BAB II

LANDASAN TEORI

2.1. Firewall

Firewall adalah sebuah *hardware* maupun *software* yang didesain untuk mengontrol aliran data yang masuk maupun yang keluar dalam suatu jaringan komputer, secara umum firewall berfungsi sebagai *layer* atau lapisan pelindung untuk menahan serangan-serangan yang sifatnya mengancam keamanan jaringan (Stiawan, D. 2005).

2.1.1. Keuntungan Adanya Firewall

Firewall dapat mengerjakan banyak hal untuk jaringan dan server, pada prakteknya ada beberapa keuntungan utama yang dapat diperoleh dengan adanya firewall, antara lain sebagai berikut (Stiawan, D. 2005) :

1. Firewall Sebagai Fokus Keputusan Security.

   Firewall dalam hal ini bertindak sebagai *choke point*, semua *traffic* (lalu lintas data) yang masuk dan keluar harus melewati "pos pemeriksaan". Oleh karena itu, firewall dapat memilah-milah traffic yang disetujui dan mana yang tidak. Firewall memberi perlindungan *security* jaringan, menjamin semua traffic sesuai dengan yang ditentukan.

2. Firewall Mendukung Security Policy.

   Beberapa service internet seringkali tidak aman, dan kebanyakan orang tidak menyadarinya. Maka firewall dapat menjadi polisi penyelamat atas *service-service* ini, menjamin *security policy* situs-situs suatu jaringan tetap terjaga.

3. Firewall Mencatat Log Aktivitas Internet.

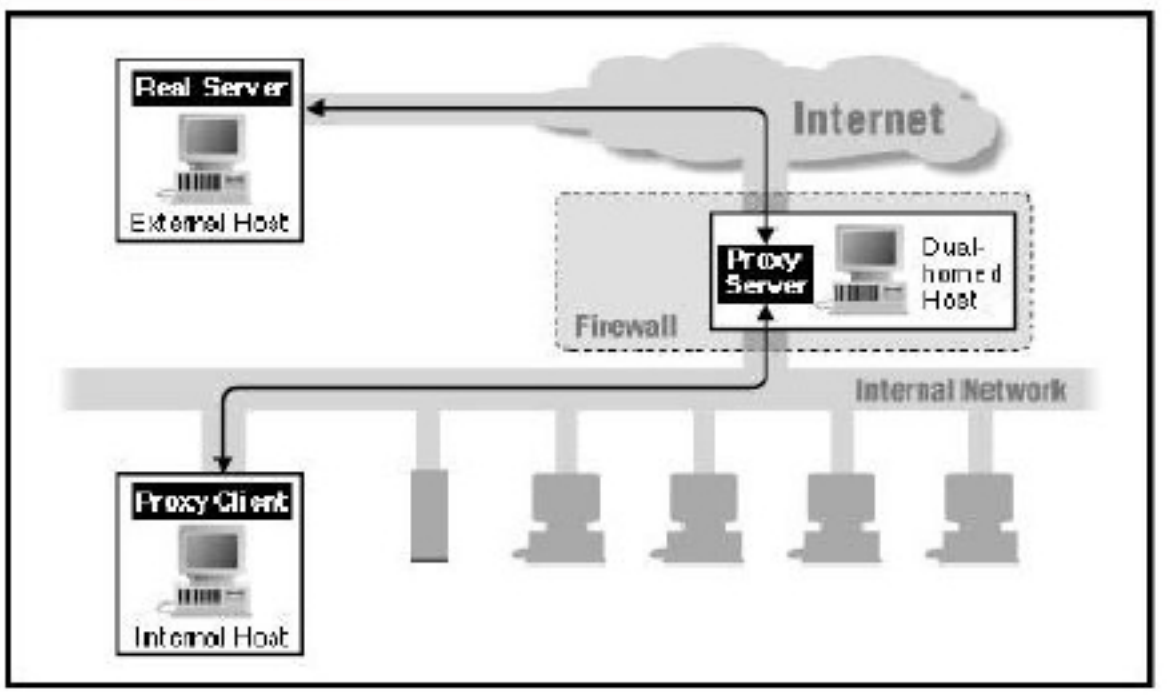
   Oleh karena semua traffic melewati firewall, maka firewall merupakan tempat yang tepat untuk mengoleksi informasi tentang sistem dan jaringan yang digunakan beserta beragam penyalahgunaannya. Sebagai sebuah point tunggal dari akses-akses, firewall dapat merekam apa-apa yang terjadi di antara jaringan terproteksi dengan jaringan *external*.

2.1.2. Tipe Firewall

A. Application Level (Proxy)

Merupakan aplikasi khusus berjalan pada *host* firewall, aplikasi ini menangani *request-request* untuk service-service internet dari *user* dan melewatkannya ke service yang sebenarnya. Proxy menyediakan koneksi pengganti dan bertindak selaku *gateway* terhadap service-service tersebut. Oleh karena itu Proxy sering juga disebut *gateway* level aplikasi .

(Dermawan, I . http://4sucktie.tripod.com).

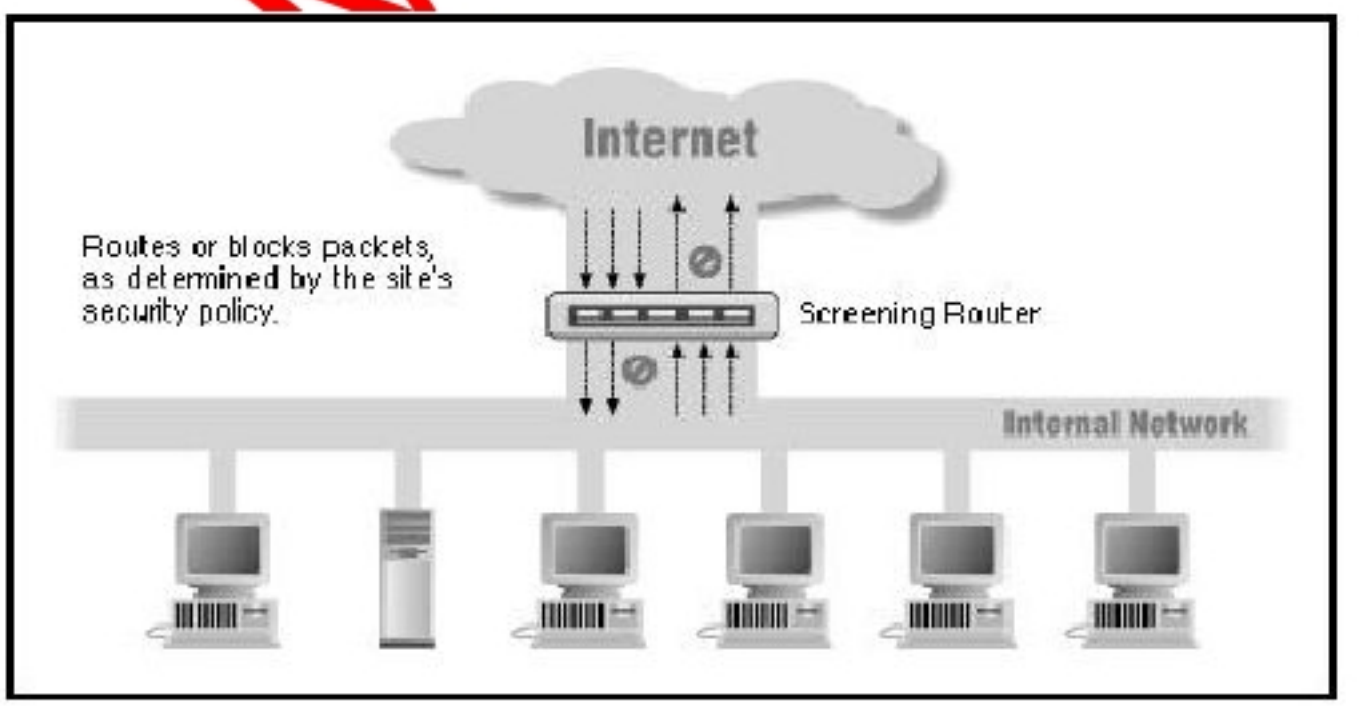


Gambar 2.1 Penggunaan proxy server pada *dual-homed host*

Proxy menghubungkan user jaringan internal dengan service pada internet. User dan service tersebut tidak dapat berkomunikasi secara langsung. Masing-masing berhubungan dengan proxy dan proxy yang menangani hubungan antara user dan service dibelakang layar. Proxy server dapat membatasi apa saja yang dapat dilakukan oleh user, karena proxy server dapat memutuskan apakah suatu request dari user diperbolehkan atau ditolak. Pada gambar 2.1 dapat dilihat bahwa semua request-request terhadap service internet yang dilakukan oleh proxy client harus melewati *dual-homed-host* yang selain berfungsi sebagai proxy server juga berfungsi sebagai *screening router* (packet filtering).

B. Packet Filtering

Sistem packet filtering melakukan packet routing antara jaringan internal dengan jaringan eksternal secara selektif. Sistem ini melewatkan atau memblok packet data yang lewat sesuai dengan aturan yang telah ditentukan. Router pada sistem ini disebut *Screening Router*. (Dermawan, I . http://4sucktie.tripod.com).



Gambar 2.2 Packet Filtering dengan menggunakan *screening router*

Untuk mengetahui bagaimana cara kerja dari packet filtering, kita harus mengetahui perbedaan router biasa dengan sebuah *Screening Router.* Router biasa hanya melihat alamat IP tujuan dari suatu paket data dan mengarahkannya ke jalur terbaik agar paket data tersebut sampai ke tujuannya. Bila router tidak dapat melakukannya, paket data akan dikembalikan ke sumbernya.

Screening Router tidak hanya menentukan apakah router dapat melewatkan suatu paket data atau tidak, tetapi juga menerapkan suatu aturan yang akan menentukan apakah paket data tersebut akan dilewatkan atau tidak. Penyaringan ini didasarkan pada :

1. IP sumber dan IP tujuan dari paket data.
2. Port sumber dan port tujuan dari data.
3. Protokol yang digunakan (TCP, UDP, ICMP dan sebagainya).
4. Tipe pesan ICMP.

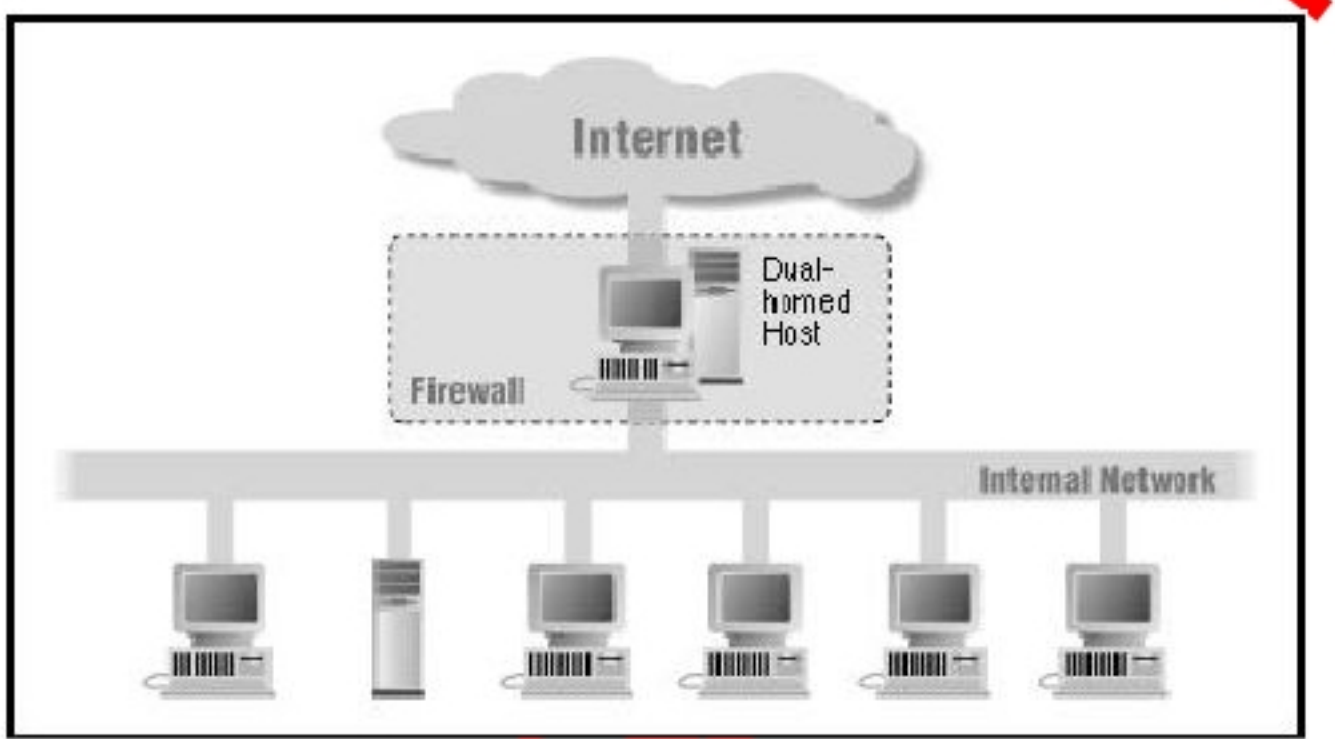
Pada gambar 2.2 dapat dilihat bahwa tidak semua paket data yang melewati *screening router* diijinkan untuk masuk maupun keluar, namun disesuaikan dengan kebijakan keamanan yang dipasang pada *screening router* tersebut.

