BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah Singkat Perusahaan

PT. Perkebunan Nusantara IX (PERSERO) PG. Pangka didirikan pada jaman kolonial Belanda pada tahun 1832 oleh NV. MIJTOT EXPLOITATIE DERT SUIKER FABRIKEN, dikelola oleh NV KOSY dan SUCIER yang bekedudukan di Surakarta. Pada waktu pendiriaanya ribuan tenaga kerja pribumi dikerahkan dengan sistem kerja paksa oleh pemerintah kolonial Belanda. Berdasarkan UU Nomor 86 Tahun 1958, PG Pangka telah diambil oleh Pemerintah Republik Indonesia atau istilah popular pada saat itu “di Nasionalisasi”.

Nama Pangka diambil dari nama desa di mana pabrik ini didirikan yaitu desa Pangkah. Desa ini diberi nama Pangkah karena dahulu terdapat sebuah pohon kelapa yang mempunyai cabang tujuh atau dalam bahasa jawa berpang tujuh jadi disebut Desa Pangkah.

Pada awalnya di Kabupaten Tegal terdapat tujuh pabrik gula, namun karena kurangnya bahan baku akhirnya enam pabrik ditidurkan dan hanya Pabrik Gula Pangkah yang masih berdiri sampai sekarang. Adapun PP yang mengatur berdirinya PG.Pangka adalah :

1. PP No. 24 tahun 1958 dan PP No.19 tahun 1959 tentang Nasionalis PG. Kemudian statusnya berubah menjadi Pusat Perkebunan Negara Baru (PPN Baru).

2. PP No. 22 tahun 1973 tentang PTP XV (PERSERO)
3. PP No. 11 tahun 1981 penggabungan PTP XV-XVI(PERSERO) kantor direksi berada di Surakarta.
4. PP No.14 tahun 1996 tentang peleburan antara PTP XV-XVI dengan PTP XVIII menjadi Nusantara IX (PERSERO).

2.1.2 Visi dan Misi Perusahaan

Visi dan misi merupakan sebuah curahan dari sebuah tujuan dalam sebuah lembaga baik dalam lembaga pendidikan organisasi-organisasi kemasyarakatan bahkan dalam industri pun terdapat visi dan misi yang ada pada PG.Pangka. Adapun visi dan misi PG.Pangka antara lain :

1) Visi

   Menjadi perusahaan agro bisnis dan agro industry yang berdaya saing tinggi dan tumbuh berkembang bersama mitra.

2) Misi

   Adapun misi PG.Pangka yaitu:

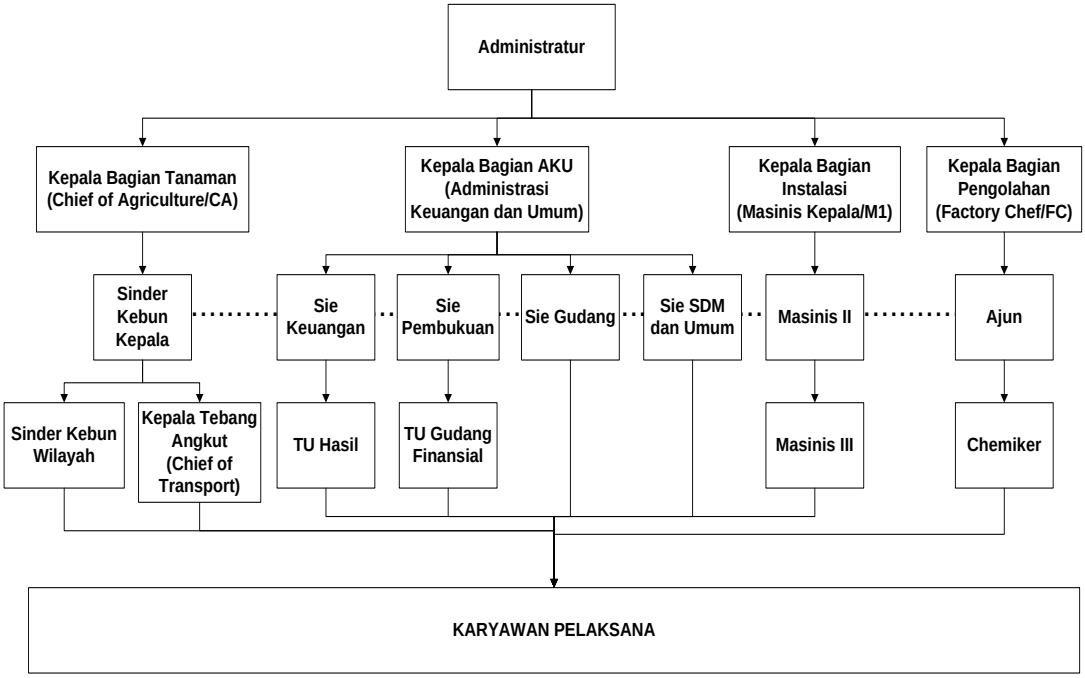
   a) Memproduksi dan memasarkan produk kretek, kopi, kakao, gula dan tetes ke pasar domestik dan Internasional secara professional untuk menghasilkan pertumbuhan laba.
   b) Meningkatkan kesejahteraan karyawan, menciptakan lingkungan kerja yang sehat serta menyelenggarakan pelatihan guna menjaga motivasi karyawan dalam upaya meningkatkan produktifitas kerja.
   c) Mengembangkan produk hilir agrowisata dan usaha lainnya untuk mendukung kinerja perusahaan.

d) Membangun sinergi dengan mitra usaha strategis dan masyarakat lingkungan usaha untuk mewujudkan kesejahteraan bersama.
e) Mendukung program pemerintah dalam pemenuhan kebutuhan gula nasional.
f) Memperdayakan seluruh sumber daya perusahaan dan potensi lingkungan guna mendukung pembangunan ekonomi AAS. Melalui penciptaan lapangan kerja.

2.1.3 Struktur Organisasi

Struktur Organisasi adalah suatu susunan dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalam menjalankan kegiatan operasional untuk mencapai tujuan. Struktur Organisasi menggambarkan dengan jelas pemisahan kegiatan pekerjaan antara yang satu dengan yang lain dan bagaimana hubungan aktivitas dan fungsi dibatasi. Dalam struktur organisasi yang baik harus menjelaskan hubungan wewenang siapa melapor kepada siapa.

Beberapa keuntungan yang dapat diperoleh dari penggunaan bagan organisasi adalah dapat memperlihatkan karakteristik utama dari suatu perusahaan tersebut, tentang gambaran pekerjaan dan hubungan-hubungan yang ada didalam perusahaan serta digunakan untuk merumuskan rencana kerja yang ideal sebagai pedoman untuk dapat mengetahui siapa bawahan dan atasannya,. Struktur keorganisasian pada PG.Pangka dapat dilihat pada bagan struktur organisasi (gambar 2.1) berikut ini :



Gambar 2.1 Struktur organisasi pada PG.Pangka

2.1.4 Deskripsi Tugas

Definisi tugas digunakan untuk mengetahui tugas, wewenang, tanggung jawab dari masing-masing bagian. Definisi tugas yang ada di PG.Pangka adalah sebagai berikut :

1. Administratur
   a. Menjalankan tugas manajerial secara keseluruhan di Pabrik Gula, bertanggung jawab secara langsung kepada Direktur Utama secara administrasi dan teknis. Bertanggung jawab kepada Direksi dalam fungsinya masing-masing dan mempunyai wewenang yang sebanding dalam menjalankan tugas-tugas perusahaan.
   b. Memimpin bagian-bagian yang ada di unit produksi antara lain : bagian tanaman, bagian pabrik, bagian pengolahan dan bagian Administrasi Keuangan dan Umum.

