

Mengintegrasikan sebanyak mungkin aktifitas penanganan dan mengkoordinasikan sistem operasi yang meliputi agen penerimaan, penyimpanan, produksi, inspeksi, perawatan dan transportasi.

3. Penyederhanaan

Menyederhanakan penanganan dengan cara mengurangi, menghilangkan, menggabungkan pemindahan atau peralatan yang tidak perlu.

4. Memanfaatkan ruangan

Memanfaatkan volume ruangan seoptimal mungkin.

5. Ukuran satuan

Tingkatkan jumlah, ukuran, berat atau tingkat aliran

6. Pemilihan peralatan

Dalam pemilihan peralatan, pertimbangkan semua aspek penanganan material, pemindahan dan metode yang digunakan

7. Standardisasi

Standarisasi metode penanganan, jenis, ukuran peralatan.

8. Kemampuan adaptasi

Gunakan metode dan peralatan yang dapat menjalankan berbagai macam tugas dan penerapan dengan baik

9. Bobot mati

Mengurangi perbandingan bobot mati dari peralatan penanganan yang bergerak terhadap beban yang dibawa.

10. Utilisasi

Rencanakan pemakaian peralatan penanganan dan tenaga kerja secara optimum.

#### 11. Perawatan

Rencanakan perawatan pencegahan dan perbaikan dari semua peralatan penanganan

#### 12. Keusangan

Ganti metode dan peralatan yang usang jika ada metode peralatan yang lebih efisien akan meningkatkan operasi.

#### 13. Pengawasan

Gunakan aktifitas-aktifitas penanganan material untuk memastikan pengendalian produksi, pengendalian persediaan dan penanganan lain.

#### 14. Kapasitas

Gunakan peralatan penanganan untuk membantu dalam mencapai kapasitas produksi yang diinginkan.

#### 15. Efektifitas

Tentukan efektifitas kinerja penanganan dalam bentuk biaya persatuan yang ditangani.

#### 16. Keamanan

Tetapkan metode dan peralatan yang sesuai untuk keamanan penanganan.

### **2.2.3. Lintasan Aliran *Material Handling***

Ukuran jarak dalam masalah fasilitas lokasi merupakan elemen penting dalam memformulasikan model analitik dalam perencanaan fasilitas. Ukuran jarak dapat dikategorikan sebagai berikut:

### 1. *Rectilinier*

Jarak diukur mengikuti aliran *orthogonal* atau tegak-lurus dari satu tempat ketempat lain. Ukuran ini disebut juga sebagai Jarak Manhattan, karena beberapa jalan di kota tersebut dibangun paralel antara jalan yang satu dengan yang lain.

### 2. *Euclidian* atau garis lurus

Jarak diukur dengan lintasan garis lurus antara satu titik dengan yang lain. Segmen jalur *konveyor* yang menghubungkan dua stasiun kerja menggambarkan jarak *Euclidian*.

### 3. Jarak lintasan aliran

Jarak diukur sepanjang lintasan aliran tetap yang menghubungkan dua titik. Jarak lintasan ini lebih panjang bila dibandingkan dengan jarak *Euclidian* atau *Rectilinier*.

## 2.3. *Artificial Neurul Network*

*Artificial neural network* merupakan suatu bentuk arsitektur yang terdistribusi paralel dengan sejumlah besar *nodes* dan hubungan antara *nodes* tersebut. Tiap titik hubungan dari satu node ke node yang lain mempunyai harga-harga yang diasosiasikan dengan bobot. Dalam nilai bobot itulah *artificial neural network* menyimpan ilmu pengetahuannya. Setiap *node* sendiri memiliki suatu nilai yang diasosiasikan sebagai nilai aktivasi *nodes*.

### 2.3.1. Konsep Dasar *Artificial Neural Network*

#### 2.3.1.1. *Artificial Neural Network*

Tiap-tiap *artificial neural network* terdiri dari koleksi *neuron* yang dikelompokkan dalam suatu *layer* (lapisan). Struktur *layer*, secara umum ada 3 *layer* [KUS99], yaitu :

1. *Input layer* :

Mengubah suatu sampel yang dimasukkan ke dalam bentuk *signal* yang dapat dimengerti sistem dan diteruskan ke dalam jaringan untuk diproses.

2. *Hidden (intermediate) layer*.

*Nodes* yang berada dilapisan ini disebut hidden unit, yaitu unit-unit yang tidak berhubungan secara langsung dengan dunia luar. Lapisan inilah yang membuat jaringan saraf memiliki sifat nonlinear karena terjadi proses komputasi sistem jaringan [FAU94].

3. *Output layer*

Mengubah *signal* yang keluar dari proses menjadi dapat ditafsirkan sesuai dengan kasus yang dikehendaki.

#### 2.3.1.2. Struktur Jaringan.

*Neuron-neuron* dapat dihubungkan dengan beberapa cara. Secara konvensional, tiap-tiap PE akan melakukan komputasi secara seri. Namun dapat juga dihubungkan secara paralel (seperti kerja otak), dimana PE dapat bekerja bersama-sama dalam suatu waktu [KUS99].

