BAB 2
LANDASAN TEORI

Persaingan yang semakin ketat dan tuntutan pelanggan yang semakin tinggi terhadap kualitas mendorong sektor industri untuk terus meningkatkan pemantauan dan optimisasi terhadap proses produksinya. Kunci sukses pemantauan dan optimisasi proses adalah dengan mengidentifikasi variabel *input* dan *output* yang kritis dan memahami interaksi antar keduanya (Rodriguez & Tobias, 2007).

Berbagai metode telah dikembangkan untuk mengidentifikasi pola dan hubungan antar variabel di dalam suatu set data, salah satunya adalah *data mining*. Dalam bab ini akan dijelaskan konsep dasar *data mining* dan teknik yang digunakan dalam penelitian ini yaitu *Frequency Analysis (FA), Principal Component Analysis (PCA)* dan *Neural Network* (*NN*).

2.1 Konsep Dasar *Data Mining*

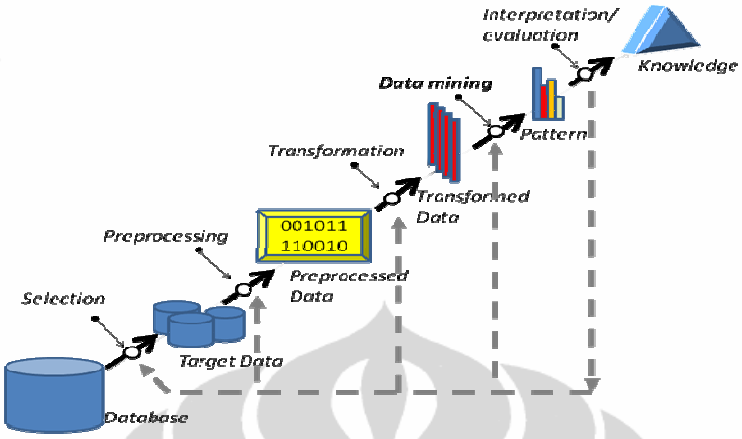
2.1.1 Pengertian *Data Mining*

George Marakas *(2003)* dan Larry Adams (2002) mendefinisikan *Data Mining* (DM) sebagai bagian dari proses *Knowledge Discovery in Database (KDD*) yang menggunakan analisis statistik dan teknik pemodelan, untuk mengungkap pola-pola dan hubungan tersembunyi dalam suatu database yang sangat besar (Feng, 2003). StatSoft Inc.(2008) mendefinisikan DM sebagai proses analitik yang didesain untuk mengeksplorasi data yang berjumlah sangat besar guna mencari pola-pola yang konsisten dan atau hubungan sistematik antar variabel dan kemudian memvalidasi temuan dengan menerapkan model (pola-pola yang terdeteksi) pada subset data yang baru. Proses DM dapat dilakukan secara automatis atau manual (Kantardzic, 2003).

Menurut Wang (2007), DM adalah *data-driven* dan *knowledge-extracted*. Oleh karena itu DM berbeda dengan statistik tradisional, *On-line Analytical Processing* (OLAP) atau pun query yang merupakan *user-driven* dan *information-related*.

Data mining..., Kholisul Fatikhin, FT UI, 2010

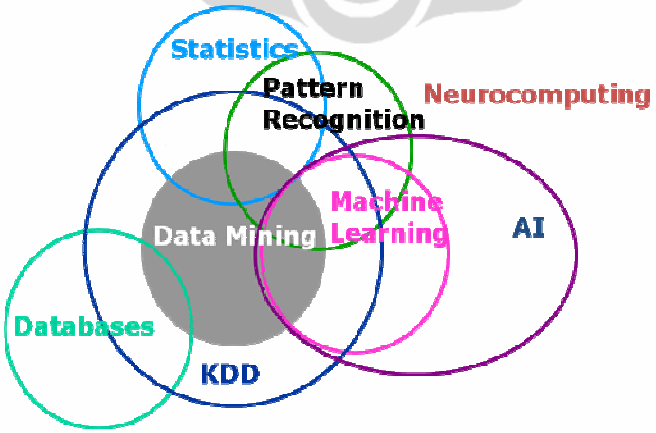
Data mining merupakan salah satu langkah dari keseluruhan proses KDD seperti tampak pada gambar 2.1 di bawah ini.



Gambar 2.1 Posisi *Data Mining* dalam Proses KDD

(sumber: Platon & Amazouz., 2007)

Menurut Berry & Linoff (1997), Hirji (2001), Smyth, Pregibon & Falautsos (2002), DM merupakan bidang yang multi-disiplin yaitu berhubungan dengan disiplin ilmu lainnya seperti *Artificial Intelligence*, *Database*, *Data Visualization*, *Mathematics*, *Operation Research*, *Pattern Recognition dan Statistics* (Feng, 2003). Hubungan DM dan disiplin ilmu lainnya dapat dilihat pada gambar 2.2 di bawah ini.