2. Kepala Bagian Tanaman
    a. Bertanggung jawab terhadap produktivitas tebu dan pencapaian areal yang telah ditentukan oleh Direksi.
    b. Membina hubungan baik dengan petani tebu pola kemitraan.
    c. Bertanggung jawab terhadap persediaan bahan baku secara kuantitas dan kualitas yang sesuai SOP.
    d. Bertanggung jawab pemasokkan tebu untuk memenuhi kapasitas giling di dalam pabrik
    e. Dalam pelaksanaannya kepala tanaman di bantu oleh para sinder kepala dan staf administrasi tanaman.
    f. Dalam musim giling kepala tanaman dalam penyediaan tebu dibantu oleh staf tebang dan angkut.
3. Kepala Bagian Instalasi
    a. Bertanggung jawab dalam masalah-masalah yang berhubungan dengan teknis, pengadaan dan pemeliharaan mesin-mesin perusahaan.
    b. Memimpin dan mengkoordinasikan para pembantunya mulai dari karyawan pelaksana sampai dengan karyawan staff (pimpinan).
    c. Bertanggung jawab atas kelancaran proses giling.
4. Kepala Bagian Pengolahan
    a. Bertanggung jawab terhadap kelancaran proses pengolahan gula.
    b. Bertanggung jawab terhadap produktivitas gula.
    c. Bertanggung jawab atas kualitas produk yang dihasilkan.

d. Dalam pelaksanaan tugasnya dibantu oleh chemiker dan karyawan pelaksana.

5. Kepala Administrasi Keuangan dan Umum
   a. Bertanggung jawab atas kelancaran pelayanan kepada semua bagian yang ada di PG.Pangka
   b. Bertanggung jawab terhadap kelancaran pengadaan barang dan penyediaan dana operasional.
   c. Bertanggung jawab di bidang administrasi tenaga kerja, keuangan.
   d. Bertanggung jawab terhadap laporan keuangan kepada Direksi.

2.2 Landasan teori

2.2.1 Jaringan Saraf Tiruan

JST adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologi di dalam otak. JST dapat digambarkan sebagai model matematis dan komputasi untuk fungsi aproksimasi nonlinear, klasifikasi data, cluster dan regresi non parametrik atau sebagai sebuah simulasi dari koleksi model saraf biologi.

Valluru B.Rao dan Hayagriva V.Rao (1993) mendefinisikan jaringan saraf sebagai sebuah kelompok pengolahan elemen dalam suatu kelompok yang khusus membuat perhitungan sendiri dan memberikan hasilnya kepada kelompok kedua atau berikutnya. Setiap sub kelompok menurut gilirannya harus membuat perhitungan sendiri dan memberikan hasilnya untuk subgroup atau kelompok yang belum melakukan perhitungan. Pada akhirnya sebuah kelompok dari satu

atau beberapa pengolahan elemen tersebut menghasilkan keluaran *(output)* dari jaringan.

Metode jaringan saraf tiruan (JST) memiliki karakteristik yang menyerupai jaringan saraf biologi dalam memproses informasi (Marimin, 2005). JST dapat menyimpan pengetahuan pola kejadian masa lampau melalui proses pelatihan yang kemudian pengetahuan tersebut digunakan untuk memperkirakan kejadian yang akan terjadi dimasa yang akan datang.

Tiga hal yang sangat menentukan keandalan sebuah JST adalah pola rangkaian *neuron-neuron* dalam jaringan yang disebut dengan arsitektur jaringan. Algoritma untuk menentukan bobot penghubung yang disebut dengan algoritma pelatihan, dan persamaan fungsi untuk mengolah masukan yang akan diterima oleh *neuron* yang disebut dengan fungsi aktivasi (Fausett, 1994).

Jaringan saraf tiruan seperti manusia, belajar dari suatu contoh karena mempunyai karakteristik yang adaptif, yaitu dapat belajar dari data-data sebelumya dan mengenal pola data yang selalu berubah. Selain itu, JST merupakan system yang tak terpogram, artinya semua keluaran atau kesimpulan yang ditarik oleh jaringan didasarkan pada pengalamannya selama mengikuti proses pembelajaran / pelatihan.

Hal yang ingin dicapai dengan melatih JST adalah untuk mencapai keseimbangan antara kemampuan *memorisasi* dan *generalisasi*. Yang dimaksud kemampuan memorisasi adalah kemampuan JST untuk mengambil kembali secara sempurna sebuah pola yang telah dipelajari. Kemampuan *generalisasi* adalah kemampuan JST 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 JST itu diinputkan informasi baru yang belum pernah dipelajari, maka JST itu masih akan tetap dapat memberikan tanggapan yang baik, memberikan keluaran yang paling mendekati (Puspitaningrum, Diyah 2006).

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

JST menyerupai otak manusia dalam dua hal yaitu :

a) Pengetahuan diperoleh jaringan melalui proses belajar.
b) Kekuatan hubungan antar sel saraf (neuron) yang dikenal sebagai bobot-bobot sinaptik digunakan untuk menyimpan pengetahuan.

Kelebihan JST antara lain :

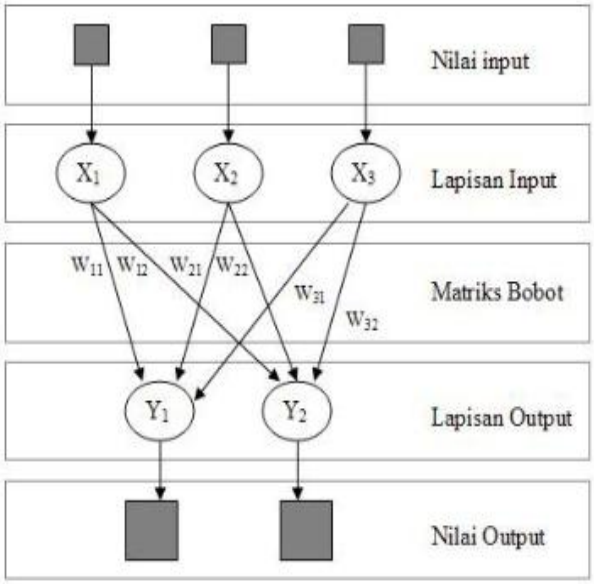
1. Mampu mengakuisisi pengetahuan walau tidak ada kepastian.
2. Mampu melakukan generalisasi dan ekstraksi dari suatu pola data tertentu.
3. JST dapat menciptakan suatu pola pengetahuan melalui pengaturan diri atau kemampuan belajar (*self organizing*).
4. Memiliki *fault tolerance*, gangguan dapat dianggap sebagai *noise* saja.
5. Kemampuan perhitungan secara paralel sehingga proses lebih singkat.

2.2.1.1 Arsitektur (model) JST

Dalam JST ada beberapa model yang digunakan, antara lain:

2.2.1.1.1 Jaringan dengan lapisan tunggal (Single Layer Net)

Jaringan dengan lapisan unggal hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi.



