

BAB II

LANDASAN TEORI

2.1 UD Dwi Mulya Plastik Sidoarjo

UD Dwi Mulya Plastik merupakan suatu badan usaha yang bergerak dibidang produksi barang-barang rumah tangga berbahan dasar bijih plastik. Badan usaha ini berlokasi di jl. Kesatriaan no. 33 Sidokepong, RT 28 RW 07 Buduran, Sidoarjo. UD Dwi Mulya Plastik ini didirikan oleh Bapak H.M Sudarto di Sidoarjo pada akhir tahun 2008. Produk-produk yang dihasilkan antara lain : timba cor, pot bunga, wakul telur, serta waskom dengan ukuran 12 dan 13. Saat ini, daerah pemasaran UD Dwi Mulya Plastik meliputi kota-kota di Jawa Timur, dan Bali.

2.2 Aplikasi

Aplikasi merupakan sekumpulan perintah program yang dibuat untuk melakukan pekerjaan-pekerjaan tertentu (khusus). Dalam pembuatan aplikasi *file text* dibutuhkan beberapa komponen seperti *label*, *textbox*, dan beberapa *command*, dimana aplikasi *file text* merupakan sebuah program yang dapat menyimpan *text* atau tulisan ke dalam *extention .txt*. Pada pembuatan aplikasi tentu tersedia menu untuk melakukan *print form*, dimana *print form* sendiri merupakan komponen untuk mencetak *form* ke *file*, *previw* atau *printer*. Komponen tersebut terdapat pada *ToolBox Printing* (Hendrayudi, 2009: 143)

2.3 Permintaan

Menurut Gilarso (2007), dalam ilmu ekonomi istilah permintaan (*demand*) mempunyai arti tertentu, yaitu selalu menunjuk pada suatu hubungan tertentu antara jumlah suatu barang yang akan dibeli orang dan harga barang tersebut. Permintaan adalah jumlah dari suatu barang yang mau dan mampu dibeli pada berbagai kemungkinan harga, selama jangka waktu tertentu, dengan anggapan hal-hal lain tetap sama (=ceteris paribus).

Menurut Danniell (2004), permintaan dipengaruhi oleh beberapa faktor, antara lain:

1. Harga

Hubungan harga dengan permintaan adalah hubungan yang negatif. Artinya bila yang satu naik maka yang lainnya akan turun dan begitu juga sebaliknya. Semua ini berlaku dengan catatan faktor lain yang mempengaruhi jumlah permintaan dianggap tetap.

2. Harga Barang Lain

Terjadinya perubahan harga pada suatu barang akan berpengaruh pada permintaan barang lain. Harga barang lain dapat meliputi harga barang substitusi, komplemen, dan independen. Salah satu contoh barang substitusi, bila harga kopi naik, biasanya permintaan teh akan naik. Barang komplementer contohnya roti dengan keju. Apabila keduanya dipakai secara bersamaan sehingga dengan demikian bila salah satu dari harga barang tersebut naik, pada umumnya akan mempengaruhi banyaknya konsumsi barang komplementernya. Barang independen adalah barang yang tidak dipengaruhi oleh harga barang yang lain.

3. Selera

Selera merupakan variabel yang mempengaruhi besar kecilnya permintaan. Selera dan pilihan konsumen terhadap suatu barang bukan saja dipengaruhi oleh struktur umum konsumen, tetapi juga karena faktor adat dan kebiasaan setempat, tingkat pendidikan, atau lainnya.

4. Jumlah Penduduk

Semakin banyaknya jumlah penduduk makin besar pula barang yang dikonsumsi dan makin naik permintaan. Penambahan jumlah penduduk mengartikan adanya perubahan struktur umur. Dengan demikian, bertambahnya jumlah penduduk adalah tidak proporsional dengan pertambahan jumlah barang yang dikonsumsi.

5. Tingkat Pendapatan

perubahan tingkat pendapatan akan mempengaruhi banyaknya barang yang dikonsumsi. Secara teoretis, peningkatan pendapatan akan meningkatkan konsumsi. Bertambahnya pendapatan, maka barang yang dikonsumsi tidak hanya bertambah kuantitasnya, tetapi kualitasnya juga meningkat.

2.4 Peramalan Permintaan

Peramalan permintaan merupakan tingkat permintaan produk-produk yang diharapkan akan terealisasi untuk jangka waktu tertentu, pada waktu tertentu pada masa yang akan datang. Aktivitas peramalan permintaan ini biasanya dilakukan oleh manajer yang nantinya akan menjadi masukan penting dalam pengambilan keputusan dan perencanaan produksi. Aktivitas peramalan permintaan bukan aktivitas yang bertujuan untuk mengukur permintaan produksi di masa yang akan datang secara pasti, melainkan sekedar usaha untuk

mengurangi kemungkinan terjadinya hal yang berlawanan antara keadaan yang sungguh-sungguh terjadi di kemudian hari dengan apa yang menjadi hasil peramalan. Dengan kata lain, hasil dari aktivitas peramalan adalah melakukan minimalisasi ketidakpastian yang mungkin terjadi di masa yang akan datang.

2.5 Peramalan

2.5.1 Pengertian Peramalan

Definisi peramalan sendiri sebenarnya beragam. Berikut beberapa definisi tentang peramalan (Santoso, 2009: 7):

1. Perkiraan munculnya sebuah kejadian di masa depan berdasarkan data yang ada di masa lampau.
2. Proses menganalisis data historis dan data saat ini untuk menentukan *trend* di masa mendatang.
3. Proses estimasi dalam situasi yang tidak diketahui.
4. Pernyataan yang dibuat tentang masa depan.
5. Penggunaan ilmu dan teknologi untuk memperkirakan situasi di masa depan.
6. Upaya sistematis untuk mengantisipasi kejadian atau kondisi di masa depan.

Dari beberapa definisi diatas, dapat disimpulkan bahwa peramalan berkaitan dengan upaya memperkirakan apa yang terjadi di masa depan, berbasis pada metode ilmiah (ilmu dan teknologi) serta dilakukan secara sistematis. Walaupun demikian, kegiatan peramalan tidaklah semata-mata berdasarkan prosedur ilmiah atau terorganisir karena ada kegiatan peramalan yang menggunakan intuisi (perasaan) atau lewat diskusi informal dalam sebuah grup.

