



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Pakar

Sistem pakar atau yang biasa disebut *expert system* merupakan alat komputerisasi yang dirancang untuk meningkatkan kualitas dan ketersediaan pengetahuan yang dibutuhkan oleh para pengambil keputusan di berbagai bidang. Sistem pakar juga merupakan sistem yang dapat mengadopsi pengetahuan manusia ke komputer. Ada beberapa definisi dari para ahli (Kusumadewi, 2003) mengenai sistem pakar, antara lain:

1. Menurut Durkin

Sistem pakar adalah suatu *program* komputer yang dirancang untuk memodelkan kemampuan penyelesaian masalah yang dilakukan oleh seorang pakar.

2. Menurut Ignizio

Sistem pakar adalah suatu model dan prosedur yang berkaitan dalam suatu domain tertentu yang mana tingkat keahliannya dapat dibandingkan dengan keahlian seorang pakar.

3. Menurut Girattano dan Riley

Sistem pakar adalah suatu sistem komputer yang bisa menyamai atau meniru kemampuan seorang pakar.

Menurut Copley (2011), sebuah sistem pakar terdiri dari tiga bagian, yaitu:

1. User interface

Pada bagian ini, *non expert user* bisa bertanya kepada sistem pakar dan menerima *output* dari sistem. *User interface* harus dirancang sangat sederhana agar mudah digunakan.

2. Knowledge base

Sekumpulan fakta dan aturan yang dibentuk dari informasi yang diberikan oleh ahli pakar.

3. Inference engine

*Inference engine* bertindak sebagai *search engine*, memeriksa *knowledge base* untuk informasi yang sesuai dengan permintaan *user*.

Sistem pakar dikembangkan oleh manusia untuk menyelesaikan permasalahan-permasalahan yang spesifik (khusus). Ciri-ciri yang menggambarkan sistem pakar adalah sebagai berikut (Kusrini, 2006):

1. Terbatas pada *domain* keahlian tertentu,
2. Berdasarkan pada kaidah atau *rule* tertentu,
3. Dapat digunakan dalam berbagai jenis komputer,
4. Mudah dimodifikasi, yaitu dengan menambah atau menghapus suatu kemampuan dari basis pengetahuannya,
5. Sistem dapat mengaktifkan kaidah secara searah yang sesuai, dituntun oleh dialog dengan pemakai.

Beberapa kelebihan yang dapat diambil dengan adanya sistem pakar, antara lain (Kusumadewi, 2003):

1. Membuat seorang yang awam bekerja seperti layaknya seorang pakar,

2. Meningkatkan produktivitas akibat meningkatnya kualitas hasil pekerjaan, meningkatnya kualitas pekerjaan ini disebabkan meningkatnya efisiensi kerja,
3. Menghemat waktu kerja,
4. Menyederhanakan pekerjaan,
5. Merupakan arsip terpercaya dari sebuah keahlian, sehingga bagi pemakai sistem pakar seolah-olah berkonsultasi langsung dengan sang pakar, meskipun mungkin sang pakar telah tiada,
6. Memperluas jangkauan dari keahlian seorang pakar. Sebuah sistem pakar yang telah disahkan, akan sama saja artinya dengan seorang pakar yang tersedia dalam jumlah besar (dapat diperbanyak dengan kemampuan yang persis sama), dapat diperoleh dan dipakai di mana saja.

Di samping adanya kelebihan, sistem pakar juga memiliki beberapa kekurangan, antara lain (Kusumadewi, 2003):

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal,
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya,
3. Sistem pakar tidak 100% benar.

Beberapa kategori dan *area* permasalahan yang terdapat pada sistem pakar, yaitu (Arhami, 2005):

1. Interpretasi  
Membuat kesimpulan atau deskripsi dari sekumpulan data mentah.

2. **Prediksi**  
Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu.
3. **Diagnosis**  
Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati.
4. **Desain**  
Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu.
5. **Perencanaan**  
Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu.
6. ***Debugging dan Repair***  
Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi.
7. **Instruksi**  
Mendeteksi dan mengoreksi defisiensi dalam pemahaman *domain* subjek.
8. **Pengendalian**  
Mengatur tingkah laku suatu *environment* yang kompleks.
9. ***Selection***  
Mengidentifikasi pilihan terbaik dari sekumpulan kemungkinan.
10. ***Simulation***  
Pemodelan interaksi antara komponen-komponen sistem.

