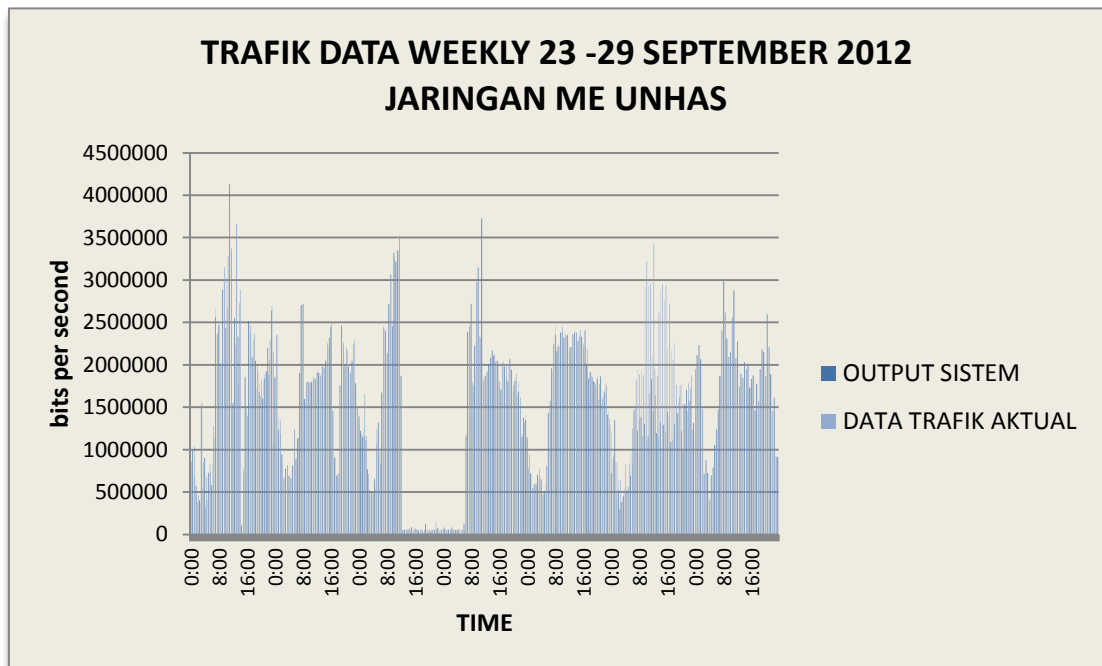
N : Jumlah data masukan

P : Nilai aktual data trafik Senin, 24 September 2012

a : Nilai hasil prediksi trafik sistem

P_{max} : Nilai maksimal data aktual trafik Senin 24 September 2012

P_{min} : Nilai minimum data aktual trafik Senin 24 September 2012



Gambar VI.2 Trafik data *weekly* 23-29 september 2012 jaringan ME UNHAS

Berikut adalah nilai RMSE per hari dari hasil simulasi sistem.

Table VI.6 nilai RMSE per hari dari hasil simulasi sistem

HARI	RMSE
Senin	0.073738
Selasa	0.018721
Rabu	0.027653
Kamis	0.017462
Jum'at	0.041056
Sabtu	0.11246
Minggu	0

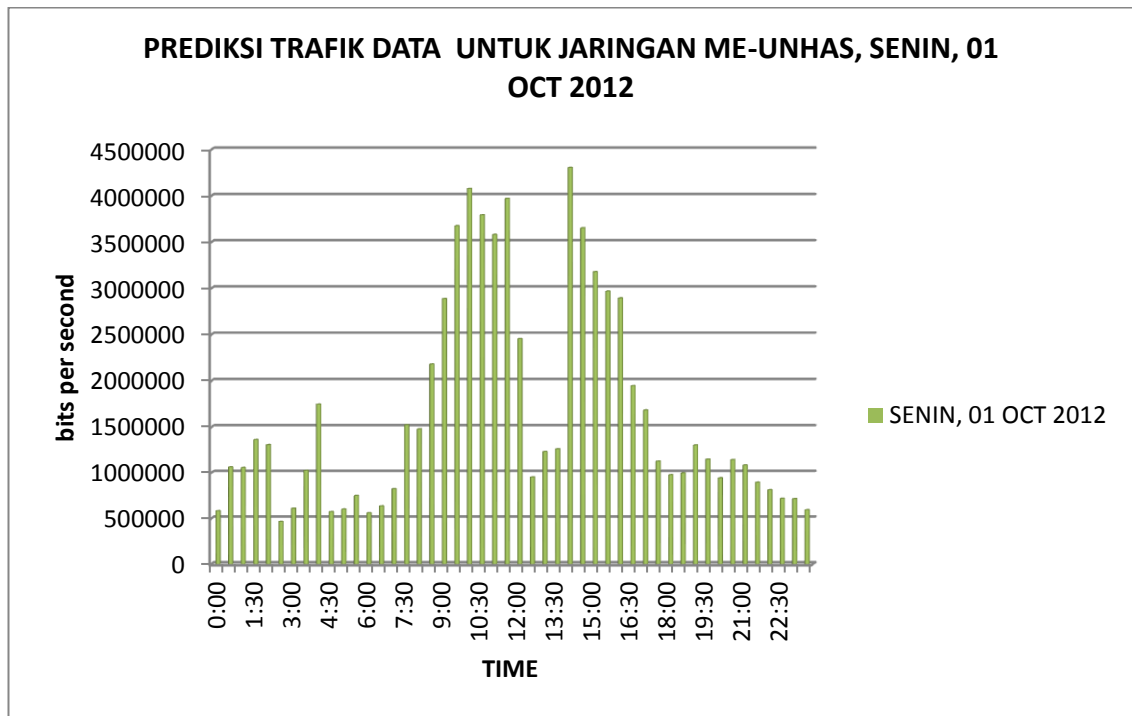
Dari hasil simulasi sistem, banyak data sebagai sample adalah 336 data selama satu minggu. Untuk mengetahui layak tidaknya sistem digunakan untuk prediksi selanjutnya dilakukan tahap validasi. Banyaknya data yang valid 303

data. Berdasarkan rumus $= \text{jumlah data benar} / \text{jumlah keseluruhan} \times 100\%$, maka diperoleh persentase untuk simulasi sistem sebesar 90.18%.

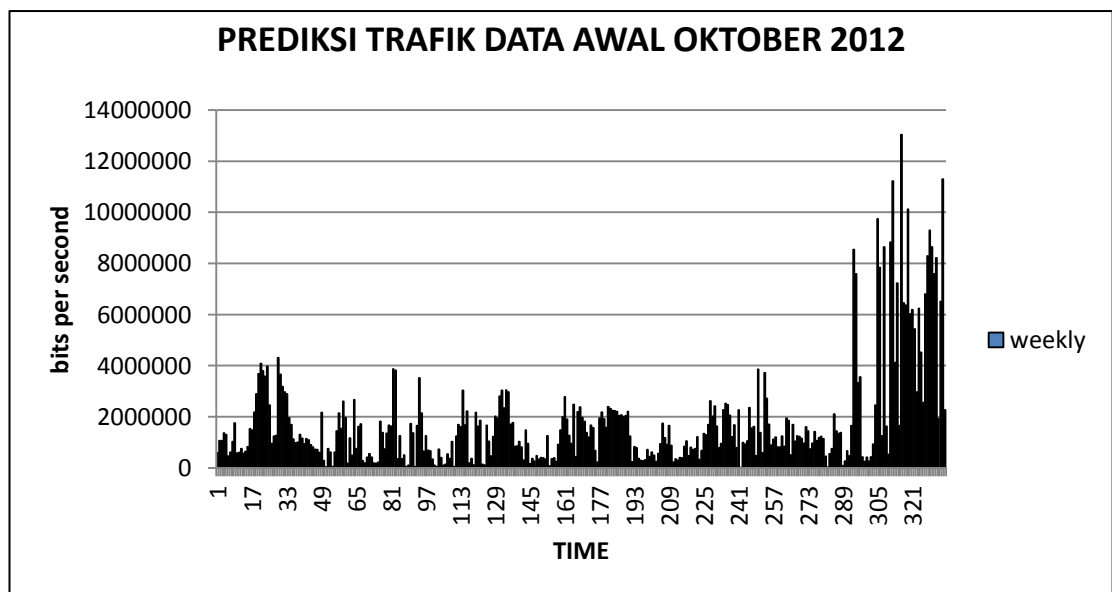
IV.3 Analisis Hasil Prediksi Trafik Data Mingguan bulan Oktober 2012 Menggunakan Metode *Radial Basis Function Neural Network*

Berdasarkan hasil simulasi sistem sebelumnya dengan menggunakan jaringan RBFNN, diperoleh nilai RMSE yang sangat kecil untuk setiap data pelatihan dan persentase keberhasilan sistem untuk data pelatihan 90.18%. Hal ini memperlihatkan bahwa sistem prediksi dapat diuji coba untuk data inputan yang baru. Sama halnya dengan data pelatihan, data uji digunakan untuk memperoleh nilai data trafik untuk awal bulan Oktober 2012. Sebagai variable input adalah nilai trafik data 4 minggu sebelumnya yakni di minggu ke-2, ke-4, ke-6, dst terhitung sejak akhir July 2012.

Data input setelah di normalisasi dengan range [0,1] , kemudian dilakukan training data menggunakan jaringan yang sama dengan jaringan pada data pelatihan. Hasil training atau simulasi untuk prediksi trafik awal Oktober 2012 untuk hari Senin dan hasil training data trafik selama seminggu dapat dilihat seperti pada gambar berikut ini



Gambar VI.3 prediksi trafik data untuk jaringan ME UNHAS senin 01/10/2012
 Dari hasil prediksi untuk kebutuhan bandwidth awal bulan oktober 2012, dapat dilihat bahwa



Gambar VI.4 gambar prediksi trafik data awal oktober 2012

BAB V

PENUTUP

V.1 KESIMPULAN

Dari penelitian prediksi trafik komunikasi data dengan menggunakan jaringan syaraf tiruan *Radial Basis Fuction* maka dapat disimpulkan bahwa :

1. Metode *Radial Basis Function* baik digunakan dalam model prediksi data time series. Dari hasil simulasi data latih diperoleh tingkat keberhasilan terhadap sistem jaringan syaraf tiruan *Radial Basis Function* sebesar 90.18%.
2. Kebutuhan trafik setiap harinya berbeda-beda namun pola data setiap harinya memiliki karakteristik yang sama, khusus untuk hari libur, kebutuhan trafik data jauh lebih sedikit dibanding hari-hari biasanya.
3. Hasil prediksi dengan metode *Radial basis function neural network* untuk awal bulan Oktober 2012 diperkirakan kebutuhan *bandwidth* pada minggu tersebut sebesar 472 Mb.

V.2 SARAN

1. Penelitian ini masih perlu dilakukan pengembangan, maka dari itu perlu dilakukan penelitian lanjutan dengan menggunakan metode kepakaran yang lain seperti DTW, SVM, *Back propagation* atau mengkombinasikan metode-metode yang sudah diteliti dengan metode-metode yang lain.
2. Penelitian dapat dikembangkan lebih lanjut dengan menggunakan data

lebih banyak dan bervariasi agar data yang dihasilkan lebih akurat lagi.

3. Untuk pengembangan Tugas Akhir ini nantinya data dapat dikategorikan berdasarkan waktu, misalnya data jumlah trafik pada saat hari kerja, data jumlah trafik saat akhir pekan, dan data jumlah trafik pada saat libur hari besar.
4. Perlu adanya software yang lebih baik lagi dalam merekam trafik data agar data yang dihasilkan lebih banyak.

DAFTAR PUSTAKA

- [1] Stalling, William. 2001. *Dasar-Dasar Komunikasi Data*. Jakarta: Penerbit Salemba Teknika
- [2] Ariyus, Dony. 2008. *Komunikasi Data*. Yogyakarta: Penerbit Andi
- [3] Gunawan, Arif Hamdani. 2004. *Komunikasi Data Via Cable Network*. Jakarta: Penerbit Salemba Teknika
- [4] Mauro, Douglas R dn Kevin J. Schmidt, 2005, *Essential SNMP*, edisi ke-2, O'Reilly, Sebastopol, CA.
- [5] Behrouz A. Forouzan, *Data Communication and Networking Fourth Edition*, Mcgraw Hill, 2007
- [6] Drs. Jek Siang, Jong, M. Sc. 2009. *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB*. Yogyakarta : Penerbit Andi
- [7] Hermawan, Arief. 2006. *Jaringan Syaraf Tiruan Teori dan Aplikasi*. Yogyakarta : Penerbit Andi
- [8] Hery Purnomo, Mauridhi ; Kurniawan, Agus. 2006. *Supervised Neural Networks dan Aplikasinya*. Yogyakarta : Graha Ilmu
- [9] Away, Gunaidi Abdia. 2010. *Matlab Programming*. Bandung: Penerbit Informatika
- [10] Djarwanto PS., S.E. 1990. *Pokok-Pokok Metode Riset dan Bimbingan Teknis Penulisan Skripsi*. Yogyakarta: Liberty Yogyakarta
- [11] JJ Siang. 2005. *Jaringan Saraf Tiruan dan Pemogramannya menggunakan MATLAB*. Yogyakarta: Andi Yogyakarta