Tabel 2.1 Ciri Sebuah Kegiatan Peramalan

Aspek	Peramalan
Fokus	Data di masa lalu
Tujuan	Menguji perkembangan saat ini dan relevansinya di masa mendatang
Metode	Proyeksi berdasar ilmu statistik, diskusi, dan review program
Orang yang terlibat	Pembuat keputusan, petugas administrasi, praktisi, analisis
Frekuensi	Regular (teratur)
Kriteria Keberhasilan	Tidak sekedar akurasi, namun bersifat pembelajaran

Dari Tabel 2.1 dapat dilihat bahwa peramalan adalah kegiatan yang bersifat teratur. Peramalan juga berupaya memprediksi masa depan dengan menggunakan tidak hanya metode ilmiah, namun juga mempertimbangkan hal-hal yang bersifat kualitatif, seperti perasaan, pengalaman seseorang dan lainnya.

2.5.2 Jenis Data Pada Kegiatan Peramalan

Data yang akan diprediksi secara umum dapat dibagi menjadi dua tipe, yakni: data kualitatif dan data kuantitatif. Data kualitatif adalah data yang bukan merupakan bilangan, tetapi berupa ciri-ciri, sifat-sifat, keadaan, atau gambaran dari kualitas objek yang diteliti. Sedangkan data kuantitatif adalah data yang berupa bilangan yang nilainya bisa berubah-ubah atau bersifat variatif.

Data kuantitatif dapat dibagi menjadi 2 bagian, antara lain: (Santoso, 2009: 13)

1. Data *time series*

Data *times series* adalah data yang ditampilkan berdasarkan waktu, seperti bulanan, data harian, data mingguan atau jenis waktu yang lain. Ciri data *time series* adalah adanya rentang waktu tertentu, dan bukannya data pada satu waktu tertentu.

2. *Data cross-sectional*

Data cross-sectional adalah data yang tidak berdasar waktu tertentu, namun pada satu (titik) waktu tertentu.

2.5.3 Tahapan Peramalan

Menurut Santoso (Santoso, 2009: 10), untuk mendapatkan hasil peramalan yang baik dan dapat menjawab masalah yang ada secara efektif, kegiatan peramalan sebaiknya mengikuti tahapan baku berikut ini :

1. Perumusan masalah dan pengumpulan data

Tahap pertama yang sebenarnya penting dan menentukan keberhasilan peramalan adalah menentukan masalah tentang apa yang diprediksi. Formulasi masalah yang jelas akan menuntun pada ketetapan jenis dan banyaknya data yang akan dikumpulkan. Dapat saja masalah telah ditetapkan, namun data yang relevan tidak tersedia, hal ini akan memaksa diadakannya perumusan ulang atau mengubah metode peramalan.

2. Persiapan data

Setelah masalah dirumuskan dan data telah terkumpul, tahap selanjutnya adalah menyiapkan data hingga dapat diproses dengan benar. Hal ini diperlukan, karena dalam praktek ada beberapa masalah berkaitan dengan data yang telah terkumpul, antara lain :

- a. Jumlah data terlalu banyak
- b. Jumlah data justru sedikit
- c. Data harus diproses terlebih dahulu
- d. Data tersedia namun rentang waktu data tidak sesuai dengan masalah yang ada

e. Data tersedia namun cukup banyak data yang hilang (*missing*), yakni data yang tidak lengkap.

3. Membangun model

Setelah data dianggap memadai dan siap dilakukan kegiatan prediksi, proses selanjutnya adalah memilih model/metode yang tepat untuk melakukan peramalan pada data tersebut.

4. Implementasi model

Setelah metode peramalan ditetapkan, maka model dapat diterapkan pada data dan dapat dilakukan prediksi pada data untuk beberapa periode ke depan.

5. Evaluasi peramalan

Hasil peramalan yang ada kemudian dibandingkan dengan data aktual. Metode peramalan tidak dapat memprediksi data di masa depan secara tepat yang ada adalah ketepatan prediksi. Untuk itu, pengukuran kesalahan peramalan dilakukan untuk melihat apakah metode yang telah digunakan sudah memadai untuk memprediksi data.

2.5.4 Pengukur Kesalahan Peramalan

Berdasarkan Lincoln Arsyad dalam Peramalan Bisnis (1994), notasi dasar peramalan adalah sebagai berikut :

Y_t = Nilai data runtut waktu periode t

\hat{Y}_t = Nilai peramalan dari Y_t

$e_t = Y_t - \hat{Y}_t$ = Residual atau kesalahan peramalan

Salah satu cara untuk mengevaluasi teknik peramalan adalah menggunakan penjumlahan kesalahan tersebut. *Mean Absolute Deviation* (MAD)

mengukur akurasi peramalan dengan merata-ratakan kesalahan peramalan (nilai absolutya).

$$\text{MAD} = \frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)}{n} \quad 1$$

Mean Squared Error (MSE) merupakan metode alternatif dalam mengevaluasi suatu teknik peramalan. Setiap kesalahan residual dikuadratkan, kemudian dijumlahkan dan dibagi dengan jumlah observasi.

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

Mean Absolute Percentage Error (MAPE) dihitung dengan menemukan kesalahan absolut setiap periode, kemudian membaginya dengan nilai observasi pada periode tersebut, dan akhirnya merata-ratakan persentase absolut ini.

$$\text{MAPE} = \frac{\sum_{t=1}^n \frac{(Y_t - \hat{Y}_t)}{Y_t}}{n} \quad 3$$

Mean Percentage Error (MPE) digunakan untuk menentukan apakah suatu metode peramalan bias atau tidak. MPE dihitung dengan cara menemukan kesalahan setiap periode, dan kemudian membaginya dengan nilai sebenarnya pada periode tersebut, dan kemudian merata-ratakan persentase kesalahan tersebut.

$$\text{MPE} = \frac{\sum_{t=1}^n \frac{(Y_t - \hat{Y}_t)}{Y_t}}{n} \quad 4$$

2.6 Jaringan Syaraf Tiruan

2.6.1 Pengertian Jaringan Syaraf Tiruan

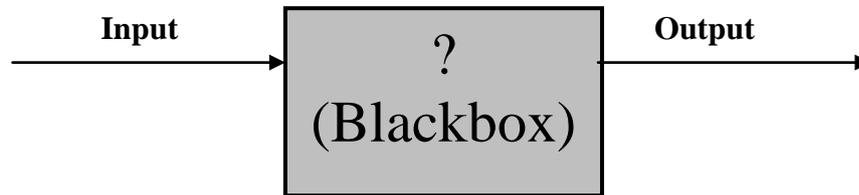
Jaringan saraf tiruan adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi (Siang, 2009: 2). Menurut Irawan (2007: 136) Jaringan syaraf tiruan dibentuk dari generalisasi model matematik jaringan syaraf biologis, berdasarkan asumsi bahwa :

1. Pemrosesan informasi terjadi pada suatu bentuk prosesor sederhana yang disebut *neuron*.
2. Sinyal dilewatkan antar *neuron* melalui sebuah koneksi tertentu.
3. Masing-masing koneksi berasosiasi dengan bobot (*weight*) tertentu.
4. Masing-masing *neuron* menerapkan fungsi aktivasi tertentu pada masing-masing jumlah sinyal *input* untuk menentukan sinyal keluaran .