## 11. *Monitoring*

Membandingkan hasil pengamatan dengan kondisi yang diharapkan.

### 2.1.1 **Komponen Sistem Pakar**

Sistem pakar terdiri dari beberapa komponen, antara lain (Buliali, dkk., 2007):

#### 1. **Basis Pengetahuan**

Menurut Martin dan Oxman (1988), pengetahuan merupakan kemampuan untuk membentuk model *mental* yang menggambarkan obyek dengan tepat dan mempresentasikannya dalam aksi yang dilakukan terhadap suatu obyek. Basis pengetahuan mengandung pengetahuan untuk pemahaman, formulasi, dan penyelesaian masalah yang dapat berasal dari pakar, jurnal, majalah, dan sumber pengetahuan lain.

#### 2. *Workplace*

*Workplace* merupakan *area* dari sekumpulan memori kerja yang digunakan untuk menyimpan data hasil observasi dan data lainnya yang dibutuhkan selama pengolahan yang dilakukan berada di dalam memori komputer dan kesimpulan yang dicapai.

#### 3. **Mesin Inferensi**

Mesin inferensi merupakan perangkat lunak yang melakukan penalaran dengan menggunakan pengetahuan yang ada untuk menghasilkan suatu kesimpulan atau hasil akhir. Dalam komponen ini dilakukan pemodelan proses berpikir seperti manusia.

#### 4. Akuisisi Pengetahuan (*Knowledge Acquisition*)

Akuisisi pengetahuan merupakan transformasi keahlian dalam menyelesaikan masalah dari sumber pengetahuan ke dalam program komputer. Terdapat empat metode utama dalam akuisisi pengetahuan, yaitu: wawancara, analisis *protocol*, observasi pada pekerjaan pakar dan induksi aturan dari contoh.

#### 5. Antarmuka Pengguna (*User Interface*)

Antarmuka pengguna merupakan perangkat lunak yang menyediakan media komunikasi antara pengguna dengan sistem. Antarmuka menerima informasi dari pemakai dan mengubahnya ke dalam bentuk yang dapat diterima oleh sistem dan menyajikannya ke dalam bentuk yang dapat dimengerti oleh pengguna.

#### 6. Fasilitas Penjelasan

Fasilitas penjelasan adalah komponen tambahan yang akan meningkatkan kemampuan sistem pakar. Komponen ini menggambarkan penalaran sistem kepada pemakai dengan cara menjawab pertanyaan-pertanyaan.

#### 7. Perbaikan Pengetahuan

Pakar memiliki kemampuan untuk menganalisis dan meningkatkan kinerjanya serta kemampuan untuk belajar.

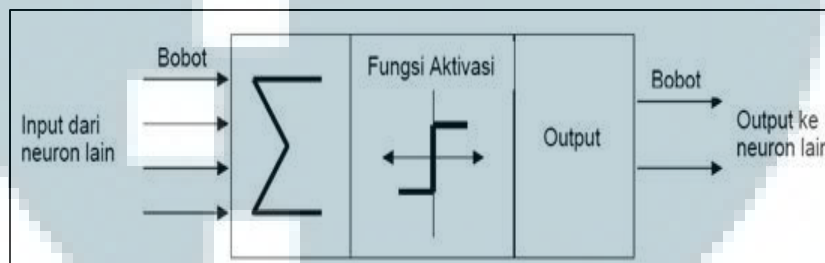
### 2.2 Neural Network

Jaringan saraf tiruan sederhana pertama kali diperkenalkan oleh McCulloch dan Pitts di tahun 1943. McCulloch dan Pitts menyimpulkan bahwa kombinasi beberapa *neuron* sederhana menjadi sebuah sistem *neural* akan meningkatkan kemampuan komputasinya (Siang, 2005).