### 2.3.1.2.1. Processing Elements

*Artificial neural network* terdiri dari beberapa *neuron* tiruan yang disebut dengan *processing elements*. Masing-masing *neuron* menerima input, memproses input, dan mengirim sinyal output (gambar 2.2). Secara umum informasi unit pengolah pada jaringan saraf terdiri atas dua bagian yaitu masukan dan keluaran. Yang berhubungan dengan masukan dari unit pengolah adalah fungsi  $f$  yang berfungsi untuk meyatukan informasi, aktivasi, atau data-data dari sumber luar atau unit pengolah yang lainnya. Biasanya fungsi yang digunakan adalah fungsi linear dengan masukan  $x_j$  yaitu :

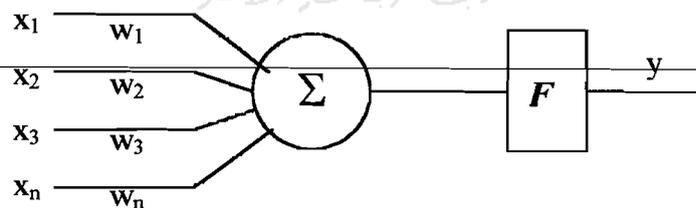
$$f_i = net_i = \sum_{j=1}^m w_{ij} x_{j_i} \dots \dots \dots (2.1)$$

Keterangan :  $i$  = Unit pengolah yang ke- $i$  pada lapisan tersembunyi

$j$  = Masukan ke- $j$

$m$  = Jumlah unit pengolah

$net_i$  = Penjumlahan berbobot pada unit pengolah yang ke- $i$



**Gambar 2.4 Informasi pemrosesan pada *artificial neural network***

Keterangan :

$x_1 \dots x_n$  = Masukan

$F$  = Fungsi aktivasi

$w_1 \dots w_n$  = Bobot

$y$  = Keluaran

$\Sigma$  = Jumlah perkalian  $x_n$  dengan  $w_n$

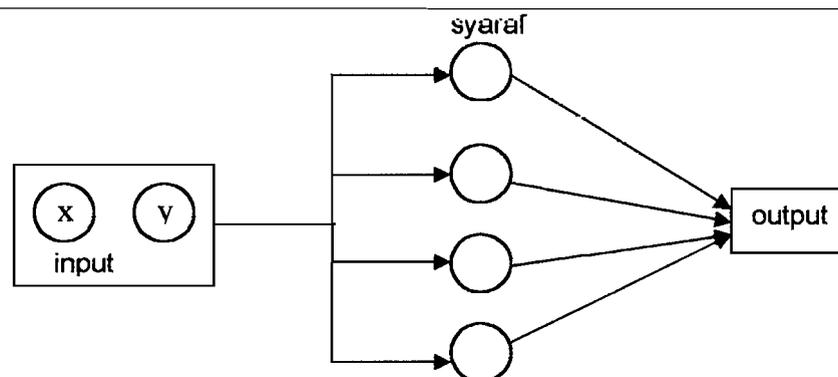
### 2.3.1.2.2. Inisialisasi Bobot dan Bias.

Pemilihan bobot awal mempengaruhi kondisi pelatihan yang dapat menyebabkan terjadinya global atau lokal minimum, jika hal tersebut terjadi maka *artificial neural network* akan kesulitan untuk mencapai kondisi konvergen. Nilai bobot awal sebaiknya tidak terlalu besar, sebaliknya jika bobot awal terlalu kecil akan menyebabkan pembelajaran menjadi lambat. Besar nilai bobot awal secara umum adalah nilai random antara  $-0.5$  s/d  $0.5$  atau antara  $-1$  s/d  $1$ .

### 2.3.1.3. Arsitektur *Kohonen Self-Organizing Maps*

Pola ketetanggaan syaraf yang terdapat dalam Kohonen bermacam-macam. Pola ketetanggaan ini ditentukan pada awal dan tergantung dari keinginan pemodel sehingga dapat berbeda-beda sesuai dengan keinginan dan semuanya dapat dipergunakan [FAU94].

Arsitektur dari jaringan Kohonen dapat digambarkan sebagai berikut.



**Gambar 2.5** Arsitektur Jaringan Syaraf

### 2.3.1.3.1. Algoritma *Kohonen Self-Organizing Maps*

Jaringan syaraf tiruan yang dikembangkan oleh kohonen telah diaplikasikan ke dalam beberapa bidang permasalahan, salah satunya adalah masalah *clustering*. Pokok permasalahan dari jaringan kohonen dalam menyelesaikan masalah *clustering* adalah membiarkan jaringan syaraf belajar sendiri mengorganisasikan dirinya [ANO03]. Untuk dapat melakukan hal ini maka pada jaringan Kohonen diberikan pola secara terus-menerus (*continue*) dan acak (*random*) sehingga diperoleh jaringan Kohonen yang stabil. Seperti kebanyakan jaringan syaraf maka dalam penyelesaian masalah *clustering* jaringan syaraf yang digunakan berjumlah dua kali dari jumlah input, dimana setiap syaraf yang ada terhubung dengan syaraf yang lainnya termasuk dengan dirinya sendiri. Bobot dari setiap syaraf yang ada tergantung dari jarak syaraf tersebut dengan syaraf yang lainnya.