Jaringan syaraf tiruan ditentukan oleh 3 (tiga) hal (Siang, 2009: 3) :

1. Pola hubungan antar *neuron* (disebut arsitektur jaringan)
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*)
3. Fungsi aktivasi

Jaringan syaraf tiruan dapat dianggap sebagai semacam kotak hitam (*black box*) yang memetakan masukan (*input*) ke suatu keluaran (*output*) tertentu. Kita tidak pernah tahu isi dari kotak hitam tersebut. Sebagai suatu model adaptif yang memiliki kemampuan belajar, jaringan syaraf tiruan mampu membuat generalisasi dan menyimpan hasil belajar tersebut untuk menghasilkan keluaran (*output*) sebagaimana kita harapkan atau tidak kita harapkan tetapi tetap sesuai dengan kaidah-kaidah jaringan syaraf tiruan (Irawan, 2007: 137)



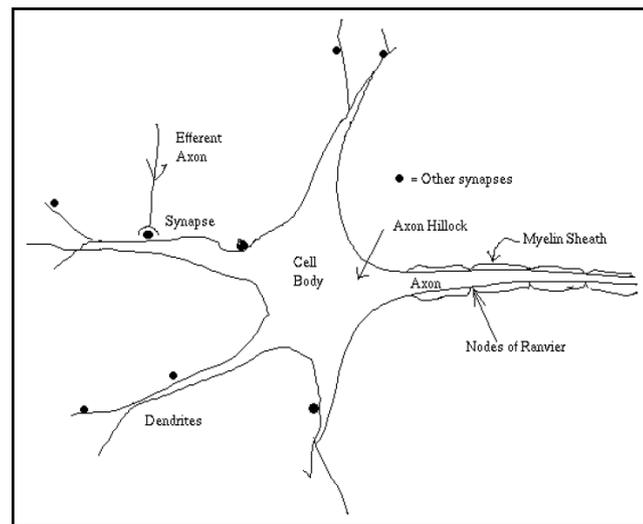
Gambar 2.1 *Artificial Neural Network*

Secara umum berdasarkan algoritma belajar, jaringan syaraf tiruan dapat dibedakan atas :

1. *Supervised learning*, yaitu metode belajar JST dengan pengawasan, dalam hal ini membandingkan suatu masukan dengan suatu *output* tertentu yang sudah ditentukan.
2. *Unsupervised learning*, yaitu metode belajar JST tanpa pengawasan dalam hal ini tidak diperlukan adanya suatu keluaran sebagai acuan.

2.6.2 Model Sel Syaraf

Sistem syaraf biologis merupakan sel-sel syaraf yang disebut *neuron*. Tidak kurang dari 100 milyar neuron mendukung fungsi otak manusia dewasa. Setiap *neuron* berhubungan dengan neuron lainnya melalui sambungan khusus yang disebut dengan sinapsis. Sebuah neuron dapat berhubungan dengan beberapa hingga 10000 *neuron* lain, sehingga masing-masing neuron dapat mengirimkan impuls listrik kepada sel tujuan sampai sekitar 10000 sel (Irawan, 2007: 138)



Gambar 2.2 Fisiologi Sel Syaraf

Sinapsis adalah daerah sambungan khusus antar neuron. Diperkirakan dalam otak manusia terdapat sekitar 1014 sinapsis, yang dipakai dalam komunikasi antar sel. Dalam sistem syaraf, pola interkoneksi sel ke sel beserta fenomena komunikasi antar unit pemroses tersebut menentukan kemampuan komputasi neural (*neurocomputing*). (Dayhoff, 1990: 136)

2.6.3 Pemodelan Sel Syaraf

Variabel bobot, W_{ij} , diambil untuk memodelkan pengaruh modelasi ion Ca^{2+} dan enzim pada sinapsis. Variabel W_{ij} disebut sebagai bobot koneksi yang menghubungkan sel j ke sel i . dengan variabel tersebut, fungsi dari sinapsis, V_p , dapat dimodelkan sebagai fungsi linier seperti:

$$V_p = X_j : W_{ij} \quad 1$$

Dimana X_j adalah potensial aksi (masukan). Selanjutnya dalam dendrit, semua potensial sinapsis yang dihasilkan akan dijumlahkan dan menghasilkan potensial aktivasi untuk sel i :

$$\text{net}_i = \sum_{j=1}^n X_j \cdot W_{ij} \quad 2$$

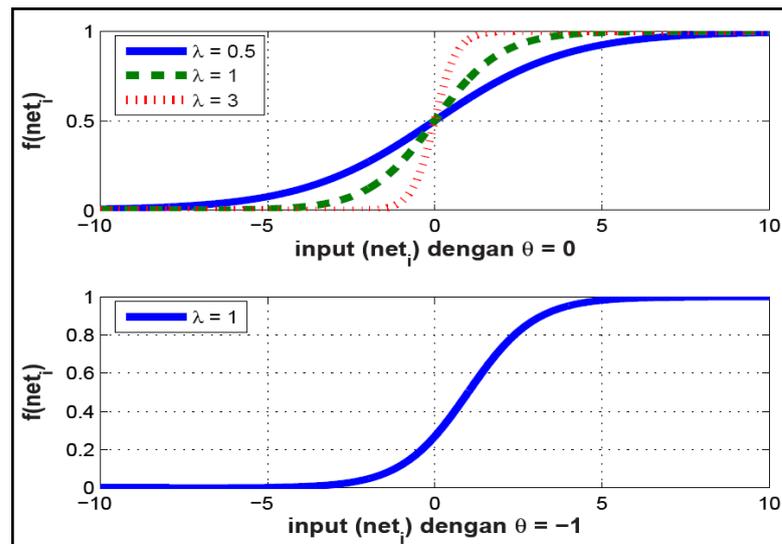
2.6.4 Fungsi Aktivasi

Fungsi aktivasi mentransformasikan semua sinyal masukan (hasil penjumlahan dari dendrit) ke suatu nilai tertentu yang disebut sebagai potensial aksi atau dengan kata lain fungsi aktivasi mentransformasikan nilai aktivasi yang tak terbatas (*infinite*) menjadi terbatas (*finite*) dalam range tertentu. Dalam jaringan syaraf tiruan terdapat bermacam-macam model fungsi aktivasi, antara lain: *binary* dan *bipolar threshold*, *linear threshold*, *binary* dan *bipolar sigmoidal*, dan *gaussian*. Fungsi sigmoid mirip dengan keadaan *neuron* yang sesungguhnya. Karena itu fungsi sigmoid umum dipakai dalam model-model jaringan syaraf tiruan.

