

BAB I

PENDAHULUAN

1.1. Latar Belakang

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*. Dalam menentukan sewa *bandwidth* harus dengan kualitas yang baik. Apabila pemberian *bandwidth* lebih besar dari kebutuhan yang sebenarnya akan mengakibatkan pemborosan *bandwidth*. Sedangkan apabila pemberian *bandwidth* lebih rendah dari kebutuhan sebenarnya, pengaksesan bagi konsumen menjadi lebih lambat yang 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.

Kenyataan bahwa masalah yang dihadapi dalam dunia nyata khususnya di bidang industri adalah masalah-masalah kompleks yang melibatkan banyak variabel dan memerlukan keputusan yang cepat menyediakan peluang yang besar bagi aplikasi jaringan saraf tiruan (*artificial neural network, ANN*). Jaringan saraf tiruan menawarkan metode penghitungan secara paralel dan terdistribusi sehingga memungkinkan penghitungan dengan melibatkan banyak variabel dengan proses hitung yang cepat. Jaringan saraf tiruan dikembangkan berdasarkan prinsip kerja jaraingan saraf pada manusia dimana untuk bisa mahir dalam ketrampilan tertentu harus melalui tahapan pembelajaran atau pelatihan dahulu. Pengembangan metode ini memang juga diarahkan untuk aplikasi penghitungan dengan komputer (Kaparthi, 1991).

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, 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*.

1.2. Perumusan Masalah

Berdasarkan latar belakang diatas, permasalahan yang dapat dirumuskan adalah peramalan kebutuhan *bandwidth* pada jaringan komputer dengan metode jaringan saraf tiruan *backpropagation*.

1.3. Batasan Masalah

Dalam penelitian ini ada beberapa pembatasan masalah yang dilakukan yaitu:

1. Pada penelitian ini di gunakan Jaringan saraf tiruan *backpropagation* dalam peramalan kebutuhan *bandwidth*.
2. Peramalan dibatasi pada keadaan penggunaan *bandwidth* normal, tidak pada saat hari libur dan keadaan penggunaan *bandwidth* yang indensial atau keadaan khusus.
3. Variabel data yang digunakan untuk peramalan kebutuhan *bandwidth* pada jaringan komputer adalah data penggunaan *bandwidth* pada jaringan komputer POLITEKNIK NEGERI SRIWIJAYA.
4. Program yang digunakan untuk perancangan sistem adalah *Borland C++ Builder*.

1.4. Keaslian Penelitian

Penelitian yang berkaitan dengan prediksi beban pada *bandwidth* dalam jaringan komputer telah dilakukan antara lain *Demand Forecast and Performance Prediction in Peer-Assisted On-Demand Streaming Systems*” dalam penelitian ini, peneliti menggunakan teknik analisis *time-series* untuk secara otomatis

memprediksi populasi *online*, *peer upload* dan *bandwidth server* permintaan di setiap saluran video, berdasarkan pembelajaran dari kedua faktor manusia dan dinamika sistem dari pengukuran *online*. Mekanisme yang diusulkan dievaluasi pada data set besar diperoleh dari internet komersial *video-on-demand system* (Niu dkk,2011).

Pada penelitian dengan Jaringan Saraf Tiruan (JST) *Perceptron* mengusulkan sebuah pendekatan berbasis prediksi kinerja jaringan. Mekanisme jaringan saraf tiruan telah diuji pada file klasik trace dan dibandingkan dengan sistem *Network Weather Service* (NWS) untuk kinerja. Perkembangan terakhir teknologi komputer, seperti layanan web, Kotak, *peer-to-peer*, dan *mobile* komputing adalah komputasi berbasis jaringan. Penerapan mereka tergantung pada ketersediaan *bandwidth* jaringan yang mendasarinya. Namun, sumber daya jaringan dibagi dan *bandwidth* jaringan yang tersedia bervariasi dengan waktu. Tidak ada solusi yang memuaskan tersedia untuk prediksi kinerja jaringan. Kurangnya prediksi membatasi penerapan komputasi berbasis jaringan, terutama untuk komputasi grid di mana pengolahan *remote* bersamaan sangat penting. Hasil pengujian menunjukkan pendekatan jaringan saraf tiruan selalu menyediakan prediksi peningkatan dibandingkan NWS (Li, 2006).

Manajemen lalu lintas multimedia dalam jaringan nirkabel, yang mempertimbangkan untuk memberikan perbedaan tingkat penyedia *Quality of Service* (QoS). Secara khusus, fokus penelitian pada mengatur lalu lintas downlink di kedua ATM nirkabel dan skenario WiFi, mengacu ke jaringan infrastruktur akses nirkabel dimana seorang koordinator pusat mengambil keputusan penjadwalan bagi pengguna ponsel. Penjadwalan didasarkan pada Jaringan Saraf Tiruan (JST) dengan penguatan belajar. Jaringan saraf tiruan dilatih dari contoh-contoh untuk berperilaku sebagai “optimal” penjadwalan, menurut model Kritikus Aktor. Hasil yang diperoleh dalam penjadwalan suara, video dan lalu lintas Web memungkinkan untuk menunjukkan peningkatan kapasitas yang signifikan yang dapat dicapai oleh skema peneliti sehubungan dengan teknik lain yang sebelumnya diusulkan dalam literatur (Fiengo, 2006).

Perbedaan penelitian ini dengan penelitian-penelitian sebelumnya yaitu pada proses analisis sistem yang dirancang dan output sistem informasi, penelitian terdahulu menggunakan model perceptron, analisis *timeseries*, dan manajemen lalu lintas multimedia. Pada penelitian ini merancang suatu sistem peramalan kebutuhan *bandwidth* pada jaringan komputer dengan metode jaringan saraf tiruan *backpropagation*.

1.5. Tujuan Penelitian

Penelitian ini bertujuan untuk membuat sistem informasi untuk meramal kebutuhan *bandwidth* pada jaringan komputer.

1.6. Manfaat Penelitian

Dengan adanya sistem ini diharapkan dapat memberi manfaat sebagai berikut:

1. Sistem informasi peramalan kebutuhan *bandwidth* ini dapat dijadikan sebagai acuan untuk penentuan kebutuhan *bandwidth* di Politeknik Negeri Sriwijaya.
2. Untuk mengukur tingkat dukungan *bandwidth* pada Politeknik Negeri Sriwijaya.

BAB II TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

Layanan *video-on-demand system* didistribusi sangat besar-besaran di *internet*. Peramalan permintaan dan prediksi kinerja secara otomatis, jika diterapkan, dapat membantu perencanaan kapasitas dan kontrol kualitas sehingga *Bandwidth server* yang cukup selalu dapat diberikan kepada masing-masing saluran video tanpa menimbulkan sisa. Pada penelitian ini, peneliti menggunakan teknik analisis *time-series* untuk secara otomatis memprediksi populasi *online*, *peer upload* dan *Bandwidth server* permintaan di setiap saluran *video*, berdasarkan pembelajaran dari kedua faktor manusia dan dinamika sistem dari pengukuran *online*. Mekanisme yang diusulkan dievaluasi pada data set besar diperoleh dari internet komersial *video-on-demand system* (Niu dkk, 2011).

Jaringan saraf tiruan (JST) Perceptron diusulkan untuk pendekatan berbasis prediksi kinerja jaringan. Mekanisme jaringan saraf tiruan telah diuji pada file klasik trace dan dibandingkan dengan unjuk kerja sistem *Network Weather Service* (NWS). Perkembangan terakhir teknologi komputer, seperti layanan web, Kotak, *peer-to-peer*, dan *mobile* komputing adalah komputasi berbasis jaringan. Penerapan mereka tergantung pada ketersediaan *Bandwidth* jaringan yang mendasarinya. Namun, sumber daya jaringan dibagi dan *Bandwidth* jaringan yang tersedia bervariasi dengan waktu. Tidak ada solusi yang memuaskan tersedia untuk prediksi kinerja jaringan. Kurangnya prediksi membatasi penerapan komputasi berbasis jaringan, terutama untuk komputasi grid di mana pengolahan remote bersamaan sangat penting. Hasil pengujian menunjukkan pendekatan jaringan saraf tiruan selalu menyediakan prediksi peningkatan dibandingkan NWS (Li dkk, 2006).

