

The Prediction of Bandwidth On Need Computer Network Through Artificial Neural Network Method of Backpropagation

^aIkhthison Mekongga, ^bRahmat Gernowo, ^cAris Sugiharto

^aDepartement of Computer Engineering
Sriwijaya State of Polytechnic,
Palembang, Indonesia 30139

^bJurusan Fisika, Fakultas Sains dan Matematika
Universitas Diponegoro, Semarang

^cProgram Studi Teknik Informatika, Fakultas Sains dan Matematika
Universitas Diponegoro, Semarang

Abstract

The need for bandwidth has been increasing recently. This is because the development of internet infrastructure is also increasing so that we need an economic and efficient provider system. This can be achieved through good planning and a proper system. The prediction of the bandwidth consumption is one of the factors that support the planning for an efficient internet service provider system. Bandwidth consumption is predicted using ANN. ANN is an information processing system which has similar characteristics as the biological neural network. ANN is chosen to predict the consumption of the bandwidth because ANN has good approachability to non-linearity. The variable used in ANN is the historical load data. A bandwidth consumption information system was built using neural networks with a backpropagation algorithm to make the use of bandwidth more efficient in the future both in the rental rate of the bandwidth and in the usage of the bandwidth.

Keywords: Forecasting, Bandwidth, Backpropagation

1. Pendahuluan

Lembaga pendidikan tinggi di negara - negara berkembang harus berhadapan dengan tuntutan terhadap pertumbuhan infrastruktur internet. Berkembangnya jaringan *Local Area Network* (LAN) dan jaringan *internet* yang semakin membesar membutuhkan adanya suatu metode yang dapat meramal kebutuhan *bandwidth* pada jaringan komputer serta dapat mengidentifikasi kebutuhan akan *bandwidth* pada jaringan komputer. *Bandwidth* adalah kapasitas transmisi dalam menyalurkan paket data dari suatu media komunikasi pada jaringan komputer yang menentukan berapa banyak informasi yang dapat ditransmisikan dalam satu satuan waktu, atau dapat juga dikatakan bahwa semakin besar *bandwidth* bisa menghasilkan komunikasi yang lebih cepat.

Tujuan dari peramalan kebutuhan *bandwidth* pada jaringan komputer adalah untuk mengidentifikasi kebutuhan *bandwidth* yang akan terjadi, baik pada jaringan LAN maupun pada jaringan koneksi *internet* sehingga dapat meningkatkan kualitas layanan akademik dan menentukan berapa besar biaya yang akan dikeluarkan untuk sewa *bandwidth*, apabila sewa *bandwidth* lebih besar dari kebutuhan yang sebenarnya mengakibatkan pemborosan *bandwidth*, kemudian apabila sewa *bandwidth* lebih rendah dari kebutuhan sebenarnya, menyebabkan akses internet menjadi lebih lambat, akibatnya merugikan pihak pengguna. Penggunaan *internet*

secara bersama dapat mempengaruhi performansi jaringan seiring dengan peningkatan jumlah pengguna. Performansi jaringan memegang peranan penting dalam pengaturan kebutuhan *bandwidth* untuk tiap layanan aplikasi *internet* yang beraneka-ragam. Ketersediaan *bandwidth* jaringan merupakan faktor penting dalam memilih layanan web (Foster, 2003). Pada dasarnya besarnya kebutuhan *bandwidth* mempresentasikan kapasitas dari koneksi, semakin tinggi kebutuhan *bandwidth*, umumnya akan diikuti oleh kinerja yang lebih baik. Salah satu solusi yang paling efektif untuk mengatasinya adalah dengan mengelola pemakaian *bandwidth* yang menghasilkan suatu kualitas layanan lalu lintas aliran data yang baik dan berkualitas.

Aplikasi, khususnya di bidang industri dari jaringan saraf tiruan sekarang ini telah meluas ke berbagai sendi-sendi industri untuk meningkatkan efisiensi dan produktifitas antara lain pendiagnosa sistem produksi, memonitor kondisi perkakas, optimisasi desain, deteksi kemacetan lini produksi, pengendalian proses, pembentukan group *technology/cellular manufacturing*, sistem inspeksi dan pengendalian kualitas dll. Aplikasi dari segi pengolahan informasi antara pengenalan pola, klasifikasi data dan pola, manajemen pengendalian, dll (Zhang, 1995).

Pemanfaatan jaringan saraf tiruan *backpropagation* telah digunakan untuk peramalan keuangan di pasar saham, peramalan beban listrik dalam jaringan listrik,

• Alamat e-mail : mekongga@polstri.ac.id

prediksi kesalahan dalam pengendalian proses, kontrol panggilan masuk, alokasi kapasitas link dalam jaringan ATM dan prediksi kongesti jaringan. Sebuah multi resolusi belajar jaringan saraf tiruan dibangun untuk memprediksi trafik video VBR untuk mengontrol *bandwidth* dinamis menggunakan jejak dunia nyata video VBR lalu lintas.

Pada penelitian ini dikembangkan sistem jaringan saraf tiruan *backpropagation* untuk peramalan kebutuhan *bandwidth* pada jaringan komputer. Sumber data yang diperlukan adalah data trafik harian yang kemudian data tersebut akan dibelajarkan pada sistem perangkat lunak yang sudah dirancang. *Software* pendukung untuk merancang program digunakan *Borland C++ Builder*.

2. Kerangka Teori

2.1 Jaringan Saraf Tiruan (JST)

Suatu jaringan saraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi, hal ini terinspirasi oleh model kerja otak biologis. Jaringan saraf tiruan (JST) atau *neural network* adalah suatu metode komputasi yang meniru sistem jaringan saraf biologis. Metode ini menggunakan elemen perhitungan non-linier dasar yang disebut *neuron* yang diorganisasikan sebagai jaringan yang saling berhubungan, sehingga mirip dengan jaringan saraf manusia. Jaringan saraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran (Yani, 2005). Layaknya *neuron* biologi, jaringan saraf tiruan juga merupakan sistem yang bersifat "fault tolerant" dalam 2 hal. Pertama, dapat mengenali sinyal *input* yang agak berbeda dari yang pernah diterima sebelumnya. Sebagai contoh, manusia sering dapat mengenali seseorang yang wajahnya pernah dilihat dari foto atau dapat mengenali seseorang yang wajahnya agak berbeda karena sudah lama tidak menjumpainya. Kedua, tetap mampu bekerja meskipun beberapa *neuron*nya tidak mampu bekerja dengan baik. Jika sebuah *neuron* rusak, *neuron* lain dapat dilatih untuk menggantikan fungsi *neuron* yang rusak tersebut.