1. Fungsi Aktivasi *Binary Sigmoid*

Fungsi aktivasi *binary threshold* dirumuskan dengan:

$$f(\text{net}_i) = \frac{1}{1+e^{-\lambda \text{net}_i}} \quad 3$$

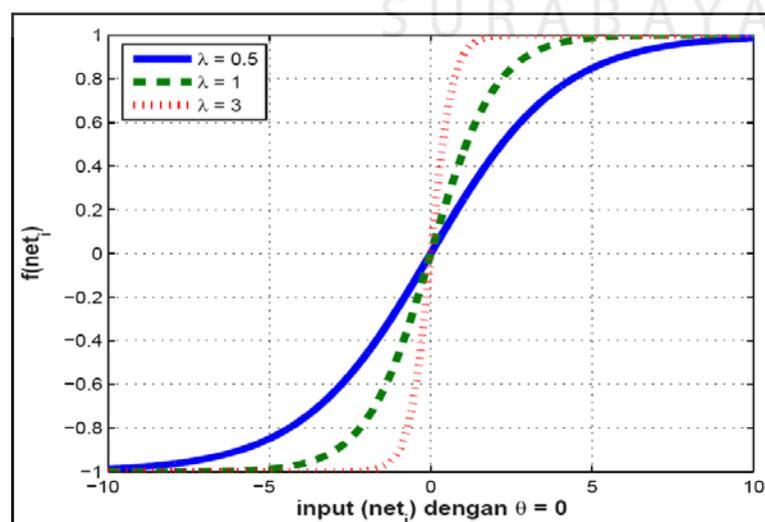
Gambar 2.3 Fungsi Aktifasi *Binary Sigmoid*

2. Fungsi Aktifasi *Bipolar Sigmoid*

Fungsi aktifasi *bipolar sigmoid* dirumuskan dengan:

$$f(\text{net}_i) = 1 - \frac{2}{1 + e^{\lambda \text{net}_i}} \quad 4$$

Dimana lamda adalah faktor penguatan. Fungsi aktifasi *bipolar sigmoid* rentang nilai $f(\text{net}_i)$ yang terletak antara -1 dan 1.

Gambar 2.4 Fungsi Aktifasi *Bipolar Sigmoid*

2.6.5 Metode *Error Back Propagation*

Kelemahan jaringan syaraf tiruan yang terdiri dari layer tunggal membuat perkembangan jaringan syaraf tiruan menjadi terhenti pada sekitar tahun 1970-an. Jaringan syaraf tiruan dengan layer tunggal memiliki keterbatasan dalam pengenalan pola. Kelemahan ini bisa ditanggulangi dengan menambahkan satu/beberapa layer tersembunyi diantara layer masukan dan keluaran (Siang, 2009: 97)

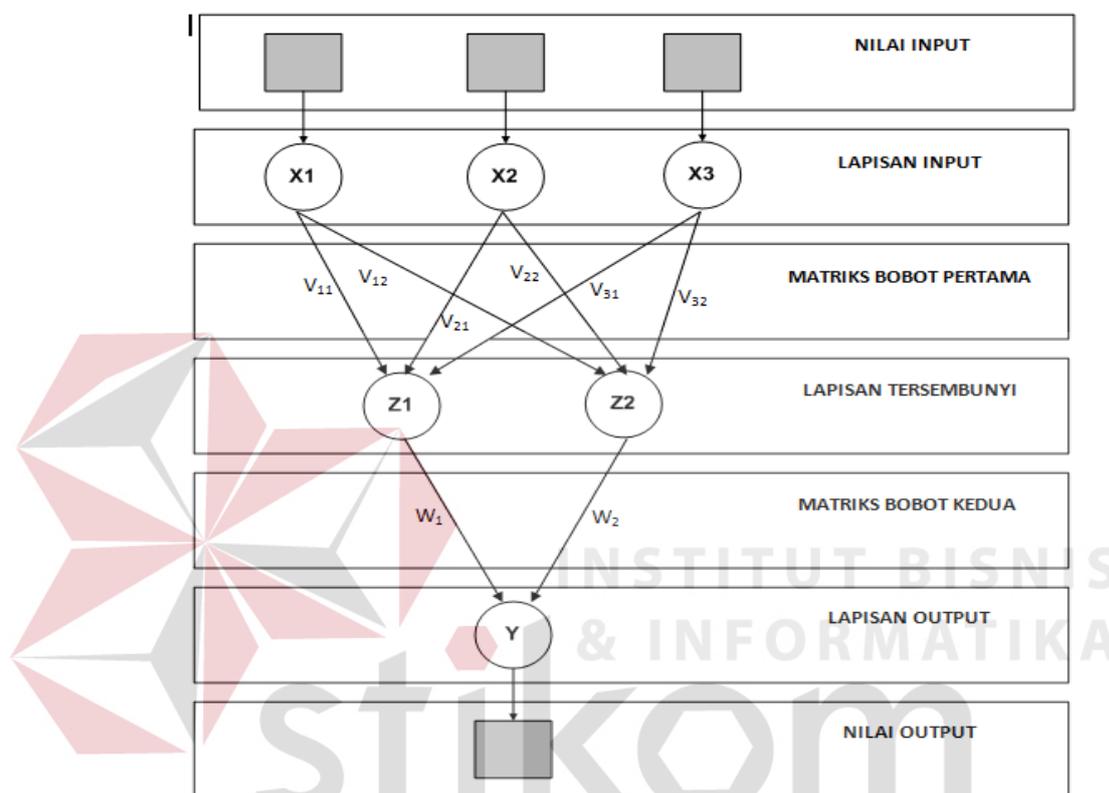
Pembelajaran menggunakan algoritma Delta yang disebut *error back propagation training algorithm*, argumen masukan diumpankan secara arah maju sedangkan proses pembelajaran selain memanfaatkan perambatan arah maju juga memanfaatkan perambatan arah umpan balik.