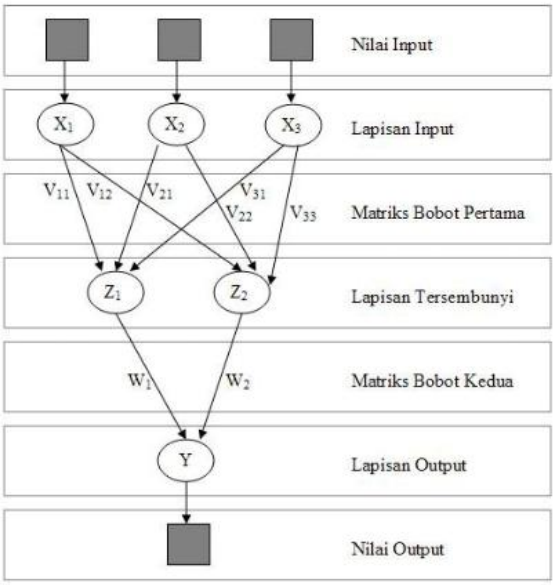
Gambar 2.2 Jaringan saraf dengan lapisan tunggal

2.2.1.1.2 Jaringan dengan banyak lapisan (Multilayer Net)

Dalam model ini sebenarnya merupakan model JST satu lapisan yang jumlahnya banyak dan prinsip kerja dari model JST ini sama dengan JST satu lapisan. Output dari tiap lapisan sebelumnya dalam model ini merupakan input dari lapisan sebelumnya.

Arsitektur jaringan yang lebih *kompleks* Menurut Rumelhart, et al. (1996) terdiri dari layar *input*, beberapa layar tersembunyi dan layar *output*. Jaringan dengan banyak lapisan lebih sering digunakan karena dapat menyelesaikan

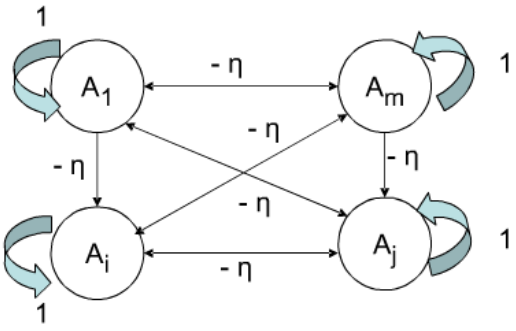
masalah yang lebih *kompleks* dibandingkan jaringan layar tunggal, meskipun proses pelatihannya lebih komplek dan lebih lama (Haykin, 1999).



Gambar 2.3 Jaringan saraf dengan banyak lapisan

2.2.1.1.3 Jaringan dengan lapisan kompetitif (*competitive layer net*)

Merupakan jenis jaringan saraf yang memiliki bobot yang telah ditetapkan dan tidak memiliki proses pelatihan. Digunakan untuk mengetahui neuron pemenang dari sejumlah neuron yang ada. Nilai bobot untuk diri sendiri tiap neuron adalah 1, sedangkan untuk neuron lain adalah bobot random negatif. Berikut ini menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot - η.



Gambar 2.4 Jaringan saraf dengan lapisan kompetitif

2.2.1.2 Komponen Jaringan Saraf Tiruan

Dengan mengambil ide dari jaringan saraf manusia, komponen-komponen pada JST adalah :

a) Neuron Tiruan(*artificial neuron*)

JST disusun oleh unit dasar yang disebut neuron tiruan yang merupakan elemen pemrosesan dalam jaringan, dimana semua proses perhitungan dilakukan disini.

b) Lapisan (*layer*)

JST disusun oleh kumpulan neuron yg berhubungan dan dikelompokkan pd lapisan2 (layer). Terdapat tiga lapisan, yaitu : lapisan masukan(*input layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*).

c) Masukan(*input*)

JST hanya dapat memproses data masukan berupa data numerik, sehingga apabila masalah melibatkan data kualitatif seperti grafik, *image*, sinyal atau suara data harus ditransformasikan dulu ke dalam data numerik yg ekuivalen sebelum dapat diproses oleh JST.

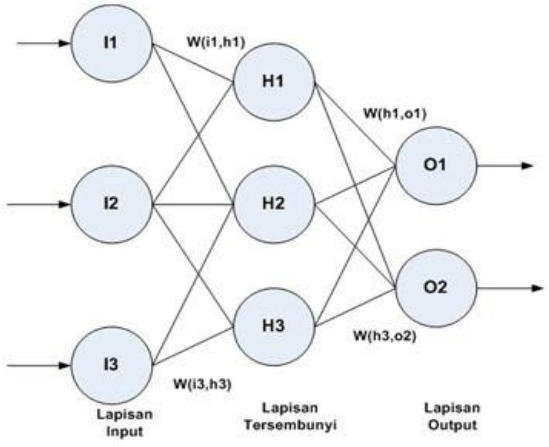
d) Keluaran (*output*)

Keluaran dari JST adalah pemecahan terhadap masalah. Data keluaran merupakan data numerik.

e) Bobot(*weight*)

Bobot pada JST menyatakan tingkat kepintaran sistem. Walaupun sebenarnya bobot tersebut hanya sebuah deretan angka-angka saja. Bobot sangat penting untuk JST, dimana bobot yang optimal akan memungkinkan sistem menterjemahkan data masukan secara benar dan menghasilkan keluaran yang diinginkan.

Ada beberapa tipe jaringan saraf, namun demikian hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan saraf juga terdiri dari beberapa neuron dan ada hubungan antara neuron-neuron tersebut. Neuron-neuron tersebut akan mentransformasikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron neuron yang lain. Pada jaringan saraf hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut.



Gambar 2.5 struktur neuron jaringan saraf tiruan

Neuron buatan sebenarnya mirip dengan sel neuron biologis. Neuron-neuron buatan tersebut bekerja dengan cara yang sama pula dengan neuron-neuron biologis. Informasi (disebut dengan: input) akan dikirim ke neuron dengan bobot kedatangan tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini 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, tapi jika tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya.

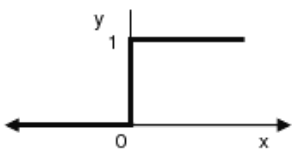
2.2.1.3 Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan saraf tiruan, yaitu:

1. Fungsi undak biner (hard limit)

   Jaringan dengan lapisan tunggal sering menggunakan fungsi undak untuk mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output biner (0 atau 1). Fungsi undak biner (hard limit) dirumuskan sebagai berikut :

$$y = \begin{cases} 0, & jika\ x \le 0 \\ 1, & jika\ x > 0 \end{cases}$$

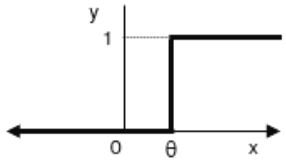


Gambar 2.6 Fungsi aktivasi : undak biner (hard limit)

2. Fungsi undak biner (Threshold)

Fungsi undak biner dengan menggunakan nilai ambang sering juga disebut dengan nama fungsi nilai ambang (threshold) atau fungsi *Heaviside*. Fungsi undak biner (Threshold) dirumuskan sebagai berikut :

$$y = \begin{cases} 0, & jika\ x < \theta \\ 1, & jika\ x \geq \theta \end{cases}$$



Gambar 2.7 Fungsi aktivasi : undak biner (Threshold)

3. Fungsi bipolar (symetric hard limit)

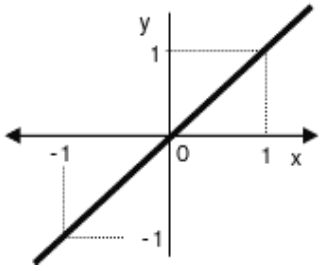
Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner, hanya saja output yang dihasilkan berupa 1, 0 atau -1.