### 2.2.1 Komponen Neural Network

Ide mendasar dari *Neural Network* adalah mengadopsi mekanisme berpikir sebuah sistem atau aplikasi yang menyerupai otak manusia, baik untuk pemrosesan berbagai sinyal elemen yang diterima, toleransi terhadap kesalahan atau *error* dan juga *parallel processing*.

*Neural Network* memiliki beberapa tipe, namun hampir semua tipe memiliki komponen yang sama. *Neural Network* memiliki beberapa *neuron* yang mempunyai hubungan satu dengan yang lainnya. *Neuron-neuron* tersebut akan mentransformasikan informasi ke *neuron* yang lain. Hubungan ini dikenal dengan sebutan bobot.



Gambar 2.1 Struktur *Neuron* pada *Neural Network* (Sumber: Suhartono, <http://socs.binus.ac.id/2012/07/26/konsep-neural-network/> )

Beberapa istilah yang terdapat di dalam *Neural Network*:

1. Neuron  
Sel saraf tiruan yang merupakan elemen pengolahan *Neural Network* untuk menerima dan memproses *input*.
2. Bobot  
Nilai matematika dari koneksi, mentransfer data antar lapisan. Bobot digunakan untuk mengatur jaringan sehingga menghasilkan *output* yang diinginkan.



### 3. Fungsi Aktivasi

Fungsi yang menggambarkan hubungan antara tingkat aktivasi internal (*summation function*) yang mungkin berbentuk *linear* atau *non linear*.

### 4. *Summation function*

Fungsi yang digunakan untuk mencari rata-rata bobot dari semua elemen *input*.

### 5. *Output*

Solusi atau hasil pemahaman jaringan terhadap data *input*.

### 6. Momentum

Koefisien untuk mempercepat konvergensi dari algoritma *error backpropagation*.

### 7. Jaringan

Kumpulan *neuron* yang saling terhubung dan membentuk lapisan.

### 8. *Input*

Berhubungan dengan sebuah atribut tunggal dari sebuah pola atau data lain.

### 9. *Hidden layer*

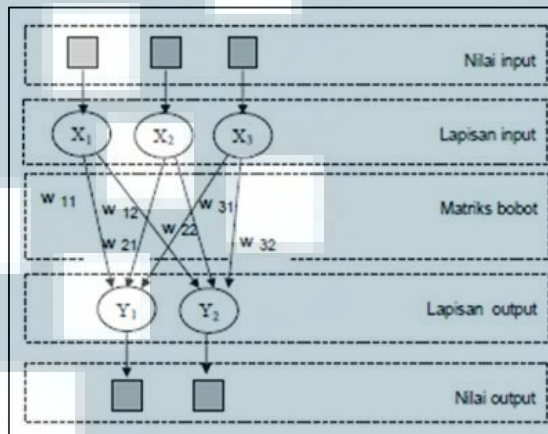
Lapisan yang tidak secara langsung berinteraksi dengan dunia luar.

## 2.2.2 Arsitektur Jaringan

Jaringan saraf tiruan dirancang dengan menggunakan suatu aturan yang bersifat menyeluruh (*general rule*) pada seluruh model jaringan memiliki konsep dasar yang sama. *Neural network* memiliki beberapa macam arsitektur berdasarkan jumlah *layer* yang dimiliki, antara lain (Hermawan, 2006):

1. Jaringan dengan lapisan tunggal (*single layer net*)

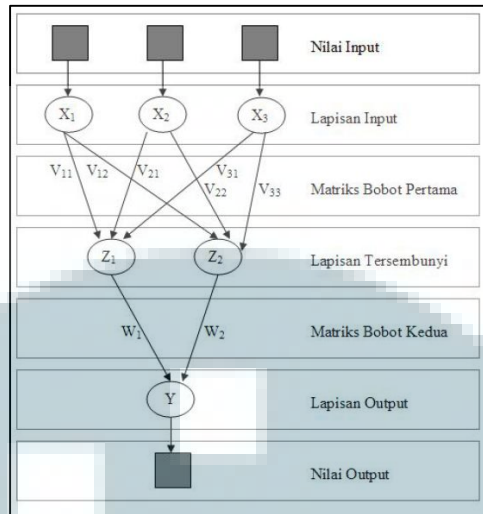
Jaringan saraf satu lapisan pertama kali dirancang oleh Widrow dan Holf pada tahun 1960. Jaringan lapisan tunggal merupakan jaringan yang hanya memiliki satu buah lapisan dengan bobot-bobot yang terhubung. Jaringan ini hanya menerima nilai *input* dan secara langsung mengolahnya untuk menjadi nilai *output* tanpa melalui lapisan tersembunyi.