2.1.3 Arsitektur Firewall

A. Dual-Homed Host

Arsitektur ini dibuat disekitar komputer dual-homed host, yaitu komputer yang memiliki paling sedikit dua *interface* jaringan (NIC). Untuk mengimplementasikan tipe arsitektur ini, fungsi routing pada *host* ini dinon-

aktifkan. Sistem di dalam firewall dapat berkomunikasi dengan *dual-homed host,* tetapi kedua sistem ini tidak dapat berkomunikasi secara langsung. (Dermawan, I . http://4sucktie.tripod.com).

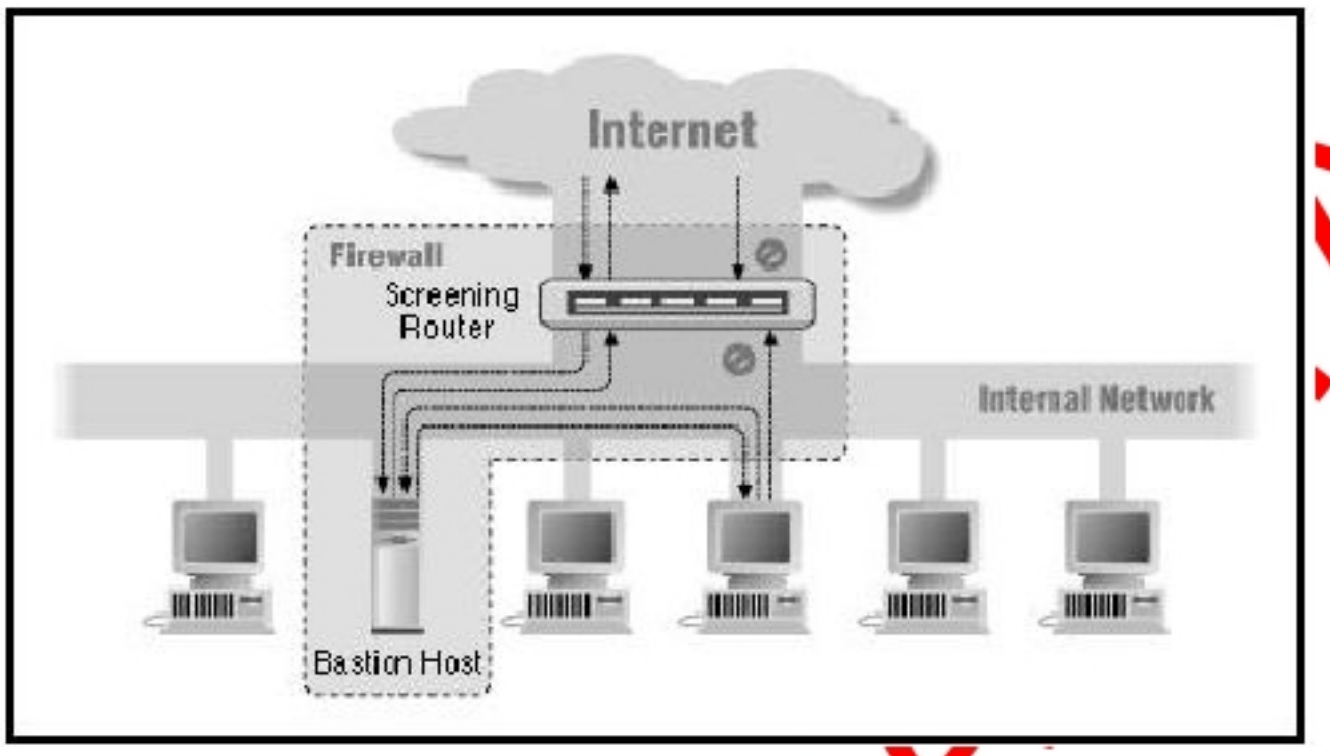


Gambar 2.3 Arsitektur *dual-homed host*

Dual-homed host dapat menyediakan service dengan menyediakan proxy pada *host* tersebut. Pada gambar 2.3 dapat dilihat bahwa *dual-homed host* memisahkan jaringan internal dan internet secara fisik menggunakan dua buah NIC sehingga paket data yang masuk dan keluar akan diseleksi pada *dual-homed host* sebelum diijinkan masuk ataupun keluar dari jaringan internal.

B. Screened Host

Arsitektur ini menyediakan service dari sebuah host pada jaringan internal dengan menggunakan router yang terpisah. Pengamanan utama dilakukan dengan packet filtering. (Dermawan, I . http://4sucktie.tripod.com).



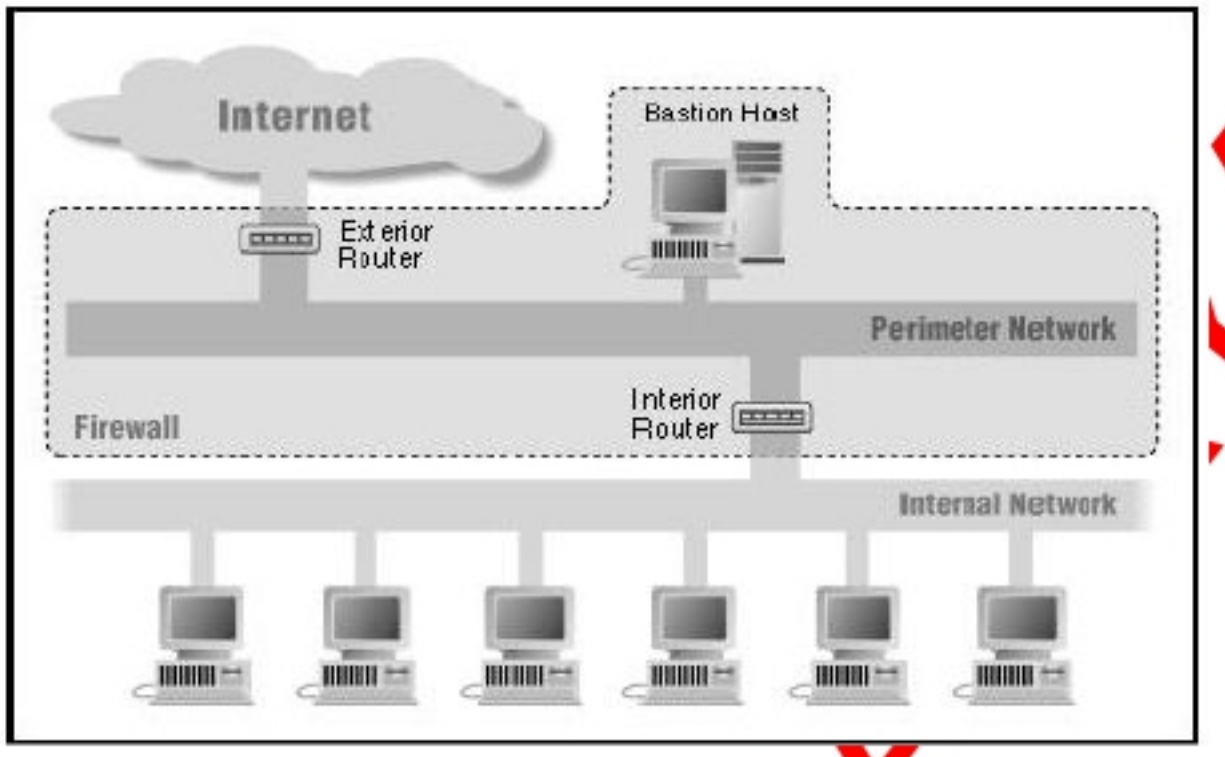
Gambar 2.4 Arsitektur *screened host*

Bastion Host berada dalam jaringan internal. Pada gambar 2.4 dapat dilihat bahwa packet filtering pada screening router dikonfigurasi sehingga hanya bastion host yang dapat melakukan koneksi ke internet (misal mengantarkan e-mail yang datang) dan hanya tipe-tipe koneksi tertentu yang diperbolehkan. Tiap sistem ekternal yang mencoba untuk mengakses sistem internal harus berhubungan dengan *host* ini terlebih dulu. *Bastion host* diperlukan untuk tingkat keamanan yang tinggi.

C. Screened Subnet

Arsitektur ini menambahkan sebuah layer pengamanan tambahan pada arsitektur *screened host*, yaitu dengan menambahkan sebuah jaringan perimeter, yaitu jaringan pengaman tambahan yang mengisolasi jaringan internal dari jaringan internet.

(Dermawan, I . http://4sucktie.tripod.com).



Gambar 2.5 Arsitektur *screened subnet*

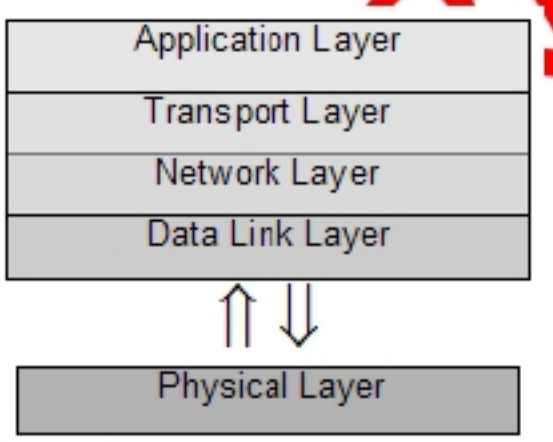
Pada gambar 2.5 dapat dilihat bahwa jaringan perimeter mengisolasi *bastion host* sehingga tidak langsung terhubung ke jaringan internal. Arsitektur *screened subnet* yang paling sederhana memiliki dua buah *screening router*, masing-masing terhubung ke jaringan perimeter. Router pertama terletak di antara jaringan perimeter dan jaringan internal, dan router kedua terletak di antara jaringan perimeter dan jaringan ekternal (biasanya internet). Untuk masuk ke jaringan internal dengan tipe arsitektur *screened subnet*, seorang *intruder* (penyusup) harus melewati dua buah router tersebut sehingga jaringan internal akan relatif lebih lama.

2.2. Model TCP/IP

TCP/IP (*Transmission Control Protocol / Internet Protocol*) adalah sekelompok protokol yang mengatur komunikasi data komputer di Internet. Komputer-komputer yang terhubung ke internet berkomunikasi dengan protokol

ini. Karena menggunakan bahasa yang sama, yaitu protocol TCP/IP, perbedaan jenis komputer dan sistem operasi tidak menjadi masalah. Jadi, walaupun menggunakan sistem operasi dan jenis komputer yang berbeda selama protokol komunikasi yang digunakan sama yaitu TCP/IP komputer tersebut dapat berhubungan dengan komputer di belahan dunia manapun yang juga terhubung ke Internet menggunakan protokol TCP/IP (Heywood, D. 1996).

Arsitektur TCP/IP didefinisikan dalam lima layer / lapisan fungsi dalam arsitektur protokol yaitu seperti pada gambar 2.6 di bawah ini :



Gambar 2.6 Model TCP/IP

Mulai dari atas, lapisan-lapisan dari model TCP/IP yaitu :

A. Application Layer

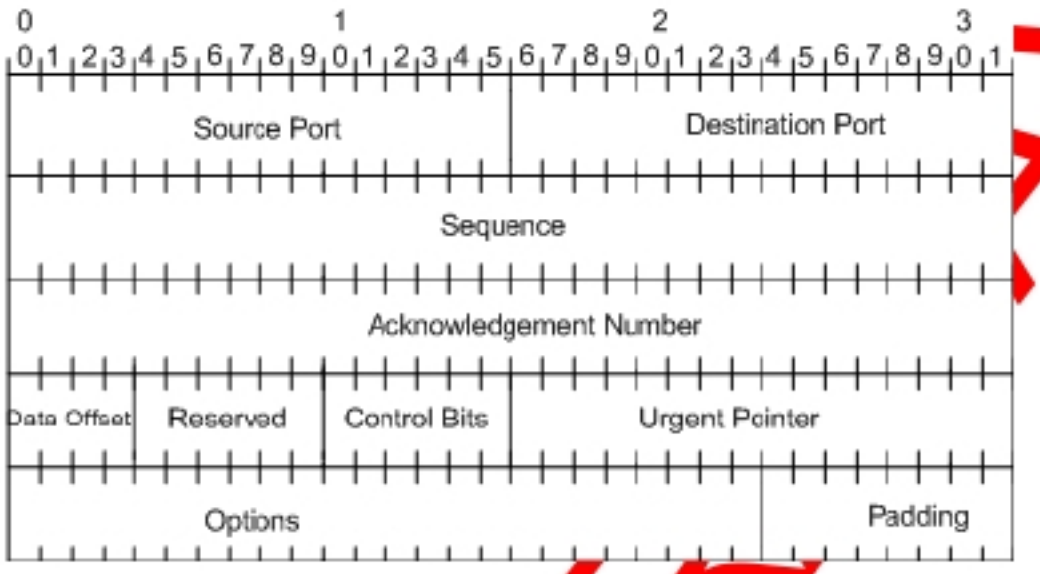
Aplication layer terdapat di puncak model TCP/IP yang berfungsi untuk memberikan pelayanan kepada pemakai komputer untuk dapat berkomunikasi dengan komputer lain melalui jaringan menggunakan protokol seperti TELNET, FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), DNS (*Domain Name Service*) (Heywood, D. 1996).