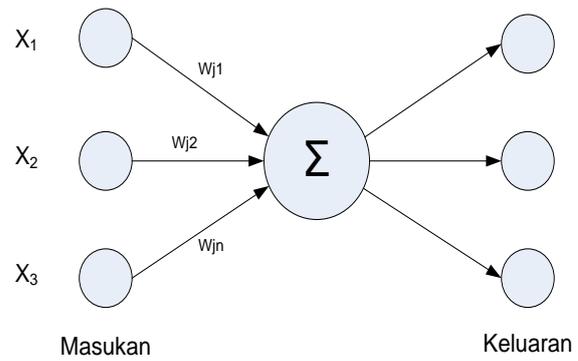
Jaringan saraf tiruan telah dikembangkan sebelum adanya suatu komputer konvensional yang canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun.

Jaringan saraf tiruan ditentukan oleh 3 hal (Siang, 2004):

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

2.2.1 Model Neuron

Satu sel saraf terdiri dari tiga bagian, yaitu: fungsi penjumlah (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*).



Gambar 1 Model Neuron (Hermawan, 2006)

Jika dilihat, *neuron* buatan di atas mirip dengan sel *neuron* biologis. Informasi (*input*) akan dikirim ke *neuron* dengan bobot tertentu. *Input* ini akan diproses oleh suatu fungsi yang akan menjumlahkan nilai-nilai bobot yang ada. Hasil penjumlahan kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap *neuron*. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka *neuron* tersebut akan diaktifkan, jika tidak, maka *neuron* tidak akan diaktifkan. Apabila *neuron* tersebut diaktifkan, maka *neuron* tersebut akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua *neuron* yang berhubungan dengannya. Sehingga dapat disimpulkan bahwa *neuron* terdiri dari 3 elemen pembentuk, yaitu:

1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan yang bernilai negatif akan memperlemah sinyal yang dibawa. Jumlah, struktur, dan pola hubungan antar unit-unit tersebut akan menentukan arsitektur jaringan.
2. Suatu unit penjumlah yang akan menjumlahkan *input-input* sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang akan menentukan apakah sinyal dari *input neuron* akan diteruskan ke *neuron* lain atau tidak.

2.2.2 Konsep Dasar Jaringan Saraf Tiruan

Setiap pola-pola informasi *input* dan *output* yang diberikan kedalam jaringan saraf tiruan diproses dalam *neuron*. *Neuron-neuron* tersebut terkumpul di dalam lapisan-lapisan yang disebut *neuron layers*. Lapisan-lapisan penyusun jaringan saraf tiruan tersebut dapat dibagi menjadi 3, yaitu:

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 *output*nya tidak dapat secara langsung diamati.
3. Lapisan *output*
Unit-unit di dalam lapisan *output* disebut unit-unit *output*. *Output* dari lapisan ini merupakan solusi jaringan saraf tiruan terhadap suatu permasalahan.

2.2.3 Metode Pelatihan/Pembelajaran Jaringan Saraf Tiruan

Cara berlangsungnya pembelajaran atau pelatihan jaringan saraf tiruan dikelompokkan menjadi 3 yaitu (Puspitaningrum, 2006):

a) *Supervised learning* (pembelajaran terawasi)

Pada metode ini, setiap pola yang diberikan kedalam jaringan saraf tiruan telah diketahui *output*-nya. Selisih antara pola *output* aktual (*output* yang dihasilkan) dengan pola *output* yang dikehendaki (*output target*) yang disebut *error* digunakan untuk mengoreksi bobot jaringan saraf tiruan sehingga jaringan saraf tiruan mampu menghasilkan *output* sedekat mungkin dengan pola target yang telah diketahui oleh jaringan saraf tiruan. Contoh algoritma jaringan saraf tiruan 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 jaringan saraf tiruan 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 jaringan saraf tiruan yang menggunakan metode ini yaitu: algoritma RBF. Metode algoritma yang baik dan sesuai dalam melakukan pengenalan pola-pola gambar adalah algoritma *Backpropagation* dan *Perceptron*. Untuk mengenali teks berdasarkan tipe *font* akan digunakan algoritma *Backpropagation*.

2.2.4 Algoritma Umum Jaringan Saraf Tiruan

Algoritma pembelajaran/pelatihan jaringan saraf tiruan (Puspitaningrum, 2006):

Dimasukkan n contoh pelatihan kedalam jaringan saraf tiruan, lakukan :

1. Inisialisasi bobot-bobot jaringan. Set $i = 1$.
2. Masukkan contoh ke-i (dari sekumpulan contoh pembelajaran yang terdapat dalam set pelatihan) kedalam jaringan pada lapisan *input*.
3. Cari tingkat aktivasi unit-unit *input* menggunakan algoritma aplikasi
If \rightarrow kinerja jaringan memenuhi standar yang ditentukan sebelumnya (memenuhi syarat untuk berhenti)
Then \rightarrow exit
4. Update bobot-bobot dengan menggunakan aturan pembelajaran jaringan.

5. If $i = n$ then reset $i = 1$
Else $i = i-1$

Ke langkah 2.

Algoritma aplikasi/inferensi jaringan saraf tiruan (Puspitaningrum, 2006): Dimasukkan sebuah contoh pelatihan kedalam jaringan saraf tiruan, lakukan:

1. Masukkan kasus kedalam jaringan pada lapisan *input*.
2. Hitung tingkat aktivasi node-node jaringan.
3. Untuk jaringan koneksi umpan maju, jika tingkat aktivasi dari semua unit *output*-nya telah dikalkulasi, maka exit.

Untuk jaringan dengan kondisi balik, jika tingkat aktivasi dari semua unit *output*-nya menjadi konstan atau mendekati konstan, maka exit. Jika tidak, kembali ke langkah 2. Jika jaringannya tidak stabil, maka exit dan fail.

2.2.5 Model Jaringan Backpropagation

Model jaringan *Backpropagation* merupakan suatu teknik pembelajaran / pelatihan *supervised learning* yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks.