Pembahasan jaringan syaraf tiruan berlapis banyak dengan algoritma *error backpropagation* dibagi 2 bagian yaitu, membahas algoritma pembelajaran arah maju terlebih dahulu kemudian menyusul pembelajaran arah mundur/balik dengan algoritma *error backpropagation* itu sendiri (Muis, 2006: 166)

Algoritma *error back propagation* memerlukan parameter lain untuk mendukung pemrosesan, yaitu kelajuan pembelajaran c dan pemberian bobot awal. Kelajuan pembelajaran c yang berpengaruh kepada kecepatan proses pembelajaran, bila ditetapkan terlalu kecil akan menyebabkan panjangnya proses pembelajaran sebaliknya penetapan c yang terlalu besar akan menyebabkan keluaran berfluktuasi pada garis target (tidak konvergen) (Muis, 2006: 166)

Secara garis besar, prinsip kerja jaringan syaraf tiruan berlapisan banyak dengan algoritma *error back propagation* adalah sebagai berikut: masukan dan keluaran didefinisikan sebagai y_j dan o_k , indeks j dan k menunjukkan *neuron*

neuron masukan ke 1,2,...,.....J atau *neuron* keluaran ke 1,2,...,.....K, Sedangkan w_{jk} menunjukkan bobot koneksi antara *neuron* masukan j dengan *neuron* keluaran k (Muis, 2006: 167)



Gambar 2.5 Jaringan syaraf tiruan dengan lapisan banyak

Pada Gambar 2.5, lapisan input memiliki 3 unit neuron, yaitu X_1 , X_2 , dan X_3 yang terhubung langsung dengan lapisan tersembunyi (*hidden layer*) yang memiliki 2 unit neuron tersembunyi, yaitu Z_1 dan Z_2 . Hubungan *neuron-neuron* pada lapisan *input* dan lapisan *output* tersebut ditentukan oleh bobot V_{11} , V_{12} , V_{21} , V_{22} , V_{31} , dan V_{32} . Kemudian, 2 unit neuron tersembunyi Z_1 dan Z_2 terhubung langsung dengan lapisan *output* yang memiliki 1 unit *neuron* Y yang besarnya ditentukan oleh bobot W_1 dan W_2 .

Back error propagation (propagasi balik kesalahan) memiliki 2 proses utama yaitu:

1. Propagasi Maju

Pada lapisan dalam (*hidden layer*). Masukan dari setiap node (*neuron*) pada lapisan dalam berasal dari penjumlahan node *input* sebagai berikut:

$$\text{net}_j^h = \sum_{i=1}^{N_i} W_{ji}^h X_i + \theta_j^h \quad 5$$

Sehingga keluaran dari fungsi aktivasi *sigmoid* pada *hidden layer* adalah:

$$H_j = f_j^h(\text{net}_j^h) \quad 6$$

Pada lapisan keluaran (*output layer*). Masukan dari setiap node (*neuron*) pada lapisan keluaran berasal dari penjumlahan node pada lapisan dalam sebagai berikut:

$$\text{net}_k^o = \sum_{j=1}^{N_j} W_{kj}^o H_j + \theta_k^o \quad 7$$

Sehingga keluaran dari fungsi aktivasi *sigmoid* pada *hidden layer* adalah:

$$O_k = f_k^o(\text{net}_k^o) \quad 8$$

Sampai disini proses propagasi maju telah selesai. Selanjutnya hasil keluaran pada *output layer* dibandingkan dengan target, apabila terdapat kesalahan (selisih antara target dan keluaran aktual) maka nilai kesalahan tersebut dipropagasikan balik dengan tujuan untuk meng-*update* seluruh bobot yang ada pada *multilayer perceptron*.

2. Propagasi Mundur

Pada lapis keluaran, kesalahan pada sebuah node keluaran didefinisikan sebagai:

$$\delta_k = Y_k - O_k \quad 9$$

Dimana Y_k adalah nilai target/acuan atau nilai keluaran yang diinginkan dan O_k adalah keluaran aktual dari multilayer perceptron. Maka *error square* pada lapis keluaran adalah :

$$\begin{aligned} E &= \frac{1}{2} \sum_{k=1}^{N_k} \delta_k^2 \\ &= \frac{1}{2} \sum_{k=1}^{N_k} (Y_k - O_k)^2 \end{aligned} \quad 10$$

Proses updating dari nilai bobot dimulai dengan cara meminimalkan turunan dari *error* tersebut diatas terhadap variabel bobot pada node keluaran:

$$\frac{\partial E}{\partial W_{kj}^o} = -(Y_k - O_k) \frac{\partial f_k^o}{\partial (net_k^o)} \cdot \frac{\partial (net_k^o)}{\partial W_{kj}^o} \quad 11$$

Sisi kanan dari persamaan diatas dapat dijabarkan sebagai berikut:

$$\begin{aligned} \frac{\partial (net_k^o)}{\partial W_{kj}^o} &= \left[\frac{\partial}{\partial W_{kj}^o} \sum_{j=1}^{N_j} W_{kj}^o H_j + \theta_k^o \right], \\ &= H_j \end{aligned} \quad 12$$

Dengan demikian persamaan (proses updating) dapat ditulis kembali menjadi:

$$-\frac{\partial E}{\partial W_{kj}^o} = (Y_k - O_k) f_k^{o'}(net_k^o) \cdot H_j \quad 13$$

Dari persamaan 16 dapat diketahui bahwa syarat agar persamaan perubahan bobot dapat dipecahkan, fungsi aktivasi sel harus bersifat dapat diturunkan (*diflerentiable*). Persamaan diatas juga berarti bahwa parameter jaringan yaitu bobot-bobot koneksi dari sel j ke sel i, W_{ij} , harus diubah sebanding dengan negatif gradien fungsi kesalahan terhadap perubahan bobot:

$$\Delta W_{kj}^o = -\eta \cdot \frac{\partial E}{\partial W_{kj}^o} \quad 14$$

Dimana ΔW_{kj}^o adalah perubahan bobot node j ke node k, dan η adalah konstanta kecepatan belajar ($0 \leq \eta \leq 1$). Dengan demikian persamaan diatas dapat dimodifikasi menjadi:

$$\Delta W_{kj}^o = \eta(Y_k - O_k) f_k^{o'}(net_k^o) \cdot H_j \quad 15$$

Selanjutnya bobot yang baru pada lapis keluaran dapat diupdate dengan persamaan:

$$W_{kj}^o(new) = W_{kj}^o(old) + \Delta W_{kj}^o \quad 16$$

Untuk menyederhanakan persamaan, diambil konstanta baru, ∂_k^o , yaitu:

$$\partial_k^o = (Y_k - O_k) f_k^{o'}(net_k^o) = \delta_k f_k^{o'}(net_k^o) \quad 17$$