Untuk memasukkan data dalam jaringan maka dibutuhkan lapisan lain yang merupakan lapisan input data. Lapisan ini tidak sama dengan lapisan sebelumnya baik dalam jumlah maupun karakteristiknya. Setiap syaraf yang merupakan syaraf input terhubung dengan setiap syaraf yang ada dalam kelompok pertama ( syaraf pada hidden layer ). Nilai dari hubungan ini disimbolkan dengan  $w$ .

Sehingga langkah utama dalam penyelesaian masalah *clustering* dapat diurutkan sebagai berikut [SVE00] [ANO03]:

#### 1. Langkah 1

Petakan sejumlah syaraf yang merupakan syaraf pada hidden layer berdasarkan pola yang dikehendaki. Syaraf pada kelompok ini mempunyai atribut nilai  $x$  dan nilai  $y$ .

## 2. Langkah 2

Masukkan input ke dalam jaringan ini berupa karakteristik dari masing-masing karakteristik yang bernilai  $x$  dan  $y$

## 3. Langkah 3

Inisialisasikan semua bobot yang ada dengan pembangkitan bilangan acak (*Random*)

## 4. Langkah 4

Pilih lokasi secara acak dan gunakan karakteristik dari lokasi tersebut sebagai syaraf input. ( nilai  $x$  dan  $y$  )

## 5. Langkah 5

Cari syaraf pada kelompok kedua yang mempunyai nilai

$$d_j = \sum_i (w_{ij} - x_i)^2 \dots\dots\dots (2.2)$$

keterangan :

$w_{ij}$  = bobot neuron  $j$  dari data input  $i$

$d_j$  = jarak neuron terhadap bobot  $kc-i$

## 6. Langkah 6

Perbaharui bobot syaraf dengan rumus:

$$w_{ij}xbaru = w_{ij}xlama + \alpha (x_i - w_{ij}xlama) \dots (2.3)$$

$$w_{ij}ybaru = w_{ij}ylama + \alpha (y_i - w_{ij}ylama) \dots (2.4)$$

keterangan :

$w_{ij}$  = bobot neuron  $j$  dari data input  $i$

$\alpha$  = laju pembelajaran jaringan.

hal ini dilakukan pada semua syaraf dimana *alpha* parameter pembelajaran

7. Langkah 7

Lakukan pengurangan terhadap *alpha* dengan momentum dan kemudian hitung kembali nilai bobot ( $r$ ).

8. Langkah 8

Kembali ke langkah ke 2 dan lakukan perulangan sebanyak mungkin.

### 2.3.2. Karakteristik *Artificial Neural Network*

*Artificial neural network* memiliki beberapa sifat khas, seperti proses belajar dan perhitungan. Penyelesaian masalah dengan *artificial neural network* melalui proses belajar dari contoh-contoh. Biasanya pada jaringan saraf buatan diberikan sebuah himpunan pola pelatihan, dari contoh inilah jaringan saraf melakukan proses belajar. *Artificial neural network* adalah suatu sistem yang dibangun secara bertahap yang memiliki prinsip menyerupai otak manusia. *Artificial neural network* dapat digunakan dalam berbagai hal seperti pencocokan pola (*Pattern matching*) dan pengelompokan (*Classification*), misalkan untuk memprediksi harga saham, identifikasi penyakit dll.

*Artificial neural network* memiliki jumlah interkoneksi unit pengolah (*node*) yang banyak, unit-unit pengolah tersebut terhubung secara paralel. Setelah jaringan saraf buatan diberi pelatihan, dengan sifat ini jaringan saraf akan bisa menanggapi pola masukan dengan benar walaupun ada beberapa unit pemrosesnya yang mengalami kerusakan.

### **2.3.2.1. Dinamika Perhitungan (*Comutational Dynamic*)**

*Artificial neural network* melakukan perhitungan dengan cara berikut ini. Setelah jaringan dilatih atau diprogram (misal pengaturan atas bobot pada suatu nilai tertentu), unit-unit jaringan mengubah posisinya dalam suatu aturan sesuai dengan macam aturan yang diterapkan.

### **2.3.2.2. Dinamika Belajar (*Learning/Training Dynamic*)**

Salah satu karakteristik yang paling menarik dari *artificial neural network* adalah kemampuan untuk belajar. Pada dasarnya belajar ini dilakukan untuk mengatur nilai dan hubungan antara *node* (bobot). Pada proses belajar ini nilai bobot disesuaikan dengan sampel-sampel yang dimasukkan melalui lapisan masukan sehingga nantinya pada suatu saat dimana kesalahan output jaringan sudah dibawah toleransi, diharapkan nilai bobot sudah tetap dan proses belajar selesai. Diharapkan bobot hubungan sudah diwakili oleh karakteristik dari pola input.