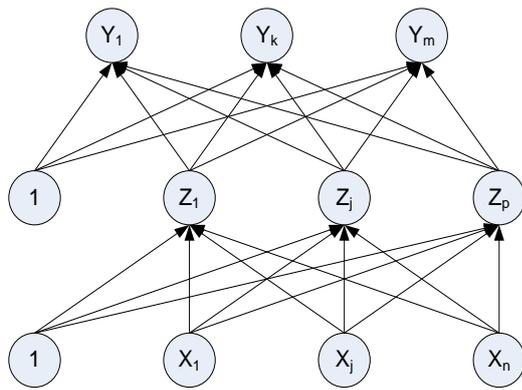
Didalam jaringan *backpropagation*, setiap unit yang berada di lapisan *input* terhubung dengan setiap unit yang ada di lapisan tersembunyi. Setiap unit yang ada di lapisan tersembunyi terhubung dengan setiap unit yang ada di lapisan *output*. Jaringan ini terdiri dari banyak lapisan (*multilayer network*). Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit lapisan tersembunyi untuk selanjutnya diteruskan pada unit-unit di lapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan respon sebagai keluaran jaringan saraf tiruan. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (*backward*) pada lapisan tersembunyi kemudian dari lapisan tersembunyi menuju lapisan masukan (Puspitaningrum, 2006).

Tahap pelatihan ini merupakan langkah untuk melatih suatu jaringan saraf tiruan, yaitu dengan cara melakukan perubahan bobot. Sedangkan penyelesaian masalah akan dilakukan jika proses pelatihan tersebut telah selesai, fase ini disebut fase pengujian (Puspitaningrum, 2006).

2.2.6 Arsitektur Jaringan Backpropagation

Setiap unit di dalam layer *input* pada jaringan *backpropagation* selalu terhubung dengan setiap unit yang berada pada layer tersembunyi, demikian juga setiap unit pada layer tersembunyi selalu terhubung dengan unit pada layer *output*. Jaringan *backpropagation* terdiri dari banyak lapisan (*multilayer network*) yaitu:

1. Lapisan *input* (1 buah), yang terdiri dari 1 hingga n unit *input*.
2. Lapisan tersembunyi (minimal 1 buah), yang terdiri dari 1 hingga p unit tersembunyi.
3. Lapisan *output* (1 buah), yang terdiri dari 1 hingga m unit *output*.



Gambar 2. Arsitektur *Backpropagation* (Siang,2004)

2.2.6 Pelatihan Jaringan *Backpropagation*

Aturan pelatihan jaringan *backpropagation* terdiri dari 2 tahapan, *feedforward* dan *backward propagation*. Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah *vector feature* yang disebut dengan *vektor input* yang diasosiasikan dengan sebuah *output* yang menjadi target pelatihannya. Dengan kata lain set pelatihan terdiri dari vektor *input* dan juga vektor *output* target. Keluaran dari jaringan berupa sebuah vektor *output* aktual. Selanjutnya dilakukan perbandingan antara *output* aktual yang dihasilkan dengan *output* target dengan cara melakukan pengurangan diantara kedua *output* tersebut. Hasil dari pengurangan merupakan *error*. *Error* dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot yang ada dengan mempropagasikannya kembali.

Setiap perubahan bobot yang terjadi dapat mengurangi *error*. Siklus setiap perubahan bobot (*epoch*) dilakukan pada setiap set pelatihan hingga kondisi berhenti dicapai, yaitu bila mencapai jumlah *epoch* yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui. Algoritma pelatihan jaringan *backpropagation* terdiri dari 3 tahapan yaitu:

1. Tahap umpan maju (*feedforward*)
2. Tahap umpan mundur (*backpropagation*)
3. Tahap pengupdatean bobot dan bias.

Secara rinci algoritma pelatihan jaringan *backpropagation* dapat diuraikan sebagai berikut :

Langkah 0:

Inisialisasi bobot-bobot, konstanta laju pelatihan (α), toleransi *error* atau nilai bobot (bila menggunakan nilai bobot sebagai kondisi berhenti) atau set maksimal *epoch* (jika menggunakan banyaknya *epoch* sebagai kondisi berhenti).

Langkah 1: Selama kondisi berhenti belum dicapai, maka lakukan langkah ke-2 hingga langkah ke-9.

Langkah 2: Untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8.

Tahap I : Umpan Maju (*feedforward*)

Langkah 3: Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi diatasnya.

Langkah 4: Masing-masing unit di lapisan tersembunyi (dari unit ke-1 hingga unit ke-p) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya.

Langkah 5: Masing-masing unit *output* ($y_k, k=1,2,3,\dots,m$) dikalikan dengan bobot dan dijumlahkan serta ditambahkan dengan biasnya.

Tahap II : Umpan Mundur (*backward propagation*)

Langkah 6: Masing-masing unit *output* ($y_k, k=1,2,3,\dots,m$) menerima pola target t_k sesuai dengan pola masukan/*input* saat pelatihan dan kemudian informasi kesalahan/*error* lapisan *output* (δ_k) dihitung. δ_k dikirim ke lapisan dibawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias (ΔW_{jk} dan ΔW_{ok}) antara lapisan tersembunyi dengan lapisan *output*.

Langkah 7: Pada setiap unit di lapisan tersembunyi (dari unit ke-1 hingga ke-p; $i=1..n; k=1..m$) dilakukan perhitungan informasi kesalahan lapisan tersembunyi (δ_j). δ_j kemudian digunakan untuk menghitung besar koreksi bobot dan bias (ΔV_{ji} dan ΔV_{jo}) antara lapisan *input* dan lapisan tersembunyi.

Tahap III : Perubahan Bobot dan Bias

Langkah 8: Masing-masing unit *output*/keluaran ($y_k, k=1,2,3,\dots,m$) dilakukan pengupdatean bias dan bobotnya ($j=0,1,2,\dots,p$) sehingga menghasilkan bobot dan bias baru. Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai dengan unit ke-p dilakukan pengupdatean bobot dan bias.

Langkah 9 : Uji kondisi berhenti (akhir iterasi)

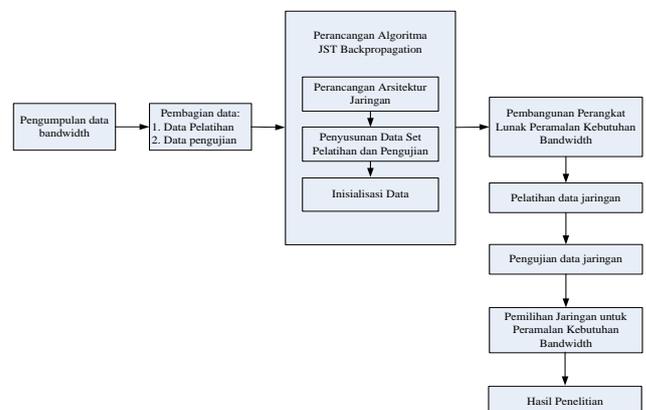
3. Metodologi

3.1 Jalan Penelitian

Proses pembuatan sistem peramalan kebutuhan *bandwidth* pada jaringan komputer Politeknik Negeri Sriwijaya dengan metode jaringan saraf tiruan *backpropagation* diterangkan pada penjelasan sebagai berikut:

3.2 Prosedur Penelitian

Prosedur penelitian peramalan kebutuhan *bandwidth* pada jaringan komputer Politeknik Negeri Sriwijaya ditunjukkan seperti pada gambar 3.