B. Transport Layer

Lapisan ini berfungsi untuk mengatur alur pengiriman *byte stream* atau paket data yang dikirimkan untuk sampai ke tujuan secara berurutan diantara dua *node* / *host* yang saling berhubungan. Protokol yang digunakan pada lapisan ini adalah TCP (*Transmission Control Protokol*) yang merupakan protokol *reliable connection-oriented* (*reliable* artinya TCP menerapkan proses deteksi kesalahan paket dan pengiriman data ulang, *connection-oriented* berarti sebelum melakukan pertukaran data, dua aplikasi pengguna TCP harus melakukan *handshake* (proses untuk membangun sebuah koneksi) terlebih dahulu yang mengijinkan sebuah *byte stream* yang berasal pada suatu mesin untuk dikirimkan tanpa kesalahan ke sebuah mesin yang ada di internet. TCP memecah aliran *byte stream* menjadi pesan – pesan diskret dan meneruskannya ke internet layer. Pada mesin tujuan, protokol TCP penerima merakit kembali pesan-pesan yang diterima menjadi *byte stream*. Untuk memastikan diterimanya pesan-pesan yang dikirim, TCP menggunakan nomor urutan segmen dan *acknowledgment.* Karena tiap oktet atau komposisi pada suatu segmen memiliki nomor urutan yang memungkinkan tiap oktet yang dikirimkan untuk dijawab oleh penerima. Saat TCP mengirimkan sebuah segmen, ia akan menyimpan duplikat dari segmen tersebut pada antrian dimana segmen tersebut akan tetap ada disana sampai sinyal *acknowledgment* diterima. Setelah diterimanya signal *acknowledgment* maka segmen duplikat akan dibandingkan dengan segmen yang diterima jika hasilnya tidak sama maka oleh penerima segmen tersebut minta dikirimkan ulang dan jika hasilnya sama maka penerima akan meminta pengiriman segmen selanjutnya (Heywood, D. 1996).

B.1 TCP Header

Pada saat IP Menyusun datagram IP, header TCP mengikuti header IP pada datagram. Gambar 2.7 menunjukkan format dari header TCP.



Gambar 2.7 Format header TCP

Segmen TCP disusun dalam 16 bit word. Bila sebuah segmen berisi jumlah oktet yang ganjil, maka akan ditambahkan oktet aktif yang berisi nol agar segmen TCP yang dikirimkan tetap dalam kombinasi 16 bit word . Field dalam header TCP adalah sebagai berikut :

1. Source Port (16 bit). Menunjukkan port pada modul TCP pengirim.
2. Destination Port (16 bit). Menunjukkan port modul TCP penerima.
3. Sequence Number (32 bit). Menunjukkan posisi urutan dari oktet data pertama pada segmen.
4. Acknowledgment Number (32 bit). Berisi nomor urutan berikutnya yang diharapkan oleh pengirim segmen. TCP menandakan bahwa field ini aktif dengan men-set bit ACK, yang selalu di set setelah sebuah hubungan dari kedua sisi terjadi (pengirim dan penerima segmen).

5. Data Offset (4 bit). Berisi jumlah 32 bit word pada header TCP. Options akan ditambahkan oktet bernilai 0 bila dibutuhkan untuk melengkapi 32 bit word ini.
6. Reverse (6 bit). Selalu berisi nol. Dicadangkan untuk penggunaan mendatang.
7. Control Bit (6 bit). Enam bit kontrol sebagai berikut :
    - URG. Saat di set (1), control bit akan mengaktifkan field urgent pointer, saat di clear (0), field tersebut dinonaktifkan.
    - ACK. Saat di set akan mengaktifkan field Acknowledgment Number.
    - PSH. Memulai fungsi push yang memaksa TCP untuk mengirimkan data yang sedang diantrikan saat ini ke tujuan, hal ini akan memaksa penerima untuk membuang buffer yang ada saat ini dan mengirim data yang diterima tersebut ke lapisan diatasnya.
    - RST. Memaksa hubungan di reset.
    - SYN. Melakukan sinkronisasi nomor urutan untuk hubungan. Bit ini di set saat sebuah segmen meminta hubungan dibuka.
    - FIN. Menyatakan berakhir atau ditutupnya hubungan.
8. Window (16 Bit). Berisi jumlah oktet, dimulai dari oktet yang disebutkan pada field Acknowledgment Number, yang dapat diterima saat ini oleh pengirim segmen.
9. Checksum (16 Bit). Checksum untuk error control yang meliputi header dan field data, hal ini tidak termasuk penambahan data 0 sesudah option agar nomor oktet menjadi genap.
10. Urgent Pointer (16 bit). Adalah bilangan positif berisi posisi dari Sequence Number pada segmen.

11. Option (variabel). Option tersedia untuk berbagai fungsi, termasuk akhir daftar option, no-option, ukuran segmen maksimum, dan ukuran option data segmen maksimum.
12. Padding (variabel). Oktet bernilai 0 yang ditambahkan pada header untuk memastikan bahwa header berukuran kelipatan 32-bit word.

C. Network Layer / Internet Layer

Internet Protocol (IP) beroperasi pada lapisan ini untuk menentukan rute yang tidak bergantung kepada media transmisi jaringan yang digunakan. Untuk mengirimkan pesan pada suatu internetwork, tiap jaringan harus secara unik diidentifikasi oleh alamat jaringan. Ketika jaringan menerima suatu pesan dari lapisan yang lebih atas, lapisan network akan menambahkan header pada pesan yang termasuk alamat asal dan tujuan jaringan. Kombinasi dari data dan lapisan network disebut paket. Informasi alamat jaringan digunakan untuk mengirimkan pesan ke jaringan yang benar. Setelah pesan tersebut sampai pada jaringan yang benar, lapisan data link dapat menggunakan alamat *node* untuk mengirimkan pesan ke *node* tertentu (Heywood, D. 1996).

C.1 IP Header

Gambar 2.8 menunjukkan format dari header IP yang terdiri dari beberapa field -field.

5. Identification (16 bit). Field identifikasi digunakan untuk membantu proses penggabungan kembali pecahan dari sebuah datagram.
6. Flag (3 bit). Field ini berisi tiga control flag.
   - Bit 0. dicadangkan, harus 0.
   - Bit 1 (DF). 0 = bisa dipecah menjadi fragmen; 1 = tidak boleh dipecah.
   - Bit 2 (MF). 0 = fragmen terakhir; 1 = masih ada fragmen lagi.
7. Fragment Offset / Posisi fragmen (13 bit). Untuk datagram yang dipecah, menunjukkan pasisi fragmen ini dalam datagram.
8. Time To Live / Waktu Hidup (8 bit). Menunjukkan waktu maksimum bagi sebuah datagram untuk berada pada jaringan. Bila field ini memiliki nilai 0, datagram akan dibuang. Field ini dimodifikasi selama tahap pemrosesan header IP dan umumnya dihitung dalam detik. Namun tiap modul IP yang menangani datagram harus mengurangi Time To Live ini dengan 1. mekanisme ini memastikan bahwa datagram yang tidak terkirim suatu saat akan dibuang.
9. Protocol (8 bit). Protokol lapisan yang lebih atas yang berhubungan dengan bagian data dari datagram.
10. Header Checksum (16 bit). Sebuah nilai checksum untuk header saja. Nilai ini harus dihitung ulang tiap kali sebuah proses header dimodifikasi.
11. Source Address (32 bit). Alamat IP dari *host* yang mengirimkan datagram.
12. Destination Address (32 bit). Alamat IP dari *host* yang merupakan tujuan akhir dari datagram.
13. Option (0 sampai 11, 32 bit). Dapat berisi 0 atau lebih pilihan. Pilihan yang tersedia dapat dilihat pada RFC 791 (RFC (Request For Comments) yang

diterbitkan oleh IAB (Internet Activities Board) yang merupakan komite independen para peneliti dan profesional yg mengerti teknis, kondisi dan evolusi sistem internet.

D. Network Access Layer

Data Link Layer bertanggung jawab untuk menyediakan komunikasi dari *node* ke *node* pada satu jaringan lokal. Untuk menyediakan pelayanan ini, lapisan data link harus melakukan dua fungsi. Yang pertama lapisan tersebut harus menyediakan mekanisme pengalamatan berdasarkan *physical address* yang memungkinkan pesan–pesan untuk dikirimkan ke *node* yang benar. Yang kedua lapisan ini juga menerjemahkan pesan–pesan dari lapisan yang lebih tinggi menjadi bit–bit yang dapat ditransmisikan oleh lapisan physical. Ketika lapisan data link menerima suatu pesan untuk ditransmisikan, lapisan ini membentuk pesan tersebut menjadi frame data (Heywood, D. 1996). Gambar 2.9 menyajikan format dari suatu frame, yang terdiri atas :

- Start Indicator. Pola bit tertentu yang menunjukkan awal dari suatu frame data.
- Source Address. Alamat dari *node* pengirim juga dimasukan sehingga jawaban pesan – pesan dapat diketahui dengan benar.
- Destination Address. Setiap *node* dikenali oleh satu alamat. Lapisan data link dari pengirim menambahkan alamat tujuan pada frame. Lapisan data link dari penerima melihat alamat tujuan untuk mengenali pesan – pesan yang harus diterimanya.
- Control. Dalam banyak hal, informasi kontrol tambahan harus dimasukan. Informasi yang dimasukan ditentukan oleh setiap protokol.

- Data. Field ini mengandung semua data yang diteruskan ke lapisan data link.

  Error Control. Field ini mengandung informasi yang memungkinkan *node* yang menerima untuk menentukan apakah terjadi kesalahan selama pengiriman. Pendekatan yang umum adalah *Cylic Redundancy Checksum* (CRC), yang merupakan nilai yang telah dihitung yang merangkum semua data pada frame. *Node* yang mengirim menghitung checksum dan menyimpannya di frame. Penerima menghitung kembali checksum tersebut. Jika CRC hasil perhitungan penerima sesuai dengan nilai CFC dalam frame, dapat dianggap bahwa frame tersebut telah dikirimkan tanpa kesalahan (Heywood, 1996).

Start Indicator	Source Address	Destination Address	Control	Data	Error Control

Gambar 2.9 Contoh frame data

2.3. IP Address

IP Address adalah sebuah alamat yang diberikan ke peralatan jaringan untuk mengakses internet atau ke suatu jaringan komputer dengan menghubungkan protokol TCP/IP. Setiap komputer yang terhubung ke jaringan komputer harus mempunyai alamat yang unik. Subnetmask berfungsi untuk membedakan bagian mana dari IP Address tersebut yang disebut *net-id* dan bagian mana yang disebut *host-id. Net-id* menyatakan bahwa *host* tersebut menjadi anggota dari suatu kelompok yang ada dalam suatu jaringan. Sedangkan *host-id* menyatakan nomor identitas yang membedakan antara *host* satu dengan lainnya dalam suatu *net-id*.

IP Address terdiri dari bilangan biner sepanjang 32 bit yang dibagi atas 4 segmen. Tiap segmen terdiri atas 8 bit yang berarti memiliki nilai desimal dari 0 sampai 255. Rentang alamat yang bisa digunakan ditunjukkan pada tabel 2.1 :

Tabel 2.1 IP Address

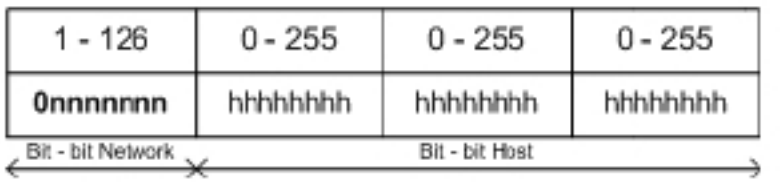
Mulai	00000000. 00000000. 00000000. 00000000
Sampai	11111111. 11111111. 11111111. 11111111

Ada sebanyak 2^{32} kombinasi alamat IP Address yang bisa dipakai di seluruh dunia (walaupun pada kenyataan ada sejumlah IP Address yang digunakan untuk keperluan khusus). Untuk memudahkan pembacaan dan penulisan, IP Address biasanya direpresentasikan dalam bilangan desimal. Jadi rentang alamat diatas dapat dirubah menjadi address 0.0.0.0 sampai dengan 255.255.255.255. nilai desimal dari IP Address inilah yang dikenal dalam pemakaian sehari-hari.