4. Fungsi bipolar dengan threshold

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner dengan threshold, hanya saja output yang dihasilkan berupa 1, 0 atau -1.

5. Fungsi linear (identitas)

Fungsi linear memiliki nilai output yang sama dengan nilai inputnya. Fungsi linear dirumuskan sebagai y = x.



Gambar 2.8 Fungsi aktivasi : Linear (identitas)

6. Fungsi saturating linear

   Fungsi ini akan bernilai 0 jika inputnya kurang dari - 1/2 dan akan bernilai 1 jika inputnya lebih dari 1/2. Sedangkan jika nilai input terletak antara -1/2 dan 1/2, maka outputnya akan bernilai sama dengan nilai input ditambah ½.

7. Fungsi simetric saturating linear

   Fungsi ini akan bernilai -1 jika inputnya kurang dari -1 dan akan bernilai 1 jika inputnya lebih dari 1. Sedangkan jika nilai input terletak antara -1 dan 1 maka outputnya akan bernilai sama dengan nilai inputnya.

8. Fungsi sigmoid biner

   Fungsi ini digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode backpropagation. Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan saraf yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan saraf yang nilai outputnya 0 atau 1.

   $y = f(x) = \frac{1}{1+e^{-\sigma x}}$ …… (1)

   dengan: $f'(x) = \sigma f(x)[1-f(x)]$ …… (2)

9. Fungsi sigmoid bipolar

   Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja output dari fungsi ini memiliki range antara -1 sampai 1.

   $y = f(x) = \frac{1-e^{-x}}{1+e^{-x}}$ ……..(3)

   dengan: $f'(x) = \frac{\sigma}{2}[1+f(x)][1-f(x)]$ ……(4)

2.2.1.4 *Sum Square Error* dan *Root Mean Square Error*

Perhitungan kesalahan merupakan pengukuran bagaimana jaringan dapat belajar dengan baik sehingga jika dibandingkan dengan pola yang baru akan dengan mudah dikenali. Kesalahan pada keluaran jaringan merupakan selisih antara keluaran sebenarnya (*current output*) dan keluaran yang diinginkan (*desired output*). Selisih yang dihasilkan antara keduanya biasanya ditentukan dengan cara dihitung menggunakan suatu persamaan.

Sum Square Error (SSE) dihitung sebagai berikut :

1. Hitung keluaran jaringan saraf untuk masukan pertama.
2. Hitung selisih antara nilai keluaran jaringan saraf dan nilai target / yang diinginkan untuk seiap keluaran.
3. Kuadratkan setiap keluaran kemudian hitung seluruhnya. Ini merupakan kuadrat kesalahan untuk contoh latihan.

Adapun rumusnya adalah :

$$SSE = \sum_{p} \sum_{j} (T_{jp} - X_{jp})^2$$

Dengan :

T_{jp} : nilai keluaran jaringan saraf

X_{jp} : nilai target / yang ingin diinginkan untuk setiap keluaran

Root Mean Square Error (RMS *Error*) dihitung sebagai berikut :

1. Hitung SSE.
2. Hasilnya dibagi dengan perkalian antara banyaknya data pada pelatihan dan banyaknya keluaran, kemudian diakarkan.

Rumus :

$$\text{RMS Error} = \sqrt{\frac{\sum_p \sum_j (T_{jp} - X_{jp})^2}{n_p \, n_o}}$$

Dengan :

T_{jp} : nilai keluaran jaringan saraf

X_{jp} : nilai target / yang ingin diinginkan untuk setiap keluaran

n_p : jumlah seluruh pola

n_o : jumlah keluaran

2.2.1.5 JST *Backpropagation*

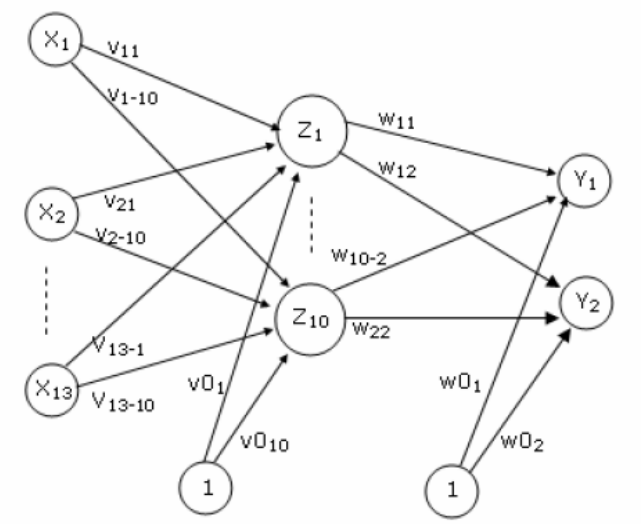
Perambatan galat mundur (*backpropagation*) adalah sebuah metode sistematik untuk pelatihan *multiplayer* jaringan saraf tiruan. Metode ini memiliki dasar matematis yang kuat, obyektif dan algoritma ini mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah kuadrat galat error melalui model yang dikembangkan (*training set*) (Brace, 1997).

a. Dimulai dengan lapisan masukan, hitung keluaran dari setiap elemen pemroses melalui lapisan luar.
b. Hitung kesalahan pada lapisan luar yang merupakan selisih antara data aktual dan target.
c. Transformasikan kesalahan tersebut pada kesalahan yang sesuai di sisi masukan elemen pemroses.
d. Propagasi balik kesalahan-kesalahan ini pada keluaran setiap elemen pemroses ke kesalahan yang terdapat pada masukan. Ulangi proses ini sampai masukan tercapai.

e. Ubah seluruh bobot dengan menggunakan kesalahan pada sisi masukan elemen dan luaran elemen pemroses yang terhubung.

2.2.1.5.1 Arsitektur Jaringan Metode Backpropagation

Jaringan saraf terdiri dari 3 lapisan, yaitu lapisan masukan/input terdiri atas variabel masukan unit sel saraf, lapisan tersembunyi, dan lapisan keluaran/output. (Brace, 1997).



Gambar 2.9 Arsitektur jaringan *backpropagation*

Keterangan :

X = Masukan (input)

J = 1..n (n = 10)

V = Bobot pada lapisan tersembunyi

W = Bobot pada lapisan keluaran

n = Jumlah unit pengolah pada lapisan tersembunyi

b = Bias pada lapisan tersembunyi dan lapisan keluaran

k = Jumlah unit pengolah pada lapisan keluaran

Y = Keluaran hasil

Tujuan dari perubahan bobot untuk setiap lapisan, bukan merupakan hal yang sangat penting. Perhitungan kesalahan merupakan pengukuran bagaimana jaringan dapat belajar dengan baik. Kesalahan pada keluaran dari jaringan merupakan selisih antara keluaran aktual (*current output*) dan keluaran target (*desired output*). Langkah berikutnya adalah menghitung nilai SSE *(Sum Square Error)* yang merupakan hasil penjumlahan nilai kuadrat error neuron1 dan neuron2 pada lapisan output tiap data, dimana hasil penjumlahan keseluruhan nilai SSE akan digunakan untuk menghitung nilai RMSE *(Root Mean Square Error)* tiap iterasi (Kusumadewi, 2002).