Gambar 3. Prosedur penelitian peramalan kebutuhan *bandwidth* pada jaringan komputer

Berikut ini penjelasan prosedur penelitian peramalan kebutuhan *bandwidth* yang telah digambarkan pada gambar 4.

1. Data penggunaan *bandwidth* diperoleh dari administrator jaringan komputer di Politeknik Negeri Sriwijaya, dikumpulkan menjadi satu kesatuan kemudian dilakukan tahap pembagian data, yaitu data pelatihan dan data pengujian. Data pelatihan digunakan ketika melatih jaringan, sedangkan data pengujian digunakan ketika jaringan di validasi.
2. Perancangan algoritma Jaringan Saraf Tiruan (JST) *Backpropagation*.
3. Model proses yang digunakan dalam pembangunan perangkat lunak ini adalah model sekuensial linier atau disebut juga dengan model air terjun (*waterfall*). Model sekuensial linier meliputi aktivitas sebagai pada gambar berikut



Gambar 4. Model Sekuensial Linier

4. Hasil pelatihan diuji dengan data pelatihan dan data pengujian. Menggunakan data pelatihan, data diuji untuk melihat kemampuan jaringan dalam mengenali pola data yang diberikan. Data pengujian digunakan untuk melihat kemampuan jaringan dalam meramal kebutuhan *bandwidth*.
5. Hasil pelatihan dan pengujian dengan berbagai kasus data inialisasi yang berbeda-beda, dipilih jaringan yang optimum untuk melakukan peramalan. Jaringan optimum yang dipilih merupakan jaringan yang dapat mengenali pola data pelatihan dan nilai akurasi peramalan yang optimum.
6. Hasil dari penelitian yang berupa tulisan dalam bentuk dokumen teknis, jurnal dan tesis.

3.3 Membangun Arsitektur Jaringan Saraf tiruan

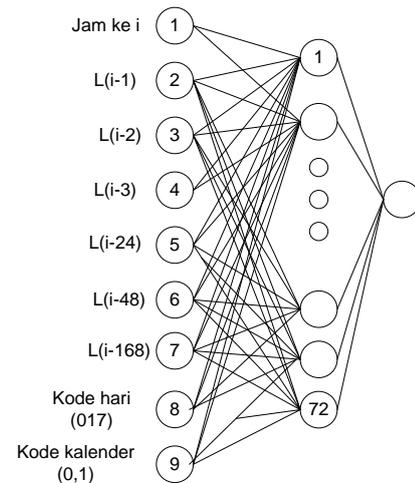
Struktur jaringan saraf tiruan yang digunakan dalam peramalan kebutuhan *bandwidth* pada jaringan komputer Politeknik Negeri Sriwijaya adalah Model Jaringan Saraf Tiruan untuk peramalan kebutuhan *bandwidth*.

Model Jaringan Saraf Tiruan untuk peramalan kebutuhan *bandwidth* menggunakan struktur jaringan saraf tiruan 3 layer/lapisan. Struktur jaringan saraf tiruan 3 layer terdiri dari 1 layer Input, 1 layer Hidden dan 1 layer Output. Pada tesis ini masing-masing layer mempunyai net struktur sebagai berikut: Net struktur: 9-72-1(9 unit layer input, 72 unit layer hidden, 1 unit layer output) :

- Jam ke i: Jam yang akan di ramal besar *bandwidth*nya
- L(i-1) : Data beban 1 jam sebelumnya
- L(i-2) : Data beban 2 jam sebelumnya
- L(i-3) : Data beban 3 jam sebelumnya
- L(i-24) : Data beban pada jam ke i pada 1 hari sebelumnya
- L(i-48) : Data beban pada jam ke i pada 2 hari sebelumnya
- L(i-168) : Data beban pada jam ke i pada 1 minggu sebelumnya
- Hari(0~7): Nama hari yang akan di ramal beban *bandwidth*nya, jika awal minggu dihitung dari hari senin, maka hari senin nilainya 1, selasa nilainya 2, rabu

nilainya 3, kamis nilainya 4, jumat nilainya 5, sabtu nilainya 6 dan minggu nilainya 7.

- Kode kalender: jika biasa 0, jika libur 1



Gambar 5. Struktur JST Untuk Peramalan Kebutuhan *Bandwidth*

3.4 Sistem Perangkat Lunak

Sistem perangkat lunak dalam proses peramalan kebutuhan *bandwidth* sebatas fase pembelajaran jaringan (*Learning*) dan fase penggunaan (*Mapping*). Pada implementasi *software* untuk menjalankan *software* system peramalan kebutuhan *bandwidth* pada Politeknik Negeri Sriwijaya menggunakan program *Borland C++ Builder*

3.4.1. Fase Pembelajaran JST Backpropagation Error

Pada fase ini dilakukan proses adaptasi bobot untuk masing-masing bobot antara layer input dan layer hidden, layer hidden dan layer output. Adaptasi bobot dilakukan secara terus menerus sampai dicapai keadaan *Error* yang paling minimum.

Perhitungan output jaringan dilakukan untuk mendapatkan informasi mengenai jalannya proses pembelajaran, apakah proses pembelajaran sudah memenuhi target atau belum, jika tidak proses pembelajaran dilanjutkan dengan cara mengubah nilai bobot jaringan. Pada proses ini ada dua tahapan yaitu tahap *feed forward* dan tahap *back forward*, seperti yang telah dijelaskan diatas.

Untuk lebih jelasnya dari proses Program pembelajaran jaringan yang digunakan dalam peramalan kebutuhan *bandwidth* dapat dilihat pada *Flowchart*/gambar 6.