Pada manajemen lalu lintas multimedia dalam jaringan nirkabel, mempertimbangkan untuk memberikan perbedaan tingkat penyedia *Quality of Service (QoS)*. Menangani tugas penjadwalan lalu lintas dengan syarat banyak tujuan pada kesalahan dilakukan oleh saluran radio. Secara khusus,

fokus penelitian pada mengatur lalu lintas downlink di kedua *ATM nirkabel* dan skenario *WiFi*, mengacu ke jaringan infrastruktur akses *nirkabel* dimana seorang koordinator pusat mengambil keputusan penjadwalan bagi pengguna ponsel di sel-nya. Penjadwalan didasarkan pada jaringan saraf tiruan (JST) dengan penguatan belajar. Jaringan saraf tiruan dilatih dari contoh-contoh untuk berperilaku sebagai “optimal” Penjadwalan, menurut model Kritikus Aktor. Hasil yang diperoleh dalam penjadwalan suara, video dan lalu lintas Web memungkinkan untuk menunjukkan peningkatan kapasitas yang signifikan yang dapat dicapai oleh skema peneliti sehubungan dengan teknik lain yang sebelumnya diusulkan dalam literature (Fiengo dkk, 2007).

Berkembangnya jaringan LAN dan jaringan internet membutuhkan adanya suatu sistem manajemen jaringan yang mengatur dan mengidentifikasi kebutuhan akan sistem lokal. Metode penelitian yang digunakan adalah pengamatan terhadap *backbone* jaringan lokal. Identifikasi serta analisa mengenai performa dan total trafik. Dari hasil pengamatan didapat bahwa dari dua jaringan yang ada, yaitu jaringan akses internet dan jaringan intranet, jaringan intranet tidak dimanfaatkan secara optimal sebagai jalur akses ke server yang seharusnya bisa diakses secara lokal. Pada penelitian ini menghasilkan beberapa topologi baru yang dapat mengatasi lambatnya koneksi akses ke server lokal serta pembatasan *Bandwidth* yang terpusat. Topologi ini diharapkan bisa memenuhi kebutuhan akan akses intranet yang cepat (Agus dkk, 2010).

2.2. Landasan Teori

2.2.1. Teori Peramalan

Secara umum pengertian peramalan adalah tafsiran. Namun dengan menggunakan teknik-teknik tertentu maka peramalan bukan hanya sekedar tafsiran. Ada beberapa definisi tentang peramalan, diantaranya:

- a. Peramalan atau forecasting diartikan sebagai penggunaan teknik-teknik statistik dalam bentuk gambaran masa depan berdasarkan pengolahan angka-angka historis (Buffa dkk, 1996).
- b. Peramalan merupakan bagian integral dari kegiatan pengambilan keputusan manajemen (Makridakis dkk, 1999)
- c. Peramalan adalah prediksi, rencana, atau estimasi kejadian masa depan yang tidak pasti. Selain itu peramalan juga dapat diartikan sebagai penggunaan teknik-teknik statistik dalam membentuk gambaran masa depan berdasarkan pengolahan angka-angka historis.

Metode peramalan merupakan cara memperkirakan apa yang akan terjadi pada masa depan secara sistematis dan pragmatis atas dasar data yang relevan pada masa yang lalu, sehingga dengan demikian metode peramalan diharapkan dapat memberikan objektivitas yang lebih besar. Selain itu metode peramalan dapat memberikan cara pengerjaan yang teratur dan terarah, dengan demikian dapat dimungkinkannya penggunaan teknik penganalisaan yang lebih maju. Dengan penggunaan teknik-teknik tersebut maka diharapkan dapat memberikan tingkat kepercayaan dan keyakinan yang lebih besar, karena dapat diuji penyimpangan atau deviasi yang terjadi secara ilmiah.

2.2.2. Jenis-jenis Peramalan

Berdasarkan sifatnya, peramalan dibedakan atas dua macam yaitu :

a. Peramalan Kualitatif

Peramalan kualitatif adalah peramalan yang didasarkan atas pendapat suatu pihak, dan datanya tidak bisa direpresentasikan secara tegas menjadi suatu angka atau nilai. Hasil peramalan yang dibuat sangat bergantung pada orang yang menyusunnya. Hal ini penting karena hasil peramalan tersebut ditentukan berdasarkan pemikiran yang instuisi, pendapat dan pengetahuan serta pengalaman penyusunnya.

b. Peramalan Kuantitatif

Peramalan kuantitatif adalah peramalan yang didasarkan atas data kuantitatif masa lalu (data historis) dan dapat dibuat dalam bentuk angka yang biasa disebut sebagai data time series (Jumingan, 2009).

Hasil peramalan yang dibuat sangat bergantung pada metode yang dipergunakan dalam peramalan tersebut. Baik tidaknya metode yang dipergunakan ditentukan oleh perbedaan atau penyimpangan antara hasil ramalan dengan kenyataan yang terjadi. Semakin penyimpangan antara hasil ramalan dengan kenyataan yang akan terjadi maka semakin baik pula metode yang digunakan.

2.2.3. Jangka Waktu Peramalan

Jangka waktu peramalan dapat dikelompokkan menjadi tiga kategori, yaitu (Heizer, 1996):

1. Peramalan jangka pendek, peramalan untuk jangka waktu kurang dari tiga bulan.
2. Peramalan jangka menengah, peramalan untuk jangka waktu antara tiga bulan sampai tiga tahun.
3. Peramalan jangka panjang, peramalan untuk jangka waktu lebih dari tiga tahun.

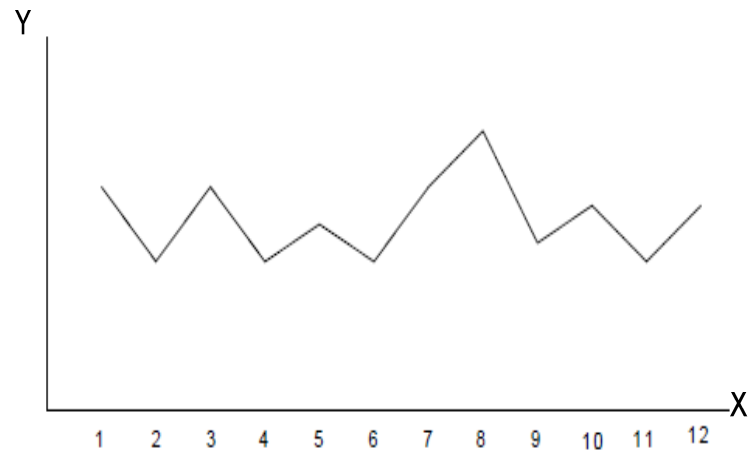
2.2.4. Jenis-jenis Pola Data

Langkah penting dalam memilih suatu metode deret berkala (time series) yang tepat adalah dengan mempertimbangkan jenis pola data, sehingga metode yang paling tepat dengan pola tersebut dapat diuji. Pola data dapat dibedakan menjadi empat jenis, yaitu: (Makridakis, 1999)

1. Pola Horizontal (H) atau Horizontal Data Pattern

Pola data ini terjadi bilamana data berfluktuasi di sekitar nilai rata-rata yang konstan. Suatu produk yang penjualannya tidak meningkat atau

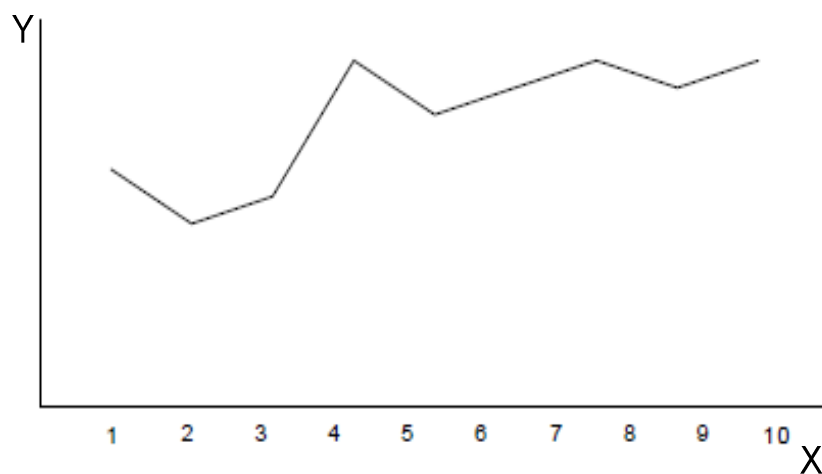
menurun selama waktu tertentu termasuk jenis ini. Bentuk pola horizontal ditunjukkan seperti gambar 2.1.