2.3.1 Pembagian Kelas IP Address

Ada 3 kelas IP Address yang utama dalam TCP/IP, yakni kelas A, B, C. *Internet Protokol* menentukan pembagian jenis kelas ini dengan menguji beberapa bit pertama dari IP Address. Penentuan kelas dilakukan dengan cara berikut :

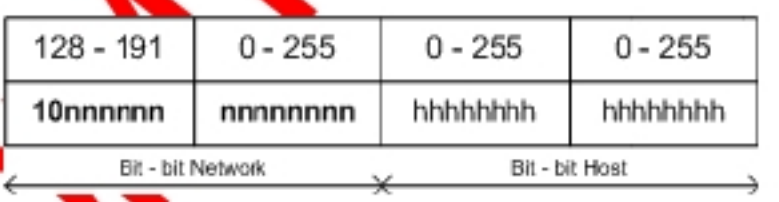
- Jika bit pertama dari IP Address adalah 0, alamat tersebut merupakan network kelas A. Bit ini dan 7 bit berikutnya (8 bit pertama) merupakan bit network sedangkan 24 bit terakhir merupakan bit *host*. Dengan demikian hanya ada 126 network untuk kelas A, yakni dari nomor 1. s/d 126.xxx.xxx.xxx, dan setiap network dapat menampung 16.277.214 *host.*



Gambar 2.10 IP Address Kelas A

Pada gambar 2.10 dapat dilihat *netid* memiliki rentang nilai biner dari 00000001 sampai dengan 01111110 dimana tanda "n" melambangkan nilai dari bilangan biner untuk *netid*. Sedangkan untuk *hostid* memiliki range biner dari 000000000 sampai dengan 11111111 dimana tanda "h" melambangkan nilai dari bilangan biner untuk *hostid*.

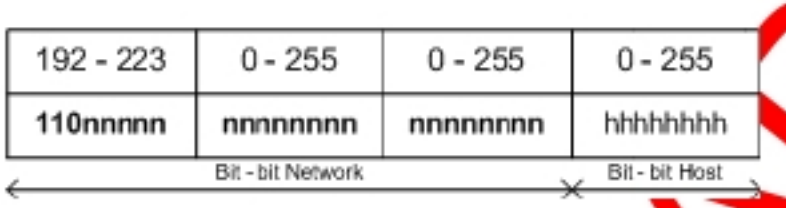
- Jika dua bit pertama dari IP Address adalah 10, alamat tersebut merupakan network kelas B. Dua bit ini dan 14 bit berikutnya (16 bit pertama) merupakan bit network, sedangkan 16 bit terakhir merupakan bit *host*. Dengan demikian terdapat 16.384 *netid*, yakni dari network 128.0.xxx.xxx s/d 191.255.xxx.xxx. setiap network kelas B mampu menampung 65.543 *host*



Gambar 2.11 IP Address Kelas B

Pada gambar 2.11 dapat dilihat *netid* memiliki rentang nilai biner dari 10000000 sampai dengan 10111111 pada segmen pertama, dimana tanda "n" melambangkan nilai dari bilangan biner untuk *netid*. Sedangkan untuk *hostid* memiliki range biner dari 000000000 sampai dengan 11111111 dimana tanda "h" melambangkan nilai dari bilangan biner untuk *hostid*.

- Jika tiga bit pertama dari IP Address adalah 110, alamat tersebut merupakan network kelas C. Tiga bit ini dan 21 bit berikutnya (24 bit pertama) merupakan bit network, sedangkan 8 bit terakhir merupakan bit *host*. Dengan demikian terdapat 2.097.152 *netid*, yakni dari network 192.0.0.xxx s/d 223.255.255.xxx. setiap network kelas C mampu menampung 254 *host*.



Gambar 2.12 IP Address Kelas C

Pada gambar 2.12 dapat dilihat *netid* memiliki rentang nilai biner dari 11000000 sampai dengan 11011111 pada segmen pertama, dimana tanda "n" melambangkan nilai dari bilangan biner untuk *netid*. Sedangkan untuk *hostid* memiliki rentang biner dari 000000000 sampai dengan 11011111 dimana tanda "h" melambangkan nilai dari bilangan biner untuk *hostid*.

Selain ketiga kelas diatas, ada dua kelas lagi yang ditujukan untuk pemakaian khusus, yakni kelas D dan kelas E. Jika 4 bit pertama adalah 1110, IP Address untuk kelas D yang digunakan untuk *multicast*, yakni sejumlah komputer yang bersama-sama memakai suatu aplikasi. Salah satu contoh penggunaan multicast address yang sedang berkembang saat ini di internet adalah aplikasi untuk *real-time video conference* yang melibatkan lebih dari dua *host* (multipoint), menggunakan *Multicast Backbone* (MBone). Kelas terakhir adalah kelas E (4 bit pertama adalah 1111 atau sisa dari seluruh kelas). Pemakaiannya dicadangkan untuk kegiatan eksperimental. (Tanenbaum, 2000).

2.4. Neural Network

2.4.1. Konsep dan Definisi

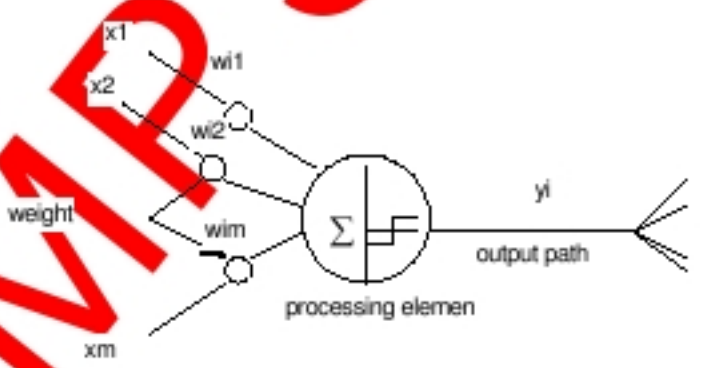
Neural Network (Jaringan Saraf Tiruan) dapat pula diartikan sebagai suatu jaringan dengan susunan tertentu yang terdiri dari elemen-elemen yang diharapkan dapat berhubungan dengan keadaan nyata sama seperti sistem saraf biologis, (Kohonen, 1988). Neural Network juga dipandang sebagai suatu sistem yang terdiri dari elemen-elemen pengolah yang mempunyai kemampuan memperbaiki dirinya dengan belajar. Neural Network juga suatu model yang sangat sederhana bila dibandingkan dengan kerumitan jaringan saraf biologis. Pada saat ini banyak peneliti telah mengembangkan Neural Network sebagai alat bantu dalam banyak bidang seperti pada pengenalan pola, diagnosa, pengontrolan, pengolahan informasi perencanaan dan sebagainya. Elemen-elemen struktural dari Neural Network secara umum adalah :

- Simpul-simpul
- Lapisan-lapisan
- Sambungan antar simpul
- Paket bobot sambungan

Simpul pada Neural Network mirip dengan *neuron*. Neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan. Simpul-simpul tersebut diatur dalam lapisan-lapisan dan melewatkan data melalui sambungan antar simpul. Sambungan antar simpul ini mirip dengan gabungan antar axon, synapsis dan dendrite. Dendrite merupakan percabangan berupa serat-serat dari bodi sel. Axon merupakan penghubung lebih lanjut dari body sel dan pembawa sinyal dari neuron. Akhir dari axon terpisah dalam bentuk

uraian-uraian serabut, setiap serabut berakhir pada suatu organ yang disebut synapsis, tempat neuron memberikan sinyal pada neuron lainnya. Penerimaan akhir pada hubungan tersebut dapat terjadi pada dendrite atau bodi sel. Gambaran hubungan secara skematik antara sistem neuron biologis dan sistem neuron buatan terlihat pada gambar 2.13 dan gambar 2.14 dibawah ini : (Siang, J. J. 2005)

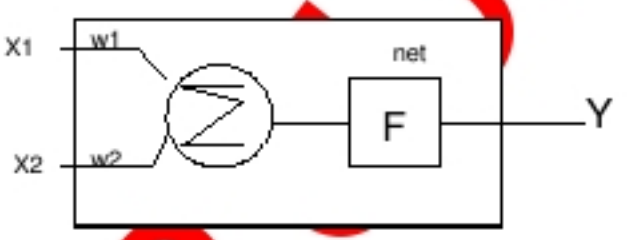
Gambar 2.13 Sistem neuron secara biologis



Gambar 2.14 Sistem neuron secara buatan (Mc Culloch dan Pitts)

Setiap sambungan mempunyai sebuah nilai yang dapat diubah-ubah besarnya yang disebut bobot. Bobot sambungan ini mirip dengan kekuatan synaptis dari jaringan saraf biologis. Simpul-simpul dianggap sebagai elemen perhitungan atau elemen proses karena didalamnya terjadi penjumlahan masukan-masukan yang diterima terlebih dahulu dikalikan dan dengan bobot sambungan

yang dibawa tiap masukan. Hasil penjumlahan itu disebut dengan *net input* atau nilai masukan untuk suatu simpul. *Net input* kemudian dimasukkan ke dalam suatu fungsi yang disebut fungsi transfer atau fungsi aktivasi untuk menghasilkan suatu nilai yang dinamakan nilai aktivasi. Nilai aktivasi ini disebut juga dengan keluaran simpul. Untuk simpul-simpul pada lapisan masukan, nilai aktivasinya adalah data atau nilai masukan itu sendiri tanpa memasukkannya lagi ke dalam fungsi aktivasi, karena fungsi lapisan masukan hanya menerima dan melewatkan data yang masuk. Berbeda dengan lapisan-lapisan yang lain karena harus lebih dahulu mengolah masukan yang masuk kepadanya. Model dari suatu elemen pemroses sederhana dapat dilihat pada gambar 2.15 dibawah ini :



Gambar 2.15 Neuron buatan

Simpul Y menerima masukan dari simpul-simpul X1 dan X2 (Simpul X disebut simpul masukan). Bobot-bobot sambungan dari X1 dan X2 ke simpul Y masing-masing adalah w1 dan w2 maka net input (y_in) atau nilai masukan simpul untuk simpul Y adalah :

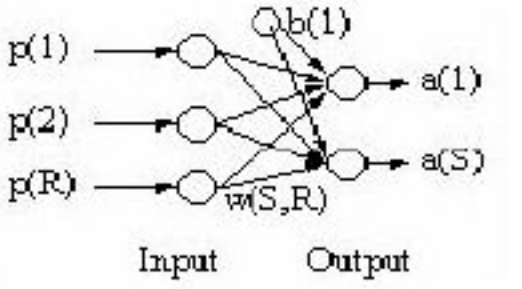
$$y\_in = x_1 w_1 + x_2 w_2$$

Dan nilai aktivasi (y) untuk simpul Y adalah y = f(y_in) dengan f adalah fungsi aktivasi atau fungsi transfer dari simpul Y.

Secara umum terdapat tiga jenis neural network berdasarkan jenis networknya, yaitu : (Siang, J. J. 2005)

A. *Single Layer Neural Network*

Dalam jaringan ini, sekumpulan masukan neuron dihubungkan langsung dengan sekumpulan keluarannya. Gambar 2.16 menunjukkan arsitektur jaringan dengan R unit masukan (p1, p2, ... pR) , b adalah bias (b1) dan S buah unit keluaran (a1, ... aS). Dalam jaringan ini, semua unit masukan dihubungkan dengan semua unit keluaran, meskipun dengan bobot yang berbeda-beda. Tidak ada unit masukan yang dihubungkan dengan unit masukan lainnya. Demikian pula dengan unit keluaran. Besaran w (S,R) menyatakan bobot hubungan antara unit ke-R dalam masukan dengan unit ke-S dalam keluaran. Selama proses pelatihan, bobot-bobot tersebut akan dimodifikasi untuk meningkatkan keakuratan hasil. Model semacam ini tepat digunakan untuk pengenalan pola karena kesederhanaannya.