LAMPIRAN

OUTBOUND

Jum'at

```
clc;
clear;
%load target
load jumat_target.mat
j=jumat_target'

%load data 1
load jumat_data01.mat
P1=jumat_data01'
P=P1
size(P)

%normalisasi
a=min(min(P));
b=max(max(P));
p=[(P-a)/(b-a)];
t=[(j-a)/(b-a)];

%load data 2
load jumat_data02.mat
P2=jumat_data02';
R=P2;
size(R)

%normalisasi
c=min(min(R));
d=max(max(R));
q=[(R-c)/(d-c)];

%bentuk jaringan RBF
%[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
[net,tr] = newrb(p,t,0,0.3,25,1)

%bobot input,lapisan,dan bias
Bobot_input = net.IW{1,1}
Bobot_bias_input = net.b{1,1}
Bobot_Lapisan = net.LW{2,1}
Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
Y1 = sim(net,p)
ytt=((b-a)*Y1)+ a;
ytt=abs(ytt);
```

```

ytt=round(ytt);

%data uji

Y2 = sim(net,q)
yqq=((d-c)*Y2)+ c;
yqq=abs(yqq);
yqq=round(yqq);

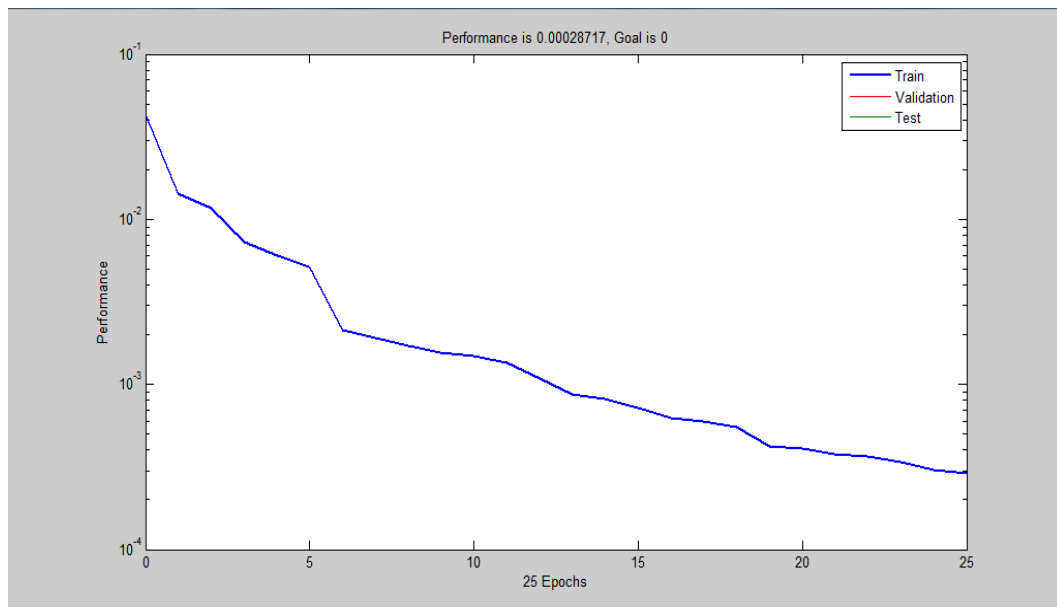
%menggambar grafik perbandingan aktual,RBF

n=1:length(Y1);
figure
C=plot(n,j, '-r*',n,ytt, '-bo')
set(C, 'LineWidth',1.5)
title('Prediksi data jumat
1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
grid;

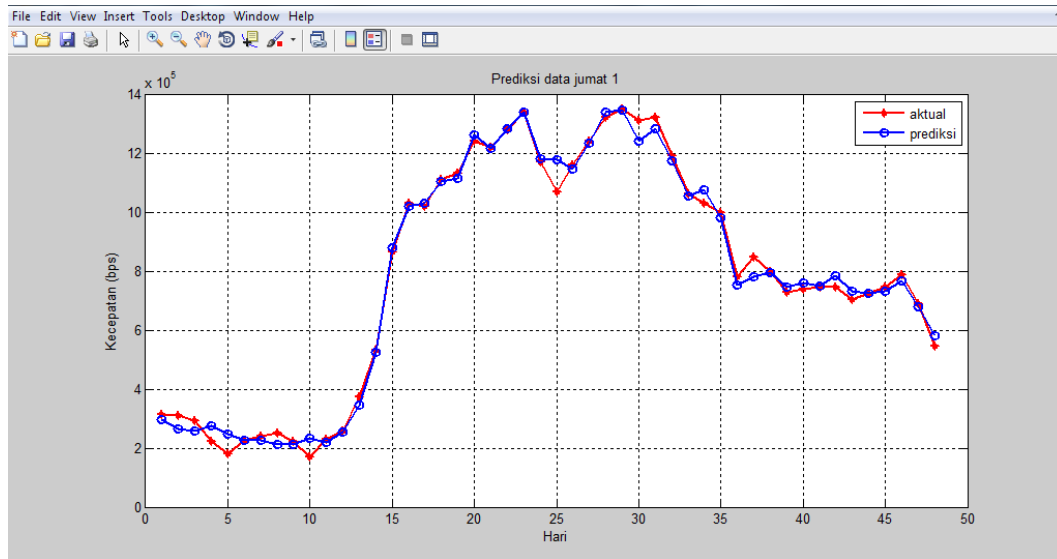
%menampilkan RMSE
rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
disp(['Root Meant Square Error = ' num2str(rmse)])

maka didapatkan nilai epoch

```



Dan grafiknya :



NEWRB, neurons = 0, MSE = 0.0426808
 NEWRB, neurons = 2, MSE = 0.0117328
 NEWRB, neurons = 3, MSE = 0.00724135
 NEWRB, neurons = 4, MSE = 0.00606054
 NEWRB, neurons = 5, MSE = 0.00518527
 NEWRB, neurons = 6, MSE = 0.00213175
 NEWRB, neurons = 7, MSE = 0.00192093
 NEWRB, neurons = 8, MSE = 0.00170004
 NEWRB, neurons = 9, MSE = 0.00154827
 NEWRB, neurons = 10, MSE = 0.00147464
 NEWRB, neurons = 11, MSE = 0.0013623
 NEWRB, neurons = 12, MSE = 0.00109063
 NEWRB, neurons = 13, MSE = 0.000874285
 NEWRB, neurons = 14, MSE = 0.000817355
 NEWRB, neurons = 15, MSE = 0.000719241
 NEWRB, neurons = 16, MSE = 0.000622921
 NEWRB, neurons = 17, MSE = 0.000592787

NEWRB, neurons = 18, MSE = 0.00055254
NEWRB, neurons = 19, MSE = 0.000422932
NEWRB, neurons = 20, MSE = 0.000408825
NEWRB, neurons = 21, MSE = 0.000380524
NEWRB, neurons = 22, MSE = 0.000366776
NEWRB, neurons = 23, MSE = 0.0003383
NEWRB, neurons = 24, MSE = 0.00030169
NEWRB, neurons = 25, MSE = 0.00028717

net =

Neural Network object:

architecture:

numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]

numOutputs: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

subobject structures:

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

functions:

adaptFcn: (none)
divideFcn: (none)
gradientFcn: (none)
initFcn: (none)
performFcn: 'mse'
plotFcns: {}
trainFcn: (none)

parameters:

adaptParam: (none)
divideParam: (none)
gradientParam: (none)
initParam: (none)
performParam: (none)
trainParam: (none)

weight and bias values:

IW: {2x1 cell} containing 1 input weight matrix

LW: {2x2 cell} containing 1 layer weight matrix

b: {2x1 cell} containing 2 bias vectors

other:

name: ''

userdata: (user information)

tr =

epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25]

perf: [1x26 double]

Bobot_input =

0.6760	0.0656	0.6896	0.5976
0.0903	0.0904	0.3758	0.5498
0.9140	0.1442	0.3724	0.8846
0.6972	0.1406	0.8455	0.6242
0.2873	0.1295	0.7536	0.7783
0.8521	0.1084	0.6115	0.4186
0.2738	0.0909	0.3752	0.3830
0.5937	0.0638	0.5201	0.7623
0.9067	0.1313	0.8812	0.6136

0.5002	0.0549	0.8776	0.8633
0.1635	0.1318	0.6432	0.4489
0.2733	0.0279	0.5720	0.4234
0.1158	0.0326	0.1808	0.0833
0.8539	0.1199	0.5421	0.7198
1.0000	0.0888	0.6064	0.6348
0.3003	0.0567	0.4732	0.3134
0.1020	0.0271	0.1435	0.1285
0.7891	0.1202	0.6654	0.3926
0.2655	0.0642	0.2967	0.3426
0.0881	0.0252	0.1482	0.0865
0.4800	0.1262	0.8109	0.7464
0.3416	0.0921	0.5249	0.3224
0.2243	0.0400	0.2255	0.2395
0.9691	0.1058	0.4391	0.5764
0.1073	0.1515	0.5213	0.5551

Bobot_bias_input =

2.7752

2.7752

2.7752

2.7752

2.7752

2.7752

2.7752

2.7752

2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752

Bobot_Lapisan =

Columns 1 through 12

0.1712	0.0150	0.7158	0.6485	0.2728	2.0169	-
5.1934	0.4468	-0.1120	0.2716	0.4407	-0.6228	

Columns 13 through 24

1.9744	-0.9889	0.5585	0.1221	2.4790	-1.7558
6.6520	-3.4063	-0.6105	1.2502	-3.7783	-0.4175

Column 25

0.5237

Bobot_Bias_lapisan =

0.1697

Y1 =

Columns 1 through 12

0.0651	0.0484	0.0435	0.0534	0.0377	0.0275
0.0278	0.0204	0.0203	0.0302	0.0233	0.0425

Columns 13 through 24

0.0899	0.1863	0.3734	0.4488	0.4547	0.4933
0.4981	0.5776	0.5526	0.5879	0.6185	0.5345

Columns 25 through 36

0.5315	0.5153	0.5626	0.6176	0.6225	0.5658
0.5875	0.5305	0.4681	0.4789	0.4285	0.3076

Columns 37 through 48

0.3226	0.3294	0.3028	0.3100	0.3054	0.3230
0.2964	0.2921	0.2951	0.3143	0.2681	0.2150

Y2 =

Columns 1 through 12

0.0837	0.1254	0.0419	0.0323	0.0207	0.0200
0.0174	0.0168	0.0142	0.0199	0.0308	0.0455

Columns 13 through 24

0.0755	0.2285	0.3829	0.3757	0.3205	0.2793
0.3122	0.2354	0.2107	0.1809	0.1979	0.1927

Columns 25 through 36

0.2217	0.2591	0.2142	0.2583	0.2207	0.2813
0.3287	0.2257	0.2693	0.3259	0.2897	0.2703

Columns 37 through 48

0.2538	0.2648	0.2510	0.2627	0.2704	0.2699
0.2750	0.2913	0.2823	0.2882	0.2849	0.2544

C =

232.0023

233.0018

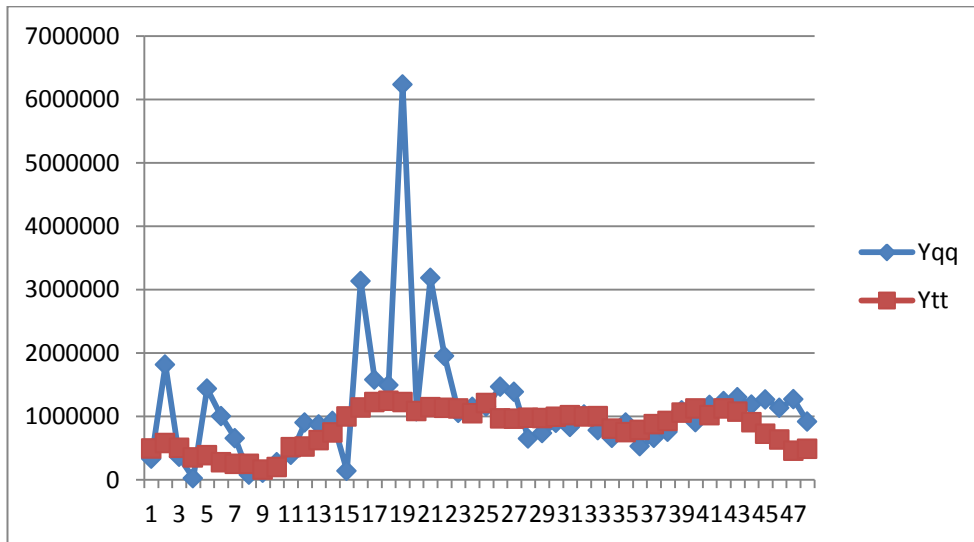
Root Meant Square Error = 0.027053

Berikut nilai Yqq dan Ytt pada prediksi hari jum'at

prediksi jumat	
Yqq	Ytt
287830	297732
368080	266264
207313	256943
188976	275607
166554	246051
165264	226913
160202	227480
158947	213627
154071	213293
165006	231978
186010	218918
214245	255218
272089	344385
566640	525775
863904	877859
850110	1019881
743678	1030994
664420	1103605
727707	1112615
579907	1262267
532320	1215270
474930	1281698
507720	1339298
497722	1181216
553487	1175542
625637	1144970
539097	1234018
624069	1337598

551562	1346700
668387	1240042
759496	1280953
561328	1173533
645285	1056096
754231	1076429
684474	981605
647064	754148
615340	782382
636437	795138
609881	745119
632501	758677
647380	750012
646305	783145
656073	732981
687477	724943
670189	730607
681527	766776
675199	679832
616425	579865

Grafiknyan :