Gambar 2.1 Pola Data Horizontal

2. Pola Trend (T) atau Trend Data Pattern

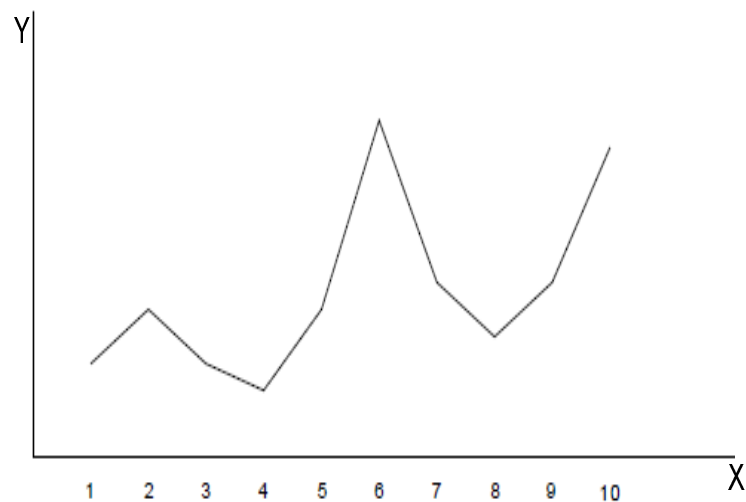
Pola data ini terjadi bilamana terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Contohnya penjualan perusahaan, produk bruto nasional (GNP) dan berbagai indikator bisnis atau ekonomi lainnya, selama perubahan sepanjang waktu. Bentuk pola trend ditunjukkan seperti gambar 2.2.



Gambar 2.2 Pola Data Trend

3. Pola Musiman (S) atau Seasonal Data Pattern

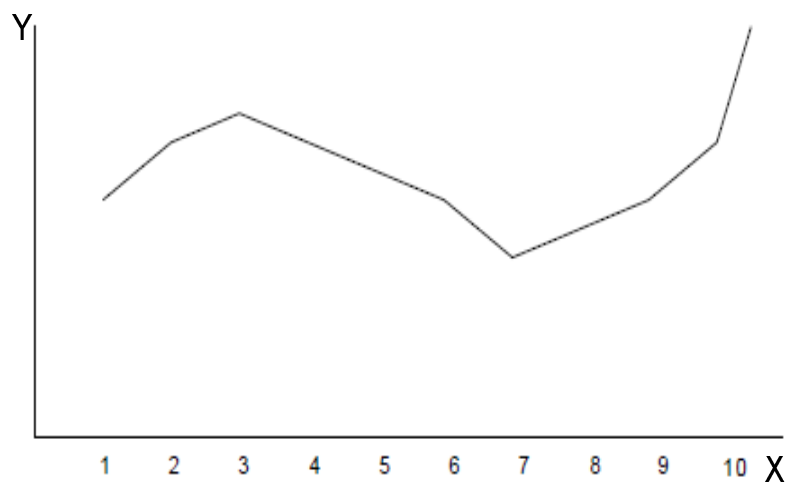
Pola data ini terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulan atau hari-hari pada minggu tertentu). Penjualan dari produk seperti minuman ringan, es krim dan bahan bakar pemanas ruang semuanya menunjukkan jenis pola ini. Bentuk pola musiman ditunjukkan seperti gambar 2.3.



Gambar 2.3 Pola Data Musiman

4. Pola Siklis (S) atau Cycled Data Pattern

Pola data ini terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis. Contohnya penjualan produk seperti mobil, baja. Bentuk pola siklis ditunjukkan seperti gambar 2.4.



Gambar 2.4 Pola Data Siklis

2.2.5. Data Berkala (*Time Series*)

Data berkala adalah data yang dikumpulkan dari waktu ke waktu, untuk menggambarkan perkembangan suatu kegiatan. Analisis data berkala memungkinkan kita untuk mengetahui perkembangan suatu atau beberapa kejadian serta hubungan/pengaruhnya terhadap kejadian lainnya. Pola gerakan data atau nilai-nilai variabel dapat diikuti atau diketahui dengan adanya data berkala, sehingga data berkala dapat dijadikan sebagai dasar untuk:

1. Pembuatan keputusan pada saat ini,
2. Peramalan keadaan perdagangan dan ekonomi pada masa yang akan datang,
3. Perencanaan kegiatan untuk masa depan.

Gerakan-gerakan khas dari data *time series* dapat digolongkan ke dalam empat kelompok utama, yang sering disebut komponen-komponen *time series*:

- 1) Gerakan jangka panjang atau sekuler merujuk kepada arah umum dari grafik *time series* yang meliputi jangka waktu yang panjang.
- 2) Gerakan siklis (*cyclical movements*) atau variasi siklis merujuk kepada gerakan naik-turun dalam jangka panjang dari suatu garis atau kurva trend. Siklis yang demikian dapat terjadi secara periodik ataupun tidak, yaitu dapat ataupun tidak dapat mengikuti pola yang tepat sama setelah interval-interval

waktu yang sama. Dalam kegiatan bisnis dan ekonomi, gerakan-gerakan hanya dianggap siklis apabila timbul kembali setelah interval waktu lebih dari satu tahun.

- 3) Gerakan musiman (*seasonal movements*) atau variasi musim merujuk kepada pola-pola yang identik, atau hampir identik, yang cenderung diikuti suatu *time series* selama bulan-bulan yang bersangkutan dari tahun ke tahun. Gerakan-gerakan demikian disebabkan oleh peristiwa-peristiwa yang berulang-ulang terjadi setiap tahun.
- 4) Gerakan tidak teratur atau acak (*irregular or random movements*) merujuk kepada gerakan-gerakan sporadis dari *time series* yang disebabkan karena peristiwa-peristiwa kebetulan seperti banjir, pemogokan, pemilihan umum, dan sebagainya. Meskipun umumnya dianggap bahwa peristiwa-peristiwa demikian menyebabkan variasi-variasi yang hanya berlangsung untuk jangka pendek, namun dapat saja terjadi bahwa peristiwa-peristiwa ini demikian hebatnya sehingga menyebabkan gerakan-gerakan siklis atau hal lain yang baru.

2.2.6. Pengukuran Peramalan

Teknik peramalan tidak selamanya selalu tepat karena teknik peramalan yang digunakan belum tentu sesuai dengan sifat datanya, atau disebabkan oleh kondisi di luar bisnis yang mengharuskan bisnis itu menyesuaikan diri. Oleh karena itu, perlu diadakan pengawasan peramalan sehingga dapat diketahui sesuai atau tidaknya teknik peramalan yang digunakan. Sehingga dapat dipilih dan ditentukan teknik peramalan yang lebih sesuai dengan cara menentukan batas toleransi peramalan atas penyimpangan yang terjadi (Jumingan, 2009).

Pada prinsipnya, pengawasan peramalan dilakukan dengan membandingkan hasil peramalan dengan kenyataan yang terjadi. Penggunaan teknik peramalan yang menghasilkan penyimpangan terkecil adalah teknik peramalan yang paling sesuai untuk digunakan (Jumingan, 2009).

Besarnya *error* peramalan dihitung dengan mengurangi data riil dengan besarnya ramalan.

$$Error (E) = X_t - F_t$$

Keterangan:

X_t = Data riil periode ke-t

F_t = Ramalan periode ke-t

Dalam menghitung *error* peramalan digunakan MAPE (*Means Absolute Percentage Error*)

Mean Absolute Percentage *Error* (MAPE) merupakan nilai tengah kesalahan persentase absolute dari suatu peramalan.

$$MAPE = \frac{\sum_{t=1}^n \frac{|X_t - F_t|}{X_t}}{n} \quad (1)$$

Keterangan:

X_t = Nilai data periode ke-t

F_t = Ramalan periode ke-t

n = Banyaknya data

2.3 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 *neuronnya* tidak mampu bekerja dengan baik. Jika sebuah *neuron* rusak, *neuron* lain dapat dilatih untuk menggantikan fungsi *neuron* yang rusak tersebut.