Gambar 2.2 *Single Layer Net* (Sumber: Hermawan.2006:39)

2. Jaringan dengan banyak lapisan (*multilayer net*)

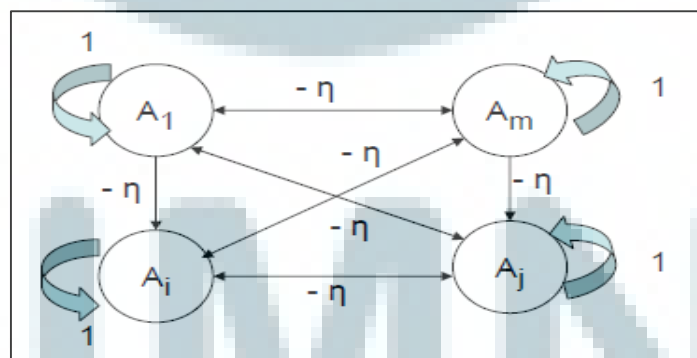
*Multilayer net* merupakan jaringan dengan banyak lapisan yang memiliki satu atau lebih lapisan yang terletak di antara lapisan *input* dan lapisan *output* seperti yang terlihat pada Gambar 2.3. *Multilayer net* ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan *single layer net* walaupun memiliki tingkat kerumitan yang tinggi serta membutuhkan waktu yang lama dalam proses pelatihannya.



Gambar 2.3 *Multilayer Net* (Sumber: Hermawan.2006:40)

3. Jaringan dengan lapisan kompetitif (*competitive layer net*)

Hubungan antar *neuron* pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. *Competitive layer net* sering disebut *feedback loop* karena unit *output* ada yang memberi informasi terhadap unit masukan. Gambar 2.4 menunjukkan salah satu contoh arsitektur jaringan lapisan kompetitif yang memiliki bobot  $-\eta$ .



Gambar 2.4 *Competitive Layer Net* (Sumber: Hermawan.2006:41)

### 2.2.3 Backpropagation

Algoritma *Backpropagation* ini pertama kali diperkenalkan oleh Paul Werbos pada tahun 1974. Pada tahun 1982, David Parker mengemukakan kembali

algoritma ini. Kemudian metode ini dipopulerkan oleh Rumelhart dan McClland pada tahun 1986. Pada *Backpropagation* ini, arsitektur jaringan menggunakan *multilayer net*.

Algoritma *Backpropagation* merupakan salah satu algoritma yang sering digunakan untuk menyelesaikan masalah-masalah yang rumit. Hal ini dimungkinkan karena jaringan dengan algoritma ini dilatih dengan menggunakan metode belajar terbimbing. Pada jaringan diberikan sepasang pola yang diinginkan. Ketika suatu pola diberikan pada jaringan, bobot-bobot diubah untuk memperkecil perbedaan pola keluaran dan pola yang diinginkan.

Pelatihan *backpropagation* meliputi tiga fase (Siang, 2005). Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi. Berikut merupakan penjelasan dari fase-fase tersebut:

1. Propagasi maju

Selama propagasi maju, sinyal masukan dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan. Berikutnya, keluaran jaringan dibandingkan dengan target yang harus dicapai. Selisih antara target *value* dan

*output node value* adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap baris dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi (Siang, 2005).

## 2. Propagasi mundur

Berdasarkan kesalahan  $t_k - y_k$ , dihitung faktor  $\partial_k$  ( $k=1,2,\dots,m$ ) yang dipakai untuk mendistribusikan kesalahan di unit  $y_k$  ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\partial_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung  $\partial_j$  di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor  $\partial$  di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung (Siang, 2005).