SABTU

```
clc;
clear;
%load target
load sabtu_target.mat
j=sabtu_target'

%load data 1
load sabtu_data01.mat
P1=sabtu_data01'
P=P1
size(P)

%normalisasi
a=min(min(P));
b=max(max(P));
p=[(P-a)/(b-a)];
t=[(j-a)/(b-a)];

%load data 2
load sabtu_data02.mat
P2=sabtu_data02';
R=P2;
size(R)

%normalisasi
c=min(min(R));
d=max(max(R));
q=[(R-c)/(d-c)];

%bentuk jaringan RBF
%[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
[net,tr] = newrb(p,t,0,0.3,25,1)

%bobot input,lapisan,dan bias
Bobot_input = net.IW{1,1}
Bobot_bias_input = net.b{1,1}
Bobot_Lapisan = net.LW{2,1}
Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
Y1 = sim(net,p)
ytt=(b-a)*Y1+ a;
ytt=abs(ytt);
ytt=round(ytt);
```

```

%data uji

Y2 = sim(net,q)
yqq=((d-c)*Y2)+ c;
yqq=abs(yqq);
yqq=round(yqq);

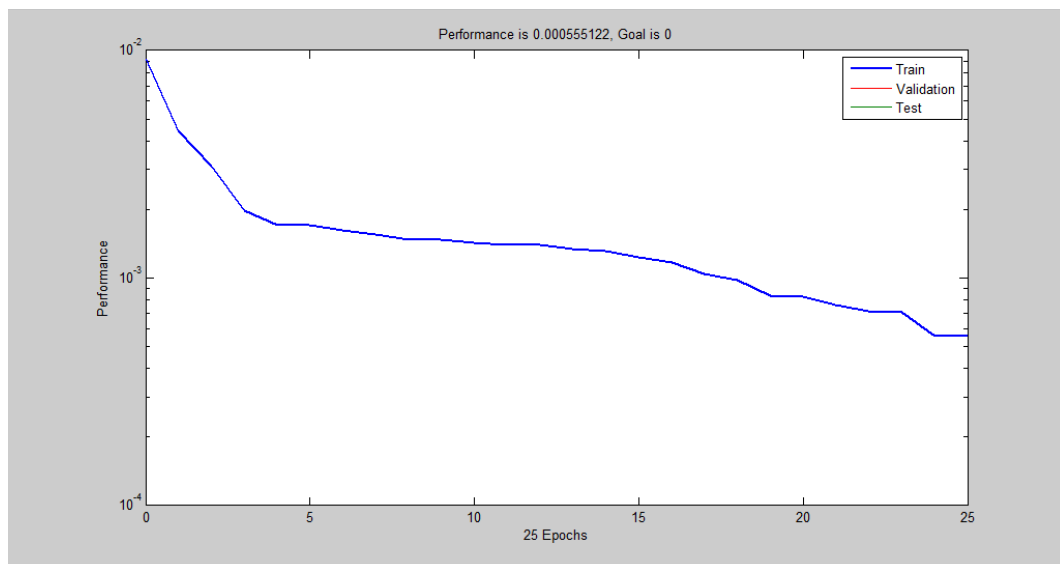
%menggambar grafik perbandingan aktual,RBF

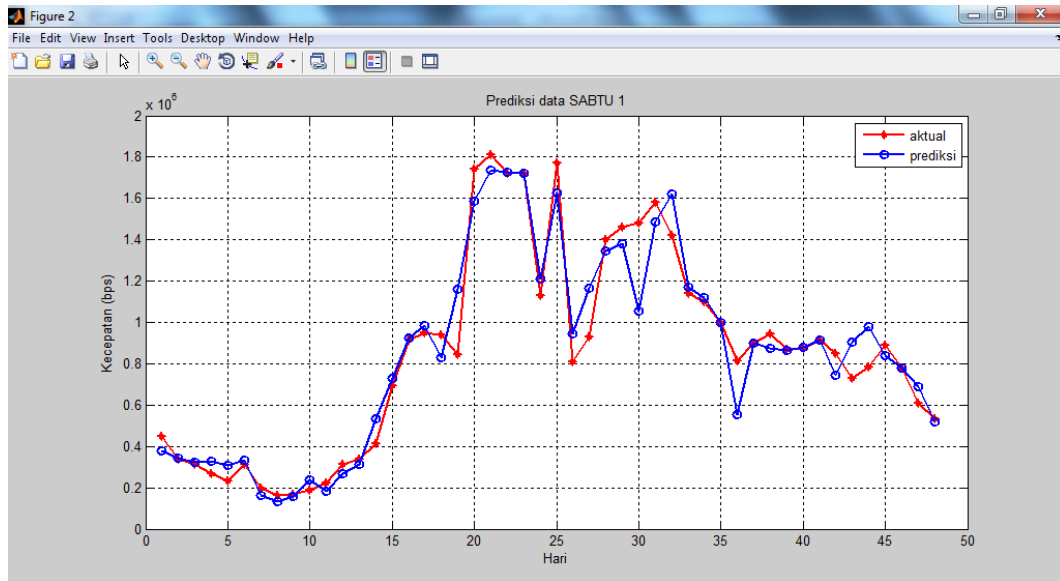
n=1:length(Y1);
figure
C=plot(n,j, '-r*',n,ytt, '-bo')
set(C, 'LineWidth',1.5)
title('Prediksi data SABTU
1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
grid;

%menampilkan RMSE
rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
disp(['Root Meant Square Error = ' num2str(rmse)])

```

maka didapatkan grafik :





```

NEWRB, neurons = 0, MSE = 0.00912809
NEWRB, neurons = 2, MSE = 0.00308902
NEWRB, neurons = 3, MSE = 0.0019818
NEWRB, neurons = 4, MSE = 0.00171262
NEWRB, neurons = 5, MSE = 0.00171145
NEWRB, neurons = 6, MSE = 0.00162157
NEWRB, neurons = 7, MSE = 0.00154923
NEWRB, neurons = 8, MSE = 0.00147733
NEWRB, neurons = 9, MSE = 0.00147273
NEWRB, neurons = 10, MSE = 0.00142721
NEWRB, neurons = 11, MSE = 0.00139974
NEWRB, neurons = 12, MSE = 0.00139972
NEWRB, neurons = 13, MSE = 0.00134286
NEWRB, neurons = 14, MSE = 0.00131029
NEWRB, neurons = 15, MSE = 0.00123599
NEWRB, neurons = 16, MSE = 0.0011644
NEWRB, neurons = 17, MSE = 0.00104471
NEWRB, neurons = 18, MSE = 0.000983945
NEWRB, neurons = 19, MSE = 0.000838721
NEWRB, neurons = 20, MSE = 0.000824266
NEWRB, neurons = 21, MSE = 0.000759695
NEWRB, neurons = 22, MSE = 0.000713984
NEWRB, neurons = 23, MSE = 0.000704833
NEWRB, neurons = 24, MSE = 0.000555155
NEWRB, neurons = 25, MSE = 0.000555122

```

net =

Neural Network object:

architecture:

```

    numInputs: 1
    numLayers: 2
    biasConnect: [1; 1]
    inputConnect: [1; 0]

```

```
layerConnect: [0 0; 1 0]
outputConnect: [0 1]
```

```
numOutputs: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)
```

subobject structures:

```
inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight
```

functions:

```
adaptFcn: (none)
divideFcn: (none)
gradientFcn: (none)
initFcn: (none)
performFcn: 'mse'
plotFcns: {}
trainFcn: (none)
```

parameters:

```
adaptParam: (none)
divideParam: (none)
gradientParam: (none)
initParam: (none)
performParam: (none)
trainParam: (none)
```

weight and bias values:

```
IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors
```

other:

```
name: ''
userdata: (user information)
```

tr =

```
epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25]
perf: [1x26 double]
```

Robot_input =

Bobot_Lapisan =

1.0e+003 *

Columns 1 through 12

-2.5490	0.0158	-0.1883	-0.2071	0.2442	0.0063	-
0.4993	0.0509	-0.0155	-0.1179	6.3309	-0.6038	

Columns 13 through 24

1.2240	-1.2031	1.5377	-0.8802	0.8075	-5.3521
0.7132	0.1086	-0.1090	-0.0947	-9.1546	9.9737

Column 25

0.0238

Bobot_Bias_lapisan =

-16.1353

Y1 =

Columns 1 through 12

0.0483	0.0419	0.0377	0.0386	0.0347	0.0393
0.0053	-0.0006	0.0044	0.0204	0.0096	0.0264

Columns 13 through 24

0.0356	0.0794	0.1179	0.1564	0.1686	0.1381
0.2030	0.2880	0.3176	0.3155	0.3142	0.2132

Columns 25 through 36

0.2960	0.1611	0.2041	0.2406	0.2473	0.1830
0.2682	0.2948	0.2055	0.1952	0.1719	0.0834

Columns 37 through 48

0.1516	0.1468	0.1451	0.1474	0.1548	0.1208
0.1523	0.1677	0.1403	0.1278	0.1101	0.0763

Y2 =

Columns 1 through 12

-0.0111	-0.1387	-0.0098	0.0608	-0.0837	-0.0224
0.0006	0.0217	0.0364	-0.0360	0.0327	0.1343

Columns 13 through 24

1.0864 2.2991 1.6952 -9.9015 -0.7079 0.0574 -
 14.8461 0.7312 -12.0327 -10.6554 1.5452 1.1687

Columns 25 through 36

0.9564 0.9145 1.1578 0.9647 0.5910 0.3957
 0.2067 -0.3123 -0.8276 -0.6821 -1.2623 -1.1029

Columns 37 through 48

-0.8117 -1.2056 -1.1713 -0.8696 -0.5230 -0.4467 -
 0.7540 -0.8821 -1.7010 -1.8676 -2.0451 -2.3446

C =

457.0023
 458.0018

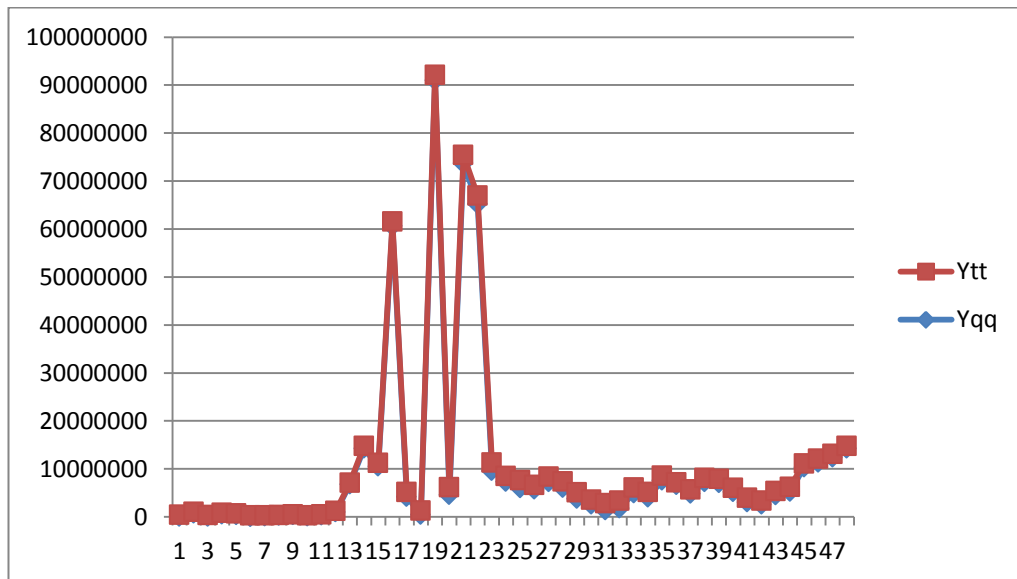
Root Meant Square Error = 0.072145

Nilai Ytt dan Yqq

prediksi Sabtu	
Yqq	Ytt
60059	377430
722931	345084
68091	323953
501322	328464
385186	308972
9379	332341
131928	160685
261114	130797
351135	155799
92768	236573
329020	182274
952048	266972
6793454	313364
14234534	534362
10528845	728831
60623002	923104
4215112	984778
480069	830954
90960642	1157993
4614611	1587071
73698900	1736609

65248347	1725784
9608648	1719523
7298982	1209891
5996051	1627526
5739132	946599
7231906	1163964
6047201	1347899
3754050	1381891
2556231	1057233
1396026	1487079
1787723	1621390
4949760	1170786
4056669	1118934
7616847	1001387
6638857	554930
4852080	898983
7268587	874659
7058572	865982
5207347	877704
3080602	915195
2612470	743221
4498181	902502
5283983	980180
10308188	841940
11330556	778931
12419513	689656
14257130	518820

Grafiknya :



MINGGU

```

clc;
clear;
%load target
load minggu_target.mat
j=minggu_target'

%load data 1
load minggu_data01.mat
P1=minggu_data01'
P=P1
size(P)

%normalisasi
a=min(min(P));
b=max(max(P));
p=[(P-a)/(b-a)];
t=[(j-a)/(b-a)];

%load data 2
load minggu_data02.mat
P2=minggu_data02';
R=P2;
size(R)

%normalisasi
c=min(min(R));
d=max(max(R));
q=[(R-c)/(d-c)];