Sum Square Error (SSE) dihitung sebagai berikut:

a. Hitung lapisan prediksi atau keluaran model untuk masukan pertama.
b. Hitung selisih antara nilai luar prediksi dan nilai target atau sinyal latihan untuk setiap keluaran.
c. Kuadratkan setiap keluaran kemudian hitung seluruhnya. Ini merupakan kuadrat kesalahan untuk contoh lain.

$$\boldsymbol{SSE = \sum_{i=1}^{N} (Dij - fj(Xi))}$$

Root Mean Square Error (RMS *Error*). Dihitung sebagai berikut:

a. Hitung SSE.
b. Hasilnya dibagi dengan perkalian antara banyaknya data pada latihan dan banyaknya luaran, kemudian diakarkan.

$$RMSE = \sqrt{\frac{SSE}{N * K}}$$

dimana

RMSE = Root Mean Square Error

SSE = Sum Square Error

N = Banyak data pada latihan

K = Banyak luaran.

2.2.1.5.2 Algoritma pelatihan pada JST backpropagation

Pelatihan suatu jaringan dengan algoritma *backpropagation* meliputi dua tahap : perambatan maju dan perambatan mundur.

Selama perambatan maju, tiap unit masukan (x_i) menerima sebuah masukan sinyal ini ke tiap-tiap lapisan tersembunyi $z_1,\ldots..,z_p$. Tiap unit tersembunyi ini kemudian menghitung aktivasinya dan mengirimkan sinyalnya (z_j) ke tiap unit keluaran. Tiap unit keluaran (y_k) menghitung aktivasinya (y_k) untuk membentuk respon pada jaringan untuk memberikan pola masukan.

Selama pelatihan, tiap unit keluaran membandingkan perhitungan aktivasinya y_k dengan nilai targetnya t_k untuk menentukan kesalahan pola tersebut dengan unit itu. Berdasarkan kesalahan ini, faktor δ_k *(k = 1,..,m) dihitung.* δ_k digunakan untuk menyebarkan kesalahan pada unit keluaran y_k kembali ke semua unit pada lapisan sebelumnya (unit-unit tersembunyi yang dihubungkan ke y_k). Juga digunakan (nantinya) untuk mengupdate bobot-bobot antara keluaran dan lapisan tersembunyi. Dengan cara yang sama, faktor ($j = 1,\ldots,p$) dihitung untuk tiap unit tersembunyi z_j. Tidak perlu untuk menyebarkan kesalahan kembali ke

lapisan masukan, tetapi δ_j digunakan untuk mengupdate bobot-bobot antara lapisan tersembunyi dan lapisan masukan.

Setelah seluruh faktor δ ditentukan, bobot untuk semua lapisan diatur secara serentak. Pengaturan bobot w_{jk} (dari unit tersembunyi z_j ke unit keluaran y_k) didasarkan pada faktor δ_k dan aktivasi z_j dari unit tersembunyi z_j. didasarkan pada faktor δ_j dan dan aktivasi x_i unit masukan. Untuk langkah selengkapnya adalah (Fausset, 1994):

a) Prosedur pelatihan

Langkah 0 : Inisialisasi bobot. (sebaiknya diatur pada nilai acak yang kecil),

Langkah 1 : Jika kondisi tidak tercapai, lakukan langkah 2-9,

Langkah 2 : Untuk setiap pasangan pelatihan, lakukan langkah 3-8,

Perambatan Maju

Langkah 3: Tiap unit masukan (x_i, $i = 1,\ldots, n$) menerima sinyal x_i dan menghantarkan sinyal ini ke semua unit lapisan di atasnya (unit tersembunyi),

Langkah 4 : Setiap unit tersembunyi (x_i, $i = 1,\ldots, p$) jumlahkan bobot sinyal masukannya,

$$Z_{inj} = Voj + \sum_{i=1}^{n} Xi\ Vij \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

v_{oj} = bias pada unit tersembunyi j aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya, $z_j = f(z_in_j)$, dan kirimkan sinyal ini keseluruh unit pada lapisan diatasnya (unit keluaran).

Langkah 5: Tiap unit keluaran (y_k, $k = 1,\ldots, m$) jumlahkan bobot sinyal masukannya,

$$\mathrm{Y_{ink}} = Wok + \sum_{i=1}^{n} Xi\,\mathrm{Vij} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

w_{ok} = bias pada unit keluaran k dan aplikasikan fungsi aktivasinya untuk menghitung sinyal keluarannya, $yk = f(y_ink)$.

Perambatan Mundur

Langkah 6: Tiap unit keluaran (y_k, $k = 1,\ldots, m$) menerima pola target yang saling berhubungan pada masukan pola pelatihan, hitung kesalahan informasinya,

$\delta_k = (t_k - y_k)\, f^{1}(y_in_k)$ …….(7)

Hitung koreksi bobotnya (digunakan untuk memperbaharui w_{jk} nantinya),

$\Delta w_{jk} = \alpha\, \delta_k\, z_j$ ……...(8)

Hitung koreksi biasnya (digunakan untuk memperbaharui w_{ok} nantinya), dan kirimkan δ_k ke unit-unit pada lapisan dibawahnya,

Langkah 7: Setiap unit lapisan tersembunyi (z_j, $j = 1,\ldots, p$) jumlahkan hasil perubahan masukannya (dari unit-unit lapisan diatasnya),

$\Delta\delta_in_j = \sum \delta_k\, w_{jk}$ ……(9)

Kalikan dengan turunan fungsi aktivasinya untuk menghitung informasi kesalahannya,

$\delta_j = \delta_in_j\, f^{1}(z_in_j)$ ……..(10)

Hitung koreksi bobotnya (digunakan untuk memperbaharui v_{oj} nanti),

$\Delta v_{ij} = \alpha\, \delta_j\, x_i$ ……...(11)

Langkah 8: Tiap unit keluaran (y_k, $k = 1..\ m$) update bias dan bobotnya ($j = 0,...,\ p$):

w_{jk} (baru) = w_{jk} (lama) + $\Delta\ w_{jk}$ ……(12)

Tiap unit lapisan tersembunyi (z_j, $j = 1,...,\ p$) update bias dan bobotnya

($i = 0,...,n$) :

v_{ij} (baru) = v_{ij} (lama) + $\Delta\ v_{ij}$ …… (13)

Langkah 9: Test kondisi berhenti.

b) Prosedure Pengujian

Setelah pelatihan, jaringan saraf *backpropagation* diaplikasikan dengan hanya menggunakan tahap perambatan maju dari algoritma pelatihan. Prosedur aplikasinya adalah sebagai berikut:

Langkah 0 **:** Inisialisasi bobot (dari algoritma pelatihan).

Langkah 1 : Untuk tiap vektor masukan, lakukan langkah 2-4.

Langkah 2 : for i = 1,…, n : atur aktivasi unit masukan xi .