Sehingga persamaan (bobot selanjutnya) dapat ditulis kembali menjadi:

$$W_{kj}^o(new) = W_{kj}^o(old) + \eta \partial_k^o \cdot H_j \quad 18$$

Kesalahan (*error*) dipropagasi ke lapisan dalam untuk mengupdate nilai bobot pada lapisan dalam. Persamaan penyesuaian bobot pada lapis dalam adalah:

$$W_{ji}^h(new) = W_{ji}^h(old) + \eta \partial_j^h \cdot X_i \quad 19$$

Dimana nilai dari ∂_j^h pada persamaan 22 adalah:

$$\partial_j^h = f_j^{h'}(net_j^h) \sum_{k=1}^{N_k} \partial_k^o \cdot W_{kj}^o \quad 20$$

Secara ringkas, algoritma *Back Error Propagation* (BEP) yang diimplementasikan pada multilayer perceptron dapat dirumuskan kedalam 5 tahapan berikut ini:

1. Inisialisasi bobot (weights) dengan bilangan acak.
2. Jika kondisi berhenti tidak terpenuhi, lakukan 2 – 9.
3. Untuk masing-masing pasangan pelatihan, lakukan langkah 3 – 8.
4. Masing-masing unit input menerima sinyal input X_i dan menyebarkan ke semua unit pada lapis di depannya.
5. Pada masing-masing unit lapis dalam jumlahkan dengan

$$net_j^h = \sum_{i=1}^{N_i} W_{ji}^h X_i + \theta_j^h \quad 21$$

Hitung fungsi aktivasi pada setiap unit lapis dalam:

$$H_j = f_j^h(net_j^h) \quad 22$$

6. Pada masing-masing unit lapis output jumlahkan dengan:

$$net_k^o = \sum_{j=1}^{N_j} W_{kj}^o H_j + \theta_k^o . \quad 23$$

Hitung fungsi aktivasi pada setiap unit lapis keluaran:

$$O_k = f_k^o (net_k^o) \quad 24$$

7. Hitung error pada masing-masing unit lapis output:

$$\partial_k^o = (Y_k - O_k) f_k^{o'} (net_k^o) \quad 25$$

Hitung kenaikan nilai bobot:

$$\Delta W_{kj}^o = \eta \partial_k^o net_j^h \quad 26$$

Hitung kenaikan nilai bobot bias:

$$\Delta W_{0j}^o = \eta \partial_k^o \quad 27$$

$$\text{Dimana } f_k^{o'} = f_k^o (net_k^o) [1 - f_k^o (net_k^o)] \quad 28$$

8. Lakukan updating semua bobot antara lapis dalam dan lapis output:

$$W_{kj}^o (new) = W_{kj}^o (old) + \Delta W_{kj}^o \quad 29$$

9. Lakukan updating semua bobot antara lapis input dan lapis dalam:

$$W_{ji}^h (new) = W_{ji}^h (old) + \Delta W_{ji}^h \quad 30$$

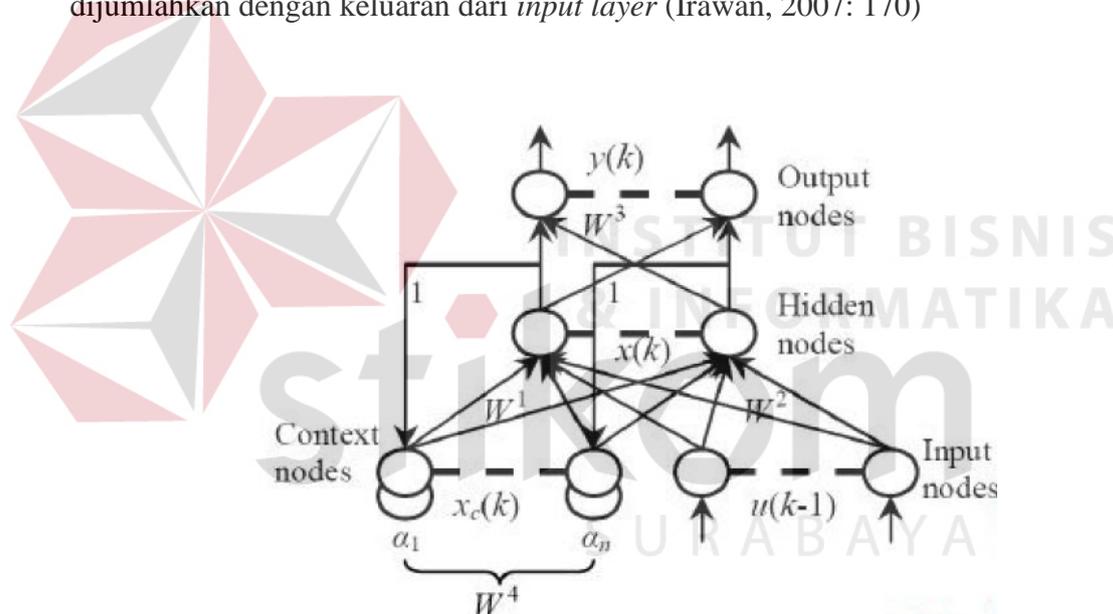
10. Uji kondisi berhenti:

$$E = \frac{1}{2} \sum_{k=1}^{N_k} (Y_k - O_k)^2 \quad 31$$

Proses pelatihan dihentikan apabila kondisi berhenti telah sesuai dengan ketelitian yang kita harapkan.