```

```

%bentuk jaringan RBF
    %[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
    [net,tr] = newrb(p,t,0,0.3,25,1)

%bobot input,lapisan,dan bias
    Bobot_input = net.IW{1,1}
    Bobot_bias_input = net.b{1,1}
    Bobot_Lapisan = net.LW{2,1}
    Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
    Y1 = sim(net,p)
    ytt=( (b-a)*Y1)+ a;
    ytt=abs(ytt);
    ytt=round(ytt);

%data uji

    Y2 = sim(net,q)
    yqq=( (d-c)*Y2)+ c;
    yqq=abs(yqq);
    yqq=round(yqq);

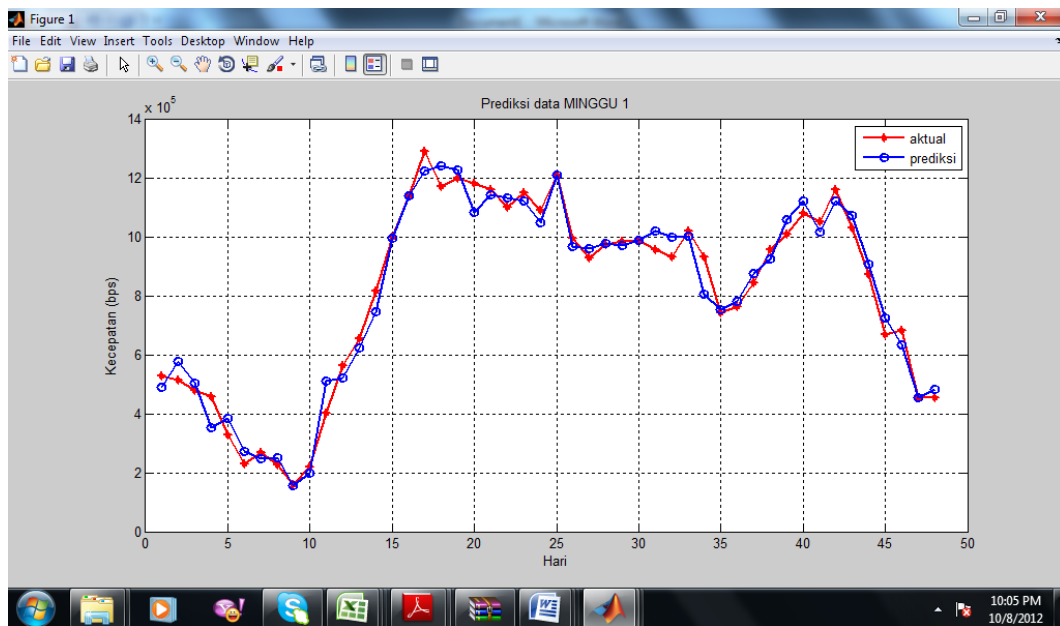
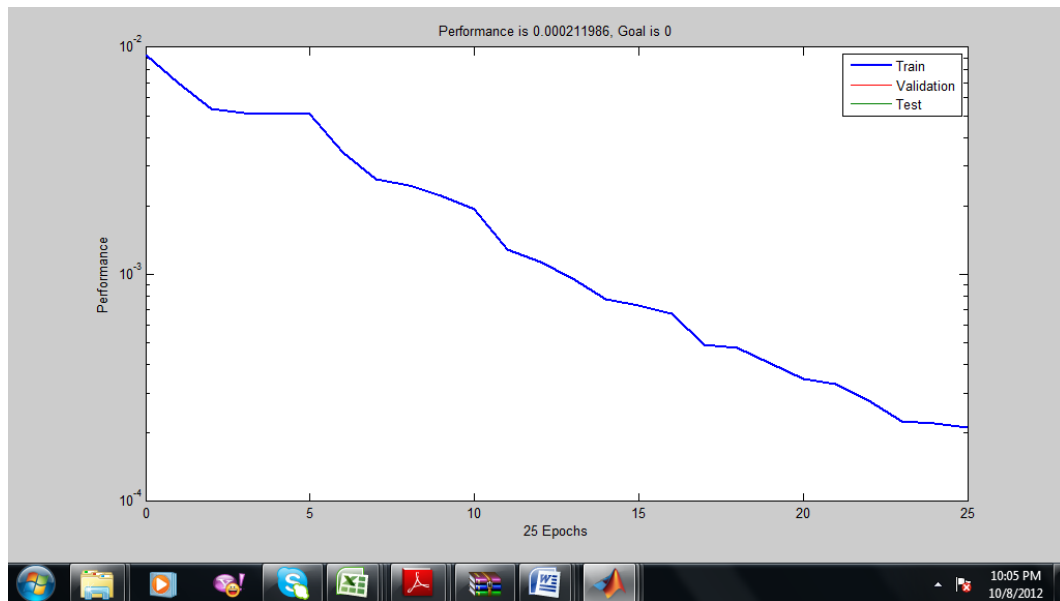
%menggambar grafik perbandingan aktual,RBF

    n=1:length(Y1);
    figure
    C=plot(n,j,'-r*',n,ytt,'-bo')
    set(C,'LineWidth',1.5)
    title('Prediksi data MINGGU
1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
    grid;

%menampilkan RMSE
    rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
    disp(['Root Meant Square Error = ' num2str(rmse)])

```

maka didapatkan grafik



```

NEWRB, neurons = 0, MSE = 0.00921849
NEWRB, neurons = 2, MSE = 0.00535492
NEWRB, neurons = 3, MSE = 0.00511997
NEWRB, neurons = 4, MSE = 0.00507302
NEWRB, neurons = 5, MSE = 0.00505907
NEWRB, neurons = 6, MSE = 0.00346165
NEWRB, neurons = 7, MSE = 0.00261249
NEWRB, neurons = 8, MSE = 0.0024549
NEWRB, neurons = 9, MSE = 0.00221658
NEWRB, neurons = 10, MSE = 0.00193319
NEWRB, neurons = 11, MSE = 0.00128017
NEWRB, neurons = 12, MSE = 0.00113718
NEWRB, neurons = 13, MSE = 0.000957039

```

```
NEWRB, neurons = 14, MSE = 0.000774649
NEWRB, neurons = 15, MSE = 0.000733154
NEWRB, neurons = 16, MSE = 0.000673898
NEWRB, neurons = 17, MSE = 0.000491465
NEWRB, neurons = 18, MSE = 0.000476396
NEWRB, neurons = 19, MSE = 0.000406645
NEWRB, neurons = 20, MSE = 0.000346571
NEWRB, neurons = 21, MSE = 0.000329835
NEWRB, neurons = 22, MSE = 0.000276922
NEWRB, neurons = 23, MSE = 0.000226174
NEWRB, neurons = 24, MSE = 0.000221451
NEWRB, neurons = 25, MSE = 0.000211986
```

```
net =
```

```
Neural Network object:
```

```
architecture:
```

```
    numInputs: 1
    numLayers: 2
    biasConnect: [1; 1]
    inputConnect: [1; 0]
    layerConnect: [0 0; 1 0]
    outputConnect: [0 1]
```

```
    numOutputs: 1 (read-only)
    numInputDelays: 0 (read-only)
    numLayerDelays: 0 (read-only)
```

```
subobject structures:
```

```
    inputs: {1x1 cell} of inputs
    layers: {2x1 cell} of layers
    outputs: {1x2 cell} containing 1 output
    biases: {2x1 cell} containing 2 biases
    inputWeights: {2x1 cell} containing 1 input weight
    layerWeights: {2x2 cell} containing 1 layer weight
```

```
functions:
```

```
    adaptFcn: (none)
    divideFcn: (none)
    gradientFcn: (none)
    initFcn: (none)
    performFcn: 'mse'
    plotFcns: {}
    trainFcn: (none)
```

```
parameters:
```

```
    adaptParam: (none)
    divideParam: (none)
    gradientParam: (none)
    initParam: (none)
    performParam: (none)
```

trainParam: (none)

weight and bias values:

IW: {2x1 cell} containing 1 input weight matrix
LW: {2x2 cell} containing 1 layer weight matrix
b: {2x1 cell} containing 2 bias vectors

other:

name: ''
userdata: (user information)

tr =

epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24 25]
perf: [1x26 double]

Bobot_input =

0.1406	0.0330	0.1568	0.3038
0.1596	0.0438	0.1082	1.0000
0.1724	0.0336	0.1056	0.3098
0.1409	0.0227	0.7362	0.1338
0.1079	0.0217	0.1264	0.0028
0.1086	0.0146	0.0227	0.0612
0.2213	0.0245	0.1489	0.2767
0.1237	0.0278	0.1120	0.3309
0.0980	0.0231	0.1134	0.0782
0.1599	0.0339	0.1657	0.3219
0.1235	0.0345	0.1760	0.2414
0.1203	0.0224	0.1601	0.0932
0.1371	0.0249	0.1346	0.2044
0.0540	0.0190	0.0461	0.0145
0.1086	0.0043	0.0288	0.0275
0.1479	0.0265	0.1204	0.1980
0.1892	0.0245	0.1628	0.2095
0.1309	0.0653	0.1522	0.2172
0.0871	0.0090	0.0413	0.0194
0.1110	0.0130	0.0127	0.0222
0.2018	0.0237	0.1007	0.2767
0.1403	0.0451	0.0994	0.2314
0.0781	0.0218	0.0930	0.0055
0.1213	0.0362	0.1306	0.3068
0.0784	0.0060	0.0500	0.0139

Bobot_bias_input =

2.7752
2.7752
2.7752
2.7752

2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752

Bobot_Lapisan =

1.0e+003 *

Columns 1 through 12

-1.4197	0.0007	0.2879	0.0004	-0.0215	0.2195
0.5559	0.1616	-0.6017	0.3869	0.9672	0.1985

Columns 13 through 24

-2.3988	-0.3897	-1.1132	3.4205	-0.7561	-0.0587
0.4492	0.2705	-0.6278	-0.6699	0.1342	0.3885

Column 25

0.6223

Bobot_Bias_lapisan =

-1.4061

Y1 =

Columns 1 through 12

0.1079	0.1345	0.1120	0.0663	0.0764	0.0422
0.0346	0.0356	0.0074	0.0203	0.1145	0.1172

Columns 13 through 24

0.1480 0.1849 0.2604 0.3030 0.3287 0.3342
0.3294 0.2860 0.3047 0.3011 0.2983 0.2762

Columns 25 through 36

0.3249 0.2512 0.2494 0.2543 0.2524 0.2583
0.2672 0.2609 0.2626 0.2029 0.1866 0.1958

Columns 37 through 48

0.2241 0.2390 0.2793 0.2983 0.2668 0.2983
0.2837 0.2334 0.1789 0.1514 0.0977 0.1061

Y2 =

Columns 1 through 12

0.0509 -0.3568 0.0570 -0.0160 -0.2853 -0.2026 -
0.1363 0.0027 0.0085 -0.0642 -0.0860 -0.1827

Columns 13 through 24

-0.1768 -0.1874 0.0149 -0.6081 0.2887 0.2725 -
1.1987 0.1937 -0.6176 -0.3833 0.1888 0.2071

Columns 25 through 36

0.2078 0.2678 0.2524 0.1114 0.1278 0.1586
0.1466 0.1838 0.1367 0.1137 0.1602 0.0889

Columns 37 through 48

0.1133 0.1322 0.1968 0.1609 0.2129 0.2246
0.2358 0.2137 0.2295 0.2038 -0.2535 -0.1856

C =

232.0029
233.0020

Root Meant Square Error = 0.04279

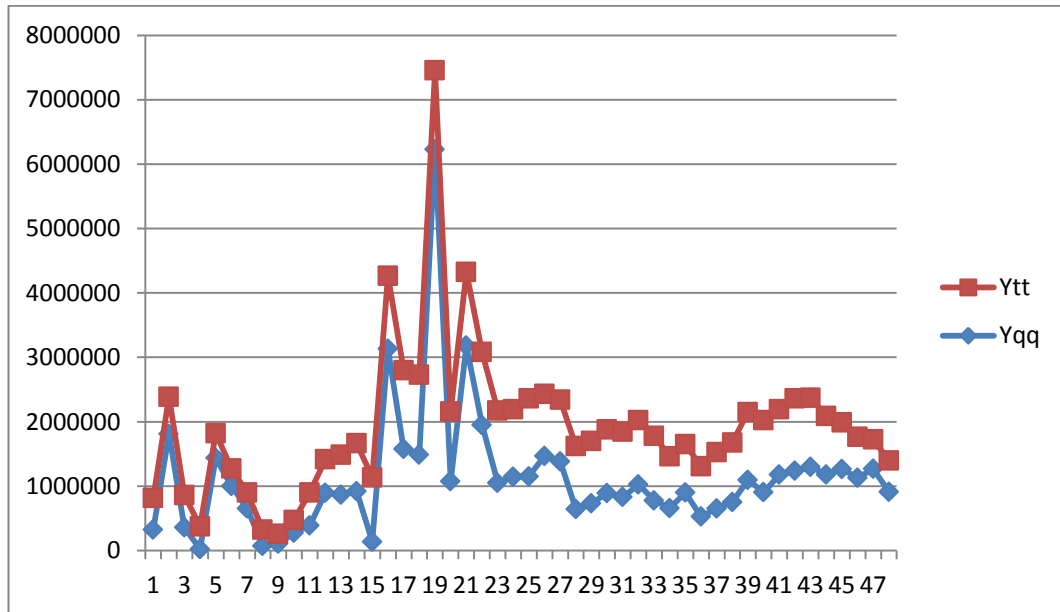
Nilai ytt dan Yqq

prediksi Minggu	
Yqq	Ytt
328110	490102
1812552	578325
359709	503434

23618	351746
1437370	385262
1003302	271851
654921	246606
74582	250150
105371	156395
276401	199394
391167	511963
898620	520731
867459	622961
923176	745361
139016	995788
3132120	1137397
1576269	1222417
1491275	1240966
6232921	1224793
1077803	1080779
3181978	1142999
1952070	1130944
1052052	1121812
1147817	1048351
1151573	1209986
1466892	965586
1385870	959578
645768	975711
731523	969415
893254	988929
830552	1018431
1025724	997589
778563	1003173
657560	805058
901518	751005
527424	781710
655526	875461
754602	924936
1093850	1058645
905276	1121693
1178657	1017339
1239937	1121706
1298472	1073373
1182836	906258

1265643	725554
1130803	634394
1270253	455982
913753	483964

Maka didapatkan grafik



SENIN

Listingnya :

```

clc;
clear;
%load target
load senin_target2.mat
j=senin_target2'

%load data 1
load senin_out1.mat
P1=senin_out1'
P=P1
size(P)

%normalisasi
a=min(min(P));
b=max(max(P));