Langkah 3 : for j = 1,…, p :

$Z_{-inj} = Voj + \sum_{i=1}^{n} Xi\ Vij$ ……(14)

Zj = f(Z_in_j)……(15)

Langkah 4 : for k = 1,…, m :

$y_{ink} = wok + \sum_{j=1}^{p} Zj\ Wjk$……(16)

y_k = f (y_in_k) ……(17)

2.2.1.5.3 *Backpropagation* dalam Peramalan

Salah satu bidang, backpropagation dapat diaplikasikan dengan baik adalah bidang peramalan (*forecasting*). Peramalan yang sering diketahui adalah peramalan besarnya penjualan, nilai tukar valuta asing, prediksi besarnya aliran sungai dan lain-lain.

Secara umum, masalah peramalan dapat dinyatakan dengan sejumlah data runtun waktu (*time series*) x_1, x_2,..., xn. Masalahnya adalah memperkirakan berapa harga xn+1 berdasarkan x_1, x_2,..., xn. Langkah-langkah membangun struktur jaringan untuk peramalan sebagai berikut.

1. Transformasi Data

Langkah awal sebelum melakukan proses pelatihan pada jaringan yang akan digunakan untuk peramalan adalah transformasi data. Sebab-sebab utama data ditransformasi adalah agar kestabilan taburan data dicapai. Selain itu berguna untuk menyesuaikan nilai data dengan *range* fungsi aktivasi yang digunakan dalam jaringan (Siang 2005:121). Ada beberapa transformasi yang digunakan, yaitu transformasi polinomial, transformasi normal dan transformasi linear.

Nilai hasil transformasi polinomial, normal dan linear dapat diperoleh dengan persamaan sebagai berikut.

a. Transformasi Polinomial

$x' = \ln x$

dengan

x'= nilai data setelah transformasi polinomial.

x = nilai data aktual.

b. Transformasi Normal

$$x_n = \frac{X0 - Xmin}{Xmax - Xmin}$$

dengan

x_n = nilai data normal.

x_0 = nilai data aktual.

x_{min} = nilai minimum data aktual keseluruhan.

x_{max} = nilai maksimum data aktual keseluruhan.

c. Transformasi Linear pada selang [a , b]

$$x' = \frac{(X - Xmin)(b - a)}{(Xmax - Xmin)} + a$$

(Jong Jek Siang.2005:121)

dengan

x'= nilai data setelah transformasi linear.

x = nilai data aktual.

x_{min} = nilai minimum data aktual keseluruhan.

x_{max} = nilai maksimum data aktual keseluruhan.

a = selang batas awal

b = selang batas akhir

2. Pembagian Data

Langkah selanjutnya setelah transformasi data adalah pembagian data. Data dibagi menjadi data pelatihan, pengujian dan validasi. Beberapa komposisi data pelatihan, pengujian dan validasi yang sering digunakan adalah sebagai berikut.

a. 80% untuk data pelatihan dan 20% untuk data pengujian dan validasi.
b. 70% untuk data pelatihan dan 30% untuk data pengujian dan validasi.
c. 2/3 untuk data pelatihan dan 1/3 untuk data pengujian dan validasi.
d. 50% untuk data pelatihan dan 50% untuk data pengujian dan validasi.
e. 60% untuk data pelatihan dan 40% untuk data pengujian dan validasi.

Aspek pembagian data harus ditekankan agar jaringan mendapat data pelatihan yang secukupnya dan data pengujian dapat menguji prestasi pelatihan yang dilakukan berdasarkan nilai MAPE data pelatihan dan pengujian. Bilangan data yang kurang untuk proses pelatihan akan menyebabkan jaringan mungkin tidak dapat mempelajari taburan data dengan baik. Sebaliknya, data yang terlalu banyak untuk proses pelatihan akan melambatkan poses pemusatan (konvergensi). Masalah *overtraining* (data pelatihan yang berlebihan) akan memyebabkan jaringan cenderung untuk menghafal data yang dimasukan daripada mengeneralisasi.

2.2.1.5.4 Perancangan Struktur Jaringan Yang Optimum

Langkah selanjutnya setelah pembagian data adalah penentuan bilangan simpul masukan, bilangan lapisan tersembunyi, bilangan simpul lapisan tersembunyi dan bilangan simpul keluaran yang akan digunakan dalam jaringan. Terdapat beberapa aturan yang dapat membantu perancangan jaringan yang optimum, yaitu sebagai berikut.

a. Bilangan simpul masukan sama dengan periode dimana data berfluktuasi.
b. Bilangan simpul keluaran sama dengan bilangan keluaran masalah.

c. Mulai dengan satu lapisan tersembunyi dan digunakan lebih dari satu lapisan tersembunyi jika diperlukan.
d. Jika menggunakan satu lapisan tersembunyi, bilangan simpul tersembunyi awal adalah 75% dari bilangan simpul masukan.

Penggunaan jaringan dengan dua atau lebih lapisan tersembunyi dalam masalah peramalan kebanyakan tidak akan memberikan pengaruh yang sangat besar terhadap prestasi jaringan untuk melakukan peramalan. Selain itu akan melambatkan proses pelatihan yang disebabkan bertambahnya simpul.

Beberapa kaedah untuk memperkirakan bilangan simpul tersembunyi yaitu sebagai berikut.

a. h = n, 2n
b. h = n/2

dengan

n = bilangan simpul masukan yang digunakan.

h = bilangan simpul tersembunyi.

Penentuan bilangan simpul tersembunyi yang terbaik diperoleh secara *trial and error* dari simpul 1 sampai 2n.

2.2.2 Metode *Multiple Regression*

Multiple regression adalah regresi dengan dua atau lebih variabel X_1, X_2, X_3, …., Xn sebagai variabel bebas dan Y sebagai variabel tak bebas, sehingga merupakan perluasan dari regresi linier sederhana. Model probabilistik regresi berganda yang melibatkan (k-1) variabel X adalah sebagai berikut (Dadan, 2004) :

$$Y = a + b_1 X_1 + b_2 X_2 + b_3 X_3 + \ldots\ldots b_k X_k \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (18)$$

Y : Variabel terikat

X_1, X_2, X_3,….,X_k = variable bebas

a_1, b_1, b_2,…. b_k = koefisien variable

Pada persamaan linear lebih dari dua variable, variable Y dipengaruhi oleh lebih dari dua variable, yaitu variable X_1, X_2, ….. X_k. Dalam hal demikian, variable Y disebut variabel terikat (*dependent variable*) dan variable-variabel X_1, X_2,…. X_k disebut variable bebas (independent variable), artinya nilai-nilai variable Y dapat ditentukan berdasarkan nilai-nilai dari variable X_1,X_2,….X_k.