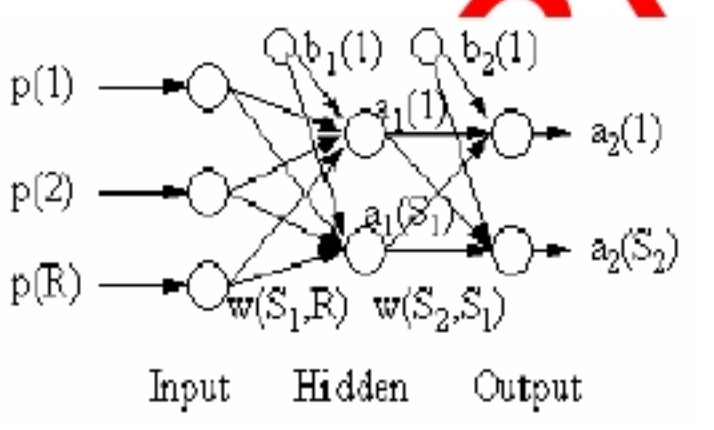


Gambar 2.16 Single layer neural network

B. *Multilayer Neural Network*

Neural network jenis ini merupakan perluasan dari *single layer network*. Dalam jaringan ini, selain unit masukan dan keluaran, ada unit-unit lain yang disebut sebagai *hidden layer* (lapisan tersembunyi). Sama seperti pada unit masukan dan keluaran, unit-unit dalam satu layer tidak saling berhubungan. Gambar 2.17 adalah jaringan dengan R buah unit masukan (p1,p2,... pR), sebuah lapisan tersembunyi yang terdiri dari S1 buah unit (a1(1),... a1(S1)), b1 adalah

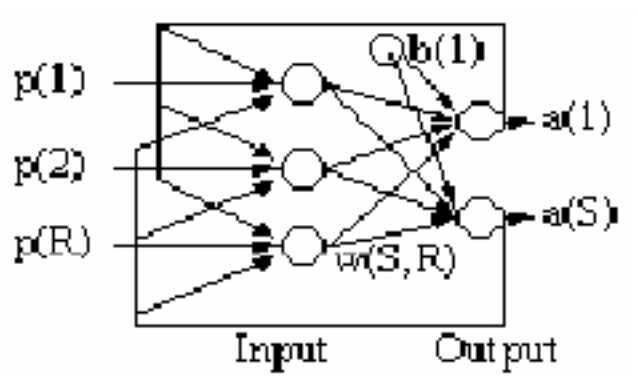
bias untuk unit masukan ke unit hidden, b2 adalah bias untuk unit hidden ke keluaran, dan S2 buah unit keluaran (a2(1),... a2(S2)). Besaran w (S1,R) menyatakan bobot hubungan antara unit ke-R dalam masukan dengan unit ke-S1 dalam hidden, sedangkan besaran w (S2,S1) menyatakan bobot hubungan antara unit ke-S1 dalam hidden dengan unit ke-S2 dalam keluaran. *Multilayer neural network* dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan *single layer neural network*, meskipun kadangkala proses pelatihan lebih kompleks dan lama.



Gambar 2.17 Multilayer neural network

C. *Recurrent Neural Network*

Neural network jenis ini memiliki ciri yaitu adanya koneksi umpan balik dari keluaran ke masukan. Kelemahannya adalah *Time Delay* akibat proses umpan balik dari keluaran ke titik masukan.



Gambar 2.18 Recurrent neural network

2.4.2. Tahap Pembelajaran

Karena Neural Network dibuat berdasarkan inspirasi dari jaringan saraf-saraf biologis (otak manusia). Salah satu kemampuan yang diharapkan adalah belajar. Kemampuan belajar tersebut ditunjukkan dengan tingkah lakunya terhadap lingkungan. Misal suatu paket masukan diperlihatkan kepada neural network lengkap dengan keluaran yang diharapkan (untuk model jaringan saraf tertentu keluaran tidak diperlihatkan) maka neural network akan mengatur dirinya sendiri (dengan metode belajar tertentu) sedemikian rupa sehingga menghasilkan tanggapan-tanggapan seperti yang telah ditetapkan.

Istilah belajar (*learning*) pada neural network adalah suatu proses pengaturan bobot-bobot sambungan antar simpul yang dilakukan dengan metode tertentu sehingga mendapatkan bobot-bobot sambungan yang diinginkan. Proses belajar terjadi karena antara keluaran yang dihasilkan jaringan tidak sama dengan keluaran yang diharapkan. Dengan kata lain bila bobot sambungan yang ada belum mampu menghasilkan keluaran yang diharapkan maka bobot akan diatur metode tertentu melalui latihan terus-menerus sehingga mendapatkan susunan bobot yang baru dan lebih baik. Dengan susunan bobot ini maka perbedaan antara keluaran jaringan dan keluaran pola akan semakin kecil. Dengan demikian berarti keluaran yang dikeluarkan jaringan akan sama dengan keluaran yang diinginkan (keluaran target) atau setidak-tidaknya mendekati. Setelah dilatih, maka suatu neural network siap diuji dan dipakai. Dan pengujian dikatakan berhasil apabila neural network sanggup menerima masukan dan mengeluarkan keluaran seperti yang diharapkan.

Dalam berlatih terdapat istilah himpunan pelatihan yaitu suatu himpunan yang terdiri dari pasangan-pasangan pola masukan dan pola keluaran yang diinginkan (*output target*). Setiap satu pola masukan dipasangkan dengan satu pola keluaran target. Istilah pola digunakan sebagai pengganti dari semua informasi atau data-data yang masuk dalam neural network. Selama proses belajar, bobot-bobot sambungan yang ada pada jaringan secara konvergen menuju suatu harga tertentu yang menandakan semakin kecilnya kesalahan (*error*) antara keluaran target dengan keluaran jaringan.

Penyediaan keluaran dan masukan ini telah membagi metode belajar pada jaringan saraf tiruan menjadi dua : (Fausett, L. 1994).

A. *Supervised Learning*

Metode ini membutuhkan pasangan pola pelatihan dari tiap pola masukan dengan pola keluaran yang diinginkan. Penyediaan masukan dan keluaran dilakukan secara eksplisit sehingga tiap pasangan dapat menunjukkan keluaran apa yang harus dihasilkan untuk masukan yang diberikan. Pola masukan dimasukkan ke dalam jaringan yang kemudian setelah diolah akan menghasilkan suatu keluaran yang disebut keluaran jaringan. Bila keluaran jaringan tidak sama dengan keluaran target, maka keluaran target harus dikurangkan dengan keluaran jaringan. Selisih dari kedua keluaran tersebut menyatakan kesalahan (*error*) yang digunakan untuk mengubah bobot-bobot sambungan, sehingga kesalahan semakin mengecil pada siklus pelatihan berikutnya.

B. *Unsupervised Learning*

Metode ini tidak memerlukan pola keluaran target untuk keluarannya, sehingga tidak ada perbandingan antara keluaran target dengan keluaran jaringan. Kumpulan pola pelatihannya hanya berupa pola masukan. Pola masukan dimasukkan dalam jaringan dan jaringan sendiri yang harus *self organizing* sedemikian rupa sehingga menghasilkan keluaran yang sama dengan masukannya.

2.4.3. Faktor – Faktor Pembelajaran

Ada beberapa parameter-parameter yang turut menentukan keberhasilan suatu proses belajar. Parameter-parameter ini meliputi inisialisasi bobot, kecepatan pembelajaran (*learning rate*), kesalahan belajar dan batas iterasi. Faktor-faktor belajar tersebut sangat berpengaruh pada proses pembelajaran pada metode backpropagation. (Fausett, L. 1994).

A. Inisialisasi bobot

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya. Bobot awal tidak boleh terlalu kecil karena akan menghasilkan nilai aktivasi yang kecil sehingga kemungkinan keluaran jaringan mendekati 0 (*zero*) dan sangat memperlambat proses pembelajaran. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi aktivasinya menjadi sangat kecil juga. Oleh karena itu bobot diisi dengan bilangan acak kecil. Salah satu pilihan yang dapat digunakan adalah memilih nilai bobot pada suatu rentang kawasan nilai antara –0,5 dan 0,5 atau beberapa interval lain yang seimbang (Fausett, L. 1994). Banyak studi

empiris membuktikan bahwa meneruskan pelatihan pada saat error mencapai harga kecil yang stabil atau datar menghasilkan bobot yang tak diinginkan. Ini dapat menyebabkan kesalahan meningkat. Untuk mengatasi hal tersebut sebaiknya nilai bobot diberi harga yang random.

B. Kecepatan Belajar (α)

Parameter kecepatan belajar menunjukkan intensitas dalam proses belajar, parameter ini juga menentukan efektifitas dan konvergensi dari pelatihan. Umumnya harga optimum dari kecepatan belajar tergantung dari permasalahan yang dihadapi dan tidak ada suatu harga yang cocok untuk semua kasus pelatihan. Nilai α yang besar dapat menyebabkan meningkatnya kecepatan dalam menentukan nilai yang diinginkan tetapi juga memungkinkan adanya suatu *overshoot* (pencapaian nilai yang terlalu cepat, sehingga jaringan menjadi tidak terlalu peka). Sedangkan nilai α yang kecil dapat menyebabkan pencapaian nilai yang diinginkan menjadi lebih lama/lamban, akan tetapi memungkinkan jaringan menjadi lebih peka. Penentuan harga kecepatan belajar dipilih pada harga antara 0 sampai dengan 1. (Fausett, L. 1994).

C. Kesalahan belajar dan batas iterasi

Tingkat keberhasilan proses belajar dari sebuah jaringan yang sedang dilatih dapat dilihat dari *error*. Error menunjukkan kemampuan pada jaringan. error tidak harus bernilai nol, tapi bernilai cukup kecil tergantung masalah yang dihadapi, karena semakin kecil nilai error berarti jaringan semakin peka. Ini berakibat kurang baik karena biasanya kita menginginkan sebuah sistem

pengenalan yang dapat mengabaikan perubahan minor sampai level tertentu. Untuk menghindari proses training jaringan terjebak dalam suatu *looping* yang terus-menerus maka sebelumnya diberikan batasan iterasi maksimum. Sedangkan untuk mendapatkan performansi sistem yang diinginkan maka diberikan batasan error minimum. Batasan ini juga merupakan upaya dalam proses penghentian iterasi training karena jaringan telah mencapai performansi tertentu yang diinginkan atau jaringan tidak mampu belajar dengan baik.

2.4.4. Algoritma Backpropagation

Sebuah jaringan *single layer neural network* yang menggunakan metode *backpropagation* terdiri dari 3 langkah pembelajaran / pelatihan yaitu :

A. Pelatihan pola input secara *feedforward*

Sebelum masuk ke dalam pelatihan feedforward terlebih dahulu dilakukan inisialisasi bobot (*w*) secara random dengan interval antara -0.5 s/d 0.5 , untuk setiap masukan (*input*) diinisialisasi dengan rumus :

$(X_i, i = 1, \ldots n) \sim$ input1, input2, …. input[n] ..... (1)

Untuk tahapan selanjutnya dalam feedforward akan dijelaskan dibawah ini :

Selama kondisi berhenti masih salah, kerjakan :

1. Untuk masing-masing pasangan latihan, lakukan : feedforward
   a. Masing-masing unit *input* (X_i , $i = 1,\ldots,n$) menerima sinyal *input* X_i dan menyebarkan sinyal ini ke semua unit lapisan atas.
   b. Masing-masing unit *output* (Y_k, $k = 1,\ldots,m$) menjumlahkan bobot sinyal *input* dengan rumus :

$$y\_in = \sum_{i=1}^{n} x_i \; w_i \qquad \text{..... (2)}$$

c. Aplikasikan fungsi aktivasi untuk menghitung sinyal keluaran (*output*) dengan rumus :

$$y_k = f(y\_in) \qquad \text{..... (3)}$$

2. Lanjut ke *backpropagation* dibawah.

B. Perhitungan dan *backpropagation* dari kumpulan kesalahan

Dalam proses ini nilai sinyal *output* setelah melalui fungsi aktivasi akan masuk dalam tahap backpropagation guna melakukan koreksi lebih lanjut (perhitungan *error*). Dalam tahap ini yang dilakukan adalah :

Untuk masing-masing pasangan pelatihan, lakukan : backpropagation.

Masing-masing unit *output* (Y_k, k = 1,…,m) menerima sebuah pola *target* yang bersesuaian dengan pola *input* pelatihan, menghitung informasi kesalahan (*error*). Digunakan rumus :

$$E = \frac{1}{2} (t_k - y_k)^2 \qquad \text{.... (4)}$$

C. Penyesuaian bobot (*update weight*)

Setelah diketahui besar error yang terjadi maka akan dilakukan perbaikan bobot dengan tujuan mendekati *error minimum*. Untuk perbaikan bobot dilakukan dengan :

1. Masing-masing unit *output* (Y_k, k = 1,…,m) memperbaiki bobot dengan rumus :

$$w_{ik}\,(new) = w_{ik}\,(old) + \alpha(t - y\_in)\, x_i \qquad \text{.... (5)}$$

2. Tes Kondisi berhenti.

Agar jaringan dapat mengenali suatu pola tertentu, maka pola masukan harus dilatih (*training*) terlebih dahulu. Dan untuk mengetahui karakteristik jaringan maka perlu dilakukan pelatihan yang berulang agar dicapai suatu struktur jaringan yang tepat dalam proses pengelompokan keluaran (*output*). Pada proses pelatihan digunakan parameter seperti konstanta kecepatan belajar dan harga inisialisasi awal bobot yang bervariasi. Istilah belajar adalah suatu proses pengaturan bobot-bobot sambungan antar simpul sehingga keluaran yang dihasilkan jaringan mendekati atau sama dengan keluaran yang diharapkan. Setelah mendapatkannya, maka akan memperoleh konfigurasi jaringan yang diharapkan sehingga dapat melakukan proses pengujian (proses pengujian hanya melibatkan tahap feed forward saja). Pengujian dikatakan berhasil apabila jaringan sanggup menerima masukan dan menghasilkan keluaran seperti yang diharapkan.