```

```

p=[ (P-a) / (b-a) ];
t=[ (j-a) / (b-a) ];

%load data 2
load senin_out2.mat
P2=senin_out2';
R=P2;
size(R)

%normalisasi
c=min(min(R));
d=max(max(R));
q=[ (R-c) / (d-c) ];

%bentuk jaringan RBF
% spread=0.1;

%[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
[net,tr] = newrb(p,t,0,0.1,25,1)

% [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
Bobot_input = net.IW{1,1}
Bobot_bias_input = net.b{1,1}
Bobot_Lapisan = net.LW{2,1}
Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
Y1 = sim(net,p)
ytt=((b-a)*Y1)+ a;
ytt=abs(ytt);
ytt=round(ytt);

Y2 = sim(net,q)
yqq=((d-c)*Y2)+ c;
yqq=abs(yqq);
yqq=round(yqq);

%menggambar grafik perbandingan aktual,RBF

n=1:length(Y1);
figure
C=plot(n,j,'-r*',n,ytt,'-bo')
set(C,'LineWidth',1.5)
title('Prediksi senin 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
grid;

%menampilkan RMSE

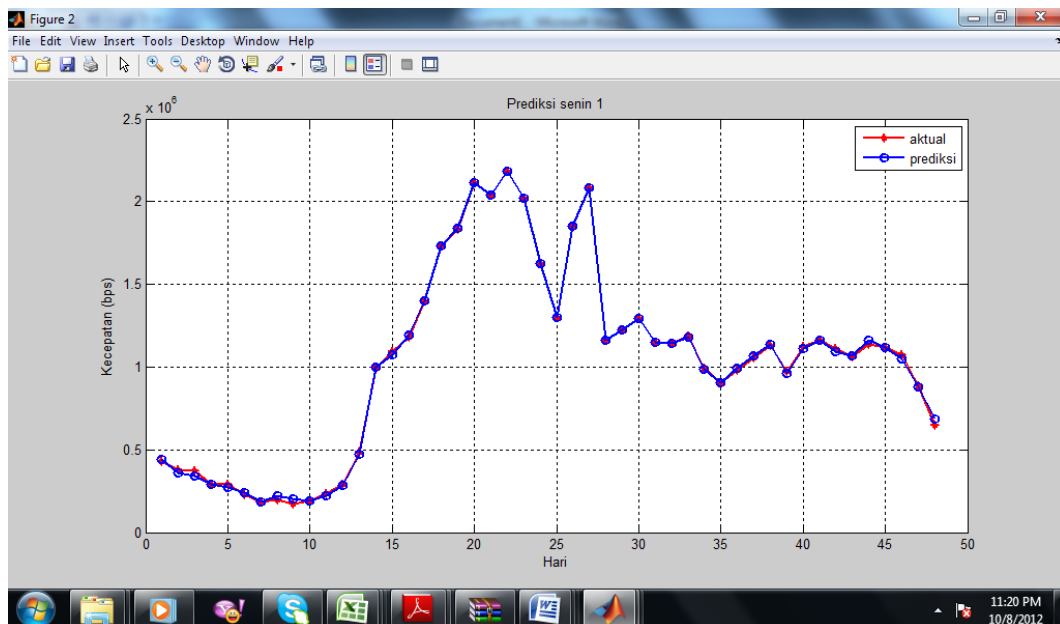
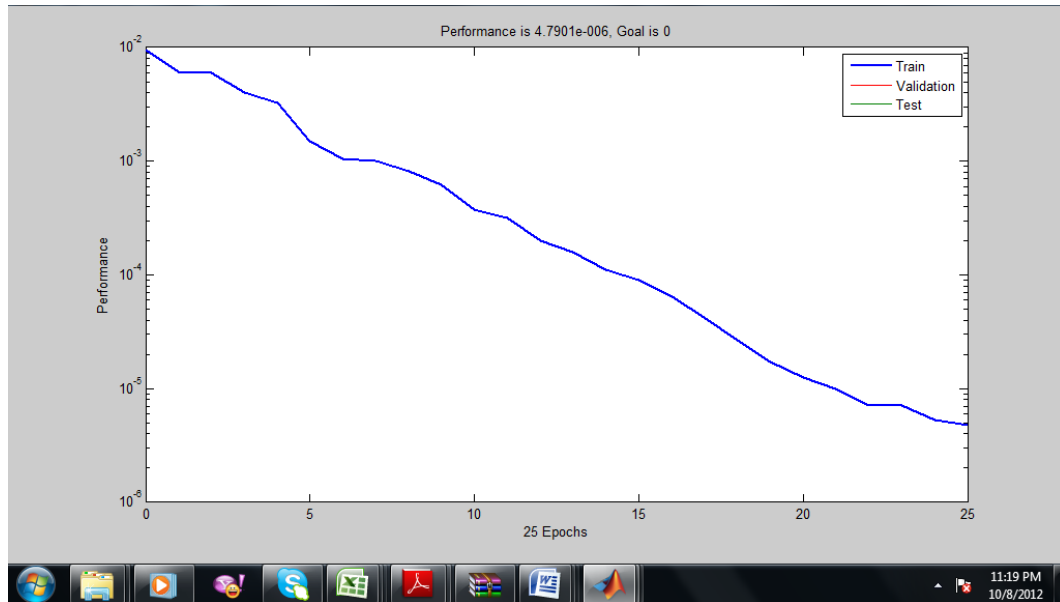
```

```

rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
disp(['Root Meant Square Error = ' num2str(rmse)])

```

maka didapatkan grafiknya :



NEWRB, neurons = 0, MSE = 0.00955207

NEWRB, neurons = 2, MSE = 0.00596048

NEWRB, neurons = 3, MSE = 0.00402613

NEWRB, neurons = 4, MSE = 0.00323521
NEWRB, neurons = 5, MSE = 0.00149223
NEWRB, neurons = 6, MSE = 0.0010592
NEWRB, neurons = 7, MSE = 0.00101258
NEWRB, neurons = 8, MSE = 0.000812728
NEWRB, neurons = 9, MSE = 0.000624934
NEWRB, neurons = 10, MSE = 0.00037513
NEWRB, neurons = 11, MSE = 0.000316039
NEWRB, neurons = 12, MSE = 0.0002017
NEWRB, neurons = 13, MSE = 0.000157989
NEWRB, neurons = 14, MSE = 0.000111634
NEWRB, neurons = 15, MSE = 9.07916e-005
NEWRB, neurons = 16, MSE = 6.46464e-005
NEWRB, neurons = 17, MSE = 4.19809e-005
NEWRB, neurons = 18, MSE = 2.66117e-005
NEWRB, neurons = 19, MSE = 1.73374e-005
NEWRB, neurons = 20, MSE = 1.25126e-005
NEWRB, neurons = 21, MSE = 9.91835e-006
NEWRB, neurons = 22, MSE = 7.09467e-006
NEWRB, neurons = 23, MSE = 7.09244e-006
NEWRB, neurons = 24, MSE = 5.25804e-006
NEWRB, neurons = 25, MSE = 4.7901e-006

net =

Neural Network object:

architecture:

numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]

numOutputs: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

subobject structures:

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

functions:

adaptFcn: (none)
divideFcn: (none)
gradientFcn: (none)
initFcn: (none)
performFcn: 'mse'

```
plotFcns: {}  
trainFcn: (none)
```

```
parameters:
```

```
    adaptParam: (none)  
    divideParam: (none)  
gradientParam: (none)  
    initParam: (none)  
performParam: (none)  
    trainParam: (none)
```

```
weight and bias values:
```

```
    IW: {2x1 cell} containing 1 input weight matrix  
    LW: {2x2 cell} containing 1 layer weight matrix  
    b: {2x1 cell} containing 2 bias vectors
```

```
other:
```

```
    name: ''  
    userdata: (user information)
```

```
tr =
```

```
    epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
21 22 23 24 25]
```


perf: [1x26 double]

Robot_input =

0.0165	0.1842	0.2870	0.2701
0.1086	0.0981	0.1294	0.1736
0.3391	0.3061	0.2755	0
0.3136	0.2582	0.1790	0.3019
0.0316	0.3138	0.3065	0.3231
0.3511	0.2960	0.1695	1.0000
0.0151	0.0533	0.1520	0.0950
0.2058	0.1956	0.1550	0.2836
0.3061	0.3027	0.1888	0.1956
0.0165	0.1139	0.2061	0.1610
0.3793	0.3185	0.1548	0
0.3055	0.3288	0.3050	0
0.2423	0.2119	0.1601	0.3633
0.1475	0.0945	0.1369	0.1221
0.0165	0.0957	0.1713	0.2364
0.1535	0.1186	0.1349	0.2370
0.0321	0.2280	0.3236	0.2720
0.0192	0.1454	0.2642	0.2417
0.0320	0.3104	0.2423	0.3183
0.1837	0.1839	0.1650	0.2566
0.0774	0.0767	0.0765	0.0708
0.0151	0.2633	0.3102	0.3154
0.0098	0.0452	0.0183	0.0404

0.0318	0.0577	0.0247	0.0418
0.0284	0.2825	0.3314	0.2967

Bobot_bias_input =

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

8.3255

Bobot_Lapisan =

Columns 1 through 12

-0.0893	-0.1837	-0.0600	0.1054	0.1979	0.2199
0.1195	-0.1870	0.1661	0.0663	0.1661	0.3891

Columns 13 through 24

0.1744	0.2204	0.0827	0.0927	0.2462	0.1796
0.0835	0.2092	-0.0246	-0.1766	-0.1256	0.1552

Column 25

0.1456

Bobot_Bias_lapisan =

-0.0109

Y1 =

Columns 1 through 12

0.0751	0.0611	0.0581	0.0498	0.0468	0.0415
0.0315	0.0376	0.0342	0.0325	0.0384	0.0483

Columns 13 through 24

0.0810	0.1711	0.1837	0.2041	0.2391	0.2954
0.3135	0.3610	0.3479	0.3727	0.3447	0.2772

Columns 25 through 36

0.2224	0.3161	0.3559	0.1988	0.2091	0.2212
0.1960	0.1957	0.2020	0.1685	0.1542	0.1696

Columns 37 through 48

0.1822	0.1944	0.1643	0.1896	0.1989	0.1870
0.1823	0.1980	0.1907	0.1786	0.1503	0.1167

Y2 =

Columns 1 through 12

0.0540	0.0591	0.0628	0.0471	0.0420	0.0493
0.0423	0.0459	0.0407	0.0351	0.0387	0.0401

Columns 13 through 24

0.0447 0.1129 0.1016 0.0366 0.0103 0.0173 -
0.0057 -0.0074 -0.0098 -0.0104 -0.0103 -0.0102

Columns 25 through 36

-0.0104 -0.0093 0.0831 -0.0093 -0.0108 -0.0011
0.0192 0.0481 0.0424 0.0563 0.1151 0.1326

Columns 37 through 48

0.1388 0.1411 0.1204 0.1315 0.1246 0.1356
0.1273 0.1225 0.1241 0.1383 0.1219 0.0927

C =

457.0090

458.0032

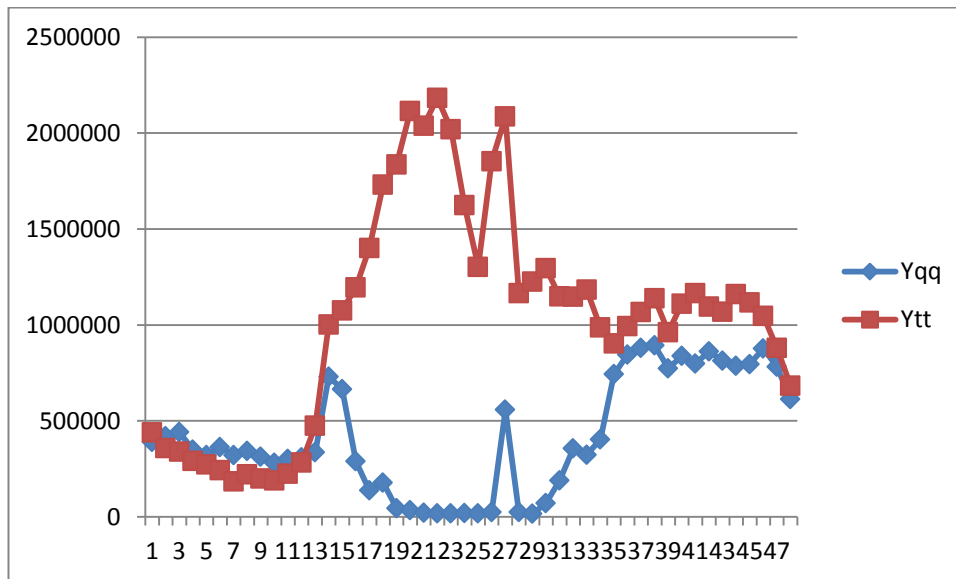
Root Meant Square Error = 0.0063797

Nilai Yqq dan Ytt

Prediksi Senin	
Yqq	Ytt
390106	439939
419427	358280
441068	340151
350288	291895
320891	273939
363127	243055
322276	184296
343231	220596
313278	200503

280947	190420
301877	224729
309781	282810
336366	474834
730556	1002458
665473	1076250
289604	1195792
137572	1401031
178202	1730652
44973	1836893
35081	2115006
21228	2038336
17597	2183594
18156	2019642
19040	1624426
17944	1302988
24267	1852101
558279	2085524
24191	1164947
15674	1224904
71779	1296065
189121	1148421
355909	1146475
323232	1183348
403540	987028
743394	903267
844515	993579
880341	1067476
893251	1139285
774023	962639
837832	1110927
797951	1165225
861988	1095647
813638	1068073
786188	1159904
795326	1117123
877071	1046425
782521	880754
613793	683475

Maka didapatkan grafik (senin) :