Syarat-syarat Regresi Linier Berganda, sebagai berikut :

a. Model regresi linier;
b. Eksistensi (X diasumsikan non stokastik);
c. Nilai rata-rata kesalahan adalah nol, atau Ė(μ/ Xi) = 0;
d. Homoskedastisitas, artinya varian kesalahan sama untuk setiap periode (homo = sama, skedastisitas = sebaran) dinyatakan dalam bentuk matematis: Var (μ/ Xi) = 0;
e. Tidak ada autokorelasi antar kesalahan (antara i dan j tidak ada korelasinya). Dinyatakan dalam bahasa matematis : Covarians (μi , μ j) = 0;
f. Antara μ dan X saling bebas, sehingga covarians (μ i, X) = 0;
g. Tidak ada multikolinieritas yang sempurna antar variabel bebas;
h. Jumlah observasi n harus lebih besar daripada jumlah parameter yang diestimasi (jumlah variabel bebas);
i. Adanya variabilitas dalam nilai X, artinya nilai X harus berbeda (tidak boleh sama semua);

j. Model regresi telah dispesifikasikan secara benar. Dengan kata lain tidak ada bias (kesalahan) spesifikasi dalam model yang digunakan dalam analisis empiris (Kuncoro, 2001:96).

2.2.3 *Forecasting* (Peramalan)

2.2.3.1 Hubungan *Forecast* dengan Rencana

Forecast adalah peramalan apa yang akan terjadi pada waktu yang akan datang, sedang rencana merupakan penentuan apa yang akan dilakukan pada waktu yang akan datang (Subagyo, 1986: 3). Dengan sendirinya terjadi perbedaan antara *forecast* dengan rencana. *Forecast* adalah peramalan apa yang akan terjadi, tetapi belum tentu bisa dilaksanakan oleh perusahaan.

2.2.3.2 Definisi dan Tujuan Forecasting

Forecasting adalah suatu usaha untuk meramalkan keadaan di masa mendatang melalui pengujian keadaan di masa lalu (Handoko, 1984:260). Dalam *kehidupan* sosial segala sesuatu itu serba tidak pasti, sukar untuk diperkirakan secara tepat. Dalam hal ini perlu diadakan *forecast*. *Forecasting* yang dibuat selalu diupayakan agar dapat meminimumkan pengaruh ketidakpastian ini terhadap perusahaan. Dengan kata lain *forecasting* bertujuan mendapatkan *forecast* yang bisa meminimumkan kesalahan meramal (forecast error) yang biasanya diukur dengan mean squared error, mean absolute error, dan sebagainya (Subagyo, 1986: 4).

2.2.3.3 Proses Peramalan

Menurut Handoko (1984: 260), proses peramalan biasanya terdiri dari langkah-langkah sebagai berikut:

a. Penentuan Tujuan

Analisis membicarakan dengan para pembuat keputusan dalam perusahaan untuk mengetahui apa kebutuhan-kebutuhan mereka dan menentukan:

1) variabel-variabel apa yang akan diestimasi,
2) siapa yang akan menggunakan hasil peramalan,
3) untuk tujuan-tujuan apa hasil peramalan akan digunakan,
4) estimasi jangka panjang atau jangka pendek yang diinginkan,
5) derajat ketepatan estimasi yang diinginkan,
6) kapan estimasi dibutuhkan,
7) bagian-bagian peramalan yang diinginkan, seperti peramalan untuk kelompok pembeli, kelompok produk atau daerah geografis.

b. Pengembangan Model

Setelah tujuan ditetapkan, langkah berikutnya adalah mengembangkan model, yang merupakan penyajian secara lebih sederhana sistem yang dipelajari. Dalam peramalan, model adalah suatu kerangka analitik yang apabila dimasukkan data masukan menghasilkan estimasi penjualan di waktu mendatang (atau variabel apa saja yang diramal). Analisis hendaknya memilih suatu model yang menggambarkan secara realistis perilaku variabel-variabel yang dipertimbangkan. Sebagai contoh, bila perusahaan ingin meramalkan jumlah produksi yang polanya berbentuk linier, model yang dipilih mungkin $Y = A + BX$, dimana Y menunjukkan besarnya jumlah produksi ; X menunjukkan unit waktu, serta A dan B adalah parameter-parameter yang menggambarkan posisi dan kemiringan garis pada grafik.

c. Pengujian Model

Sebelum diterapkan, model biasanya diuji untuk menentukan tingkat akurasi, validitas, dan reliabilitas yang diharapkan. Ini sering mencakup penerapannya pada data historis, dan penyiapan estimasi untuk tahun-tahun sekarang dengan data nyata yang tersedia. Nilai suatu model ditentukan oleh derajat ketepatan hasil peramalan data aktual.

d. Penerapan Model

Setelah pengujian, analisis menerapkan model dalam tahap ini, data historis dimasukkan dalam model untuk menghasilkan suatu ramalan. Dalam kasus model penjualan, Y = A + BX, analisis menerapkan teknik-teknik matematika agar diperoleh A dan B.

e. Revisi dan Evaluasi

Ramalan-ramalan yang telah dibuat harus senantiasa diperbaiki dan ditinjau kembali. Perbaikan mungkin perlu dilakukan karena adanya perubahan-perubahan dalam perusahaan atau lingkungannya, seperti tingkat harga produk perusahaan, karakteristikkarakteristik produk, pengeluaran-pengeluaran pengiklanan, tingkat pengeluaran pemerintah, kebijakan moneter dan kemajuan teknologi.

Evaluasi, di pihak lain, merupakan perbandingan ramalan-ramalan dengan hasil nyata untuk menilai ketepatan penggunaan suatu metodologi atau teknik peramalan. Langkah ini diperlukan untuk menjaga kualitas estimasi-estimasi di waktu yang akan datang.

2.2.3.4 Metode-metode Peramalan

Untuk mendapatkan suatu metode yang baik dalam sistem peramalan, digunakan tingkat ketelitian sebagai ukuran, semakin tinggi tingkat ketelitian yang didapatkan maka semakin baik penggunaan dari metode yang dipakai.

Metode peramalan didasarkan pada :

1. Penggunaan analisa pola hubungan antara variabel yang akan dipekirakan.
2. Penggunaan analisa pola hubungan antara variabel yang akan memperkirakan dengan variabel lain yang mempengaruhi.

Metode jaringan saraf tiruan algoritma backpropagation dilatih dalam suatu cara yang disesuaikan melalui cara komputasi. Aplikasi ini pada peramalan didasarkan pada kemampuan jaringan saraf tiruan untuk mendekati keluaran yang diinginkan dengan kombinasi dari variabel parameter yang terlibat untuk mendapatkan hasil yang sesuai dengan pola yang terbentuk.

Di bawah ini adalah daftar beberapa istilah peramalan :

1. *Accurancy* (Ketepatan)

Kriteria yang paling banyak dipakai untuk mengevaluasi unjuk kerja model dan metode-metode peramalan alternative adalah ketepatan. Ia menunjukkan tingkat kebenaran ramalan yang diukur. Ketepatan dapat diukur menggunakan dimensi seperti rata-rata kesalahan kuadrat (MSE), rata-rata kesalahan persentase absolute (MAPE) atau rata-rata kesalahan persentase bias (MPE).

2. Algorithm (Algoritma)

Suatu rangkaian aturan sistematis untuk memecahkan suatu persoalan. Satu set aturan yang digunakan dalam penerapan berbagai metode peramala kuantitatif adalah algoritma.

3. Applicability (Kemampuan untuk diterapkan)

Aplikabilitas merupakan suatu criteria yang penting di dalam metode peramalan. Aplikabilitas menunjukkan mudahnya suatu metode untuk diterapkan dalam situasi tertentu dengan pemakai peramalan yang khusus. Semakin menigkatnya permintaan dan kecanggihan metode peramalan mengikuti seringkali akan mengurangi tingkat aplikabilitas.