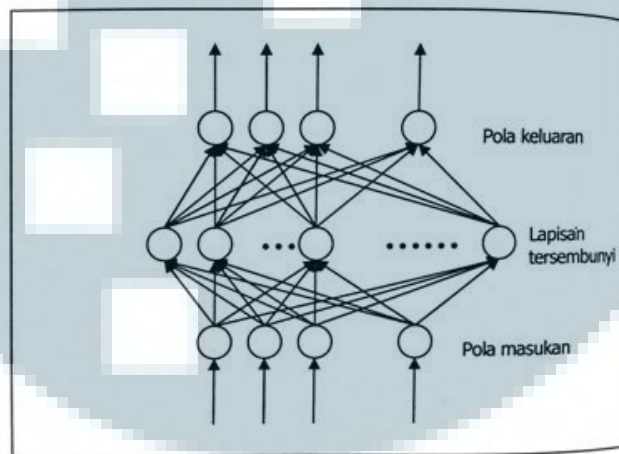
## 3. Perubahan bobot

Setelah semua faktor  $\partial$  dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor  $\partial$  *neuron* di layar atasnya.

Ketiga fase tersebut diulang-ulang terus hingga kondisi pemberhentian terpenuhi (Siang, 2005). Untuk setiap pengulangan dalam pelatihan jaringan saraf tiruan disebut *epoch*. Umumnya kondisi pemberhentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diizinkan (Siang, 2005).

Latihan ini dilakukan berulang-ulang sehingga semua pola yang dikeluarkan jaringan dapat memenuhi pola yang diinginkan. Algoritma *Backpropagation* terdiri atas dua langkah, yaitu perambatan maju dan perambatan mundur. Langkah perambatan maju dan perambatan mundur ini dilakukan pada jaringan untuk setiap pola yang diberikan selama jaringan mengalami pelatihan.

*Backpropagation* terdiri atas tiga lapisan atau lebih unit pengolah. Gambar 2.5 menunjukkan *Backpropagation* dan tiga lapisan pengolah. Bagian kiri sebagai masukan, bagian tengah disebut lapisan tersembunyi, dan bagian kanan sebagai lapisan keluaran. Ketiga lapisan ini terhubung secara penuh.



Gambar 2.5 Tiga lapisan *Backpropagation* (Sumber: Sibi, dkk., <http://www.jatit.org/volumes/Vol47No3/61Vol47No3.pdf>)

Perambatan maju dimulai dengan memberikan pola masukan ke lapisan masukan. Pola masukan ini merupakan nilai aktivasi unit-unit masukan. Dengan melakukan perambatan maju diitung nilai aktivasi pada unit-unit di lapisan berikutnya. Pada setiap lapisan, tiap unit pengelola melakukan penjumlahan berbobot dan menerapkan fungsi *sigmoid* untuk menghitung keluarannya.

Yang dilakukan pada langkah perambatan mundur adalah menghitung dan mengubah bobot-bobot pada semua interkoneksi. Perhitungan dimulai dari

lapisan keluaran dan mundur sampai lapisan masukan dengan mengelola dan mengubah bobot pada semua sambungan.

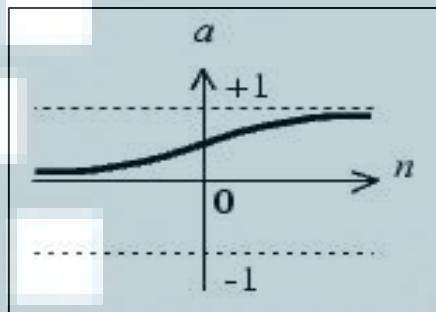
#### 2.2.4 Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang digunakan di dalam algoritma *Backpropagation*, seperti:

##### 1. Fungsi Sigmoid Biner

Fungsi ini merupakan fungsi yang umum digunakan. Fungsi ini memberikan keluaran 0 sampai 1. Data pelatihannya juga di antara 0 sampai

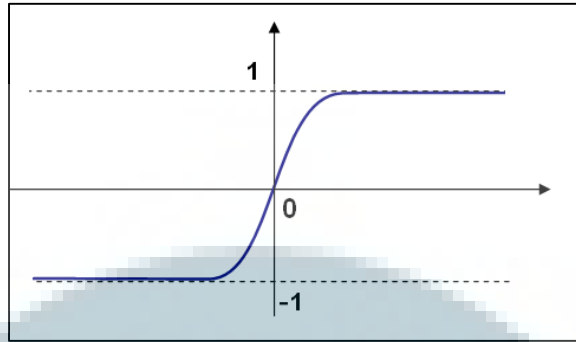
1.