2.7 Improved Elman

Pada dasarnya proses pelatihan/pembelajaran/*training* pada jaringan elman tidak berbeda dengan proses training yang menggunakan *back error propagation*. Seluruh nilai dari bobot (*weight*) di-*update* dengan menggunakan algoritma *back error propagation*. Jaringan elman memiliki umpan balik yang menghubungkan keluaran dari *hidden layer* dengan *context unit*. Pada saat t , *context unit* akan menerima input yang berasal dari keluaran *hidden layer* pada saat $t - 1$, memproses dan mengirimkan kembali ke *hidden layer* untuk dijumlahkan dengan keluaran dari *input layer* (Irawan, 2007: 170)



Gambar 2.6 Jaringan Syaraf Tiruan dengan arsitektur Elman

Para ahli telah membuktikan bahwa jaringan elman (klasik) hanya akan mencapai hasil yang baik untuk sistem orde pertama saja, sedangkan untuk sistem dengan orde yang lebih tinggi, jaringan elman klasik tidak memberikan hasil yang memuaskan. Untuk memecahkan masalah ini, maka *context unit* harus ditingkatkan untuk memperbaiki jaringan elman klasik tersebut. Pada metode improved elman, faktor konstanta α diubah menjadi vektor w . Metode improved

elman akan menghasilkan suatu performa yang meningkat dan efisiensi didalam pembelajaran yang didapat dari iterasi juga meningkat (Yusanto, 2009: 28)

Algoritma dari metode Improved Elman adalah sebagai berikut:

1. Menghitung output dengan menggunakan persamaan :

$$x(k) = f(w^1 x_c(k) + w^2 u(k-1)) \quad 1$$

$$y(k) = g(w^3 x(k)) \quad 2$$

$$x_c(k) = w^4 x_c(k-1) + x(k-1) \quad 3$$

2. Menghitung gradien $\frac{\partial x_j(k)}{\partial w_{jl}^1}$ dengan menggunakan persamaan:

$$\frac{\partial x_j(k)}{\partial w_{jl}^1} = f_j(\cdot) x_1(k-1) + w_j^4 \frac{\partial x_j(k-1)}{\partial w_{jl}^1} \quad 4$$

Dimana $k = 0, \frac{\partial x_j(k)}{\partial w_{jl}^1} = 0$

3. Dari hasil perhitungan gradien diatas, maka dapat dihitung perubahan bobot dengan menggunakan persamaan:

$$E(k) = \frac{1}{2} (y_d(k) - y(k))^T (y_d(k) - y(k)) \quad 5$$

$$\Delta w_{jl}^4 = \eta_4 \delta_j^h \sum_{i=1}^n (w_{ji}^1) x_i(k-2), j = 1, 2, \dots, n; l = 1, 2, \dots, n \quad 6$$

$$\Delta w_{ij}^3 = \eta_3 \delta_i^o x_j(k), i = 1, 2, \dots, m, j = 1, 2, \dots, n \quad 7$$

$$\Delta w_{jq}^2 = \eta_2 \delta_j^h u_q(k-1), j = 1, 2, \dots, n; q = 1, 2, \dots, n \quad 8$$

$$w_{jl}^1 = \eta_1 \sum_{i=1}^m (\delta_i^o w_{ij}^3) \frac{\partial x_j(k)}{\partial w_{jl}^1}, j = 1, 2, \dots, n; l = 1, 2, \dots, n \quad 9$$

4. Melakukan *adjustment* bobot menggunakan persamaan:

$$w^i = w^i + \Delta w^i, i = 1,2,3,4 \quad 10$$

Langkah 1 sampai dengan langkah 4 diulang sampai didapat *error* lebih rendah atau sama dengan nilai *error* yang ditentukan.

2.8 Unified Modelling Language (UML)

Menurut Nugroho (2005:16), pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Notasi-notasi baku sangat penting demi suatu alasan komunikasi. Dengan notasi-notasi pemodelan yang bersifat baku komunikasi yang baik akan terjalin dengan mudah antar anggota tim pengembang sistem/perangkat lunak dan antara anggota tim pengembang dengan para pengguna. Untuk melakukan pemodelan sistem/perangkat lunak, dalam buku ini notasi-notasi *Unified Modeling Language* (UML) yang akan digambarkan secara elektronik (dengan bantuan komputer) lewat sarana perangkat lunak. Dengan pemodelan menggunakan UML ini, pengembang dapat melakukan:

1. Tinjauan umum bagaimana arsitektur sistem secara keseluruhan.
2. Penelaahan bagaimana objek-objek dalam sistem saling mengirim pesan (*message*) dan saling bekerjasama satu sama lain.
3. Menguji apakah sistem/perangkat lunak sudah berfungsi seperti yang seharusnya.
4. Dokumentasi sistem/perangkat lunak untuk keperluan-keperluan tertentu di masa yang akan datang.

2.8.1 Diagram-diagram *Unified Modelling Language* (UML)

Menurut Sholiq (2010: 19), UML menyediakan beberapa diagram visual yang menunjukkan berbagai aspek dalam sistem. Banyaknya diagram tersebut dimaksudkan untuk memberikan gambaran yang lebih terintegrasi terhadap sistem yang akan dibangun. Beberapa diagram yang disediakan dalam UML antara lain:

1. Diagram *Use Case* Bisnis,

Diagram ini digunakan untuk mempresentasikan bisnis yang dilakukan organisasi. Diagram ini menjawab pertanyaan: “apa yang bisnis lakukan?” dan “mengapa kita membangun sistem untuk itu?”. Diagram ini digunakan untuk memodelkan aktivitas bisnis organisasi sebagai landasan pembuatan use case sistem.

2. Diagram *Use Case*,

Diagram ini menyajikan interaksi antara use case dan aktor dalam sistem yang akan dikembangkan. Use case adalah fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi oleh sistem yang akan dikembangkan tersebut menurut pandangan pemakai sistem.

3. Diagram Aktivitas,

Diagram ini mendefenisikan dari mana *workflow* dimulai, dimana *workflow* berakhir, aktivitas apa saja yang terjadi didalam *workflow*, dan apa saja yang dilakukan saat sebuah aktivitas terjadi.

4. Diagram Sekuensial,

Diagram ini digunakan untuk menunjukkan alur (*flows*) fungsionalitas yang melalui sebuah *use case* yang disusun dalam urutan waktu.

5. Diagram Kolaborasi,

Diagram ini menunjukkan informasi yang sama persis dengan diagram sekuensial, tetapi dalam bentuk dan tujuan yang berbeda. Pada diagram ini, interaksi antar obyek atau aktor ditunjukkan dengan arah panah tanpa keterangan waktu.

6. Diagram Kelas,

Diagram ini menunjukkan interaksi antar kelas-kelas dalam sistem. Kelas juga dianggap sebagai cetak biru dari obyek-obyek didalam sistem. Sebuah kelas mengandung informasi (*attribute*) dan tingkah laku (*behavior*) yang berkaitan dengan informasi tersebut.

7. Diagram *Statechart*,

Diagram ini memungkinkan untuk memodelkan bermacam-macam *state* yang mungkin dialami oleh obyek tunggal. Diagram ini digunakan untuk menggambarkan perilaku dinamik sebuah obyek tunggal.