4. Back Forecasting (Peramalan mundur)

Dalam menerapkan teknik-teknik peramalan kuantitatif yang berdasarkan nilai-nilai kesalahan lama (*past error*). Diperlukan suatu *starting value* (nilai awal) supaya perhitungan dapat dilaksanakan. Salah satu cara untuk mendapatkan nilai ini adalah dengan menerapkan metode peramalan untuk data yang dimulai dari nilai akhir ke bagian awal data. Prosedur ini dinamakan peramalan mundur.

5. *Cencus II* (Sensus II)

Metode peramalan Sensus II menguraikan suatu deret berkala menjadi komponen musim (seasonal), trend, siklus dan random yang dapat dianalisa secara terpisah kemudian digabung kembali untuk mendapatkan ramalan.

6. *Estimasi* (Penaksiran)

Penaksiran berisi penemuan nilai-nilai yang sesuai untuk parameter2 sebuah persamaan dengan suatu cara sehingga beberapa criteria dapat dioptimasi.

7. File (Berkas)

Sebuah file merupakan kumpulan data yang disusun untuk petunjuk di masa yang akan datang. Jika disimpan dalam komputer, file-file dapat disajikan dengan program komputer actual untuk melakukan meode peramalan atau menyederhanakan data historis yang digunakan dengan program-program komputer tersebut.

8. Heuristic (Heuristik)

Sebuah heuristic merupakan suatu susunan langkah atau prosedur yang menggunakan pendekatan trial dan error untuk mencapai beberapa sasaran yang diinginkan.

9. Input-Output Analysis (Analisis input-output)

Pendekatan ini digunakan untuk merencanakan sebuah analisis yang berkenaan dengan modeling sistem total dalam bentuk hubungan antar beberapa variabel.

10. Mean Absolute Percentage Error (MAPE) (Kesalahan Persentasi Absolut Rata-rata)

Rata-rata kesalahan persentasi absolute merupakan suatu nilai tengah atau rata-rata jumlah seluruh persentasi kesalahan untuk sebuah susunan data yang diberikan. Ia merupakan salah satu ukuran ketepatan yang digunakan dalam metode kuantitatif atau peramalan.

11. Mean Percentage Error(MPE) (Nilai Tengah Persentasi Kesalahan)

Nilai tengah kesalahan persentasi merupakan rata-rata dari seluruh kesalahan persentasi susunan data yang diberikan. Rata-rata ini memungkinkan

kesalahan persentasi positif dan negative untuk saling menunda. Oleh karena itu, kadang-kadang digunakan sebagai suatu ukuran bias dalam aplikasi metode peramalan.

12. Mean Squared Error (MSE) (Nilai Tengah Kesalahan Kuadrat)

Nilai tengah kesalahn kuadrat merupakn suatu ukuran ketepatan perhitungan dengan menguadratkan masing-masing kesalahan untuk masing-masing item dalam sebuah susunan data dan kemudian memperoleh rata-rata atau nilai tengah jumlah kuadrat tersebut. Nilai tengah kesalahn kuadrat memberikan bobot yang lebih besar terhadap kesalahan yang besar daripada kesalahan kecil sebab kesalahn dikuadratkan sebelum dijumlahkan.

13. Observation (Pengamatan)

Pengamatan adalah nilai dari suatu peristiwa tertentu yang ditunjukkan oleh suatu skala pengukuran nilai data tunggal. Dalam sebagian besar aplikasi peramalan sekelompok pengamatan dipakai untuk memberikan data dimana model yang dipilih sesuai.

14. Probability (Probabilitas)

Probablitas dari suatu peristiwa ditunjukkan dengan angka dari 0 sampai dengan 1. Peristiwa yang tidak mungkin terjadi mempunyai probabilitas 0. Suatu peristiwa yang pasti terjadi mempunyai probabilitas 1.

15. Statistic (Statistik)

Suatu statistik adalah bilangan sembarang yang menerangkan sifat-sifat sampel data.

2.2.4 Perancangan Sistem

Tahap perancangan disebut juga tahap pemecahan masalah, yaitu dengan menyusun suatu algoritma, alur sistem, masukan, prosedur proses, keluaran, dan database. Proses perancangan diperlukan untuk menghasilkan suatu rancangan sistem yang baik, karena dengan rancangan yang tepat akan menghasilkan sistem yang stabil dan mudah dikembangkan di masa mendatang. Berikut ini akan dijelaskan rangkaian atau ruang lingkup sistem yang akan dirancang dengan memanfaatkan alat bantu seperti :

2.2.4.1 Diagram Konteks

Diagram konteks menggambarkan hubungan antara sistem dengan entitas luarnya. Diagram konteks berfungsi sebagai transformasi dari satu proses yang melakukan transformasi data *input* menjadi data *output*. Entitas yang dimaksud adalah entitas yang mempunyai hubungan langsung dengan sistem.

Suatu diagram konteks selalu mengandung satu dan hanya satu proses saja. Proses ini mewakili proses dari seluruh sistem. Diagram konteks ini menggambarkan hubungan *input* atau *output* antara sistem dengan dunia luarnya (kesatuan luar).

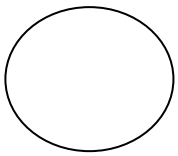
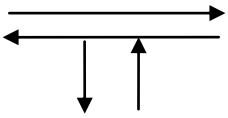
2.2.4.1 Data Flow Diagram

Data Flow Diagram (DFD–DAD/Diagram Alir Data) memperlihatkan hubungan fungsional dari nilai yang dihitung oleh sistem, termasuk nilai masukan, nilai keluaran, serta tempat penyimpanan internal. DAD adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam objek kemudian melewati proses yang mentransformasinya ke tujuan yang lain, yang ada pada objek lain.

DAD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangan lingkungan fisik dimana data tersebut mengalir.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis and design*). DFD merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur jelas.

Tabel 2.1 Diagram Arus Data (*Data Flow Diagram*)

NO	SIMBOL	NAMA	KETERANGAN
1		*External Entity* (kesatuan luar)	Menggambarkan kegiatan proses dari operasi program komputer
2		*Process* (proses)	Menggambarkan kegiatan atau kerja yang dilakukan oleh manusia atau komputer
3		*Data Flow* (arus data)	Menunjukan arus data yang mengalir antara proses, simpanan data dan kesatuan luar dari ke suatu arus

4	—————— ——————	*Data Store* (simpanan luar)	Menggambarkan suatu tempat penyimpanan data

Beberapa simbol yang digunakan dalam *Data Flow Diagram* (DFD) antara lain:

1. *External Entity* (kesatuan luar) atau *boundary* (batas sistem)

Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. *Process* (proses)

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang , mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Untuk *physical data flow diagram* (PDFD), proses dapat dilakukan oleh orang, mesin atau komputer, sedangkan untuk *logical data flow diagram* (LDFD), suatu proses hanya menunjukkan proses dari komputer. Setiap proses harus diberi penjelasan yang lengkap meliputu identifikasi proses, nama proses dam pemroses.

3. *Data Flow* (arus data)

Arus data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data strore*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

4. *Data Store* (simpanan luar)

Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa, yaitu suatu file atau *database* di sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, suatu tabel acuan manual, dan suatu agenda atau buku.