Gambar 2.2 Hubungan *Data Mining* dan Disiplin Ilmu Lainnya

(sumber: SAS Institute Inc, 2002)

2.1.2 Teknik *Data Mining*

Berdasarkan fungsinya DM dapat dibedakan menjadi dua kategori utama yaitu *discovery DM* dan *predictive DM*. *Discovery DM* digunakan untuk menemukan pola-pola di dalam set data tanpa ada pengetahuan sebelumnya mengenai pola yang ada di dalam data tersebut. Teknik yang digunakan seperti *clustering*, *link analysis*, *frequency analysis* dan sebagainya. *Predictive DM* digunakan untuk menemukan hubungan antara suatu variabel spesifik (dinamakan variabel target) dengan variabel-variabel lainnya. Teknik yang digunakan seperti *classification*, *value prediction*, *assosiation rule* dan sebagainya (Wang, 2007)

Beberapa teknik DM yang paling umum digunakan antara lain: (Kantardzic, 2003; Wang, 2005)

- Metode statistika klasik yaitu *linier, quadratic* dan *logistic discriminate analyses.*
- Teknik statistika modern yaitu *projection pursuit classification, density estimation, k-nearest neighbor, Bayesian networks.*
- *Artificial Neural Network (ANN),* yaitu model matematis yang meniru atau mensimulasikan struktur dan aspek fungsi dari jaringan saraf biologis
- *Support Vector Machine (SVM),* yaitu rangkaian metode *supervised learning* yang digunakan untuk klasifikasi dan regresi.
- *Decision Trees (DT),* yaitu tool pendukung suatu keputusan yang menggunakan grafik seperti pohon atau model keputusan yang terdiri dari konsekuensi-konsekuensi.
- *Association Rules (AR),* yaitu suatu metode riset untuk menemukan hubungan yang menarik antar variabel dalam suatu database yang besar.
- *Case Based Reasoning (CBS)*, yaitu proses untuk memecahkan suatu masalah baru berdasarkan solusi dari masalah-masalah masa lalu yang mirip
- *Fuzzy Logic System (FLS)*, yaitu sebuah bentuk dari logika nilai ganda yang terkait dengan kesimpulan dari suatu alasan (*reasoning*) secara pendekatan. Logika fuzzy mempunyai nilai kebenaran diantara 0 dan 1.
- *Genetic Algorithms (GA)*, yaitu algoritma pencarian *heuristic* yang meniru proses evolusi alam (genetika), untuk mendapatkan solusi yang optimum

Data mining..., Kholisul Fatikhin, FT UI, 2010

Dengan bertambahnya kompleksitas dari system, maka beberapa teknik DM digunakan secara bersama-sama dalam suatu penelitian. Beberapa peneliti menggunakan gabungan teknik DM ini untuk mendapatkan kelebihan dari masing-masing teknik diantaranya : (Paton & Amazouz, 2007)

- Hall Barbosa et al (2002) menggunakan *Bayesian Neural Network* untuk memprediksi kualitas dari produk destilasi untuk *REPAR refinery* di Brazil.
- Zhou (2004) mengembangkan model NN untuk memonitor proses, deteksi kegagalan (*fault*) dan skema klasifikasi pada *batch* reaktor polimerisasi dalam proses produksi *polymethylmethacrylate*. *Feedforward* NN digunakan untuk memodelkan proses dan *radial basis function* (RBF) NN digunakan untuk klasifikasi. Zhou menggunakan regresi *polynomial* untuk mereduksi dimensi dari model NN.
- Zamprogna et al (2005) mengembangkan model berdasarkan PCA dan *Partial Least Squares* (PLS) untuk memonitor proses dan untuk mendeteksi ubnormality pada proses penuangan logam (*steel casting*). PCA/PLS digunakan untuk mengidentifikasi korelasi data dalam kondisi normal. Model memberikan pemahaman yang mendalam mengenai interaksi antar parameter proses sehingga dapat digunakan untuk mendeteksi kegagalan (*ubnormality*) di dalam proses.
- Ahvenlamp et al (2005) menggunakan kombinasi NN dan *fuzzy logic* untuk memprediksi nomor Kappa dan untuk memonitor perubahan di dalam variabel proses, untuk mendeteksi kegagalan dan untuk maksud klasifikasi. Dilaporkan bahwa kombinasi ini mempunyai performa prediksi yang baik dan dapat mendeteksi perilaku *abnormal* bahkan ketika deviasinya kecil.

2.1.3 *Data Mining* dan Pengendalian Proses

Pemantauan (*monitoring*) proses menjadi bagian yang penting untuk mendapatkan produk yang terjaga kualitasnya. Dengan dilakukannya pemantauan maka variasi atau penyimpangan proses yang terjadi dapat segera diketahui sehingga bisa segera dikoreksi.

Data mining..., Kholisul Fatikhin, FT UI, 2010

Deming (1975) mengusulkan penggunaan metode statistik untuk mencari penyebab variasi yang disebut dengan *Statistical Process Control* (*SPC*). Kelemahannya, *SPC* hanya bisa dipakai untuk sejumlah kecil variabel. Faktor-faktor lain yang mempengaruhi variabel kualitas seperti parameter mesin dan peralatan tidak bisa dijelaskan dengan *SPC* (Kang, Choe & Park, 1999). Hasil statistik yang demikian ini tidak cukup digunakan untuk analisis proses kontrol industri modern (Hinckley & Barkan, 1995; Kang, Choe & Park, 1999; Adams, 2001)

Pada umumnya proses kontrol industri modern menggunakan banyak sensor untuk memonitor variabel proses. Masing-masing sensor ini akan menghasilkan ribuan data pengukuran setiap jamnya. Data yang jumlahnya ribuan ini tidak dapat dianalisis menggunakan metode tradisional (Fayyad & Stolorz, 1997). Teknik DM menjadi solusi yang tepat untuk menangani data yang besar dan multi dimensional (Fayyad & Uthurusamy, 2002).

Teknik DM yang digunakan dalam penelitian ini adalah *Frequency Analysis* (FA), *Principal Component Analysis* (PCA) dan *Neural Network* (NN). Berikut akan dijelaskan masing-masing teknik tersebut.