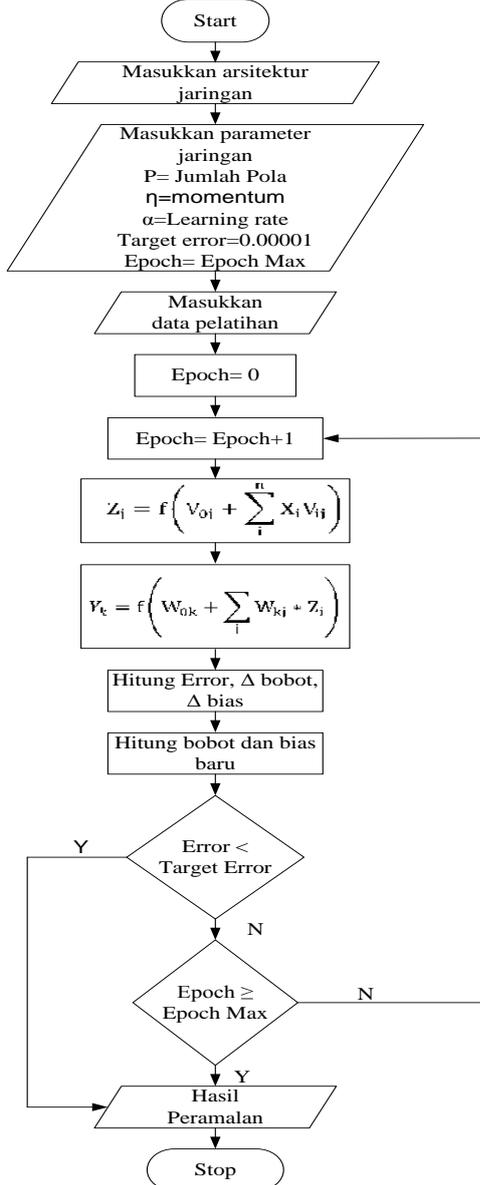
3.4.2 Fase Penggunaan (Mapping) Backpropagation Error

Pada fase penggunaan ini, pola yang akan dihitung dimasukkan pada masukan jaringan (node pada layer masukan). Pola ini akan dikomputasi dengan bobot-bobot interkoneksi hasil fase pembelajaran, kemudian hasil aktifasi sel-sel pada lapisan akhir adalah merupakan keputusan dari jaringan saraf tiruan.

3.4.3 Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan awal pengelolaan informasi pada suatu sistem. Perancangan ini

dibuat sebagai gambaran/bahan dari suatu program/aplikasi yang akan di bangun.



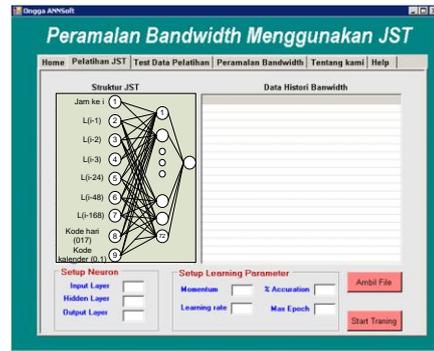
Gambar 6. Fase Pembelajaran Metode Backpropagation Error

4. Hasil dan Pembahasan

4.1 Hasil Penelitian

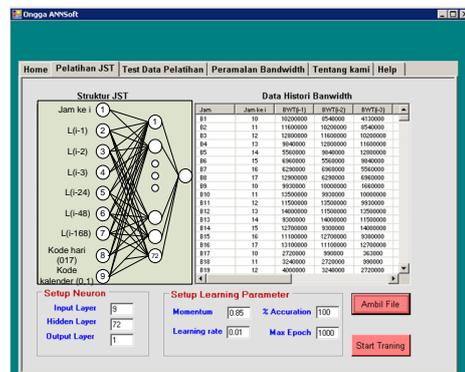
Penggunaan bahasa pemrograman Borland C++ Builder, untuk aplikasi sistem peramalan kebutuhan bandwidth pada Politeknik Negeri Sriwijaya yang menggunakan metode jaringan saraf tiruan backpropagasi ditunjukkan pada gambar-gambar sebagai berikut:

1. Menu Pelatihan JST



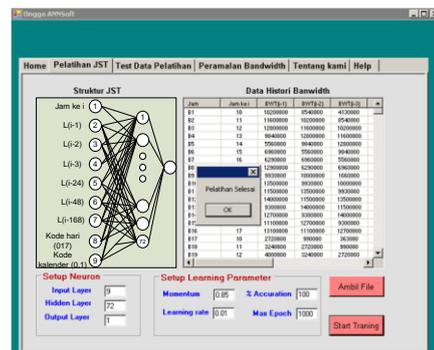
Gambar 7. Menu Pelatihan JST

Form pada gambar 7 digunakan untuk melakukan pelatihan jaringan saraf tiruan. Untuk mengambil data yang akan dilakukan pembelajaran dengan menekan tombol ambil file, setelah file diambil, data tersebut akan tampil pada kolom data historis bandwidth seperti pada gambar 8



Gambar 8. Menu Hasil Pengambilan File

Selanjutnya pilih tombol Start training untuk melakukan pembelajaran, Sistem melakukan pembelajaran sebanyak epoch yang dimasukkan, tunggu sampai ada peringatan pembelajaran telah berhasil seperti pada gambar 9.

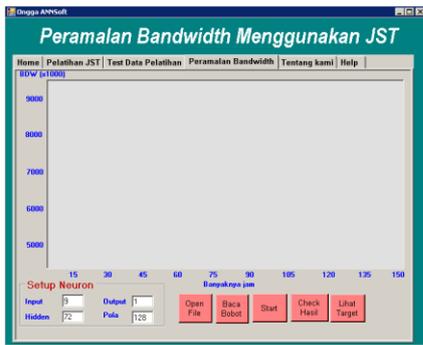


Gambar 9. Menu Pelatihan Selesai

2. Menu Peramalan Bandwidth

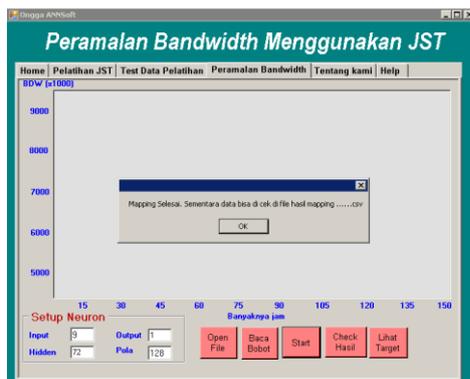
Menu peramalan bandwidth merupakan menu yang digunakan untuk meramal kebutuhan bandwidth. Untuk masuk ke halaman ini pengguna harus memilih menu peramalan bandwidth yang terdapat pada halaman utama kemudian dilanjutkan dengan memasukkan nilai pada bagian Setup Neuron yang terdiri dari empat kolom

inputan nilai yaitu: kolom *input*, kolom *hidden*, kolom *Output* dan kolom pola seperti pada gambar 10.