SELASA

LISTINGnya :

```
clc;
clear;
%load target
load selasa_targetout.mat
j=selasa_targetout'
```

```
%load data 1
load selasa_out1.mat
P1=selasa_out1'
P=P1
size(P)
```

```
%normalisasi
a=min(min(P));
b=max(max(P));
p=[(P-a)/(b-a)];
t=[(j-a)/(b-a)];
```

```
%load data 2
load selasa_out2.mat
P2=selasa_out2';
R=P2;
size(R)
```

```
%normalisasi
c=min(min(R));
```

```

d=max(max(R));
q=[(R-c)/(d-c)];

%bentuk jaringan RBF
%   spread=0.1;

%[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
[net,tr] = newrb(p,t,0,0.3,25,1)

% [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
    Bobot_input = net.IW{1,1}
    Bobot_bias_input = net.b{1,1}
    Bobot_Lapisan = net.LW{2,1}
    Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
    Y1 = sim(net,p)
    ytt=((b-a)*Y1)+ a;
    ytt=abs(ytt);
    ytt=round(ytt);

    Y2 = sim(net,q)
    yqq=((d-c)*Y2)+ c;
    yqq=abs(yqq);
    yqq=round(yqq);

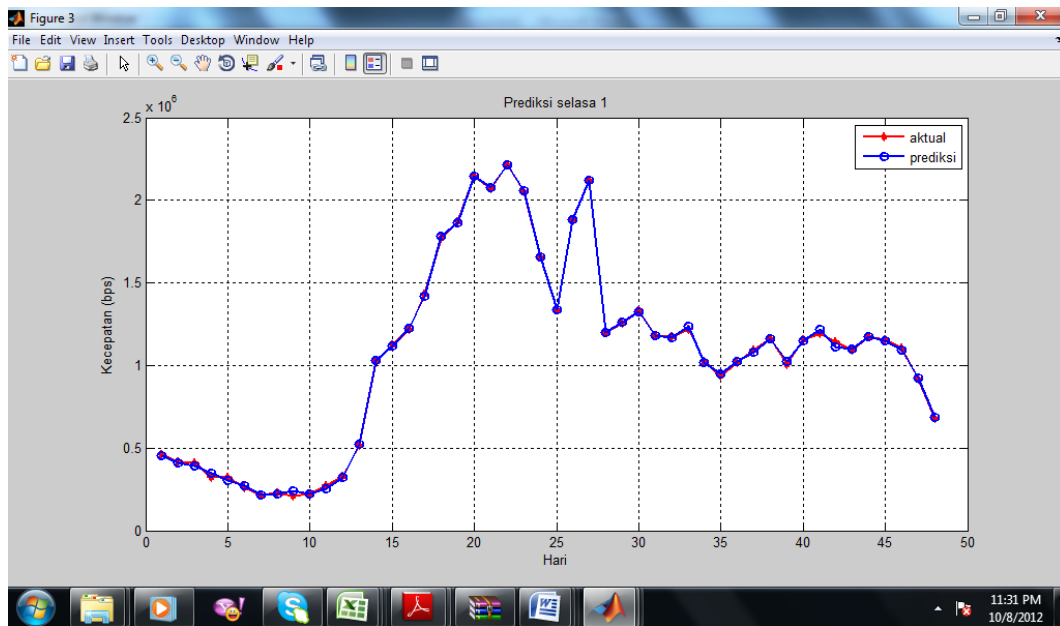
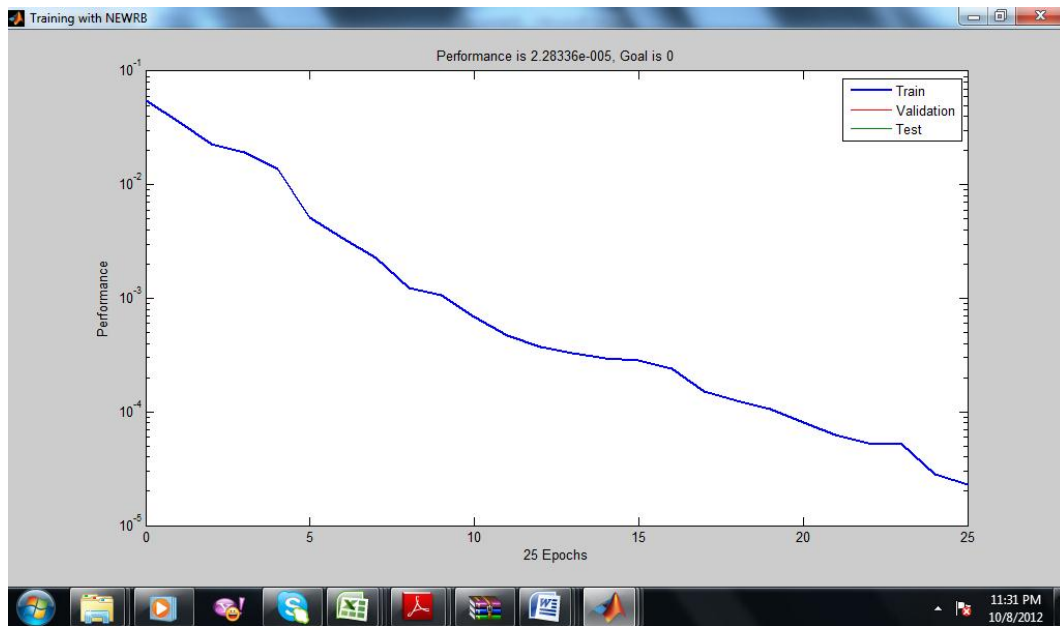
%menggambar grafik perbandingan aktual,RBF

n=1:length(Y1);
figure
C=plot(n,j,'-r*',n,ytt,'-bo')
set(C,'LineWidth',1.5)
title('Prediksi selasa 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
grid;

%menampilkan RMSE
rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
disp(['Root Meant Square Error = ' num2str(rmse)])

```

didapatkan grafik



NEWRB, neurons = 0, MSE = 0.054807
 NEWRB, neurons = 2, MSE = 0.0227326
 NEWRB, neurons = 3, MSE = 0.0193193
 NEWRB, neurons = 4, MSE = 0.0137552
 NEWRB, neurons = 5, MSE = 0.00518992

NEWRB, neurons = 6, MSE = 0.0033464
NEWRB, neurons = 7, MSE = 0.00225552
NEWRB, neurons = 8, MSE = 0.00122945
NEWRB, neurons = 9, MSE = 0.00106632
NEWRB, neurons = 10, MSE = 0.000691742
NEWRB, neurons = 11, MSE = 0.000472772
NEWRB, neurons = 12, MSE = 0.000376281
NEWRB, neurons = 13, MSE = 0.000326892
NEWRB, neurons = 14, MSE = 0.000297034
NEWRB, neurons = 15, MSE = 0.000284718
NEWRB, neurons = 16, MSE = 0.000237764
NEWRB, neurons = 17, MSE = 0.000149772
NEWRB, neurons = 18, MSE = 0.000126492
NEWRB, neurons = 19, MSE = 0.00010647
NEWRB, neurons = 20, MSE = 8.06961e-005
NEWRB, neurons = 21, MSE = 6.30512e-005
NEWRB, neurons = 22, MSE = 5.2969e-005
NEWRB, neurons = 23, MSE = 5.18919e-005
NEWRB, neurons = 24, MSE = 2.85214e-005
NEWRB, neurons = 25, MSE = 2.28336e-005

net =

Neural Network object:

architecture:

numInputs: 1
numLayers: 2
biasConnect: [1; 1]
inputConnect: [1; 0]
layerConnect: [0 0; 1 0]
outputConnect: [0 1]

numOutputs: 1 (read-only)
numInputDelays: 0 (read-only)
numLayerDelays: 0 (read-only)

subobject structures:

inputs: {1x1 cell} of inputs
layers: {2x1 cell} of layers
outputs: {1x2 cell} containing 1 output
biases: {2x1 cell} containing 2 biases
inputWeights: {2x1 cell} containing 1 input weight
layerWeights: {2x2 cell} containing 1 layer weight

functions:

adaptFcn: (none)
divideFcn: (none)
gradientFcn: (none)

```
    initFcn: (none)
  performFcn: 'mse'
    plotFcns: {}
    trainFcn: (none)
```

parameters:

```
    adaptParam: (none)
    divideParam: (none)
  gradientParam: (none)
    initParam: (none)
  performParam: (none)
    trainParam: (none)
```

weight and bias values:

```
    IW: {2x1 cell} containing 1 input weight
matrix
```

```
    LW: {2x2 cell} containing 1 layer weight
matrix
```

```
    b: {2x1 cell} containing 2 bias vectors
```

other:

```
    name: ''
```

```
    userdata: (user information)
```

tr =

epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25]

perf: [1x26 double]

Bobot_input =

0.0161	0.4214	0.6676	0.7273
0.7888	0.7135	0.6399	0.7442
0.2221	0.2695	0.3516	0.3182
0.0522	0.7319	0.7143	0.9621
0.0446	0.6569	0.7740	0.3545
0.0161	0.2530	0.4739	0.9321
0.5569	0.4878	0.3637	0.7993
0.7084	0.7679	0.7107	0.5885
0.0126	0.1080	0.3441	0.3911
0.0533	0.5263	0.7553	0.9838
0.8175	0.6891	0.3861	0.7708
0.2937	0.2439	0.3711	0.3399
0.2900	0.2013	0.3438	0.3969
0.0128	0.6108	0.7233	0.4664
0.3230	0.2318	0.2645	0.4616
0.7096	0.7053	0.4324	0.9190
0.8851	0.7431	0.3509	0.6583

2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752
2.7752

Bobot_Lapisan =

Columns 1 through 12

0.1621	0.0961	-0.0297	0.5631	1.0356	0.3179
0.2229	0.6618	0.5244	0.3115	-0.7613	3.8449

Columns 13 through 24

3.0533	-0.2910	-1.4716	0.5099	0.6213	-0.1003
3.9059	-0.0265	-0.0638	2.1449	0.1296	-11.0927

Column 25

-0.1983

Bobot_Bias_lapisan =

-0.0139

Y1 =

Columns 1 through 12

0.1573	0.1391	0.1322	0.1124	0.0966	0.0836
0.0600	0.0615	0.0695	0.0623	0.0752	0.1031

Columns 13 through 24

0.1852	0.3929	0.4283	0.4725	0.5527	0.7005
0.7325	0.8477	0.8202	0.8773	0.8118	0.6488

Columns 25 through 36

0.5174	0.7420	0.8385	0.4620	0.4876	0.5126
0.4538	0.4481	0.4782	0.3886	0.3602	0.3899

Columns 37 through 48

0.4145	0.4458	0.3893	0.4415	0.4690	0.4253
0.4201	0.4526	0.4419	0.4194	0.3493	0.2510

Y2 =

Columns 1 through 12

0.2125	0.1981	0.2101	0.1680	0.1639	0.1334
0.1285	0.1199	0.1170	0.1223	0.1179	0.1249

Columns 13 through 24

0.1998	0.3048	0.3957	0.3989	0.2463	0.0724
0.0436	0.0304	0.0025	0.0024	0.0209	0.1280

Columns 25 through 36

0.1086	0.0546	0.0221	0.0393	0.0189	0.1538
0.2087	0.3265	0.3121	0.3460	0.3715	0.4199

Columns 37 through 48

0.4274	0.4180	0.4171	0.4217	0.4097	0.4415
0.4270	0.3946	0.3937	0.4030	0.3943	0.3159

C =

682.0012

683.0007

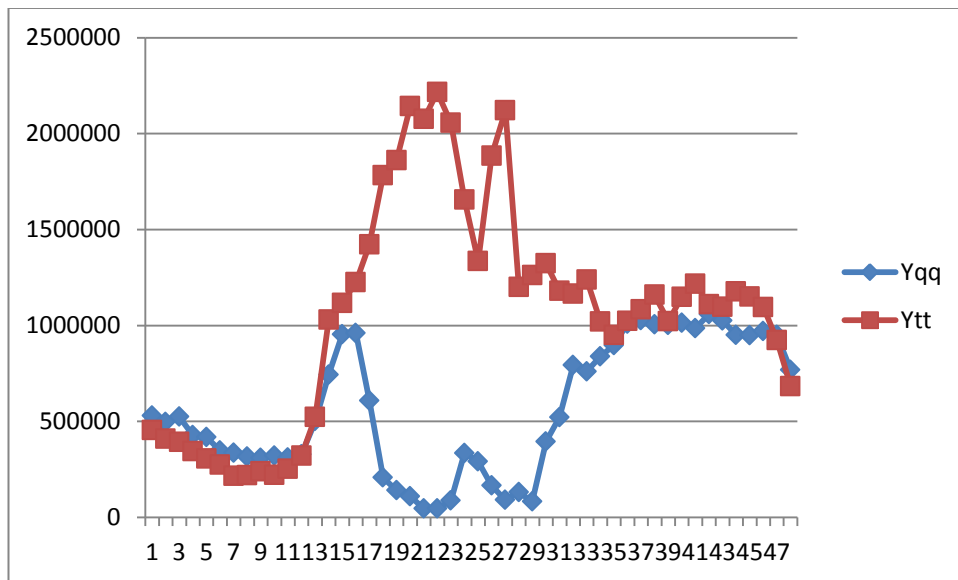
Root Meant Square Error = 0.005815

Nilai Yqq dan Ytt :

Prediksi Selasa	
Yqq	Ytt
530573	454551
497297	409899
524994	393198
427951	344736
418458	305955
348003	274299
336681	216566
316878	220230
310090	239872
322320	222104
312167	253729
328380	321926
501355	522733
743671	1030735
953645	1117291
960902	1225559
608761	1421676
207294	1783238
140727	1861474
110303	2143187
45966	2076126
45560	2215666
88425	2055528
335643	1656746
290886	1335408
166124	1884833
91110	2120683
130898	1199852
83678	1262367

395123	1323640
521883	1179785
793911	1165896
760546	1239505
838873	1020317
897620	950784
1009407	1023435
1026664	1083574
1004983	1160188
1002833	1022092
1013548	1149668
985823	1217026
1059335	1110061
1025827	1097379
951058	1176738
948822	1150569
970362	1095696
950350	924086
769376	683621

Grafiknya :



RABU

Listingnya :

```

clc;
clear;
%load target
load rabu_targetout.mat
j=rabu_targetout'

%load data 1
load rabu_out1.mat
P1=rabu_out1'
P=P1
size(P)

%normalisasi
a=min(min(P));
b=max(max(P));
p=[(P-a)/(b-a)];
t=[(j-a)/(b-a)];

%load data 2
load rabu_out2.mat
P2=rabu_out2';
R=P2;
size(R)

%normalisasi
c=min(min(R));
d=max(max(R));
q=[(R-c)/(d-c)];

%bentuk jaringan RBF
% spread=0.1;

%[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
[net,tr] = newrb(p,t,0,1,40,1)

% [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
Bobot_input = net.IW{1,1}
Bobot_bias_input = net.b{1,1}
Bobot_Lapisan = net.LW{2,1}
Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
Y1 = sim(net,p)
ytt=((b-a)*Y1)+ a;
ytt=abs(ytt);
ytt=round(ytt);

Y2 = sim(net,q)
yqq=((d-c)*Y2)+ c;