2.2 *Frequency Analysis* (FA)

Data yang diperoleh dari sensor yang dipasang pada proses produksi pada umumnya merupakan data kontinyu yang nilainya dapat berubah dari waktu waktu. Variasi yang terjadi dapat bersifat acak dan dapat pula berulang dengan periode tertentu. Untuk mengetahui ada tidaknya pola pengulangan variasi pada suatu set data runtun waktu (*time series*) maka digunakan algoritma *Fast Fourier Transform* (FFT).

Algoritma FFT digunakan untuk mengidentifikasi komponen-komponen frekuensi yang menyusun suatu signal runtun waktu. Misalkan x adalah data runtun waktu, maka transformasi Fourier dari fungsi x dapat didefinisikan sebagai berikut:

$$X(k) = \sum_{j=1}^{N} x(j)\, \omega_N^{(j-1)(k-1)} \qquad (2.1)$$

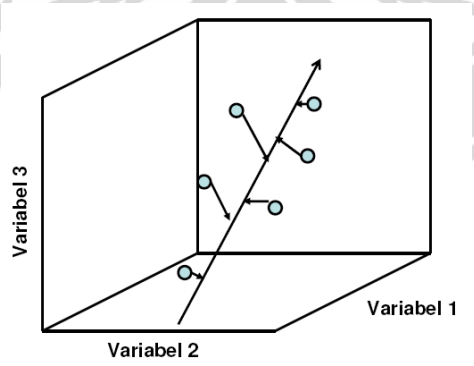
dimana $\omega_N = e^{(-2\pi i)/N}$

2.3 *Principal Component Analysis* (PCA)

Metode yang efektif untuk mendeteksi dan mendiagnosis kondisi operasi *ubnormal* pada proses dengan jumlah variabel yang banyak serta tingkat korelasi yang tinggi adalah *multivariate* SPC (MacGregor et al 1991; Martin & Moris, 1996; Wise & Gallagher, 1996). Adapun Teknik *multivariate* SPC yang paling banyak digunakan untuk pengendalian proses adalah PCA (Kourti & MacGregor 1995; Wise & Gallagher, 1996).

PCA adalah teknik yang powerful untuk mengurangi dimensi (Deng & Tian, 2006), menjelaskan struktur *variance-covariance* melalui beberapa kombinasi linier dari *original* variabel. Sasaran umumnya adalah reduksi data dan interpretasi data (Johnson & Wichern, 2002).

Untuk memahami konsep *Principal Component* (PC) dapat dilihat gambar 2.3, yang menggambarkan titik-titik awan di dalam suatu ruangan berdimensi tinggi. Sebagian besar variasi terletak diantara garis yang tidak parallel pada sembarang sumbu variabel. Garis ini, yang dianggap sebagai PC pertama, melalui rata-rata titik yang dipilih sedemikian sehingga proyeksi titik-titik terhadap garis mempunyai jarak minimal.

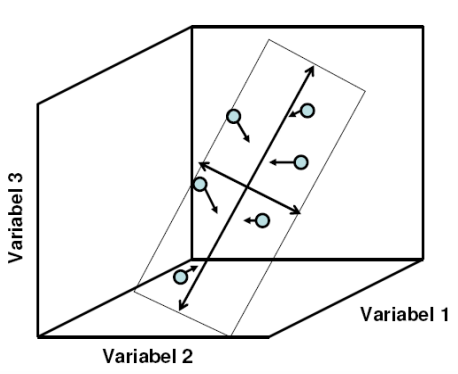


Gambar 2.3 Proyeksi Awan Data pada Satu Garis

(Sumber: Rodrigues & Randall, SAS Institute Inc., diolah kembali)

PC kedua adalah garis yang melalui rata-rata dan memiliki proyeksi jarak minimal dengan arah orthogonal dari PC pertama. PC pertama dan kedua membentuk satu bidang seperti tampak pada gambar 2.4.

Data mining..., Kholisul Fatikhin, FT UI, 2010

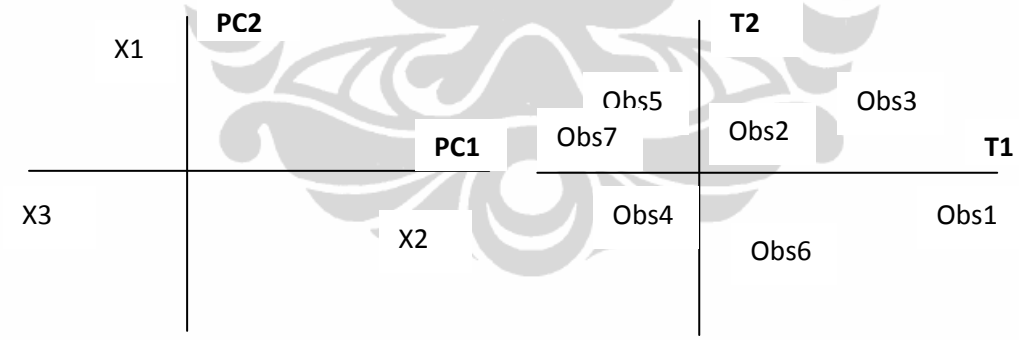


Gambar 2.4 Proyeksi Awan Data pada Satu Bidang

(Sumber: Rodrigues & Randall, SAS Institute Inc., n.d.)

Informasi yang dikandung dalam ruang selanjutnya dapat digambarkan dalam bidang 2-D seperti tampak pada gambar 2.5. Gambar (a) adalah loading plot, menunjukkan hubungan antara variabel asli dan PC. Gambar (b) adalah score plot, menunjukkan hubungan antara observasi dan PC.