8. Diagram Komponen, dan

Diagram ini menunjukkan komponen apa saja yang dibutuhkan saat proses kompilasi dan menampilkan komponen runtime apa saja yang dihasilkan sebagai hasil proses kompilasi.

9. Diagram *Deployment*.

Diagram ini menampilkan rancangan fisik jaringan tempat berbagai komponen akan diletakkan. Hanya ada satu diagram *deployment* satu untuk sistem yang sedang dibangun.

2.8.2 Elemen-elemen Pemodelan Bisnis

Menurut Sholiq (2010: 55), Elemen-elemen yang digunakan untuk membuat model bisnis adalah sebagai berikut:

1. Aktor Bisnis

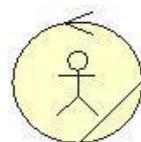
Aktor bisnis adalah seseorang atau sesuatu yang ada diluar organisasi. Orang-orang didalam organisasi, tetapi diluar bagian yang dimodelkan dapat disebut aktor bisnis. Aktor bisnis dimodelkan dengan menggunakan ikon berikut:



Gambar 2.7 Notasi aktor bisnis dalam UML

2. Pekerja Bisnis

Pekerja bisnis adalah suatu peran didalam organisasi, bukan posisi atau jabatan. Pekerja bisnis dimodelkan dengan menggunakan ikon berikut:



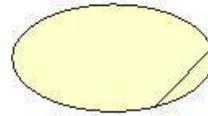
Pekerja Bisnis

Gambar 2.8 Notasi pekerja bisnis dalam UML

3. Use Case Bisnis

Use case bisnis adalah model yang digunakan untuk menggambarkan sebuah proses bisnis organisasi. Use case bisnis menginformasikan tentang aktivitas

bisnis utama yang organisasi lakukan. Use case bisnis dimodelkan dengan menggunakan ikon berikut:



Use Case Bisnis

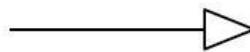
Gambar 2.9 Notasi use case bisnis dalam UML

4. Relasi Asosiasi dan Generalisasi

Ada 2 relasi yang mungkin terjadi pada pemodelan bisnis dengan UML, yaitu: asosiasi dan generalisasi. Relasi asosiasi adalah relasi antara aktor bisnis atau pekerja bisnis dan use case bisnis. Sedangkan relasi generalisasi digunakan ketika ada dua atau lebih aktor bisnis, pekerja bisnis, atau use case bisnis yang sangat serupa. Relasi asosiasi dan generalisasi dimodelkan dengan menggunakan ikon berikut:



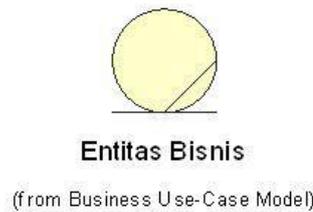
Gambar 2.10 Notasi relasi asosiasi dalam UML



Gambar 2.11 Notasi relasi generalisasi dalam UML

5. Entitas Bisnis

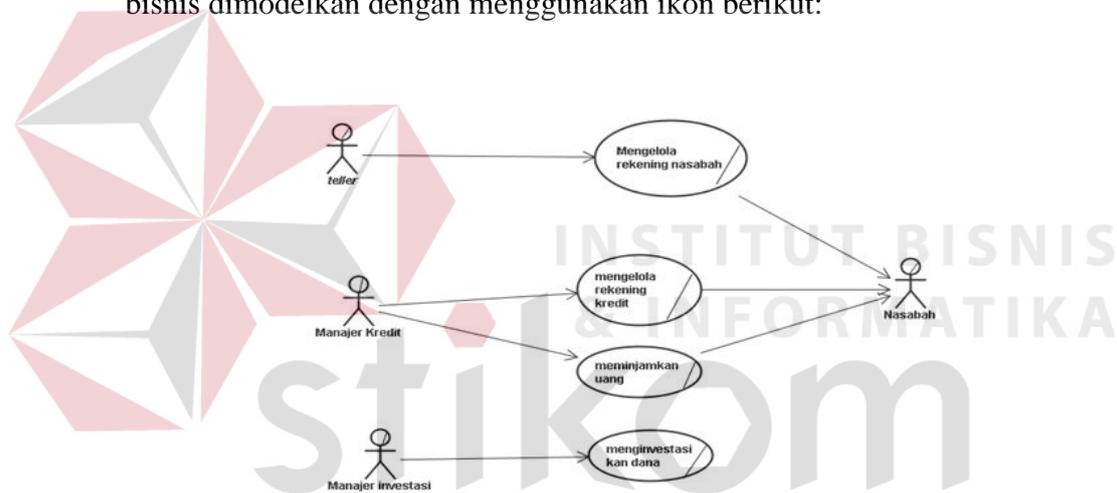
Entitas bisnis adalah obyek yang digunakan atau yang dihasilkan oleh organisasi saat melakukan aktivitas bisnis. Entitas bisnis dimodelkan dengan menggunakan ikon berikut:



Gambar 2.12 Notasi entitas bisnis dalam UML

6. Diagram *Use Case* Bisnis

Diagram *use case* bisnis menunjukkan interaksi antara aktor bisnis atau pekerja bisnis dan *use case* bisnis dalam sebuah organisasi. Diagram *use case* bisnis dimodelkan dengan menggunakan ikon berikut:

Gambar 2.13 Notasi diagram *use case* bisnis dalam UML

2.9 *Black Box Testing*

Menurut Rizky (2011), *black box testing* adalah tipe pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya. Para penguji memandang perangkat lunak seperti layaknya sebuah “kotak hitam” yang tidak penting dilihat isinya tapi cukup dikenai proses pengujian di bagian luar. Jenis pengujian ini hanya memandang perangkat lunak dari sisi spesifikasi dan kebutuhan yang telah didefinisikan pada saat awal perancangan.

Beberapa keuntungan yang diperoleh dari jenis pengujian ini antara lain:

1. Anggota tim penguji tidak harus dari seseorang yang memiliki kemampuan teknis di bidang pemrograman.
2. Kesalahan dari perangkat lunak ataupun *bug* sering ditemukan oleh komponen penguji yang berasal dari pengguna.
3. Hasil dari *black box testing* dapat memperjelas kontradiksi ataupun kerancuan yang mungkin timbul dari eksekusi sebuah perangkat lunak.
4. Proses pengujian dapat dilakukan lebih cepat dibandingkan *white box testing*.