Jaringan saraf 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, jaringan saraf tiruan merupakan sistem yang tak terprogram, artinya semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran/pelatihan. Hal yang ingin dicapai dengan melatih jaringan saraf tiruan adalah untuk mencapai keseimbangan antara kemampuan memorisasi dan generalisasi. Yang dimaksud kemampuan memorisasi adalah kemampuan jaringan saraf tiruan untuk mengambil kembali secara sempurna sebuah pola yang telah dipelajari. Kemampuan generalisasi adalah kemampuan jaringan saraf tiruan untuk menghasilkan respons yang bisa diterima terhadap pola-pola *input* yang serupa (namun tidak identik) dengan pola-pola yang sebelumnya telah dipelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam jaringan saraf tiruan itu diinputkan informasi baru yang belum pernah dipelajari, maka jaringan saraf tiruan itu masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati (Puspitaningrum, 2006).

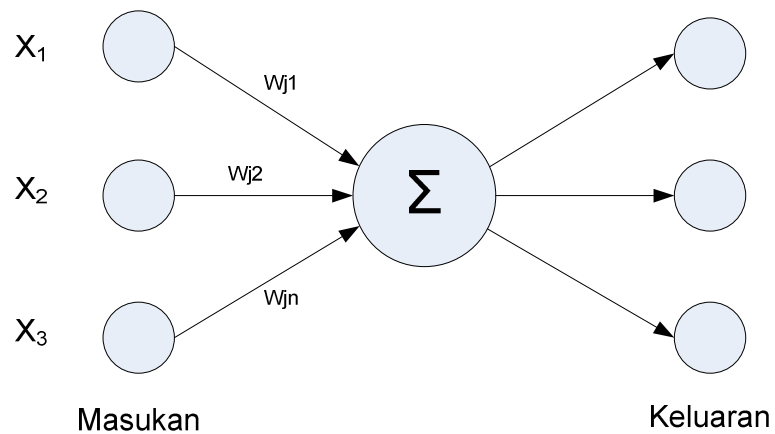
Jaringan saraf tiruan telah dikembangkan sebelum adanya suatu komputer konvensional yang canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun.

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*.

2.3.1. Model *Neuron*

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



Gambar 2.5 Model *Neuron* (Hermawan, 2006)

Jika kita lihat, *neuron* buatan diatas 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 *outputnya* 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.

2.3.2. 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 *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 jaringan saraf tiruan terhadap suatu permasalahan.

2.3.3. Arsitektur Jaringan Jaringan Saraf Tiruan

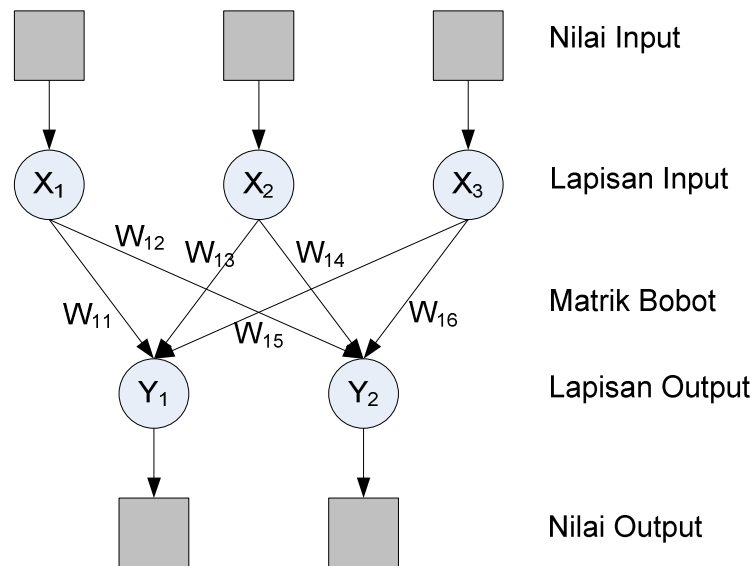
Jaringan saraf tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur jaringan saraf tiruan tersebut, antara lain (Kusumadewi, 2003):

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.

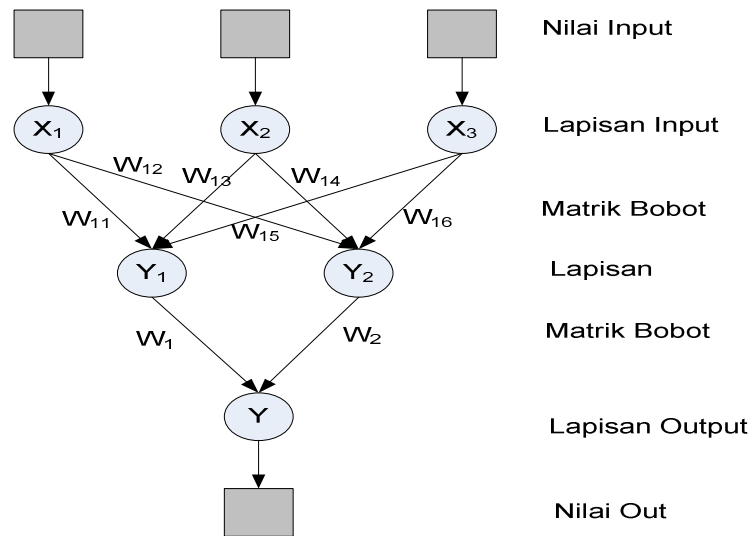
Contoh algoritma jaringan saraf tiruan yang menggunakan metode ini yaitu: *ADALINE*, *Hopfield*, *Perceptron*.



Gambar 2.6 Arsitektur layer tunggal (Hermawan, 2006)

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

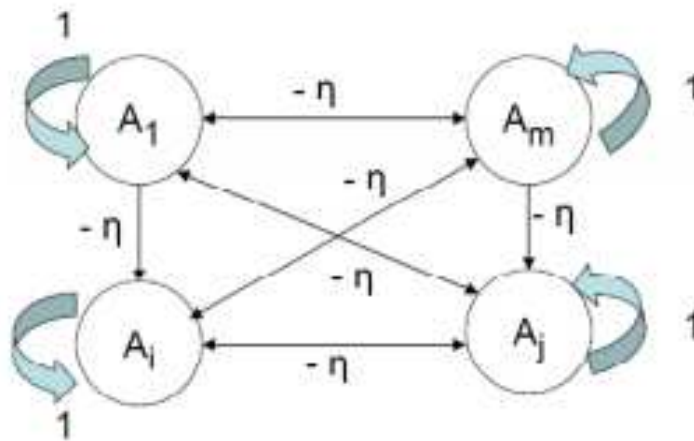
Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis layer yakni layer *input*, layer *output*, dan juga layer tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan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



Gambar 2.7 Arsitektur layer jamak (Hermawan, 2006)

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 2.8 Arsitektur layer kompetitif (Hermawan, 2006)

2.3.4. 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*.

2.3.5. Fungsi Aktivasi Jaringan Saraf Tiruan

Dalam jaringan saraf tiruan, fungsi aktivasi digunakan untuk menentukan keluaran suatu *neuron*. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya)

Beberapa fungsi aktivasi yang digunakan adalah:

a) Fungsi threshold (batas ambang)

Fungsi threshold (3) merupakan fungsi threshold biner. Untuk kasus bilangan bipolar, maka angka 0 diganti dengan angka -1. Adakalanya dalam jaringan saraf tiruan 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

Fungsi ini sering digunakan karena nilai fungsinya yang sangat mudah untuk didiferensiasikan.

c) Fungsi identitas

Digunakan jika keluaran yang dihasilkan oleh jaringan saraf tiruan merupakan sembarang bilangan riil (bukan hanya pada range $[0,1]$ atau $[1,-1]$).

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

2.3.7. 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.

Di dalam 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).

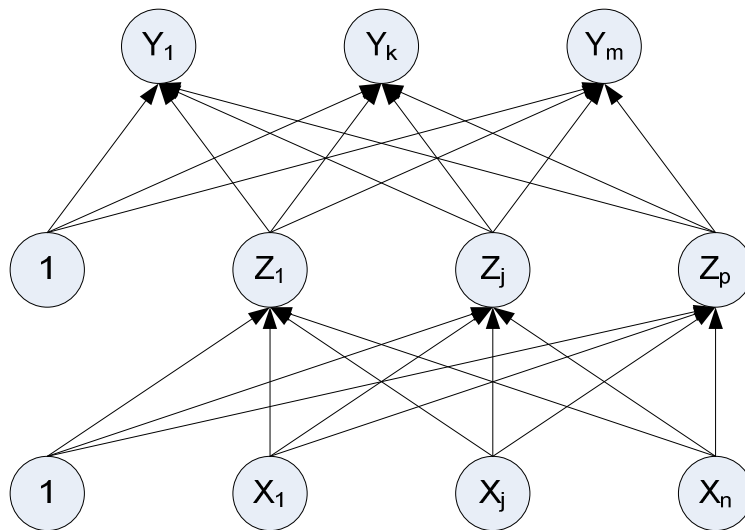
Tahap pelatihan ini merupakan langkah untuk melatih suatu jaringan saraf tiruan, yaitu dengan cara melakukan perubahan bobot. Sedangkan penyelesaian

masalah akan dilakukan jika proses pelatihan tersebut telah selesai, fase ini disebut fase pengujian (Puspitaningrum, 2006).