2.5. *Intrusion Detection System* (IDS)

2.5.1. Analisis Pendeteksian Serangan

Intrusion adalah usaha untuk masuk dan menyalahgunakan sistem yang ada. *Intrusion Detection* adalah proses untuk *monitoring event* yang terjadi pada sistem komputer atau jaringan komputer dan melakukan analisis data tersebut untuk mengetahui adanya *intrusion* terhadap mekanisme pengamanan yang ada. Dalam mengenali sebuah serangan yang dilakukan oleh hacker atau cracker,

biasanya digunakan dari data-data yang telah diperoleh. Pendekatan yang sering digunakan untuk mengenali serangan antara lain :

1. *Anomaly detection.*

   Anomaly detector mengidentifikasi perilaku tak lazim yang terjadi (anomaly) dalam host atau network. Detektor berfungsi dengan asumsi bahwa serangan itu berbeda dengan aktifitas normal dan karena itu dapat dideteksi dengan sistem yang mampu mengidentifikasikan perbedaan tersebut. *Anomaly detector* menyusun profil-profil yang merepresentasikan kebiasaan user yang normal, host, atau koneksi jaringan. Profil-profil ini di bangun dari data-data historis yang dikumpulkan dalam periode operasi normal. Kemudian detektor mengumpulkan data-data peristiwa dan menggunakan langkah-langkah yang beragam ketika aktivitas yang diamati menyimpang dari normal.

2. *Misuse Detection.*

   Detektor melakukan analisis terhadap aktifitas sistem, mencari event atau set event yang cocok dengan pola perilaku yang dikenali sebagai serangan. Pola perilaku serangan tersebut disebut sebagai signatures, sehingga misuse detection banyak dikenal sebagai *signatures-based detection*.

2.5.2. Jenis IDS

Jenis-jenis IDS didasarkan atas sumber data yang diperolehnya, dapat dibedakan menjadi tiga yaitu :

1. Network-Based IDS.

   Network-based IDS menggunakan network adapter sebagai alat untuk menangkap paket-paket yang akan dipantau. *Network Adapter* berjalan pada

mode *prosmicuous* untuk memonitor dan melakukan analisis paket-paket yang berada di dalam suatu jaringan komputer. Mode *prosmicuous* adalah mode dimana network adapter dipaksa menangkap semua paket-paket data yang melewati suatu jaringan komputer, walaupun paket-paket data tersebut tidak ditujukan pada alamat network adapter tersebut.

2. Host-Based IDS.

   Host-based IDS memperoleh informasi dari data yang dihasilkan oleh sistem pada sebuah komputer yang diamati. Data host-based IDS biasanya berupa log yang dihasilkan dengan memonitor pada sistem file, event, dan keamanan pada windows NT dan syslog pada lingkungan sistem operasi UNIX. Saat terjadi perubahan pada log tersebut, dilakukan analisis apakah sama dengan pola serangan yang ada pada basis data IDS.

3. Hibrid IDS.

   Hibrid IDS merupakan gabungan dari Network-based IDS dan Host-based IDS. Gabungan dari kedua teknologi tersebut meningkatkan resistansi jaringan terhadap serangan dan penyalahgunaan, serta meningkatkan pelaksanaan kebijakan keamanan.

2.5.3. Kelebihan dan Keterbatasan IDS

Intrusion detection system merupakan sistem pengamanan yang digunakan sebagai *secondary* sistem yang telah ada atau sistem yang dibuat untuk membackup sistem keamanan yang utama. Sistem pengamanan utama seperti firewall, enkripsi dan autentikasi sangat handal untuk digunakan.

Kelebihan dari IDS antara lain (Bace, R & Mell, P . 2000) :

1. Monitoring dan analisis sistem dan perilaku pengguna.
2. Pengujian terhadap konfigurasi keamanan sistem.
3. Memberikan titik acuan untuk pelaksanaan keamanan sistem dan penelusuran pada perubahan pada titik acuan.
4. Pengenalan serangan menurut pola yang telah diketahui.
5. Pengenalan pola aktifitas yang tidak normal.
6. Manajemen audit sistem operasi dan mekanisme *logging* pada data yang dihasilkan.
7. Memberikan peringatan pada administrator jika terjadi serangan.

IDS memiliki beberapa keterbatasan dalam aktifitasnya, antara lain :

1. IDS tidak dapat mengenali teknik baru yang belum terdapat pada basis data pola serangan yang dimiliki.

Konsep yang digunakan IDS tidak benar-benar akurat, karena sistem tersebut dapat ditumpuhkan oleh hacker. Namun bukan berarti IDS tidak valid untuk digunakan, sebuah IDS yang baik dapat meningkatkan keamanan suatu sistem yang berada pada jaringan. hal yang sangat penting untuk diperhatikan adalah bahwa IDS digunakan sebagai pelengkap sistem pengamanan yang utama.

2.6. Snort (*Lightweight Intrusion Detection System*)

Snort adalah software yang dikembangkan oleh Martin Roesch. Awalnya snort dibuat untuk unix *operating system* (sistem operasi), tetapi sekarang sudah dapat digunakan pada operating system lain seperti windows. Snort merupakan

software *open source* yang berfungsi sebagai *intrusion detection system* yang mampu mendeteksi pola (*pattern*) pada paket yang lewat dan mengirimkan *alert* (pesan kesalahan) jika pola tersebut terdeteksi. Pola-pola atau rules disimpan dalam berkas yang disebut *library* yang dapat dikonfigurasi sesuai dengan kebutuhan.

Snort dikonfigurasi dengan menggunakan *command line switches* dan *optional Berkeley Packet Filter (BPF) commands*. Pada dasarnya Snort dibuat dengan menggunakan *library libpcap* yang populer (library yang sama seperti yang digunakan oleh tcpdump untuk melakukan *packet sniffing*). Snort adalah *lightweight intrusion detection system* yang non-intrusive, mudah dikonfigurasi, menggunakan metoda yang sudah dikenal (*familiar methods*) untuk pegembangan rulenya, dan memerlukan hanya beberapa menit untuk menginstal. Dengan fasilitas yang dimilikinya snort mempunyai kemampuan untuk :

1. Mendeteksi sekaligus memberikan alert berdasarkan pola yang sesuai dengan ancaman seperti *buffer overflows*, *stealth port scan*, *CGI attacks*, *NMAP portscan*, *well-known backdoors*, dan masih banyak lagi.
2. Menggunakan fasilitas *syslog*, *WinPopUp messages*, atau file untuk memberitahukan administrator.
3. Mengembangkan rule baru dengan cepat bila pola (*attack signature*) telah diketahui.
4. Mencatat paket-paket dalam bentuk yang dapat dibaca oleh manusia dari IP address yang yang menjadi sumber gangguan dalam struktur direktori yang mempunyai hirarki (*hierarchial derectory structure*).

2.6.1. Arsitektur Snort (*Snort Architecture*)

Arsitektur Snort (*Snort architecture*) terdiri dari tiga komponen yang prinsip yaitu :

1. Packet Decoder

   Snort packet decoder mendukung Ethernet, SLIP dan PPP mediums. Packet decoder melaksanakan semua pekerjaan untuk mempersiapkan data secara praktis agar dapat digunakan oleh *detection engine*.

2. Detection Engine

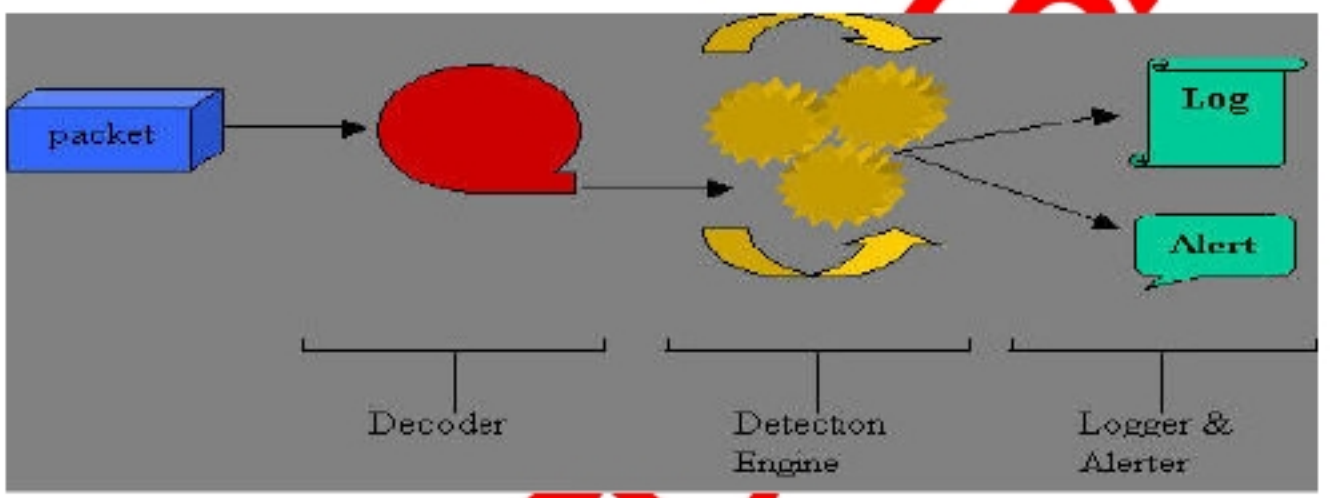
   Merupakan komponen inti dari snort, yang bertanggung jawab untuk menganalisa setiap paket berdasarkan pada snort rule yang di-*load* pada saat *runtime*. Detection engine memisahkan snort rule menjadi chain header dan chain option. Detection engine secara rekursif menganalisa setiap paket dengan menggunakan rule yang telah didefinisikan dalam file rule snort . Rule pertama yang sesuai dengan paket yang didekode akan memicu suatu aksi yang telah ditentukan dalam rule definition. Sebuah paket yang tidak sesuai dengan snort rule set tertentu akan diabaikan. Komponen kunci dari *detection engine* adalah *plugin module* seperti portscan module. Plugin module meningkatkan fungsi snort dengan menambahkan fasilitas untuk analisis.

3. Logger / Alerter

   Logging dan alerting merupakan dua subkomponen yang terpisah. Logging mengijinkan untuk mencatat (*log*) informasi yang dikumpulkan oleh packet decoder dalam format tcpdump atau yang dapat dibaca oleh manusia. Alert dapat dikonfigurasi untuk dikirim ke syslog, UNIX socket atau database.

Untuk keperluan testing alert dapat di disable. Semua log ditulis dalam folder /var/log/snort, dan alert ditulis pada file /var/log/snort/alerts.

Representasi yang sederhana dari komponen-komponen tersebut diperlihatkan pada gambar 2.19 dibawah ini :



Gambar 2.19 Arsitektur snort

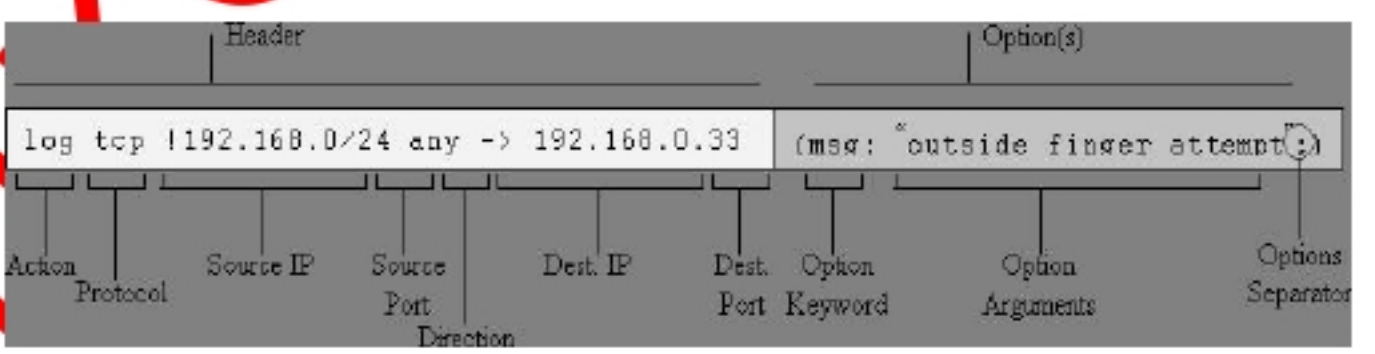
2.6.2. Aturan-Aturan pada Snort (*Snort Rules*)

Snort rule file merupakan file ASCII yang dapat dibuat dengan menggunakan text editor. Rule file terdiri dari:

1. *Variable Definitions*: definisi variabel ini dapat digunakan kembali (*reuse*) dalam membuat snort rule.
2. *Snort rule*: merupakan rule-rule aktual yang mengaktifkan intrusion detection. Dan harus konsisten dengan keseluruhan policy dari intrusion detection.
3. *Preprocessor*: merupakan sinonim dari plugins yang merupakan instrumental dalam perluasan kemampuan dari snort. Sebagai contoh, module portscan mengijinkan snort untuk mendeteksi port scan.