(a) Loading Plot (b) Score Plot



Gambar 2.5 Principal Component dalam Bidang 2-D

(Sumber: Kresta, MacGregor, Marlin, 1991)

Dalam notasi matriks, PCA dapat dijelaskan sebagai berikut: pengukuran ke-*i* pada variabel ke-*j* dinotasikan sebagai X_{ij} untuk $i = 1, 2, \ldots, n$, dimana n adalah jumlah pengukuran dan $j = 1, 2, \ldots, k$, dimana k adalah jumlah variabel. Kemudian sampel ke-*i* dapat diwakili sebagai satu vector $\boldsymbol{X}_i = [X_{i1}, X_{i2}, \ldots, X_{ik})$, dan rata-rata sampel vector adalah $\bar{\boldsymbol{X}}_n = [\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_k]$, dimana $\bar{X}_j = \frac{1}{n} \sum_{i=1}^{n} X_{ij}$.

PCA menghitung satu vector yang disebut PC pertama, yaitu satu garis regresi orthogonal yang melalui data di dalam ruangan yang berjarak ***X*** yang merupakan kombinasi linier dari kolom ***X(Xp$_1$)***, diberikan oleh *eigenvector* ***X^{T}X(p$_1$)*** yang berhubungan dengan eigenvalue paling besar . PC kedua adalah garis orthogonal dari PC pertama yang diperoleh dari *fitting* satu garis melalui residual yang tersisa setelah fitting PC pertama yang merupakan kombinasi linier dari kolom ***X(Xp$_2$)***, diberikan oleh *eigenvector* ***X^{T}X(p$_2$)*** yang berhubungan dengan *eigenvalue* terbesar berikutnya. Dengan cara yang sama dihitung PC ke – *k*.

Pada umumnya setelah dihitung PC ke A dimana A << *k*, sebagian besar variasi di dalam matrix data ***X*** sudah dapat dijelaskan. Dalam bentuk vector matrix ***X*** dapat dituliskan dengan persamaan sebagai berikut:

$$\boldsymbol{X} = t_1 p_1^T + t_2 p_2^T + \ldots + t_k p_k^T + \boldsymbol{E} \qquad (2.2)$$

dimana $\boldsymbol{t_i}$ adalah score *vector*, $\boldsymbol{p_i}$ adalah *loading vector*, k adalah jumlah PC dan **E** adalah matriks residual.

Idealnya, dimensi A dipilih sedemikian sehingga tidak ada informasi proses yang signifikan tertinggal dalam matrik residual **E**. Dengan kata lain, **E** hanya mewakili random error.

PC dapat digunakan untuk menginvestigasi faktor mana yang bertanggung jawab pada signal *out-of-control* (Bersimis et. al, 2005). Salah satunya dengan menggunakan peta kendali Hotteling T^2, dengan persamaan sebagai berikut:

$$T_i^2 = \sum_{k=1}^{c} \frac{t_{ik}^2}{s_{ik}^2} \qquad (2.3)$$

dimana t_{ik}^2 adalah score dari observasi ke – *i* untuk komponen ke – *k*, s_{ik}^2 adalah estimasi variance dari t_k .

Dalam praktek, Peta T^2 mempunyai dua kelemahan (*drawback*) utama. Pertama, peta T^2 tidak memberikan skala yang cukup baik untuk jumlah variabel yang besar (k > 20), terutama jika variabel kolinier. Kedua, cukup sulit untuk

menginterpretasikan titik di luar batas kendali (Rodriguez & Tobias, SAS Institute Inc., n.d.). Metode yang lebih baik adalah *Square of Prediction Error,* SPE(Q) (MacGregor & T.Kourti, 1996). Untuk observasi ke – *i* SPE(Q) didefinisikan sebagai:

$$SPE\ (Q) = \sum_{j=1}^{k}(x_{ij} - \hat{x}_{ij})^2 \qquad (2.4)$$

dimana x_{ij} dan $\hat{x}_{ij}$ adalah elemen-elemen dari $\boldsymbol{X}$ dan $\widehat{\boldsymbol{X}}$

Alternatif lain adalah dengan menggunakan residual (elemen dari matrik **E**) yaitu dengan menghitung jarak ke model (*distance to model*) dengan menggunakan persamaan berikut ini:

$$D\text{-}to\text{-}Model = \sqrt{\frac{\sum_{j=1}^{k}(x_{ij} - \hat{x}_{ij})^2}{DF}} \qquad (2.5)$$

dimana *degree of freedom* (DF) didefinisikan sebagai perbedaan antara jumlah variabel **X** dengan jumlah PC atau DF = M – C.

Besaran lain yang berguna untuk diagnosa PC model adalah *fraction of the explained variation* R^2X. Nilai R^2X dapat dihitung dengan mengunakan persamaan berikut:

$$R^2X = 1 - \frac{residual\ sum\ of\ squere}{sum\ of\ squares}$$

$$R^2X = 1 - \frac{\Sigma_{ij}(x_{ij} - \hat{x}_{ij})^2}{\Sigma_{ij}\, x_{ij}^2} \qquad (2.6)$$

PC dengan nilai R^2X yang lebih besar maka dikatakan lebih signifikan dibandingkan PC yang mempunyai nilai R^2X lebih kecil.