2.3.8. 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 2.9 Arsitektur *Backpropagation* (Siang,2004)

2.3.9. 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 di atasnya.

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 ; $i=1\dots n; k=1\dots m$) 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)

2.4. Jaringan Komputer

Jaringan komputer adalah dua atau lebih komputer yang saling terhubung, bisa berbagi pakai file (data, *software*) dan peralatan (*modem, scanner, CDROM, dll*) jaringan pada beberapa lokasi (*email, link, video conferences*). Komponen jaringan meliputi (Trimantaraningsih, 2008):

1. Minimal ada 2 komputer
2. Antar muka jaringan pada setiap komputer (*NIC* atau *adapter*)

3. Media koneksi (kabel, gelombang radio, dll)
4. OS jaringan
5. Antar muka jaringan yang disebut NIC (*Network Interface Card*), berupa *Adapter Card*, *PC Card* atau *Compact Flash Card* yang menyebabkan komputer atau peralatan bisa terhubung ke jaringan.

2.4.1. Local Area Network (LAN)

LAN merupakan jaringan yang meliputi area geografis yang relatif kecil. LAN dicirikan dengan kecepatan data yang relatif tinggi dan *error* yang relatif rendah. LAN menghubungkan workstation, perangkat jaringan, terminal, dan perangkat lain dalam area yang terbatas. LAN dimiliki oleh pengguna dan tidak dioperasikan lewat sambungan sewa, meskipun LAN mungkin saja memiliki pintu gerbang ke PSTN atau jaringan swasta lainnya (Trimantaraningsih, 2008).

2.4.2. Internet

Internet merupakan singkatan dari *Interconnected Networking*, yang berarti suatu jaringan komputer yang terhubung dengan luas. *Internet* berasal dari sebuah jaringan komputer yang dibuat pada tahun 1970-an yang terus berkembang sampai sekarang menjadi jaringan dunia yang sangat luas. Jaringan tersebut diberi nama ARPANET, yaitu jaringan yang dibentuk oleh Departemen Pertahanan Amerika Serikat. Kemudian, jaringan komputer tersebut diperbaharui dan dikembangkan sampai sekarang dan menjadi tulang punggung global untuk sumber daya informasi yang disebut internet (Anonim, 1999)

- Jenis Koneksi *Internet* :
 - User pribadi: mempergunakan Koneksi *dial-up modem* (menggunakan *line* telepon).
 - User institusi/*corporate*:
 - Koneksi *dial- up Analog/Digital* (ISDN).
 - Koneksi *leased- line* (permanen).
 - Koneksi VSAT (*Very Small Arpperture Terminal*).

- Alamat di *Internet*:
 - *IP address*: terdiri atas 4 angka 8 bit
Contoh: untuk *IP address server MATT* adalah 202.43.253.9
 - *Domain name*: *host.domain*
Contoh : matt.petra.ac.id
 - *User address*:
Contoh : dinsetia@matt.petra.ac.id
[dinsetia merupakan login], [matt merupakan host], [petra.ac.id merupakan *domain*, dimana ac menandakan *academic* dan id menandakan negara Indonesia].

ISP (*Internet Service Provider*) adalah badan usaha yang menyediakan fasilitas koneksi ke internet. *Server* atau *Host* adalah suatu mesin komputer yang tugasnya melayani segala aktifitas dan aplikasi *internet*. Login atau user id merupakan tanda/*e-mail address* dari orang tersebut bahwa dia terdaftar di server tersebut.

Password merupakan bagian dari pengamanan pada sistem di *internet*.

- Fasilitas *Internet*:
 - Akses komputer jarak jauh (remote login).
 - Komunikasi dengan pemakai lain :
 - Off- line: surat elektronik (e- mail), mailing list, newsgroup
 - On-line: talk, IRC (Internet Relay Chat), Internet Phone, Netmeeting.
 - Telnet
 - Akses Informasi
 - WWW (World Wide Web), search (surfing), download.
 - Pemindahan Berkas/File Transfer Protocol (FTP).
 - Mencari lokasi suatu file (Archie).

2.4.3. *Bandwidth*

Bandwidth adalah suatu ukuran rentang frekuensi maksimum yang dapat mengalir data dari suatu tempat ke tempat lain dalam suatu waktu tertentu

(Hekmat, 2005). Satuan yang dipakai untuk *Bandwidth* adalah *bit per* detik (bps) atau Byte perdetik (Bps) dimana 1Byte = 8 bit. Bit atau binary digit adalah basis angka yang terdiri dari angka 0 dan 1. Satuan ini menggambarkan seberapa banyak bit (angka 0 dan 1) yang dapat mengalir dari satu tempat ke tempat yang lain dalam setiap detiknya melalui suatu media. Sedangkan dalam sinyal analog, *bandwidth* diartikan sebagai rentang antara frekuensi tinggi dan frekuensi terendah di ukur dalam satuan Hertz (HZ). *Bandwidth* adalah konsep pengukuran yang sangat penting dalam jaringan, tetapi konsep ini memiliki kekurangan atau batasan, tidak peduli bagaimana cara mengirimkan informasi maupun media apa yang dipakai dalam penghantaran informasi. Ini akan menyebabkan batasan terhadap panjang media yang dipakai, kecepatan maksimal yang dapat dipakai, mau pun perlakuan khusus terhadap media yang dipakai, Karena faktor distorsi, *Bandwidth* dan rate data biasanya berbanding terbalik dengan jarak komunikasi.

Sedangkan batasan terhadap perlakuan atau cara pengiriman data misalnya adalah dengan pengiriman secara paralel (*synchronous*), serial (*asynchronous*), perlakuan terhadap media yang spesifik seperti media yang tidak boleh ditekuk (serat optis), pengirim dan penerima harus berhadapan langsung (line of sight), kompresi data yang dikirim, dll

Manajemen *bandwidth* adalah sebuah proses penentuan besarnya bandwidth kepada tiap pemakai dalam jaringan komputer. Besarnya *bandwidth* akan berdampak kepada kecepatan transmisi, *Bandwidth internet* disediakan oleh *provider internet* dengan jumlah tertentu tergantung sewa pelanggan. Dengan QoS dapat diatur agar user tidak menghabiskan *Bandwidth* yang di sediakan oleh *provider*. *Bandwidth* mempresentasikan jarak keseluruhan atau jangkauan di antara sinyal tertinggi dan terendah pada kanal (*band*) komunikasi. Pada dasarnya bandwidth mempresentasikan kapasitas dari koneksi, semakin tinggi kapasitas, maka umumnya akan diikuti oleh kinerja yang lebih baik, meskipun kinerja keseluruhan juga tergantung pada faktor-faktor lain, misalnya *latency* yaitu waktu tunda antara masa sebuah perangkat meminta akses ke jaringan dan masa perangkat itu memberi izin untuk melakukan transmisi.