```

```

yqq=abs(yqq);
yqq=round(yqq);

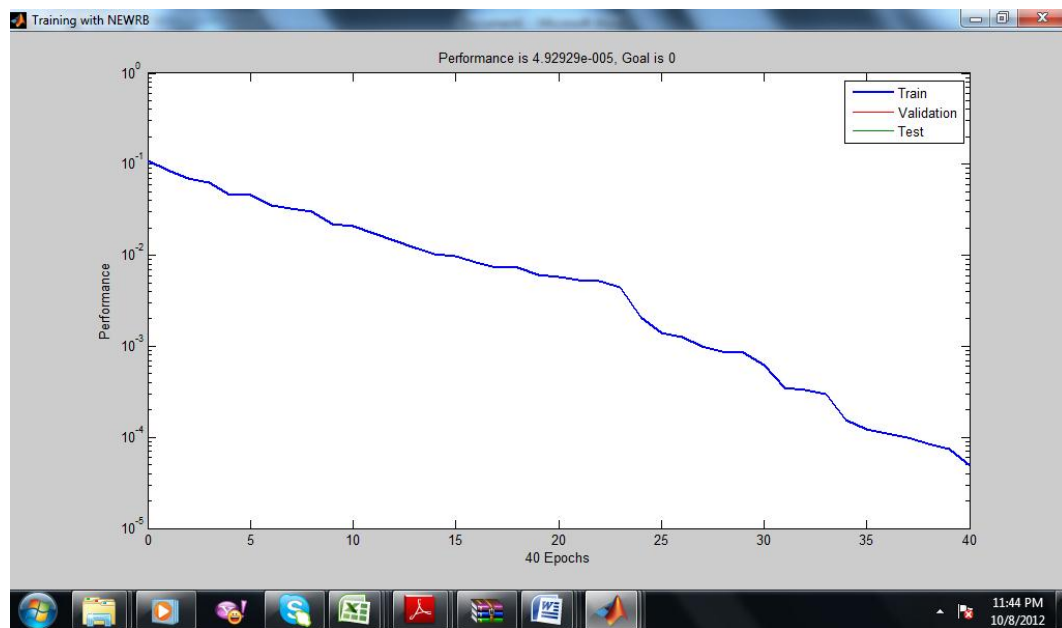
%menggambar grafik perbandingan aktual,RBF

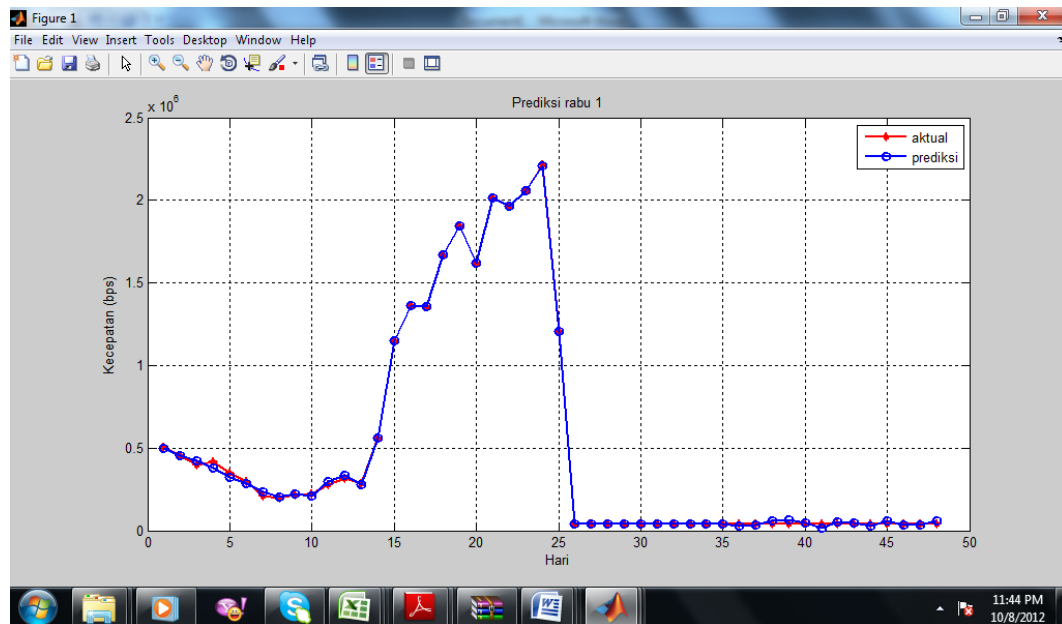
n=1:length(Y1);
figure
C=plot(n,j,'-r*',n,ytt,'-bo')
set(C,'LineWidth',1.5)
title('Prediksi rabu 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');
grid;

%menampilkan RMSE
rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));
disp(['Root Meant Square Error = ' num2str(rmse)])

```

grafik yg didapat :





NEWRB, neurons = 0, MSE = 0.109797

NEWRB, neurons = 2, MSE = 0.0691805

NEWRB, neurons = 3, MSE = 0.0629055

NEWRB, neurons = 4, MSE = 0.045829

NEWRB, neurons = 5, MSE = 0.0457044

NEWRB, neurons = 6, MSE = 0.0356485

NEWRB, neurons = 7, MSE = 0.0330355

NEWRB, neurons = 8, MSE = 0.0303033

NEWRB, neurons = 9, MSE = 0.0222442

NEWRB, neurons = 10, MSE = 0.0208013

NEWRB, neurons = 11, MSE = 0.01717

NEWRB, neurons = 12, MSE = 0.0143135

NEWRB, neurons = 13, MSE = 0.0121677

NEWRB, neurons = 14, MSE = 0.0102948

NEWRB, neurons = 15, MSE = 0.00976695

NEWRB, neurons = 16, MSE = 0.00839639

NEWRB, neurons = 17, MSE = 0.00736112

NEWRB, neurons = 18, MSE = 0.00728911
NEWRB, neurons = 19, MSE = 0.00615862
NEWRB, neurons = 20, MSE = 0.00576371
NEWRB, neurons = 21, MSE = 0.00533229
NEWRB, neurons = 22, MSE = 0.0051674
NEWRB, neurons = 23, MSE = 0.00443429
NEWRB, neurons = 24, MSE = 0.00208753
NEWRB, neurons = 25, MSE = 0.00139638
NEWRB, neurons = 26, MSE = 0.00125377
NEWRB, neurons = 27, MSE = 0.00100219
NEWRB, neurons = 28, MSE = 0.000869988
NEWRB, neurons = 29, MSE = 0.00085127
NEWRB, neurons = 30, MSE = 0.000625253
NEWRB, neurons = 31, MSE = 0.000351418
NEWRB, neurons = 32, MSE = 0.000332854
NEWRB, neurons = 33, MSE = 0.000299703
NEWRB, neurons = 34, MSE = 0.00015609
NEWRB, neurons = 35, MSE = 0.000123612
NEWRB, neurons = 36, MSE = 0.000111441
NEWRB, neurons = 37, MSE = 9.96903e-005
NEWRB, neurons = 38, MSE = 8.48226e-005
NEWRB, neurons = 39, MSE = 7.45234e-005
NEWRB, neurons = 40, MSE = 4.92929e-005

net =

Neural Network object:

architecture:

numInputs: 1

numLayers: 2

biasConnect: [1; 1]

inputConnect: [1; 0]

layerConnect: [0 0; 1 0]

outputConnect: [0 1]

numOutputs: 1 (read-only)

numInputDelays: 0 (read-only)

numLayerDelays: 0 (read-only)

subobject structures:

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

functions:

adaptFcn: (none)

divideFcn: (none)

gradientFcn: (none)

initFcn: (none)


```
performFcn: 'mse'  
plotFcns: {}  
trainFcn: (none)
```

parameters:

```
adaptParam: (none)  
divideParam: (none)  
gradientParam: (none)  
initParam: (none)  
performParam: (none)  
trainParam: (none)
```

weight and bias values:

```
IW: {2x1 cell} containing 1 input weight matrix  
LW: {2x2 cell} containing 1 layer weight matrix  
b: {2x1 cell} containing 2 bias vectors
```

other:

```
name: ''  
userdata: (user information)
```

tr =

```
epoch: [1x41 double]
```

perf: [1x41 double]

Robot_input =

0.9169	1.0000	0.8203	0.8476
0.5840	0.9007	0.9952	0.9234
0.8284	0.8080	0.7822	0.9686
0.7528	0.8318	0.4375	0.9184
0.3824	0.3901	0.7564	0.4085
0.8925	0.9151	0.3901	0.8872
0.9451	0.6893	0.6463	0.9649
0.5600	0.8697	0.4481	0.8275
0.6596	0.4756	0.7651	0.5656
0.8070	0.8941	0.7394	0.7906
0.5527	0.5049	0.6475	0.6104
0.2457	0.2634	0.4723	0.6010
0.1807	0.0971	0.4277	0.5642
0.4431	0.4762	0.7964	0.5503
0.2718	0.2710	0.6203	0.7328
0.4484	0.2449	0.6386	0.3946
0.1861	0.0513	0.3472	0.1799
0.3288	0.1408	0.2750	0.4909
0.3240	0.1652	0.2810	0.3966
0.3044	0.1655	0.2504	0.4519
0.2756	0.1678	0.4546	0.7298
0.2493	0.1759	0.2631	0.4225
0.6932	0.6971	0.8392	0.7640

0.8260	0.6306	0.5859	0.8259
0.1092	0.0758	0.1196	0.0765
0.4562	0.8369	0.4807	0.8612
0.5596	0.3732	0.7296	0.3748
0.1532	0.1065	0.1134	0.0971
0.1525	0.1075	0.1572	0.1943
0.0901	0.0347	0.1891	0.1113
0.0614	0.0698	0.0466	0.0403
0.3033	0.1660	0.2369	0.3902
0.0639	0.1491	0.0247	0.0126
0.0771	0.0671	0.0364	0.0269
0.2820	0.2119	0.2126	0.4637
0.0880	0.0809	0.0612	0.0604
0.3419	0.2233	0.1880	0.4851
0.0817	0.0904	0.0088	0.0124
0.3184	0.1819	0.2048	0.4207
0.3128	0.1914	0.2227	0.3928

Bobot_bias_input =

0.8326
0.8326
0.8326
0.8326
0.8326
0.8326
0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

0.8326

Bobot_Lapisan =

1.0e+005 *

Columns 1 through 12

0.0545	0.0137	-0.0244	0.0104	0.0224	-0.0021
0.0167	0.0172	0.0587	-0.0867	0.0632	-0.3624

Columns 13 through 24

-0.1316	0.0015	0.0789	-0.1526	0.1999	-0.2274
2.0359	-1.9750	0.1114	3.2278	-0.0385	-0.0042

Columns 25 through 36

-2.7061	0.0079	0.0085	1.9088	-1.3684	0.3710	-
0.1520	-3.6117	0.3315	1.8476	-0.8836	1.1800	

Columns 37 through 40

-0.3763	-1.9217	4.8778	-2.4071
---------	---------	--------	---------

Bobot_Bias_lapisan =

-295.0441

Y1 =

Columns 1 through 12

0.1786	0.1588	0.1420	0.1210	0.0925	0.0744
0.0495	0.0342	0.0432	0.0377	0.0827	0.0998

Columns 13 through 24

0.0710	0.2092	0.4984	0.6028	0.5983	0.7522
0.8394	0.7283	0.9221	0.8974	0.9412	1.0171

Columns 25 through 36

0.5243	-0.0443	-0.0444	-0.0444	-0.0444	-0.0443	-
0.0445	-0.0441	-0.0443	-0.0440	-0.0437	-0.0515	

Columns 37 through 48

-0.0473	-0.0355	-0.0325	-0.0400	-0.0705	-0.0370	-
0.0410	-0.0505	-0.0360	-0.0472	-0.0459	-0.0366	

Y2 =

Columns 1 through 12

-0.0614	0.1177	0.0387	0.0385	0.1841	-0.0423	-
0.0882	-0.0838	-0.2025	-0.1875	-0.1337	-0.0618	

Columns 13 through 24

-0.0503	-0.2155	0.1970	0.4485	0.2331	0.5015	-
0.3160	0.1823	-0.0960	-0.4764	-1.9980	1.2564	

Columns 25 through 36

-0.6516	-3.1384	-9.1423	-6.7618	1.2230	-1.3371	-
1.3696	-0.4694	0.0593	0.2952	0.0584	-0.2107	

Columns 37 through 48

-0.1791	-0.3459	-0.3207	-1.0242	-0.4653	-0.3128	-
0.0945	0.0047	-0.1247	-0.1094	-0.0568	0.1116	

C =

223.0034

224.0024

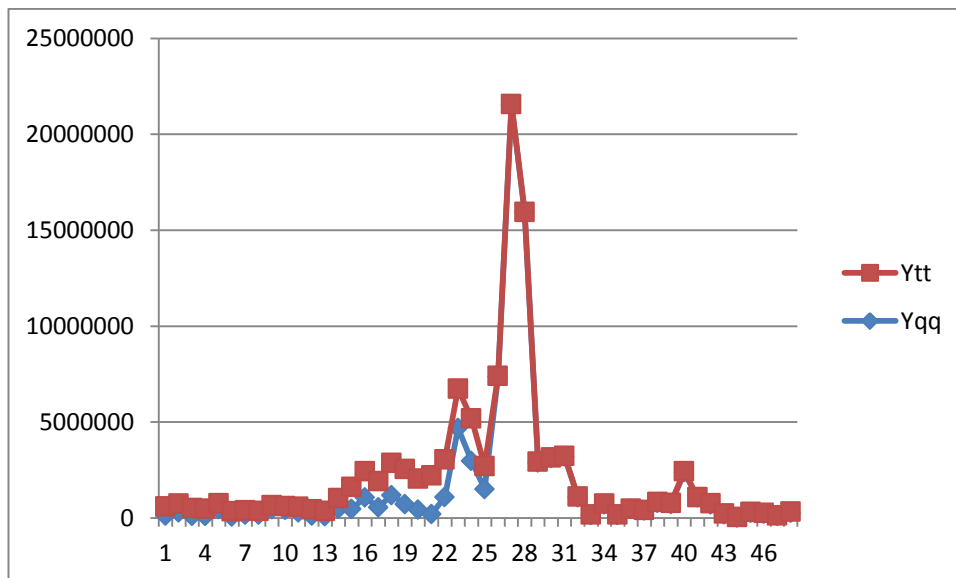
Root Meant Square Error = 0.0058814

Nilai Yqq dan Ytt :