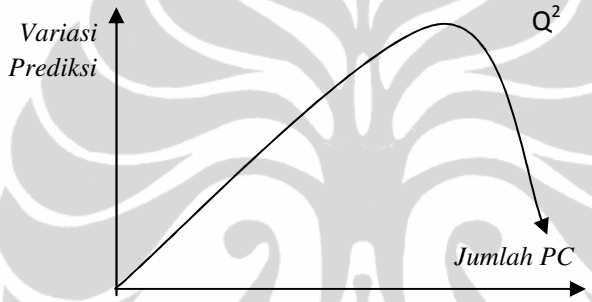
Besaran yang identik dengan R^2X adalah *farction of predicted variation* Q^2X. Besaran ini dihitung menggunakan data observasi yang tidak digunakan untuk membangun PC model.

Data mining..., Kholisul Fatikhin, FT UI, 2010

$$Q^2X = 1 - \frac{predictive\ residuals\ sum\ of\ squeres}{residual\ sum\ of\ squares\ for\ the\ previous\ component}$$

(2.7)

Nilai Q^2 akan bertambah dengan bertambahnya jumlah PC. Namun demikian, pada jumlah PC tertentu nilai Q^2 akan mencapai optimal dan kemudian akan turun kembali seperti tampak pada gambar 2.6. Titik dimana nilai Q^2 turun kembali dapat digunakan untuk mengestimasi jumlah PC. (Statsoft Inc, 2008).



Gambar 2.6 Hubungan Variasi Q^2 dan Jumlah PC

(sumber: Statsoft Inc, 2008)

Jumlah PC menentukan kompleksitas dan akurasi model. Semakin banyak jumlah PC, akurasi model menjadi semakin baik namun model menjadi semakin komplek. Tujuannya adalah untuk mendapatkan model yang tidak terlalu komplek namun cukup akurat untuk memprediksi data.

Cara lain untuk menentukan jumlah PC adalah menggunakan metode *scree test* yang diusulkan oleh cattell (1966) dimana jumlah PC ditentukan berdasarkan titik terendah lembah (*scree*) pada grafik *Eigenvalue*s.

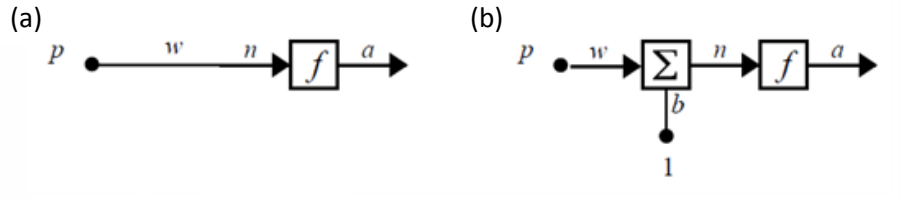
2.4 *Artificial Neural Network* (ANN)

ANN adalah model yang meniru struktur *neural* dari otak manusia. Otak pada dasarnya belajar dari pengalaman. Demikian pula ANN memerlukan suatu model training yang berupa data *input* dan data *output* dari data proses

Data mining..., Kholisul Fatikhin, FT UI, 2010

sebelumnya. Berdasarkan model training, ANN akan mengidentifikasi pola yang ada di dalam kotak hitam (*black box*).

Model paling sederhana dari *artificial neuron* tampak seperti pada gambar 2.7 di bawah ini.



Gambar 2.7 Model Sederhana *Artificial Neuron* (a) Tanpa Bias (b) dengan Bias
(sumber: Demuth & Beale, 2002)

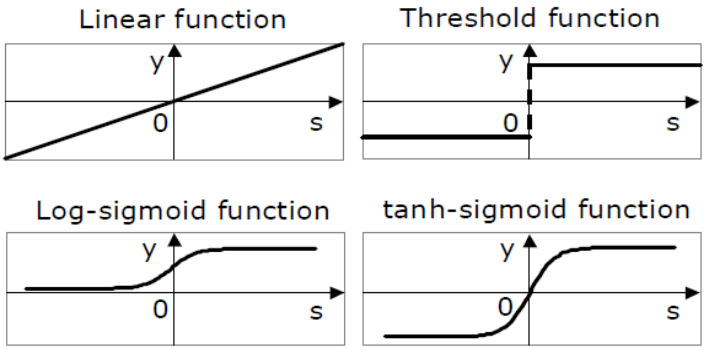
Input diwakili dengan simbol matematis (*P*). Tiap-tiap *input* dikalikan dengan bobot koneksi spesifik (*w*). Disamping itu terdapat satu ekstra *input* yang bernilai +1 dengan bobot w_{k0} yang disebut dengan bias. *Input* selanjutnya melalui satu operasi matematik (biasanya penjumlahan) dan selanjutnya hasil dari operasi ini diumpankan pada fungsi alih (*transfer function*) untuk menghasilkan suatu keluaran. Persamaan matematis untuk model di atas dapat dituliskan sebagai berikut:

$$a = f(wp) \qquad (2.8)$$

$$a = f(wp + b) \qquad (2.9)$$

Ada beberapa fungsi alih yang umum digunakan antara lain fungsi alih *linier*, *threshold*, *log-sigmiod* dan *tanh-sigmoid* seperti tampak pada gambar 2.8.