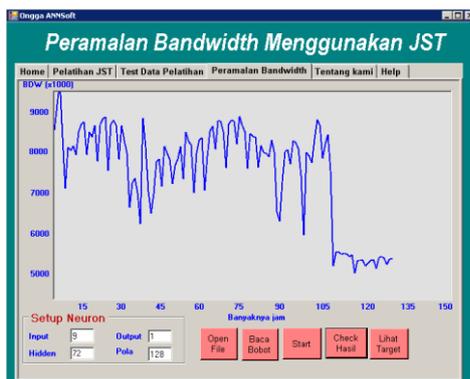


Gambar 10. Menu Peramalan Bandwidth

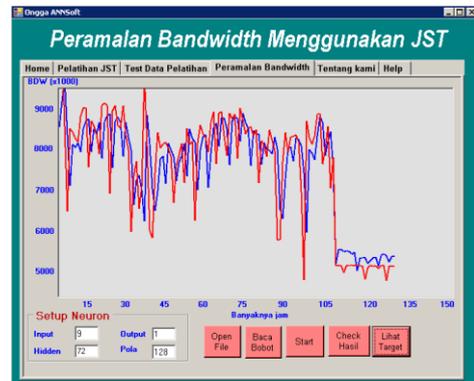
Selanjutnya ambil data yang akan dilakukan peramalan *bandwidth* dengan menekan tombol *open file*, tekan tombol *baca bobot* untuk mengambil *file* bobot untuk memulai peramalan *bandwidth*, *file* bobot didapat dari hasil pelatihan. Tampilan pada gambar 11 berikut ini merupakan tampilan peramalan kebutuhan *bandwidth* selesai dilakukan. Untuk melihat hasil dari peramalan *bandwidth* tekan tombol *check hasil*, hasilnya dapat dilihat seperti gambar 11.



Gambar 11 Menu Peramalan Bandwidth Selesai



Gambar 12 Menu Hasil Peramalan Bandwidth



Gambar 13 Grafik Perbandingan Target dengan Hasil Peramalan

4.2 Pembahasan

4.2.1 Proses Pembelajaran (Learning Process)

Data beban historis digunakan dalam proses pembelajaran untuk mendapatkan bobot yang nantinya digunakan dalam proses *mapping* atau pengujian dalam prakiraan kebutuhan *bandwidth* Politeknik Negeri Sriwijaya. Data beban yang digunakan pada proses pembelajaran adalah data beban bulan November dan Desember 2011 mulai jam 7 pagi sampai jam 17 sore.

Proses pembelajaran dilakukan berdasarkan algoritma pembelajaran, begitu juga dengan struktur data *input* yang digunakan dalam proses pembelajaran. Berikut ini disajikan contoh struktur data *input* dalam 1 hari (Tabel 1)

Tabel 1. Struktur Pola *input* peramalan *bandwidth*

Jam ke i	Input structure						Kode hari	Kode Kalender
	BWT (i-1)	BWT (i-2)	BWT (i-3)	BWT (i-24)	BWT (i-48)	BWT (i-168)		
10	10200000	8540000	4130000	11700000	2360000	8760000		0
11	11600000	10200000	8540000	7710000	846000	7880000		0
12	12800000	11600000	10200000	7990000	556000	6640000		0
13	9840000	12800000	11600000	6660000	35100	5660000		0
14	5560000	9840000	12800000	8670000	44200	8950000	2	0
15	6960000	5560000	9840000	23600000	78100	8320000		0
16	6290000	6960000	5560000	6250000	41300	7970000		0
17	12900000	6290000	6960000	11000000	54200	8610000		0
18	8190000	12900000	6290000	2600000	49700	3570000		0

Keterangan:

- Jam ke i: Jam yang akan di ramal besar *bandwidth*nya
- L(i-1) : Data beban 1 jam sebelumnya
- L(i-2) : Data beban 2 jam sebelumnya
- L(i-3) : Data beban 3 jam sebelumnya
- L(i-24): Data beban pada jam ke i pada 1 hari sebelumnya
- L(i-48): Data beban pada jam ke i pada 2 hari sebelumnya
- L(i-168): Data beban pada jam ke i pada 1 minggu sebelumnya
- Hari(0~7): Nama hari yang akan di ramal beban *bandwidth*nya, jika awal minggu dihitung dari hari senin, maka hari senin nilainya 1, selasa nilainya 2, rabu nilainya 3, kamis nilainya 4, jumat nilainya 5, sabtu nilainya 6 dan minggu nilainya 7.
- Kode kalender: jika biasa 0, jika libur 1

Data yang ditampilkan di atas adalah data pada bulan November 2011.

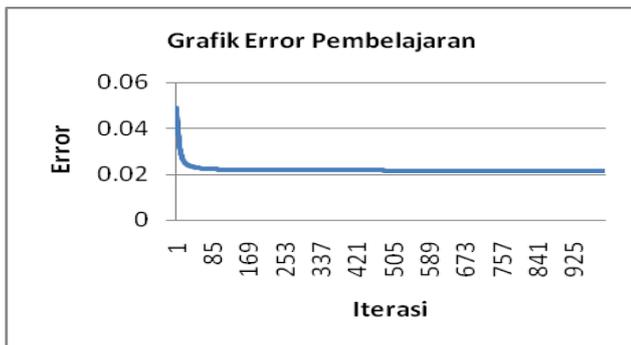
4.2.1.1 Peramalan kebutuhan bandwidth

Berikut ini resume proses pembelajaran JST untuk peramalan kebutuhan bandwidth perjam.

- *Input Layer* : 9 Unit
- *Hidden Layer*: 72 Unit
- *Output Layer*: 1 Unit (*Output* kebutuhan bandwidth per jam)
- Data yang diajarkan: Data bandwidth bulan November 2011 dan bulan Desember 2011.
- Banyaknya data : 128

Selama proses pembelajaran berlangsung, *error* dari tiap pola dapat ditampilkan atau disimpan dalam file. *Error* dari tiap pola yang diajarkan pada jaringan yang disimpan dari interaksi ke 1 sampai dengan interaksi ke 1000 dapat dilihat seperti pada gambar 14. Penyebab lamanya proses pembelajaran pada struktur jaringan yang digunakan adalah pola yang diajarkan sangat banyak, sehingga *looping* dari satu interaksi ke interaksi dipengaruhi oleh banyaknya pola dan struktur data pada pola tersebut, selain itu penyebab target *error* rata-rata jaringan sangat kecil yaitu 0.00001, semakin kecil target *error* jaringan semakin lama proses pembelajaran yang dilakukan oleh jaringan karena untuk memperkecil nilai *error*.