Gambar 2.6 Fungsi Aktivasi Sigmoid Biner (Sumber: Mathworks, <http://www.mathworks.com/help/nnet/ref/logsig.html>)

##### 2. Fungsi Sigmoid Bipolar

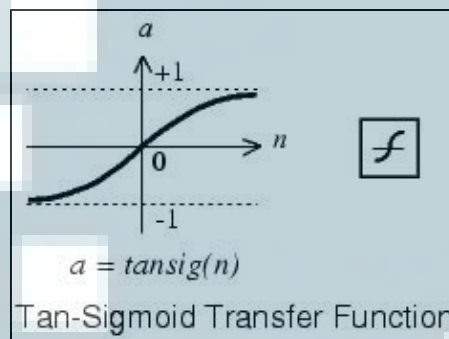
Fungsi sigmoid bipolar merupakan fungsi yang umum digunakan dan memiliki keluaran dari -1 sampai 1.



Gambar 2.7 Fungsi Aktivasi *Sigmoid Bipolar* (Sumber: OpenCV, [http://docs.opencv.org/2.4/modules/ml/doc/neural\\_networks.html](http://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html))

### 3. Fungsi Tangen Hiperbolik

Fungsi ini sangat mirip dengan sigmoid bipolar yang memiliki rentang -1 sampai 1.



Gambar 2.8 Fungsi Aktivasi Tangen Hiperbolik (Sumber: Mathworks, <http://www.mathworks.com/help/nnet/ref/logsig.html>)

## 2.3 Attention Deficit Hyperactivity Disorder

Menurut Sugiartin (2006), *Attention Deficit Hyperactivity Disorder* (ADHD) merupakan suatu kelainan perkembangan yang terjadi pada masa anak dan dapat berlangsung sampai masa remaja. ADHD adalah gangguan yang mempengaruhi perkembangan dalam peningkatan aktivitas motorik anak-anak yang tidak lazim dan cenderung berlebihan.

Istilah ADHD belum dikenal secara luas dan merupakan istilah baru, tetapi banyak orang yang menyebutnya hiperaktif bila seorang anak memperlihatkan



perilaku *over* aktif dan tidak terkendali yang telah dialami sejak lama. Istilah hiperaktif pertama kali ditulis oleh seorang *neurolog* pada tahun 1845 yang bernama Heinrich Hoffman. Beliau mengamati seorang anak yang bernama Dusty yang tinggal di Chicago, dan anak itu menunjukkan perilaku *over* aktif.

Kebanyakan dari mereka yang mengalami gangguan ini mulai membutuhkan bantuan pada usia 6-9 tahun, walaupun banyak orang tua yang mengatakan bahwa masalah pada anaknya sebenarnya telah muncul sejak anak-anak ini duduk di Taman Kanak-Kanak. Namun demikian anak ADHD memiliki tiga komponen ciri utama yang sama yaitu *inattention*, *impulsivitas*, dan *hyperactive* (Sugiarmin, 2006).

### **2.3.1 Penyebab Attention Deficit Hyperactivity Disorder**

ADHD telah banyak diteliti dan dipelajari tetapi belum ada satu pun penyebab pasti yang tampak berlaku bagi semua gangguan yang ada. Berbagai virus zat-zat kimia berbahaya yang banyak dijumpai di lingkungan sekitar, faktor genetika, masalah selama kehamilan atau kelahiran, atau apa saja yang dapat menimbulkan kerusakan perkembangan otak, berperan penting sebagai faktor penyebab ADHD ini.