Prediksi Rabu	
Yqq	Ytt
123849	496693
298567	456186
112092	421796
111800	378727
455155	320604
78887	283411
187150	232640
176780	201316
456656	219642
421285	208406
294404	300384
124787	335418
97780	276556
487293	559273
485599	1150847
1078753	1364281
570615	1355214
1203840	1669992
724278	1848237
450822	1620997
205504	2017418
1102704	1967021
4691340	2056452
2984282	2211724
1515959	1203686
7380946	40710
21541373	40531
15926783	40439
2905449	40585
3132540	40770
3209347	40334
1086143	41151
160699	40636
717166	41378
158710	41997
476071	25965
401445	34592
794827	58618

735542	64870
2394591	49535
1076388	12804
716867	55670
202020	47366
31984	28112
273086	57764
237173	34752
113112	37387
284054	56493

Grafiknya :



KAMIS

Listingnya :

```

clc;
clear;
%load target
load kumis_targetout.mat
j=kumis_targetout'

%load data 1
load kumis_out1.mat
P1=kumis_out1'
P=P1

```

```

size(P)

%normalisasi
a=min(min(P));
b=max(max(P));
p=[(P-a)/(b-a)];
t=[(j-a)/(b-a)];

%load data 2
load Kamis_out2.mat
P2=Kamis_out2';
R=P2;
size(R)

%normalisasi
c=min(min(R));
d=max(max(R));
q=[(R-c)/(d-c)];

%bentuk jaringan RBF
% spread=0.1;

%[net,tr] = newrb(P,T,GOAL,SPREAD,MN,DF)
[net,tr] = newrb(p,t,0,3,25,1)

% [net,tr] = newrb(P,target,GOAL,spread,0,31)

%bobot input,lapisan,dan bias
Bobot_input = net.IW{1,1}
Bobot_bias_input = net.b{1,1}
Bobot_Lapisan = net.LW{2,1}
Bobot_Bias_lapisan = net.b{2,1}

%simulasi
%data training
Y1 = sim(net,p)
ytt=((b-a)*Y1)+ a;
ytt=abs(ytt);
ytt=round(ytt);

Y2 = sim(net,q)
yqq=((d-c)*Y2)+ c;
yqq=abs(yqq);
yqq=round(yqq);

%menggambar grafik perbandingan aktual,RBF

n=1:length(Y1);
figure
C=plot(n,j,'-r*',n,ytt,'-bo')
set(C,'LineWidth',1.5)
title('Prediksi Kamis 1');xlabel('Hari');ylabel('Kecepatan
(bps)');legend('aktual','prediksi');

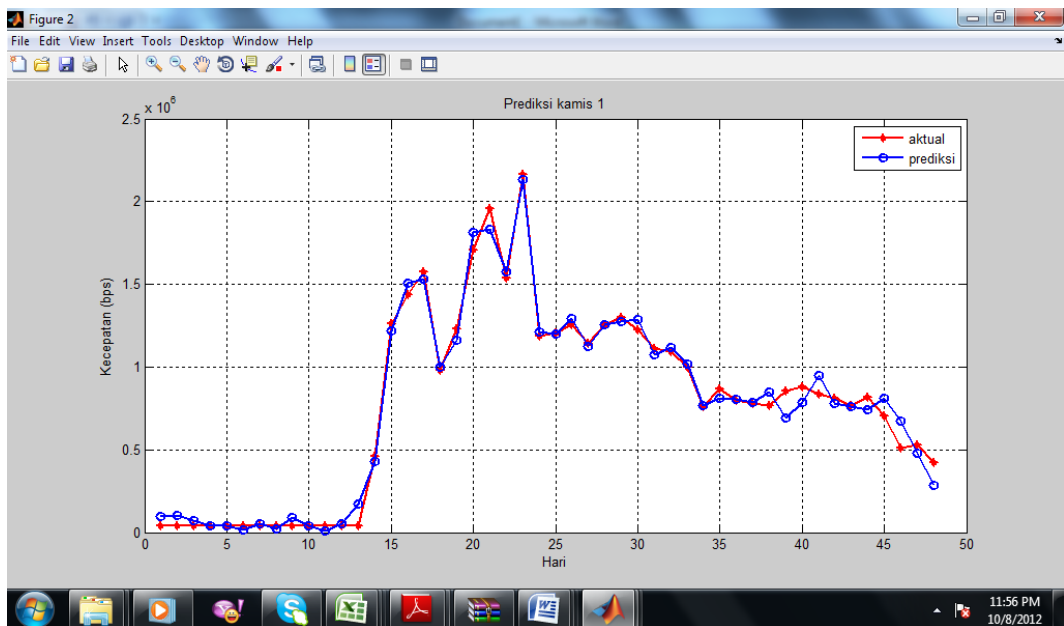
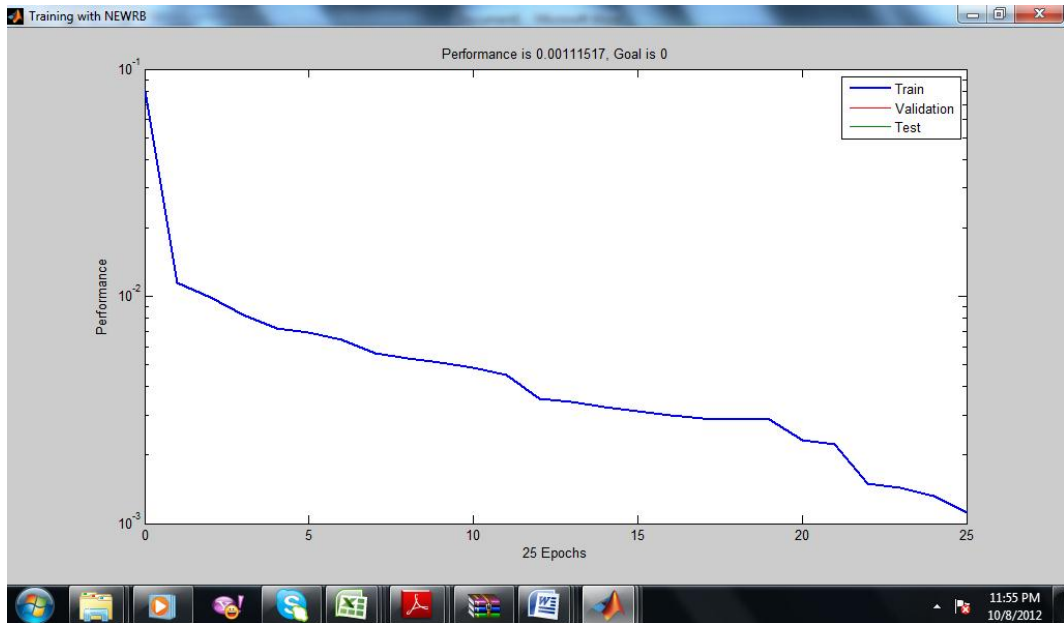
```

```
grid;
```

```
%menampilkan RMSE
```

```
rmse = (sqrt(1/48*sum((j-ytt).^2)))/(max(j)-min(j));  
disp(['Root Meant Square Error = ' num2str(rmse)])
```

grafik dari matlab :



NEWRB, neurons = 0, MSE = 0.0796076
NEWRB, neurons = 2, MSE = 0.00991672
NEWRB, neurons = 3, MSE = 0.0083172
NEWRB, neurons = 4, MSE = 0.0072616
NEWRB, neurons = 5, MSE = 0.00689846
NEWRB, neurons = 6, MSE = 0.00641666
NEWRB, neurons = 7, MSE = 0.00562462
NEWRB, neurons = 8, MSE = 0.00531991
NEWRB, neurons = 9, MSE = 0.00512123
NEWRB, neurons = 10, MSE = 0.00482702
NEWRB, neurons = 11, MSE = 0.00450072
NEWRB, neurons = 12, MSE = 0.00355826
NEWRB, neurons = 13, MSE = 0.00341649
NEWRB, neurons = 14, MSE = 0.00326473
NEWRB, neurons = 15, MSE = 0.0031066
NEWRB, neurons = 16, MSE = 0.00300334
NEWRB, neurons = 17, MSE = 0.00288882
NEWRB, neurons = 18, MSE = 0.00286977
NEWRB, neurons = 19, MSE = 0.00286674
NEWRB, neurons = 20, MSE = 0.00233775
NEWRB, neurons = 21, MSE = 0.00223474
NEWRB, neurons = 22, MSE = 0.00149242
NEWRB, neurons = 23, MSE = 0.00143247
NEWRB, neurons = 24, MSE = 0.00131617
NEWRB, neurons = 25, MSE = 0.00111517

net =

Neural Network object:

architecture:

numInputs: 1

numLayers: 2

biasConnect: [1; 1]

inputConnect: [1; 0]

layerConnect: [0 0; 1 0]

outputConnect: [0 1]

numOutputs: 1 (read-only)

numInputDelays: 0 (read-only)

numLayerDelays: 0 (read-only)

subobject structures:

inputs: {1x1 cell} of inputs

layers: {2x1 cell} of layers

outputs: {1x2 cell} containing 1 output

biases: {2x1 cell} containing 2 biases

inputWeights: {2x1 cell} containing 1 input weight

layerWeights: {2x2 cell} containing 1 layer weight

functions:

```
    adaptFcn: (none)
    divideFcn: (none)
    gradientFcn: (none)
    initFcn: (none)
    performFcn: 'mse'
    plotFcns: {}
    trainFcn: (none)
```

parameters:

```
    adaptParam: (none)
    divideParam: (none)
    gradientParam: (none)
    initParam: (none)
    performParam: (none)
    trainParam: (none)
```

weight and bias values:

```
    IW: {2x1 cell} containing 1 input weight
matrix
    LW: {2x2 cell} containing 1 layer weight
matrix
    b: {2x1 cell} containing 2 bias vectors
```

other:

name: ''

userdata: (user information)

tr =

epoch: [0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25]

perf: [1x26 double]

Bobot_input =

1.0000	0.6865	0.8508	0.9407
0.9894	0.6942	0.7647	0.8919
0.6223	0.6652	0.5123	0.6433
0.0836	0.0622	0.0240	0.0571
0.2114	0.1159	0	0.0885
0.1546	0.0704	0.0153	0.0610
0.5252	0.3383	0.7318	0.3740
0.1061	0.0763	0.0028	0.0921
0.8473	0.6032	0.4559	0.5464
0.9159	0.6126	0.4131	0.7643
0.9047	0.6233	0.8661	0.7195
0.1899	0.1609	0.4045	0.4129

0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775
0.2775

Bobot_Lapisan =

1.0e+007 *

Columns 1 through 12

-0.0200	0.0097	-0.0206	0.8246	0.2903	-1.6293
-0.0141	-0.1331	0.0133	0.0253	0.0831	0.0267

Columns 13 through 24

1.2593	0.0926	-0.0498	-0.0184	0.1869	-0.1688
0.1974	0.1778	-0.1082	-0.2125	0.7345	-1.3061

Column 25

-0.2406

Bobot_Bias_lapisan =

-4.6778

Y1 =

Columns 1 through 12

0.0006	0.0044	-0.0128	-0.0279	-0.0296	-0.0399
-0.0214	-0.0388	-0.0024	-0.0683	-0.0544	-0.0220

Columns 13 through 24

0.0368	0.1638	0.5576	0.7000	0.7141	0.4499
0.5309	0.8523	0.8632	0.7364	1.0112	0.5544

Columns 25 through 36

0.5468	0.5939	0.5123	0.5752	0.5862	0.5913
0.4861	0.5078	0.4564	0.3346	0.3539	0.3507

Columns 37 through 48

0.3429	0.3737	0.2970	0.3426	0.4245	0.3398
0.3292	0.3198	0.3538	0.2872	0.1905	0.0925

Y2 =

Columns 1 through 12

0.0212	-0.0146	-0.0180	-0.0194	-0.0305	-0.0450
-0.0491	-0.0553	-0.0837	-0.0695	-0.0315	-0.0163

Columns 13 through 24

0.0083	0.0537	0.5021	0.5434	0.6248	0.3504
0.1574	0.1208	0.7300	0.6032	0.7585	0.6767

Columns 25 through 36

0.5417	-0.2981	0.4836	0.8121	0.0294	0.2864
0.5295	0.3729	0.4518	0.3966	0.2605	0.2761

Columns 37 through 48

0.2403 0.2457 0.2237 0.2792 0.3494 0.3116
 0.3007 0.3069 0.3267 0.2870 0.1609 0.0479

C =

448.0044

449.0034

Root Meant Square Error = 0.030948

Nilai Yqq dan Ytt

Prediksi kamis	
Yqq	Ytt
157251	98630
72538	106081
64513	71595
61099	41266
34791	37872
381	17050
9158	54332
23845	19215
91286	92504
57653	39985
32330	12059
68463	53094
126711	171280
234110	426631
1296145	1218630
1393982	1504977
1586822	1533296
936989	1002039
479808	1164918
393077	1811193

1835911	1833169
1535547	1578094
1903531	2130640
1709636	1212066
1390003	1196849
598918	1291546
1252485	1127574
2030487	1253921
176632	1276151
785336	1286433
1361176	1074760
990272	1118349
1177128	1015038
1046277	770187
724092	808892
760893	802554
676249	786946
688865	848781
636867	694640
768299	786343
934602	950970
844909	780538
819108	759325
833838	740453
880823	808834
786741	674871
487992	480442
220509	283386

Grafiknya :