Pada proses pembelajaran, jaringan melakukan proses *backward* atau arus balik dengan cara mengubah nilai bobot jaringan, sebelum target *error* terpenuhi pembelajaran akan terus dilanjutkan sampai mencapai target *error*. Semakin kecil target *error* yang ditetapkan semakin akurat jaringan mengenali dan menghitung bentuk pola baru yang diujikan kepadanya. Pada tesis ini target *error* sengaja dibuat sangat kecil karena diharapkan software ini mampu digunakan untuk meramalkan kebutuhan bandwidth bulan berikutnya.



Gambar 14 Grafik Error Pembelajaran JST

Setelah proses pembelajaran dilaksanakan, proses selanjutnya adalah proses *mapping*, dimana bobot jaringan yang didapatkan dalam proses pembelajaran digunakan sebagai otak untuk menghitung *Output* jaringan yang berupa informasi data kebutuhan bandwidth hasil peramalan oleh JST.

4.2.1.2 Hasil pengujian data kebutuhan bulan Januari

Pengujian data kebutuhan bandwidth bulan Januari 2012 dari tanggal 2 Januari sampai tanggal 19 Januari 2012 dilakukan dengan cara memasukkan struktur pola *input* sesuai dengan aturan yang digunakan dalam proses pembelajaran. *Output* peramalan kebutuhan bandwidth diperoleh dari memasukkan data - data aktual jam

sebelumnya. Hasil pengujian data kebutuhan bandwidth bulan Januari 2012 akan di analisis perjam dan per hari.

4.2.1.3 Analisa Hasil Pengujian Peramalan Kebutuhan Bandwidth

Hasil pengujian data kebutuhan bandwidth perjam dari hari ke hari pada bulan Januari 2012 dengan menggunakan struktur jaringan 3 Layer terdiri dari 9 unit *input*, 72 unit *hidden* dan 1 unit *Output* dengan 1000 interaksi seperti pada Table 2 dan Table 3.

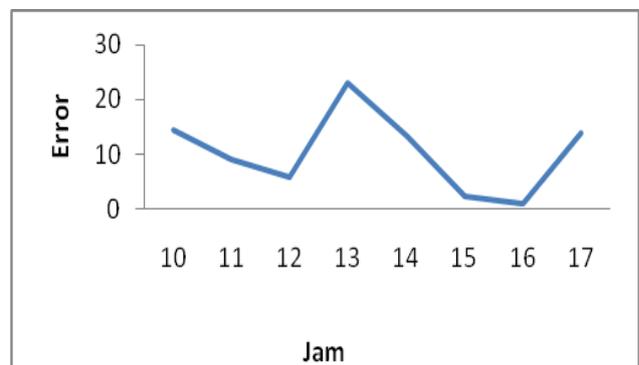
Tabel 2. Hasil Peramalan Bandwidth tanggal 2 Januari 2012

Hari	Jam	Hasil Peramalan	Data Aktual	MAPE
Senin/2/1/2012	10	9303410	11100000	16.1855
	11	10309900	12600000	18.1754
	12	11245800	9410000	19.50903
	13	9061480	6170000	46.86353
	14	6967800	9320000	25.2382
	15	8595260	9180000	6.369717
	16	8503360	8950000	4.990391
	17	8691860	8810000	1.340976
Max		11245800	12600000	46.86353
Min		6967800	6170000	1.340976
Rata-rata		9084858.75	9442500	17.33409

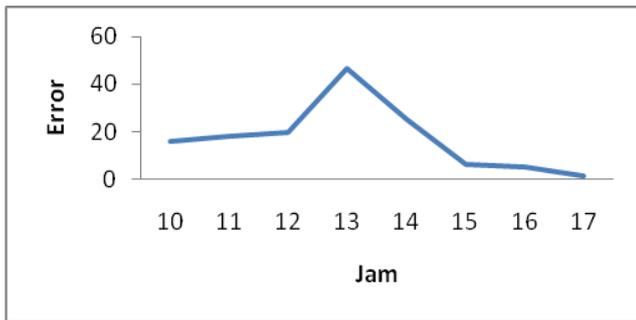
Tabel 3. Hasil Peramalan Bandwidth tanggal 3 Januari 2012

Hari	Jam	Hasil Peramalan	Data Aktual	MAPE
Selasa/3/1/2012	10	8373280	9790000	14.47109
	11	9189320	10100000	9.016634
	12	9523210	10100000	5.710792
	13	9667240	7850000	23.14955
	14	8334890	9620000	13.35873
	15	9212470	9420000	2.203079
	16	9035460	9120000	0.926974
	17	9509050	8350000	13.88084
Max		9523210	10100000	23.14955
Min		8334890	7850000	0.926974
Rata-rata		9105615	9293750	10.33971

Pada Tabel 2 diatas menunjukkan hasil peramalan kebutuhan bandwidth hari senin tanggal 2 Januari 2012 dan Tabel 3 pada hari selasa 3 Januari 2012 dari jam 10 pagi sampai jam 17 sore tiap jamnya. Persentase absolute *Error* perjam harian hasil peramalan kebutuhan bandwidth dapat dilihat pada gambar 15 dan gambar 16.



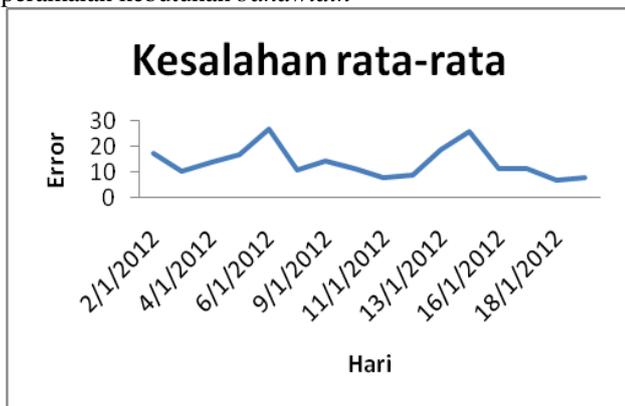
Gambar 15 Grafik Absolute Kesalahan (02/01/2012)



Gambar 16 Grafik Absolute Kesalahan (03/01/2012)

Untuk dapat menganalisa hasil peramalan kebutuhan *bandwidth* perjam dalam harian pada bulan Januari dapat dilihat pada gambar 17 dimana gambar 17 Merupakan gambar grafik perbandingan persentase absolute kesalahan dari peramalan kebutuhan *bandwidth* perjam.