4. *Include file*: bagian ini mengijinkan anda untuk mengikutkan snort rule pada file yang lain.
5. *Output modules*: bagian ini mengijinkan adminsitrator untuk menyeleksi keluaran untuk keperluan logging dan alerting. *Output modules* dijalankan bila subsistem alert atau logging dari snort dipanggil.

Snort menggunakan rule description language yang sederhana dan mudah dalam penulisannya. Kebanyakan snort rule ditulis dalam single line, sedangkan rule yang lebih dari satu baris (*multiple lines*) dapat ditulis dengan menambahkan sebuah backslash (\ ) di akhir suatu baris. Snort rule dapat dibagi dalam dua bagian (*logical section*), yaitu : rule header dan rule option. Rule header berisikan informasi dari rule action (*alert, log, pass, activate, dynamic*), protocol (*tcp, udp, icmp, ip*), source dan destination IP addrees dan netmask, source dan destination port. Sedangkan rule option berisikan pesan *alert* (yang tersedia dalam mode *full, fast, socket, syslog, smb, none*) dan informasi mengenai bagian paket yang harus diperiksa (*inspected*) untuk menentukan perlu tidaknya mengambil rule action. Gambar 2.20 di bawah ini memperlihatkan contoh format dari snort rule.



Gambar 2.20 Contoh rule pada snort

Text yang berada di luar tanda kurung adalah rule header dan text yang berada di dalam tanda kurung merupakan rule option. Sedangkan kata-kata yang terdapat sebelum tanda titik dua pada rule option disebut sebagai option keyword. Dengan demikian rule header dari contoh rule diatas adalah **log tcp ! 192.168.0/24 any -> 192.168.0.33** dan rule optionnya adalah **(msg: "outside finger attempt" ;)** sedangkan option keywordnya adalah **msg**.

Rule option section bukan suatu hal spesifik yang dibutuhkan oleh rule tertentu, rule ini digunakan untuk menerangkan definisi paket yang akan dikumpulkan atau diberi alert. Semua element yang menyusun suatu rule harus benar untuk mengindikasikan rule action yang akan diambil. Rule-rule action tersebut memberitahukan snort apa yang harus dilakukan bila menemukan paket yang memenuhi kriteria yang telah ditentukan. Contoh aksi yang dilakukan snort misalnya membangkitkan alert dengan menggunakan alert method yang ditentukan, mencatat paket atau mengabaikan paket tersebut.

Sebelum menulis snort rule direkomendasikan untuk mendokumentasi policy NIDS. Policy ini berisi penjelasan tentang aktifitas yang di inginkan untuk di catat (*log*), diabaikan (*ignore*) atau diberi alert apabila terdeteksi.

2.6.3. Menjalankan Snort (*Running Snort*)

Pada dasarnya ada tiga mode utama dimana snort dapat dikonfigurasi yaitu : mode packet sniffer, mode packet logger dan mode network intrusion detection system (NIDS).

1. Mode Sniffer

Dalam mode ini, Snort memonitor traffic paket-paket yang melewati jaringan komputer, dimana ethernet dibuat dalam mode promiscous sehingga dapat menanggkap semua trafik di dalam jaringan.

Contoh command untuk mode ini adalah :

./snort –vd

2. Mode Packet Logger

Pada saat snort run pada mode ini, ia mencatat setiap paket yang dilihatnya dan menempatkan paket-paket tersebut pada suatu tempat (misalnya ke dalam disk).

Contoh command untuk mode ini adalah:

./snort –dev –1 ./log

3. Mode NIDS (Network Intrusion Detection System)

Mode ini memungkinkan snort menganalisa trafik yang melewati jaringan untuk dicocokkan (*match*) dengan rule set yang telah didefinisikan user dan melakukan beberapa aksi tertentu berdasarkan rule action yang ada.

Contoh command untuk mode ini adalah:

./snort –dev –1 ./log –h 192.168.1.0/24 –c snort.conf

2.7. Fuzzy Logic

Fuzzy secara leksikal mengandung arti tidak jelas, samar atau kabur. Konsep ini pertama kali dikemukakan oleh L.A. Zadeh pada tahun 1965. Dalam

teori sistem pengaturan kata fuzzy dihubungkan dengan kata logika, sehingga diperoleh kata logika fuzzy yang berarti suatu logika yang samar. Dengan menentukan fenomena-fenomena di alam nyata yang mengandung sifat serba tidak tepat atau samar kita tentukan aturan yang samar-samar juga.

Fuzzy berfungsi untuk mengekspresikan tingkat keanggotaan (*degree of membership*) suatu himpunan terhadap suatu harga sebagai suatu fungsi yang berharga 0 dan 1. Hal ini tentu saja berbeda dengan teori *crisp* yang menyatakan sesuatu itu hanya dalam salah satu dari dua keadaan yakni ya atau tidak.

Perancangan pengendali logika fuzzy menggabungkan aspek pendefinisian himpunan fuzzy dengan aspek logika fuzzy yang keduanya diterapkan pada masukan dan keluaran untuk memperoleh hasil perancangan yang berbentuk suatu algoritma aturan fuzzy. Pengendali fuzzy menggunakan aturan-aturan linguistik pertama kali diaplikasikan pada tahun 1974 oleh Mamdani, yang mengunakan metode teoritis Zadeh.

Aturan diberikan dalam bentuk :

R_i = "jika x adalah A_i maka y adalah B_i"

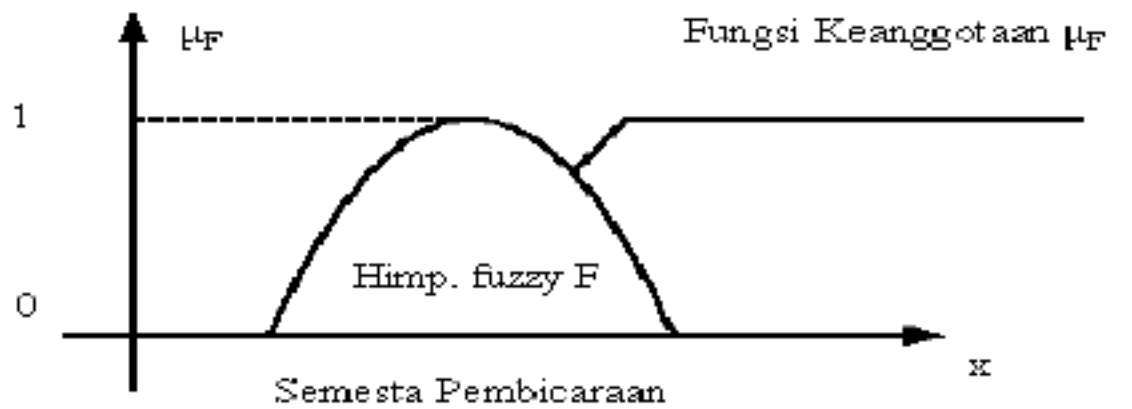
Dengan $x = (x_1, x_2, x_3, \ldots, x_n)$ adalah variabel keadaan masukan; y adalah bentuk skalar yang menggambarkan variabel keluaran; dan A_i dan B_i adalah terminologi linguistik yang ditunjukkan oleh himpunan fuzzy.

2.7.1. Himpunan *Fuzzy*

Masukan dan keluaran kontroler fuzzy menggunakan himpunan yang didefinisikan secara matematis adalah sebagai berikut :

$$F = \{(u, \mu_F(u)) | u \in U\}$$

Himpunan fuzzy F mempunyai semesta pembicaraan U, dalam semesta pembicaraan inilah terdapat anggota u yang menjadi pembicaraan. Secara matematis keanggotaan u dalam fuzzy dinyatakan dalam $\mu_F(u)$ yang bernilai dalam interval [0,1]. Range 0 sampai 1 menunjukkan tingkat keanggotaan anggota u dalam F. Bila $\mu_F(u) = 0$ menunjukkan u bukan anggota F, sedangkan bila $\mu_F(u)=1$ menunjukkan u merupakan anggota penuh F. Keanggotaan dari semua himpunan berada pada semesta pembicaraan yang ada pada gambar 2.21 dibawah ini diwakili oleh elemen-elemen pada sumbu x.



Gambar 2.21 Himpunan *fuzzy* dan fungsi keanggotaannya

2.7.2. Fungsi Keanggotaan Himpunan *Fuzzy*

Keanggotaan dalam suatu himpunan fuzzy mempunyai bentuk yang berbeda-beda. Pendefinisian suatu anggota dalam fuzzy dapat menggunakan cara numerik maupun bentuk fungsi. Pendefinisian bentuk fungsi digunakan untuk himpunan yang kontinyu, sedangkan pendefinisian numerik digunakan untuk himpunan diskrit. Fungsi keanggotaan (*membership function*) yang paling sering digunakan pada pendukung kontinyu adalah :

- Fungsi Eksponensial

$$\mu_A(u) = \exp\left[-\frac{(u-x)^2}{2\sigma^2}\right]$$

Keterangan : $\mu_A(u)$ adalah fungsi keanggotaan untuk himpunan u.

nilai x adalah masukan (*input*).

nilai σ adalah konstanta bernilai 3.14.

- Fungsi Segitiga

$$\mu_A(u) = \max\left[\min\left(\left(\frac{x-a}{b-a}\right),\left(\frac{c-x}{c-b}\right)\right),0\right]$$

Keterangan : $\mu_A(u)$ adalah fungsi keanggotaan untuk himpunan u.

nilai x adalah masukan (*input*).

nilai a adalah batas awal fungsi keanggotaan.

nilai b adalah batas fungsi keanggotaan yang bernilai 1.

nilai c adalah batas akhir fungsi keanggotaan.

- Fungsi Singleton

$$\mu_A(u) = \begin{cases} 1; u = a \\ 0; u \neq a \end{cases}$$

Keterangan : $\mu_A(u)$ adalah fungsi keanggotaan untuk himpunan u.

nilai a adalah batas fungsi keanggotaan yang bernilai 1.

- Fungsi Trapesium

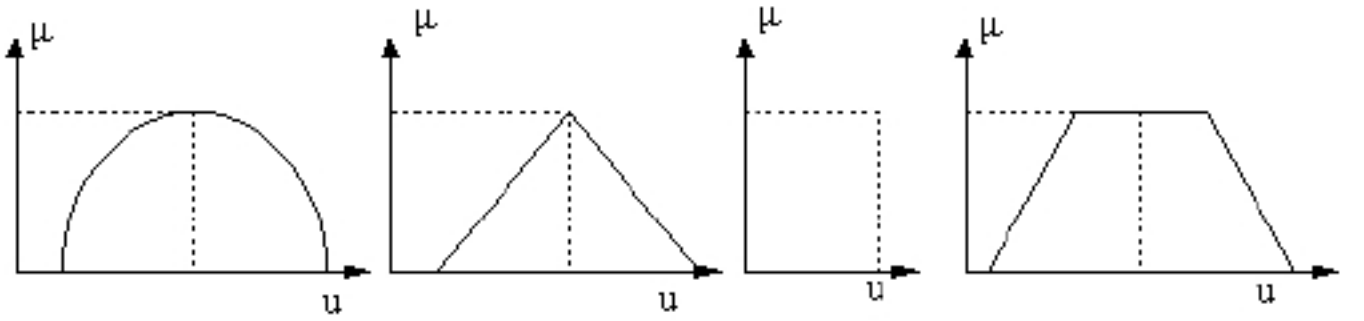
$$\mu_A(u) = \begin{cases} 1.....;0 \geq (u-a) \leq \dfrac{b}{a} \\ 2 - 2\dfrac{\sqrt{(u-a)^2}}{b}...;\dfrac{b}{2} \leq (u-a) \leq b \end{cases}$$

Keterangan : $\mu_A(u)$ adalah fungsi keanggotaan untuk himpunan u .

nilai a adalah batas kiri/awal fungsi keanggotaan.

nilai b adalah batas kanan/akhir fungsi keanggotaan.