2.5. Model JST dalam peramalan *Bandwidth*

Metode Jaringan Saraf Tiruan yang umum digunakan dalam peramalan beban adalah metode *Backpropagation*, namun model struktur inputnya berbeda satu sama lainnya. Berikut ini akan disajikan beberapa model struktur input JST *Backpropagation* yang menjadi referensi dalam tesis ini seperti pada tabel 2.1

Tabel 2.1 Tabel Struktur JST

Referensi	Struktur JST
Bao, 2002	<p><u>Jumlah Neuron Lapisan Input: 9 Unit</u> Terdiri dari</p> <p>X_1= Jam ke-i pada hari ke t</p> <p>X_2= Cuaca pada hari ke t</p> <p>X_3= $L(i,t-1)$ Beban jam ke-i hari sebelumnya(t-1)</p> <p>X_4= $L(i-1,t)$ Beban jam ke i-1</p> <p>X_5=$L(1,t)$ Beban jam ke 1</p> <p>X_6= Kode akhir pekan (0,1). 1 jika akhir pekan dan 0 jika bukan</p> <p>$X_{7,8,9}$ = Kode hari (001~111). 001 hari senin dan 111 hari minggu</p> <p><u>Jumlah Neuron Lapisan Hidden: 8 Unit</u></p> <p><u>Jumlah Neuron Output Layer: 1 Unit</u> (Data beban jam ke i+1)</p>
Murto, 1998	<p><u>Jumlah Neuron Lapisan Input: 14 Unit</u> Terdiri dari</p> <p>X_k= $L(i-k)$, dimana $k=1, 2, 3$. Data beban jam ke i-1 sampai jam ke i-3</p> <p>X_4= $L(i-24)$. Data beban jam ke i-24 (sehari sebelum)</p> <p>X_5= $L(i-168)$. Data beban jam ke i-168 (seminggu sebelum)</p> <p>$X_6 \sim X_9$= Dk. Dimana $k= 1, 2, 3, 4$ Dk adalah kode hari (senin, selasa, rabu,..., minggu)</p> <p>$X_{10} \sim X_{14}$= Kode jam (00000 ~ 10111) yaitu dari jam ke 0 sampai jam ke 23.</p> <p><u>Jumlah Neuron Lapisan Hidden: 10, 15, 20 Unit</u> (tiap banyak neuron di ujikan satu persatu)</p> <p><u>Jumlah Neuron Output Layer: 1 Unit</u> (Data beban jam ke i)</p>
Lee, 1992	<p><u>Jumlah Neuron Lapisan Input: 51 Unit</u> Terdiri dari</p> <p>$X_1 \sim X_{24}$= Data beban minggu sebelumnya selama 24 jam</p>

	<p>$X_{25} \sim X_{48}$ = Data beban 2 minggu sebelumnya selama 24 jam</p> <p>$X_{49} \sim X_{51}$ = Kode hari dalam bentuk biner (001~111)</p> <p><u>Jumlah Neuron Lapisan Hidden: 70, 90, 120 Unit</u> (tiap banyak neuron di ujikan satu persatu)</p> <p><u>Jumlah Neuron Output Layer: 24 Unit</u> (Data beban dari jam ke 1~24)</p>
--	--

Referensi	Struktur JST
Bao, 2002	<p><u>Jumlah Neuron Lapisan Input: 9 Unit</u> Terdiri dari</p> <p>X_1 = Jam ke-i pada hari ke t</p> <p>X_2 = Cuaca pada hari ke t</p> <p>X_3 = $L(i,t-1)$ Beban jam ke-i hari sebelumnya(t-1)</p> <p>X_4 = $L(i-1,t)$ Beban jam ke i-1</p> <p>X_5 = $L(1,t)$ Beban jam ke 1</p> <p>X_6 = Kode akhir pekan (0,1). 1 jika akhir pekan dan 0 jika bukan</p> <p>$X_{7,8,9}$ = Kode hari (001~111). 001 hari senin dan 111 hari minggu</p> <p><u>Jumlah Neuron Lapisan Hidden: 8 Unit</u></p> <p><u>Jumlah Neuron Output Layer: 1 Unit</u> (Data beban jam ke i+1)</p>
Murto, 1998	<p><u>Jumlah Neuron Lapisan Input: 14 Unit</u> Terdiri dari</p> <p>X_k = $L(i-k)$, dimana $k=1, 2, 3$. Data beban jam ke i-1 sampai jam ke i-3</p> <p>X_4 = $L(i-24)$. Data beban jam ke i-24 (sehari sebelum)</p> <p>X_5 = $L(i-168)$. Data beban jam ke i-168 (seminggu sebelum)</p> <p>$X_6 \sim X_9$ = Dk. Dimana $k= 1, 2, 3, 4$ Dk adalah kode hari (senin, selasa, rabu,..., minggu)</p> <p>$X_{10} \sim X_{14}$ = Kode jam (00000 ~ 10111) yaitu dari jam ke 0 sampai jam ke 23.</p> <p><u>Jumlah Neuron Lapisan Hidden: 10, 15, 20 Unit</u> (tiap banyak neuron di ujikan satu persatu)</p> <p><u>Jumlah Neuron Output Layer: 1 Unit</u> (Data beban jam ke i)</p>

BAB III

METODE PENELITIAN

3.1. Bahan Penelitian

Bahan yang dibutuhkan untuk melakukan penelitian dalam pembuatan aplikasi peramalan kebutuhan *bandwidth* pada jaringan komputer Politeknik Negeri Sriwijaya dengan metode jaringan saraf tiruan *backpropagation* adalah data histori penggunaan *bandwidth* pada jaringan komputer Politeknik Negeri Sriwijaya.

Mengumpulkan dan mengidentifikasi data histori pengguna *bandwidth*, Pengambilan dan pengolahan data pengguna *bandwidth* dilakukan dengan mengamati trafik penggunaan *bandwidth* yang ada di Politeknik Negeri Sriwijaya. Data yang di ambil berupa data histori trafik yang disediakan. Data yang diambil adalah data histori dari data harian sehingga dari data yang diperoleh tersebut akan digunakan sebagai data *input* untuk penelitian dengan menggunakan jaringan saraf tiruan *backpropagation*. Data yang diperoleh dikumpulkan dan dipisahkan menjadi 2 bagian, bagian pertama dipergunakan untuk melatih jaringan saraf tiruandan bagian ke 2 dipergunakan untuk menguji unjuk kerja jaringan saraf tiruanyaitu untuk menguji apakah jaringan menghasilkan *output* sesuai dengan yang diinginkan pada waktu *input* yang belum pernah di pelajari oleh jaringan dimasukkan. Semakin banyak data yang dapat diperoleh. Semakin baik jaringan dapat menyelesaikan masalahnya.

3.2. Alat Penelitian

Alat penelitian yang digunakan dalam proses penelitian ini antara lain sebagai berikut:

1. Perangkat keras (*hardware*) adalah peralatan dalam komputer yang secara fisik dapat dilihat. Dalam sistem komputer, perangkat keras dapat dibagi dalam empat bagian, yaitu unit masukan, unit keluaran, unit pengolah dan unit penyimpanan. Pada pembangunan sistem ini, spesifikasi minimum

perangkat keras yang digunakan adalah: *RAM 1GB, Prosesor intel Core 2 Duo, HDD 320 GB, Keyboard, Monitor, Mouse.*

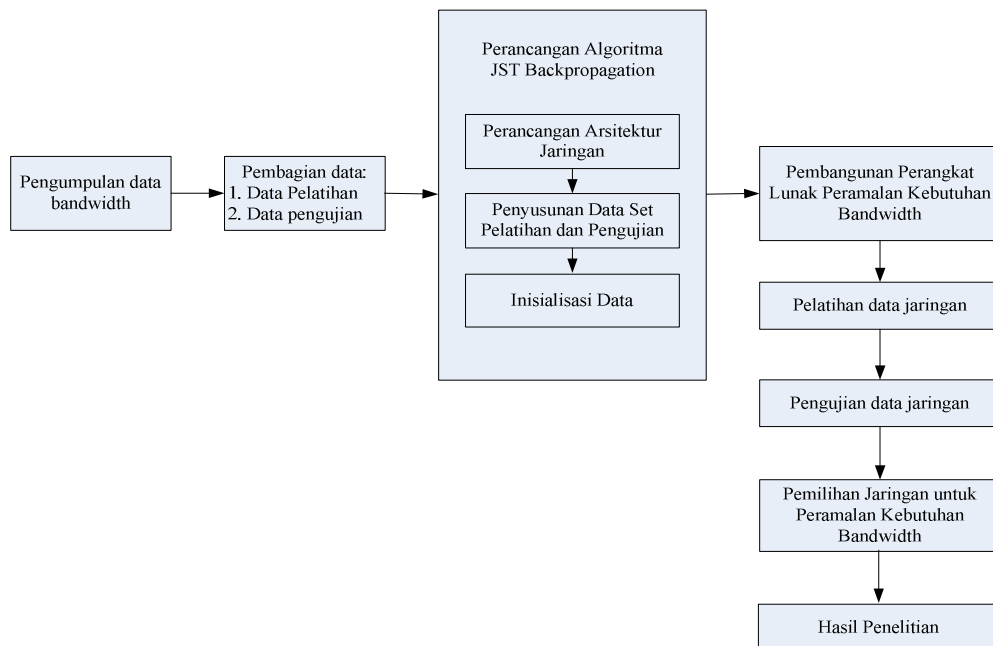
2. Perangkat lunak dalam sistem komputer merupakan serangkaian perintah dengan aturan tertentu yang mengatur operasi perangkat keras. Perangkat lunak terdiri atas tiga bagian, yaitu sistem operasi, bahasa pemrograman dan program aplikasi yang merupakan faktor penunjang dari sistem komputer. Perangkat lunak yang digunakan untuk membangun sistem ini adalah: *Borland C++ Builder dan microsoft excel*

3.3. 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:

3.3.1. Prosedur Penelitian

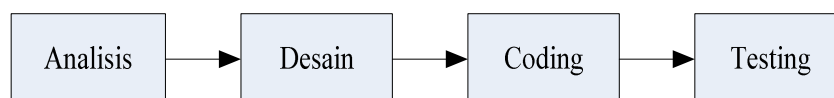
Prosedur penelitian peramalan kebutuhan *bandwidth* pada jaringan komputer Politeknik Negeri Sriwijaya ditunjukkan seperti pada gambar 3.1.