Data mining..., Kholisul Fatikhin, FT UI, 2010



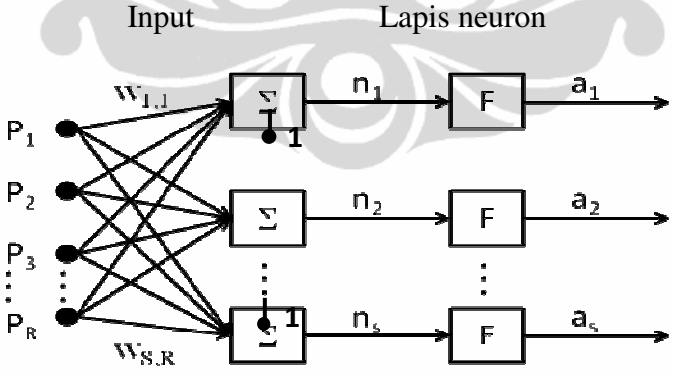
Gambar 2.8 Fungsi Alih Jaringan AN

(sumber: Demuth & Beale, 2002)

Dua atau lebih *artificial neuron* pada gambar 2.7 dapat digabung menjadi satu lapis *neuron*. Satu jaringan AN dapat terdiri dari satu atau lebih lapis. Contoh jaringan dengan satu lapis *neuron* dapat dilihat pada gambar 2.9. Persamaan matematis untuk model jaringan ini dapat dituliskan sebagai berikut:

$$\mathbf{a = f(Wp + b)} \qquad (2.10)$$

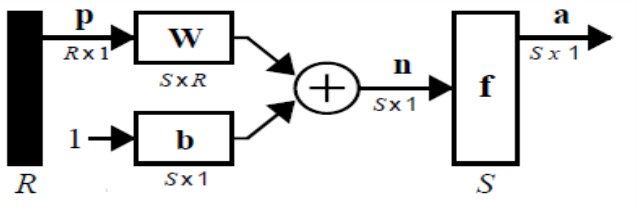
dimana $\boldsymbol{p}$ = vector input; $\mathbf{W}$ = matriks bobot.



Gambar 2.9 Satu Lapis Jaringan AN

(sumber: Demuth & Beale, 2002)

Satu lapis jaringan yang terdiri dari R elemen input dan S neuron tersebut dapat dituliskan kembali dalam bentuk notasi yang lebih sederhana seperti tampak pada gambar 2.10 di bawah ini.

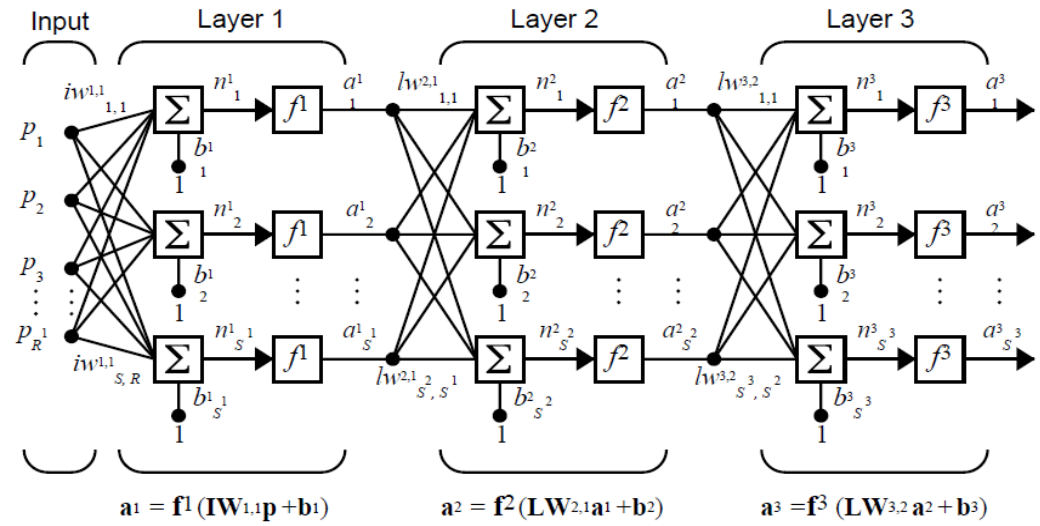


Gambar 2.10 Jaringan AN yang Disederhanakan

(sumber: Demuth & Beale, 2002)

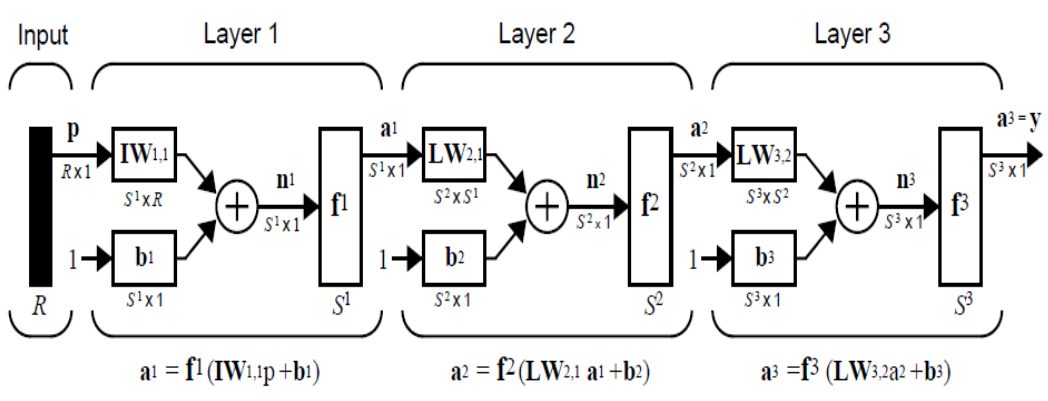
Di sini **p** adalah *vector input* dengan panjang **R**, **W** adalah matriks *S x R*, **a** dan **b** adalah *vector output* dengan panjang **S.**

Jaringan AN yang terdiri dari tiga lapis *neuron* beserta notasi sederhananya dapat dilihat pada gambar 2.11 dan 2.12. Kelebihan jaringan AN multi lapis ini adalah masing-masing lapisan dapat memiliki fungsi alih yang berbeda misalnya lapis pertama *sigmoid*, lapis kedua *linier* dan sebagainya sehingga jaringan AN dapat ditraining untuk mendekati fungsi tertentu.