Faktor yang berpengaruh terhadap munculnya ADHD, yaitu (Sugiarmin, 2006):

#### **1. Faktor genetika**

Bukti penelitian menyatakan bahwa faktor genetika merupakan faktor penting dalam memunculkan tingkah laku ADHD. Satu pertiga dari anggota keluarga ADHD memiliki gangguan, yaitu jika orang tua mengalami ADHD, maka anaknya beresiko ADHD sebesar 60%. Pada anak kembar,

jika salah satu mengalami ADHD, maka saudaranya 70%-80% juga beresiko mengalami ADHD. Beberapa penemuan menunjukkan bahwa molekul genetika gen-gen tertentu dapat menyebabkan munculnya ADHD. Jadi, temuan-temuan dari aspek keluarga, anak kembar, dan gen-gen tertentu menyatakan bahwa ADHD ada kaitannya dengan keturunan.

## 2. Faktor neurobiologis

Beberapa dugaan dari penemuan tentang *neurobiologis* menyatakan bahwa terdapat persamaan antara ciri-ciri yang muncul pada ADHD dengan muncul pada kerusakan fungsi *lobus prefrontal*. Temuan dilakukan melalui MRI (pemeriksaan otak dengan teknologi tinggi) yang menunjukkan adanya ketidaknormalan pada bagian otak depan. Bagian otak ini meliputi *korteks prefrontal* yang saling berhubungan dengan bagian dalam secara kolektif yang dikenal sebagai *basal ganglia*.

Bagian otak ini berhubungan dengan atensi, fungsi eksekutif, penundaan respons dan organisasi respons. Kerusakan-kerusakan pada daerah ini memunculkan ciri-ciri yang serupa dengan ciri-ciri pada ADHD. Informasi lain bahwa anak ADHD memiliki *korteks prefontal* yang lebih kecil dibandingkan anak yang tidak ADHD.

## 3. Identifikasi ADHD

ADHD merupakan salah satu gangguan yang tidak mudah untuk dibedakan antara anak ADHD yang tergolong ringan dengan anak normal yang sedikit lebih aktif dibanding anak yang lainnya. Oleh karena itu, ADHD merupakan suatu gangguan yang kompleks berkaitan dengan pengendalian diri dalam berbagai variasi gangguan tingkah laku. Menurut Lauer (1992), variasi

gangguan secara umum adalah gangguan pemusatan perhatian yang berkaitan dengan gangguan tingkah laku dan aktivitas kognitif, seperti misalnya berpikir, mengingat, menggambar, dan lain-lain.

Berikut ciri-ciri ADHD yang muncul pada masa anak-anak awal yang bersifat menahun dan tidak diakibatkan oleh kelainan fisik yang lain, mental maupun emosional. Ciri utama yang muncul meliputi (Sugiarmin, 2006):

a) Inatensi

Yang dimaksud adalah bahwa sebagai individu penyandang gangguan ini tampak mengalami kesulitan dalam memusatkan perhatiannya. Mereka sangat mudah teralihkan oleh rangsangan yang tiba-tiba diterima oleh alat inderanya atau oleh perasaan yang timbul pada saat itu. Dengan demikian mereka hanya mampu mempertahankan suatu aktivitas atau tugas dalam jangka waktu yang pendek, sehingga akan mempengaruhi proses penerimaan informasi dari lingkungannya.

b) Impulsifitas

Impulsifitas adalah suatu gangguan perilaku berupa tindakan yang tidak disertai dengan pemikiran. Mereka sangat dikuasai oleh perasaannya sehingga sangat cepat bereaksi. Mereka juga sulit untuk memberi prioritas kegiatan, sulit untuk mempertimbangkan atau memikirkan terlebih dahulu perilaku yang akan ditampilkannya. Perilaku ini biasanya menyulitkan dengan yang bersangkutan maupun lingkungannya.

c) Hiperaktivitas

Hiperaktivitas merupakan suatu gerakan yang berlebihan melebihi gerakan yang dilakukan secara umum anak seusianya. Biasanya sejak bayi mereka

banyak bergerak dan sulit untuk ditenangkan. Jika dibandingkan dengan individu yang aktif tapi produktif, perilaku hiperaktif tampak tidak bertujuan. Mereka tidak mampu mengontrol dan melakukan koordinasi dalam aktivitas motoriknya, sehingga tidak dapat dibedakan gerakan yang penting dan tidak penting. Gerakannya dilakukan terus menerus tanpa lelah, sehingga kesulitan untuk memusatkan perhatian.