Gambar 3.1 Prosedur Penelitian Peramalan Kebutuhan *Bandwidth* Pada Jaringan Komputer

Berikut ini penjelasan prosedur penelitian peramalan kebutuhan *bandwidth* yang telah digambarkan pada gambar 3.1

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* terdiri dari beberapa tahap yaitu:
 - a. Perancangan arsitektur jaringan untuk menentukan jumlah *node input* (simpul masukan), jumlah lapisan *layer* tersembunyi dengan jumlah nodenya (simpul tersembunyi) dan jumlah *node output* (simpul keluaran). Pada saat perancangan arsitektur jaringan ditentukan juga fungsi aktivasi yang akan digunakan. Fungsi aktivasi yang digunakan harus disesuaikan dengan algoritma yang digunakan dan karakteristik datanya.
 - b. Menyusun data set yang akan digunakan untuk pelatihan dan pengujian.
 - c. Inisialisasi data meliputi inisialisasi bobot, inisialisasi variable *learning rate*, *momentum*, maksimum *epoch* dan batas nilai toleransi pelatihan *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 3.2 berikut



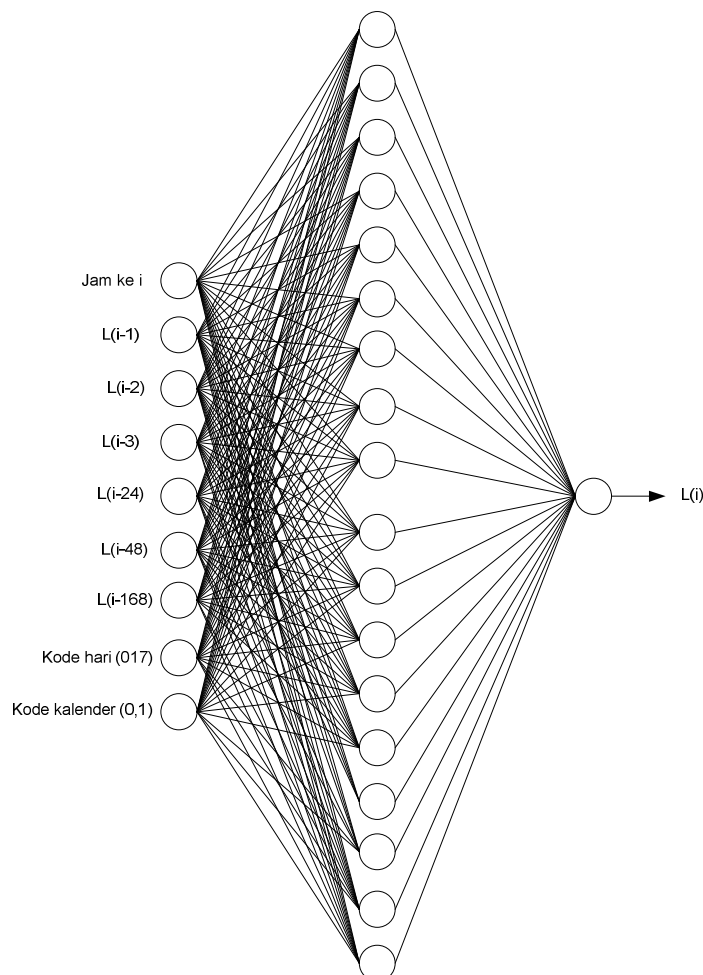
Gambar 3.2 Model Sekuensial Linier

- a. Analisis
Tahap ini merupakan tahapan menganalisa hal-hal yang diperlukan dalam pelaksanaan pembuatan perangkat lunak.
 - b. Desain
Tahap penerjemahan dari data yang dianalisis ke dalam bentuk yang mudah dimengerti oleh pengguna.
 - c. Coding
Tahap penerjemah data atau pemecahan masalah yang telah dirancang ke dalam bahasa pemrograman tertentu.
 - d. Testing
Merupakan tahapan pengujian terhadap perangkat lunak yang dibuat.
 - e. Maintenance
Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan-perubahan atau penambahan sesuai kebutuhan.
4. Proses pelatihan jaringan untuk menemukan bobot penghubung yang mendekati antara masing-masing data masukan dengan nilai keluaran yang diharapkan.
 5. 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*.
 6. 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.
 7. Hasil dari penelitian yang berupa tulisan dalam bentuk dokumen teknis, jurnal dan tesis.

3.3.2. 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) seperti pada gambar 3.3



Gambar 3.3 Struktur JST Untuk Peramalan Kebutuhan *Bandwidth*

Keterangan:

- Jam ke I : Jam yang akan di ramal besar *bandwidthnya*
- 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 *bandwidthnya*, 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

3.3.3. 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*

3.3.3.1. 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. Adapun perhitungan untuk fase pembelajaran adalah bobot awal yang telah di inisialisasi, kemudian dilakukan perhitungan seperti berikut:

Hitung harga aktivasi sel-sel pada *layer hidden* dengan persamaan 2.

$$Z_j = f(V_{0j} + \sum_i^n X_i V_{ij}) \quad (2)$$

Dimana V_{ij} adalah sel-sel pada *layer input*, Z_j adalah sel pada *layer hidden* dan X_i adalah bobot interkoneksi antara *layer input* dengan *layer hidden*. Selanjutnya hitung harga aktivasi sel-sel pada *layer output* dengan persamaan 3.

$$Y_k = f(W_{0k} + \sum_j W_{kj} * Z_j) \quad (3)$$

Dimana Y_k adalah sel-sel pada *layer output* dan W_{jk} adalah bobot interkoneksi antara *layer hidden* dengan *layer output*. Proses perhitungan harga aktivasi ini berlangsung terus sampai pola terakhir. Kemudian hitung harga δ_k yang akan digunakan untuk menghitung interkoneksi antara *layer hidden* dengan *layer output* dalam jaringan melalui persamaan 4.

$$\delta_k = [(t_k - y_k) * (1 - y_k) * y_k] \quad (4)$$

Dimana t_k merupakan hasil akhir dari jaringan yang diinginkan atau biasa disebut dengan target. Selanjutnya dilakukan pencarian harga δ_j yang akan digunakan untuk menghitung perubahan bobot interkoneksi antara *layer input* dengan *layer hidden* melalui persamaan 5.

$$\delta_j = \sum_k (\delta_k - W_{kj}) * (1 - Z_j) * Z_j \quad (5)$$

Selanjutnya dilakukan adaptasi bobot interkoneksi antara lapisan *hidden* dengan lapisan *output* dengan persamaan 6.

$$W_{kj}(t + 1) = W_{kj}(t) + \Delta W_{kj}(t) \quad (6)$$

$$\text{Dimana } \Delta W_{kj}(t + 1) = \sum (\delta_j * Z_j) + \Delta W_{kj}(t - 1)$$

Δ adalah kecepatan belajar dan α adalah momentum, kemudian dilakukan lagi adaptasi bobot interkoneksi antara *layer input* dengan *layer hidden* dengan persamaan 7.