Gambar 2.11 Jaringan AN Tiga Lapis

(sumber: Demuth & Beale, 2002)



Gambar 2.12 Jaringan AN Tiga Lapis yang Diserderhanakan

(sumber: Demuth & Beale, 2002)

Persamaan untuk model jaringan AN tiga lapis dapat ditu;iskam sebagai berikut:

$$\mathbf{a}^3 = \mathbf{f}^3\,(\mathbf{LW}^{3,2}\,\mathbf{f}^2\,(\mathbf{LW}^{2,1}\,\mathbf{f}^1\,(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3 = \mathbf{y} \qquad (2.11)$$

Dimana **p** adalah *vector input* yang terhubung pada matriks bobot input ($\mathbf{IW}^{1,1}$). yang mempunyai *source* 1 (indeks kedua) dan tujuan 1 (indeks pertama). Keluaran dari lapis pertama $\mathbf{a}^1$ diumpankan ke lapis kedua melalui matriks bobot lapis ($\mathbf{LW}^{2,1}$) dan seterusnya.

Jaringan di atas mempunyai *input* $\mathbf{R}^1$ dan *neuron* $\mathbf{S}^1$ pada lapis pertama, pada lapis kedua ada *neuron* $\mathbf{S}^2$ dan seterusnya. Pada masing-masing lapis umumnya mempunyai jumlah *neuron* yang berbeda.

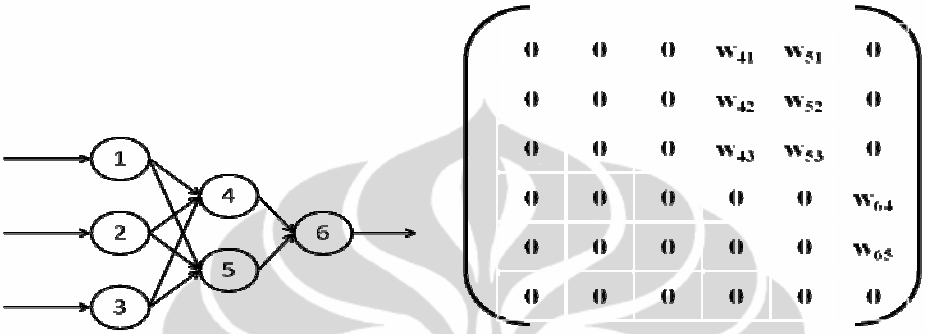
Lapis pada jaringan multi-lapis yang menghasilkan *output* disebut lapis output. Lapis lainnya disebut lapis tersembunyi. Pada contoh di atas, terdapat satu lapis output (lapis 3) dan dua lapis tersembunyi (lapis 1 dan lapis 2).

2.4.1 Arsitektur Jaringan AN

Berdasarkan struktur koneksinya, arsitektur jaringan AN dapat diklasifikasikan menjadi dua group utama yaitu jaringan *feed-forward* dan *feed-back (recurrent)*

a. Jaringan *Feed-Forward*

Yaitu jika interkoneksi matriks dibatasi hanya pada satu arah (tidak ada feed-back atau koneksi sendiri) dan tidak ada proses yang dilakukan pada lapis pertama. Jaringan *feed-forward* yang terdiri dari beberapa lapis dinamakan *multi-layer perceptron* (MLP).

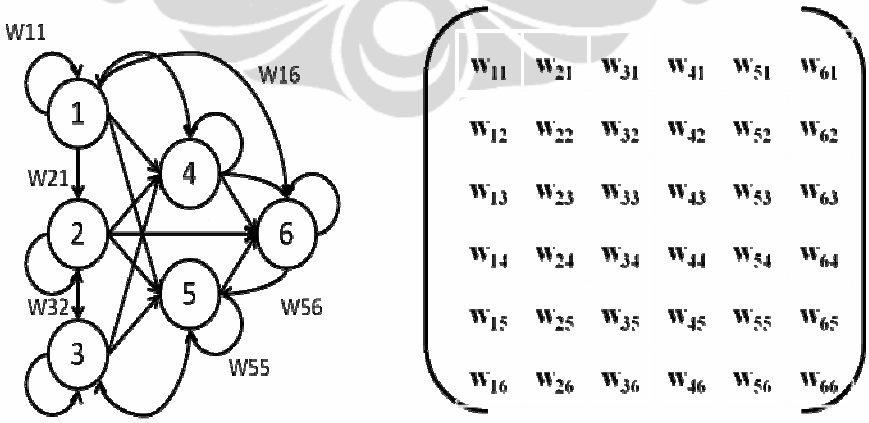


Gambar 2.13 Jaringan AN *Feed-Forward* dan Matriks Bobot

(Sumber: Kaiadi, M., 2006).

b. Jaringan *Recurrent*

Yaitu jika setiap unit dalam setiap layer saling terkoneksi seperti tampak pada gambar 2.14.



Gambar 2.14 Jaringan AN *Recurrent* dan Matriks Bobot

(Sumber: Kaiadi, M., 2006).

Data mining..., Kholisul Fatikhin, FT UI, 2010

2.4.2 Struktur Data Jaringan AN

Ada dua tipe dasar *vector input* yaitu yang terjadi secara serentak (*concurrently*) dan terjadi secara berurutan (*sequentially*). Jika jaringan AN bersifat statik (tidak mempunyai *feedback* atau *delay*) maka tipe *input* ini tidak menjadi begitu penting, semua *vector input* bisa diperlakukan sebagai *concurrent.* Dalam hal ini jaringan dapat dianggap hanya memiliki satu *vector input*.