Data pola *Output* hasil peramalan kebutuhan *bandwidth* dengan jaringan saraf tiruan mirip data aktual penggunaan *bandwidth*, melihat hasil akhir dari jaringan ini yang mempunyai kesalahan (MAPE) rata-rata 13.66551 %, Jika MAPE < 25% maka hasil simulasi dapat diterima secara memuaskan, sebaliknya jika MAPE > 25%, hasil simulasi kurang memuaskan (Oktafiri, 2001). Hal ini menunjukkan bahwa sistem informasi peramalan kebutuhan *bandwidth* dengan metode jaringan saraf tiruan dapat digunakan sebagai metode alternatif dalam peramalan kebutuhan *bandwidth*



Gambar 17 Grafik kesalahan rata-rata peramalan *bandwidth* bulan Januari 2012

Data pola *Output* hasil peramalan kebutuhan *bandwidth* dengan jaringan saraf tiruan mirip data aktual penggunaan *bandwidth*, melihat hasil akhir dari jaringan ini yang mempunyai kesalahan (MAPE) rata-rata 13.66551 %, Jika MAPE < 25% maka hasil simulasi dapat diterima secara memuaskan, sebaliknya jika MAPE > 25%, hasil simulasi kurang memuaskan (Oktafiri, 2001). Hal ini menunjukkan bahwa sistem informasi peramalan kebutuhan *bandwidth* dengan metode jaringan saraf tiruan dapat digunakan sebagai metode alternatif dalam peramalan kebutuhan *bandwidth*.

Berdasarkan Hasil Peramalan kebutuhan *bandwidth* dengan metode jaringan saraf tiruan *backpropagation*, kebutuhan *bandwidth* yang diusulkan untuk disewa Politeknik Negeri Sriwijaya pada bulan Januari 2012 adalah 10 Mbps.

5 Kesimpulan

Berdasarkan hasil penelitian dan analisa hasil penelitian, dapat diambil beberapa kesimpulan diantaranya adalah:

1. Dari hasil peramalan *bandwidth* didapatkan perhitungan terbaik menggunakan arsitektur model peramalan unit input 9 dengan *hidden layer* 72 dengan *error* (MAPE) yang didapatkan 13.6655 %.
2. Dari hasil peramalan kebutuhan *bandwidth* dengan metode jaringan saraf tiruan *backpropagation* dapat diusulkan untuk sewa *bandwidth* Politeknik Negeri Sriwijaya pada bulan Januari 2012 adalah sebesar 10 Mbps.

Daftar Pustaka

- Agus, F., Suyatno, A. dan Supianto, 2010. Optimalisasi Manajemen Bandwidth Pada Jaringan Intranet Universitas Mulawarman. Jurnal informatika mulawarman Vol 5 No. 1 Februari 2010 1.
- Anonim, 1999. Modul Training Internet Level Dasar. *Computing Center*. (<http://puskom.petra.ac.id>).
- Bao, 2002. Short-term Load Forecasting based on Neural network and Moving Average. (http://www.cs.rpi.edu/~baojie/pub/2002-05-08_stlf.pdf)
- Buffa, S., Elwood and Rakesh, Sarin, 1996. Modern Production and Operation Management, Eighth Edition, John Wiley and Sons Inc., London.
- Fiengo P., Giambene, G. and Trentin, E., 2007. Neural-based downlink scheduling algorithm for broadband wireless networks. 0140-3664/\$ - see front matter © 2006 Elsevier B.V. All rights reserved. doi: 10.1016/j.comcom.2006.08.009.
- Hermawan, A., 2006. Jaringan Saraf Tiruan Teori dan Aplikasi. Penerbit Andi. Yogyakarta.
- Foster and Kesselman, 2003. The Grid: Blueprint for a New Computing Infrastructure, Second edition, Morgan Kaufman.
- Jong, J., 2004. Jaringan Saraf Tiruan & Pemrogramannya Menggunakan Matlab. Penerbit Andi. Yogyakarta.
- Jumingan, 2009. Studi Kelayakan Bisnis, Teori dan Proposal Kelayakan. Bumi Aksara. Jakarta
- Kaparthi S and Suresh N.C., 1992. Machine-component cell formation in group technology: a neural network approach, International Journal of Production Research, 30(6): 1353-1367.
- Kusumadewi, S., 2003. Artificial Intelligence (Teknik dan Aplikasinya). Penerbit Graha Ilmu. Yogyakarta.
- Lee, K.Y. and Park, J.H., 1992. Short-term Load forecasting using an artificial Neural Network' Transaction on Power System, Vol 7
- Li, D. and Yu, W., 2006. A New Predictive Mechanism Based on Artificial Neural Network. 1-4244-0605-6/06/\$20.00 ©2006 IEEE.
- Makridakis, Spyros, Steven, C., Wheelwright, Victor, E., and Mc.Gee., 1999. Metode dan Aplikasi Peramalan. Jilid I. Edisi Kedua. Jakarta. Binarupa Aksara.
- Murto, P. and Hämmäläinen, R., 1998. Neural network models for short-term load forecasting. [citeseerx.ist.psu.edu /.../download](http://citeseerx.ist.psu.edu/.../download)
- Nandi, B., Vasarhelyi, M.A., and Ahn J.H., 1998, Network demand model and global Internet traffic forecasting. (Elsevier, 12th Biennial conference of the International Telecommunications Society held Stockholm in June 1998, 1998).
- Niu, D., Liu, Z., Li, B., and Zhao, S., 2011. Demand Forecast and Performance Prediction in Peer-Assisted On-Demand Streaming Systems in the Proceedings of IEEE INFOCOM 2011 Mini-Conference, Shanghai, China, April 10-15, 2011.

Oktafri, 2001. Aplikasi Metode Simulasi Monte Carlo Untuk Menduga Debit Aliran Sungai. Repository .ipb.ac.id / handle/123456789/29281.

Puspitaningrum, D., 2006. Pengantar Jaringan Saraf Tiruan. Penerbit Andi. Jogjakarta.

Trimantaraningsih, R. dan Muarifah, I., 2008. Implementasi Mikrotik Sebagai Manajemen *Bandwidth*.

Yani, E., 2005. Pengantar Jaringan Saraf Tiruan. Artikel kuliah.http://trirezqiarantoro.files.wordpress.com/2007/05/jaringan_saraf_tiruan.pdf

Zhang, H.C. and Huang, S.H., 1995, Application of neural network in manufacturing: a state-of-the-art. *International Journal of Production Research*, 33(3): 705-728.