$$V_{ji}(t + 1) = V_{ji}(t) + \Delta V_{ji}(t) \quad (7)$$

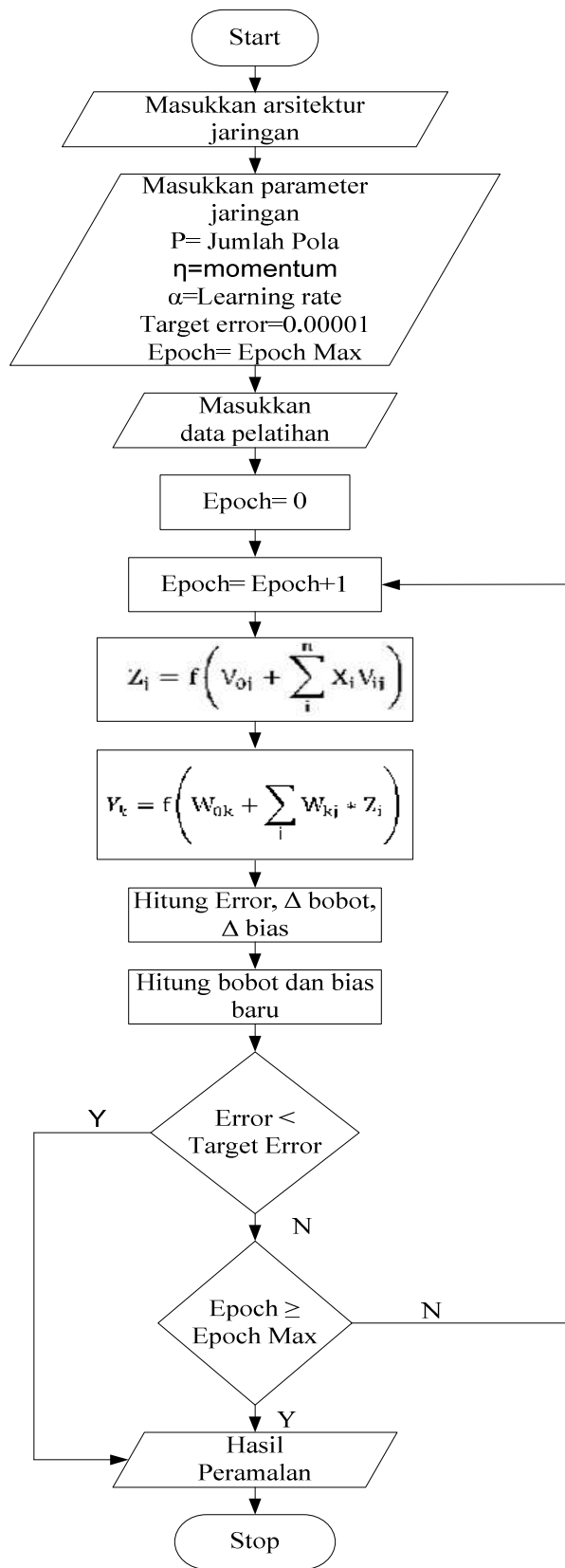
$$\text{Dimana } \Delta V_{ji}(t) = \sum(\delta_j * V) + \alpha \Delta V_{ji}(t - 1)$$

Tahap selanjutnya adalah perhitungan kesalahan antara target dengan hasil aktivasi jaringan saraf tiruan, seperti yang ditunjukkan oleh persamaan 8, dimana E adalah hasil penjumlahan *error*. Jika harga kesalahan rata-rata masih besar, proses pembelajaran di ulang dari awal begitu seterusnya sampai dicapai harga kesalahan yang minimum.

$$E = \frac{1}{n} \sum_{i=1}^n (t_k - y_k)^2 \quad (8)$$

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 pada bab sebelumnya.

Untuk lebih jelasnya dari proses Program pembelajaran jaringan yang digunakan dalam peramalan kebutuhan *bandwidth* dapat dilihat pada *Flowchart*/gambar 3.4.



Gambar 3.4 Fase Pembelajaran Metode *Backpropagation Error*

3.3.3.2. 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 aktivasi sel-sel pada lapisan akhir adalah merupakan keputusan dari jaringan saraf tiruan.

3.4. Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan pengelolaan informasi pada suatu sistem. Perancangan ini dibuat sebagai gambaran/bahan dari suatu program/aplikasi yang akan di bangun. Rancangan antarmuka yang dibangun sebagai berikut:

3.4.1 Rancangan Antarmuka Tampilan Utama

Peramalan Bandwidth Menggunakan JST					
Home	Peramalan JST	Test Data Pelatihan	Peramalan Bandwidth	Tentang Kami	Help
 Ikhtison Mekongga Universitas Diponegoro Semarang					

Gambar 3.6 *Form Menu* Tampilan Utama

Keterangan:

1. *Menu Home* berisi informasi mengenai judul penelitian, logo dan identitas

2. *Menu* Pelatihan JST berisi *form* pelatihan JST, yang terdiri dari struktur JST, data histori *bandwidth*, *setup neuron*, *setup Learning parameter*, *ambil file* dan *start training*.
3. *Menu Test Data* Pelatihan berisi *form* pola *input* dan target, hasil pelatihan, *setup data neuron*, *ambil file*, *ambil bobot*, *start test* dan *cek hasil*
4. *Menu Peramalan Bandwidth* berisi *form setup neuron*, *open file*, *baca bobot*, *start*, *cek hasil* dan *lihat target*.

3.4.2. Rancangan Antarmuka Pelatihan Jaringan Saraf Tiruan

Peramalan Bandwidth Menggunakan JST

Home	Peramalan JST	Test Data Pelatihan	Peramalan Bandwidth	Tentang Kami	Help
------	---------------	---------------------	---------------------	--------------	------

Struktur JST

text

Data Histori Bandwidth

text

Setup Neuron

Input Layer

Hidden Layer

Output Layer

Setup Learning Parameter

Momentum % Accuration

Learning rate Max Epoch

Ambil File

Start Training

Gambar 3.7 Rancangan Antarmuka Pelatihan Jaringan Saraf Tiruan

Keterangan:

1. Pada *form setup neuron* terdapat tiga kolom yang harus diisi nilainya, yaitu jumlah *input Mapping*, jumlah *hidden Mapping* dan jumlah *output Mapping*.
2. Pada *form setup Learning* parameter terdapat empat kolom yang harus diisi nilainya yaitu: nilai *momentum*, nilai *Learning rate*, *% Accuration* dan *max epoch*.

3. Tombol ambil *file*, untuk mengambil data yang akan di latih.
4. Tombol *start* untuk memulai pelatihan JST
5. Pada tampilah data historis *bandwidth* untuk menampilkan data yang akan di latih.

3.4.3. Rancang Bangun Antarmuka Test Data Pelatihan

Gambar 3.8 Rancang Bangun Antarmuka Test Data Pelatihan

Keterangan:

1. Tampilan Pola *Input* dan target akan menampilkan data *input* dan target
2. Tampilan Hasil akan menampilkan hasil *Test*.
3. Pada *form setup neuron* terdapat empat kolom yang harus diisi nilainya, yaitu jumlah *input Mapping*, jumlah *hidden Mapping*, jumlah *output Mapping* dan jumlah pola.
4. Tombol ambil *file*, untuk mengambil data yang akan di latih.
5. Tombol *start test* untuk memulai *test*.
6. Tombol ambil bobot, untuk mengambil *file* nilai bobot.
7. Tombol cek hasil, untuk melihat hasil *test*.

3.4.4. Rancang Bangun Antarmuka Peramalan *Bandwidth*

Peramalan Bandwidth Menggunakan JST					
Home	Peramalan JST	Test Data Pelatihan	Peramalan Bandwidth	Tentang Kami	Help
text					
Setup Neuron					
Input	<input type="text"/>	te	Output	<input type="text"/>	
Hidden	<input type="text"/>	Pola	<input type="text"/>		
		Open File	Baca Bobot	Start	Check Hasil
					Lihat Target

Gambar 3.9 Rancang Bangun Antarmuka *Test Data Pelatihan*

Keterangan:

1. Pada *form setup neuron* terdapat empat kolom yang harus diisi nilainya, yaitu jumlah *input Mapping*, jumlah *hidden Mapping*, jumlah *output Mapping* dan jumlah pola.
2. Tombol *open file*, untuk mengambil data yang akan di gunakan.
3. Tombol *baca bobot*, untuk mengambil nilai bobot.
4. Tombol *start* untuk memulai peramalan.
5. Tombol *cek hasil*, untuk menampilkan hasil peramalan.
6. Tombol *Target* untuk menampilkan Target yang akan diramal.