Jika jaringan AN bersifat dinamis (mengandung *delay*), *input* pada jaringan normalnya adalah *sequential* yang terjadi dalam orde waktu tertentu. Dalam kasus ini, orde *input* sangat penting karena jika orde *input* berubah maka nilai output akan berbeda.

2.4.3 Model Training Jaringan AN

Ada dua jenis training atau proses pembelajaran (*learning*) dari jaringan AN yaitu *supervised learning* dan *unsupervised learning*. Yang pertama adalah jika proses *input* dan *output* diketahui, maka jaringan AN dapat memprediksi performance berdasarkan *input* yang diberikan. Jika yang diketahui hanya *input*nya saja maka jaringan AN harus menemukan cara untuk memanage input.

Dalam fungsi pembelajaran terdapat suatu istilah yang dinamakan laju pembelajaran (*learning rate*). Dengan memilih laju pembelajaran yang rendah, hasil jaringan AN akan lebih akurat tapi proses pembelajaran akan menjadi lebih lama. Sebaliknya, jika laju pembelajaran lebih cepat maka akurasi tidak sebaik sebelumnya.

Lamanya waktu pembelajaran di dalam suatu jaringan AN selain ditentukan oleh laju pembelajaran seperti disebutkan di atas, juga ditentukan oleh kompleksitas jaringan, ukuran data, arsitektur dan *learning rule*.

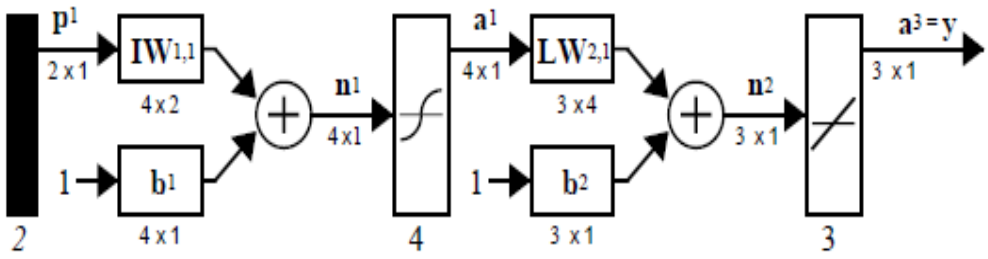
Learning rule adalah prosedur untuk memodifikasi bobot dan bias pada jaringan. Ada dua model *leraning rule* yaitu *Incremental* dan *Batch.* Yang pertama, bobot dan bias pada jaringan diupdate setiap ada input masuk ke jaringan. Yang kedua, bobot dan bias diupdate hanya sekali setelah semua input masuk ke jaringan.

Data mining..., Kholisul Fatikhin, FT UI, 2010

2.4.4 Algoritma *Backpropagation*

Backpropagation dibuat berdasarkan *Windrow-Hoff learning rule*, dimana *vector input* dan *vector target* yang berhubungan digunakan untuk mentraining jaringan. Output jaringan dibandingkan dengan target output, perbedaannya (*error*) digunakan untuk memodifikasi bobot pada jaringan. Hal ini dilakukan berulang sampai diperoleh error terkecil.

Arsitektur jaringan yang paling umum menggunakan algoritma *backpropagation* adalah jaringan *feed-forward* multi-lapis. Jaringan multi-lapis *backpropagation* ini umumnya menggunakan fungsi alih *log-sigmoid* atau *Tan-sigmoid* dan fungsi alih *liner* seperti tampak pada gambar 2.15.



$\mathbf{a}_1 = \mathbf{tansig}(\mathbf{IW}_{1,1}\mathbf{p}_1 + \mathbf{b}_1)$ $\mathbf{a}_2 = \mathbf{purelin}(\mathbf{LW}_{2,1}\mathbf{a}_1 + \mathbf{b}_2)$

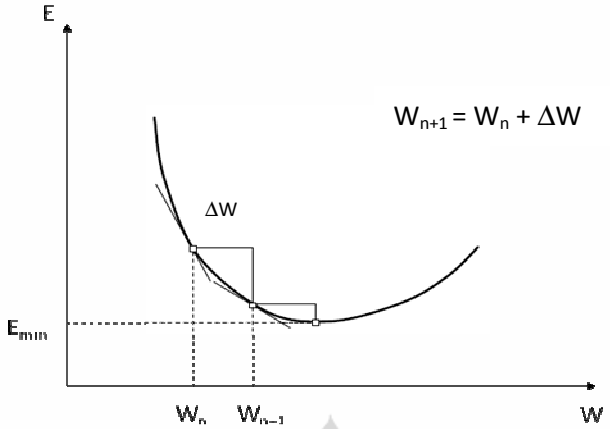
Gambar 2.15 Jaringan Multi-Lapis *Backpropagation*

(sumber: Demuth & Beale, 2002)

Ada banyak variasi algoritma *backpropagation*, yang paling sederhana adalah algoritma *gradient descent*, dimana bobot dan bias jaringan diupdate dalam arah gradien fungsi negatif seperti tampak pada gambar 2.16. Iterasi algoritma ini dapat dituliskan sebagai berikut:

$$x_{k+1} = x_k - \alpha_k g_k \qquad (2.12)$$

dimana $\boldsymbol{x}_k$ adalah vector bobot dan bias sekarang, $\boldsymbol{g}_k$ adalah gradient sekarang dan α_k adalah laju pembelajaran.



Gambar 2.16. *Gradient Descent Learning Rule*

(Sumber: Kaiadi, M., 2006).