Dengan μ merupakan derajat keanggotaan fuzzy dan u merupakan himpunan fungsi keanggotaan fuzzy. Secara grafik dapat digambarkan seperti gambar 2.22 dibawah ini :



(a) (b) (c) (d)

Gambar 2.22 Fungsi keanggotaan

(a) Eksponensial, (b) Segitiga, (c) Singleton, (d) Trapesium

2.7.3. Variabel Linguistik

Logika *Fuzzy* berusaha merepresentasikan cara berpikir manusia sebagai ganti variabel numerik yang biasa digunakan dalam pendekatan kuantitatif. Untuk itu digunakan suatu variabel linguistik, yang bersifat kualitatif dan berfungsi

untuk menyatakan himpunan fuzzy. Variabel linguistik dinyatakan dalam bentuk [X, T, U, M], dimana :

- X adalah nama variabel linguistik, misal : tekanan, speed, suhu
- T adalah himpunan dari suku X, misal : tinggi, sedang rendah
- U adalah nilai aktual dalam bentuk numerik. Misal 0 sampai 20 *psi* atau dalam bentuk skala normalisasi.
- M adalah untuk kaidah semantik untuk mengasosiasikan masing-masing nilai itu dengan artinya. Sebagai contoh, tinjau 'kecepatan' sebagai variabel linguistik.

$$V(kecepatan) = \{rendah, sedang, tinggi\}$$

Dimana V (kecepatan) dinyatakan dalam semesta pembicaraan, contohnya : Ditentukan masukan X dengan interval dari 0 sampai dengan 100. Dalam hal ini rendah terletak disekitar 0 ~ 30 rpm, sedang disekitar 50 ~ 70 rpm, serta tinggi disekitar 70 ~ 100 rpm dalam bentuk fungsi keanggotaan segitiga untuk mendefinisikannya adalah :

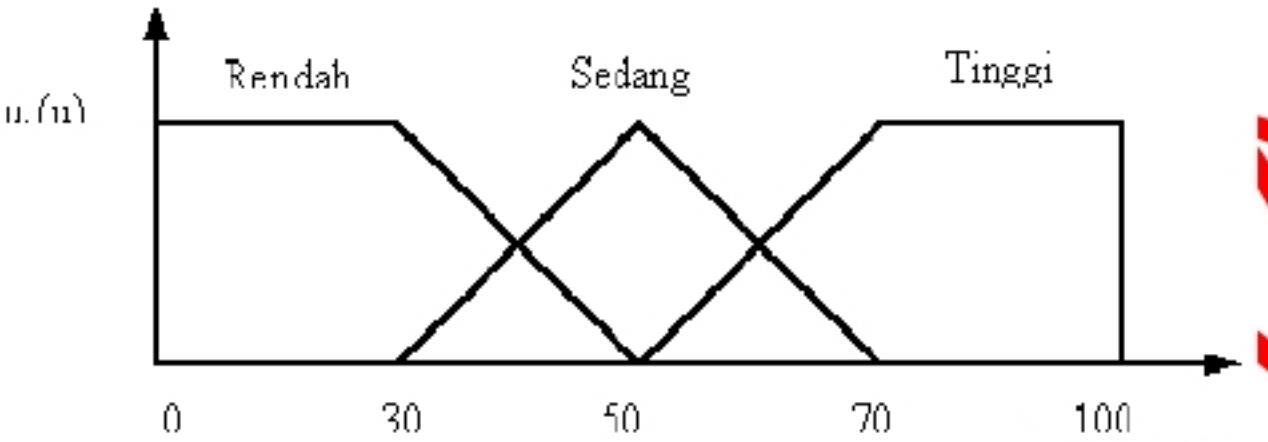
Rendah : $\mu_A(u) = \max\left[\min\left(\left(\frac{x-0}{30-0}\right),\left(\frac{50-x}{50-30}\right)\right),0\right]$

Sedang : $\mu_A(u) = \max\left[\min\left(\left(\frac{x-30}{50-30}\right),\left(\frac{70-x}{70-50}\right)\right),0\right]$

Tinggi : $\mu_A(u) = \max\left[\min\left(\left(\frac{x-50}{70-50}\right),\left(\frac{100-x}{100-70}\right)\right),0\right]$

Titik silang terletak di titik = 40 rpm dan titik 60 rpm, yaitu pendukung dengan nilai keanggotaan μ (40) dan nilai keanggotaan μ (60) sama dengan 0,5.

Fungsi keanggotaan untuk kondisi ini ditunjukkan pada gambar 2.23 dibawah ini :



Gambar 2.23 Fungsi keanggotaan segitiga

2.7.4. Operasi Himpunan *Fuzzy*

Seperti halnya set klasik, dalam set fuzzy juga didefinisikan operasi-operasi untuk mengkombinasikan dan memodifikasi himpunan *fuzzy*. Perbedaan operasi dasar fuzzy adalah bersifat gradual, sehingga definisi operasi-operasi itu menjadi berbeda.

Beberapa operasi dasar himpunan fuzzy yang digunakan adalah sebagai berikut :

- Gabungan

  Fungsi keanggotaan μ_c (u) sebagai gabungan C = A ∪ B didefinisikan dengan ekspresi :

$$\mu_C(u) = \max\{\mu_A(u), \mu_B(u)\} u \in U$$

- Irisan

  Fungsi keanggotaan μ_c (u) sebagai suatu irisan C = A ∩ B didefinisikan dengan ekspresi :

$$\mu_C(u) = \min\{\mu_A(u), \mu_B(u)\} u \in U$$

- Komplemen

  Fungsi keanggotaan dari komplemen A,

$$\mu_{\bar{A}}(u) = 1 - \mu_A(u)$$

- Relasi *Fuzzy*

Suatu n relasi fuzzy adalah himpunan fuzzy dalam U_1x U_1x…x U_n dan fungsi keanggotaannya dinyatakan sebagai berikut :

$$\mu U_1,...,U_n = \{((U_1,...,U_n)\mu_R(U_1,...,U_n))\,|\,(U_1,...,U_n)\mu_R(U_1,...,U_n)\}$$

2.7.5. Basis Aturan *Fuzzy*

Basis aturan merupakan inti dari sistem fuzzy, karena pada bagian ini berada sekumpulan aturan *IF-THEN* yang dinyatakan dalam bentuk :

IF (masukan i) THEN (keluaran j) i,j = 1,2,3,…,n

Masukan dan keluaran merupakan himpunan fuzzy yang dinyatakan dengan variabel linguistik. Karena keringkasannya, bentuk ini sering digunakan untuk mewakili kemampuan manusia untuk mengambil keputusan atas suatu kondisi yang penuh dengan ketidakpastian dan ketidaktepatan.

Contohnya : jika tekanan tinggi, maka volume kecil, dimana tinggi dan kecil adalah besaran kualitatif yang dijelaskan dalam fungsi keanggotaan.

2.7.6. Sistem Inferensi *Fuzzy*

Sistem inferensi fuzzy adalah sistem kerja komputer yang didasarkan pada konsep teori fuzzy, aturan fuzzy *if-then*, dan logika fuzzy.

Struktur dasar dari sistem inferensi fuzzy terdiri dari :

1. Fuzzifikasi yang mentransformasi masukan himpunan klasik (*crisp*) ke derajat tertentu yang sesuai dengan aturan besaran fungsi keanggotaan.
2. Basis aturan yang berisi aturan *if-then*.
3. Basis data yang mendefinisikan fungsi keanggotaan dari himpunan fuzzy.

4. Unit pengambilan keputusan yang menyatakan operasi inferensi aturan-aturan.
5. Defuzzifikasi yang mentransformasi hasil fuzzy ke bentuk keluaran yang crisp.

2.7.7. Defuzzifikasi

Untuk mengembalikan nilai besaran fuzzy menjadi nilai *non-fuzzy* (*crisp*) digunakan defuzzifikasi. Beberapa teknik defuzzifikasi diantaranya adalah :

A. *Center Average (CA)*

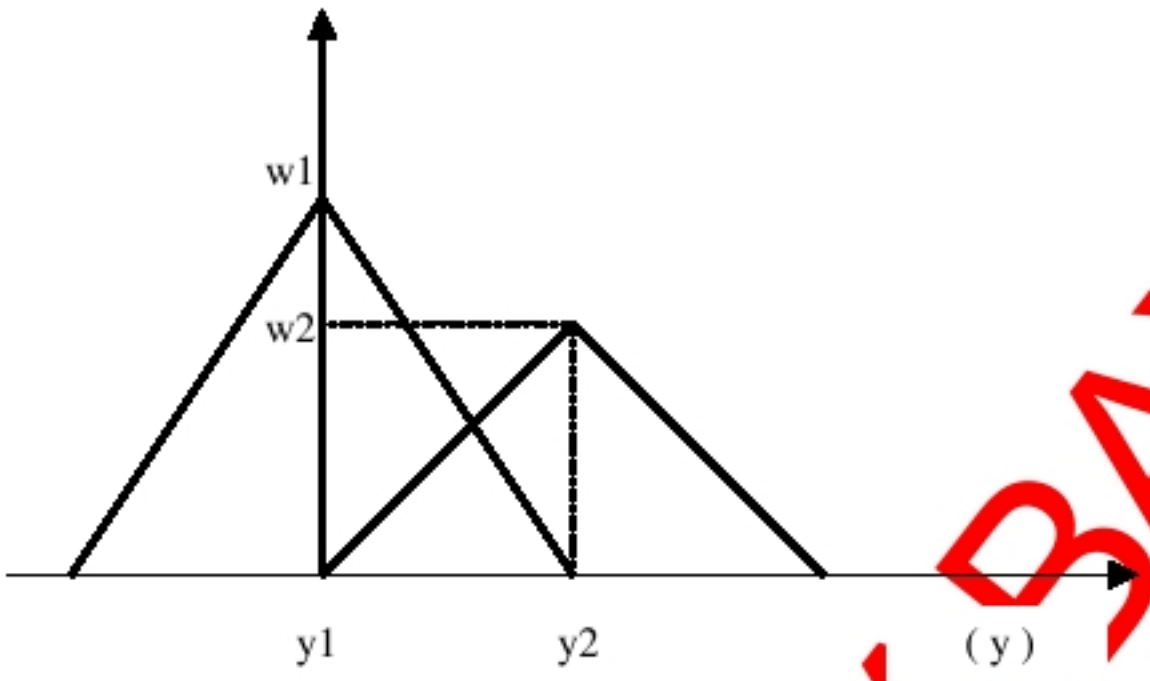
Misalkan diperoleh keluaran berupa dua fungsi keanggotaan fuzzy seperti gambar 2.24 dibawah ini. Dimana y1 dan y2 adalah nilai tengah dari masing-masing fungsi keanggotaan fuzzy, lalu w1 dan w2 adalah tinggi. Maka *CA Defuzzifier*, *Y** bisa dihitung sebagai berikut (Li-Xin Wang, 1997) :

$$Y^* = \frac{\sum_{i=1}^{2} Y_i \, W_i}{\sum_{i=1}^{2} W_i}$$

Keterangan : Y_i = nilai tengah dari masing-masing fungsi keanggotaan fuzzy

Wi = tinggi dari derajat keanggotaan fuzzy.

Gambarannya sebagai berikut :



Gambar 2.24 Center average defuzzifier

B. *Centroid of Area (COA)* atau *Center of Gravity (COG)*

Metode COA ini adalah metode yang sering digunakan. Strategi ini dibangkitkan dari pusat berat (*center of gravity*) pada aksi kontrol. Defuzzifikasi *Center of Area* (Jang, J.-S.R, Sun, C.-T & Mizutani,E., 1996) atau *Center of Gravity* (Li-Xin Wang, 1997) menggunakan rumus sebagai berikut :

$$z_{COA} = \frac{\int_z \mu_A(z) z \, dz}{\int_z \mu_A(z) dz}$$

Keterangan : $\mu_A(z)$ adalah fungsi keanggotaan untuk himpunan z.

z adalah himpunan semesta fuzzy.

dz adalah turunan dari z (himpunan fuzzy).

C. *Bisector of Area (BOA)*

Defuzzifikasi dengan metode *Bisector of Area* (Jang, J.-S.R, Sun, & Mizutani.E., 1996) menggunakan rumus sebagai berikut :

$$\int_{\alpha}^{Z} BOA\ \mu A(z)dz = \int_{\alpha}^{\beta} BOA\ \mu A(z)dz$$

Keterangan : $\mu_A(z)$ adalah fungsi keanggotaan untuk himpunan z

z adalah himpunan semesta fuzzy.

dz adalah turunan dari z (himpunan fuzzy).

D. *Mean of Maksimum (MOM)*

Metode *Mean of Maksimum* (Jang, J.-S.R, Sun, & Mizutani.E., 1996) merepresentasikan nilai titik tengah dari keluaran yang fungsi keanggotaannya maksimum. Fungsinya ditunjukkan sebagai berikut :

$$z_{MOM} = \frac{\int_{z'} z\, dz}{\int_{z'} dz}$$

Keterangan : z adalah himpunan semesta fuzzy.

dz adalah turunan dari z (himpunan fuzzy).

E. *Smallest of Maximum*

Defuzzifikasi dengan metode *Smallest of Maximum* (Jang, J.-S.R, Sun, & Mizutani.E., 1996) menggunakan aturan sebagai berikut :

z_{SOM} adalah nilai minimum dari z.

F. *Largest of Maximum*

Defuzzifikasi dengan metode *Largest of Maximum* (Jang, J.-S.R, Sun, & Mizutani.E., 1996) menggunakan aturan sebagai berikut :

z_{LOM} adalah nilai maksimum dari z.