

BAB 2

LANDASAN TEORI

2.1. Pengenalan Pola

Pengenalan Pola dapat dikatakan sebagai kemampuan manusia mengenali obyek-obyek berdasarkan ciri-ciri dan pengetahuan yang pernah diamatinya dari obyek-obyek tersebut. Tujuan dari Pengenalan Pola ini adalah mengklasifikasi dan mendeskripsikan pola atau obyek kompleks melalui pengetahuan sifat-sifat atau ciri-ciri obyek tersebut. Pola adalah entitas yang terdefinisi dan dapat diberikan suatu identifikasi atau nama misalkan mobil.

Pendekatan pengenalan pola ada tiga, yaitu secara sintaks, statistik, serta melalui jaringan saraf tiruan. Pendekatan secara sintaks adalah pendekatan dengan menggunakan aturan-aturan tertentu, misalnya baju si mamat mempunyai *rule* sebagai berikut, selalu berwarna biru, bahannya kaos, bermerek adidas, lengannya lengan panjang dan memiliki kerah. Jika ada sebuah baju dengan ciri-ciri 90% lebih dari ciri-ciri tersebut dapat dikatakan bajunya mamat dengan toleransi sekitar 10%.

Pendekatan metoda statistik adalah pendekatan dengan menggunakan data-data yang berasal dari statisik misalnya dalam sebuah pasar saham terlihat kurva penjualan tertinggi adalah saham A, kemudian disusul saham B dan saham C, apabila seseorang datang ke pasar saham tersebut maka orang tersebut dapat dikatakan sekitar 95% orang membeli saham A, karena berdasarkan kurva saham A memiliki harga tertinggi.

Pendekatan dengan pola jaringan saraf tiruan adalah pendekatan dengan menggabungkan pendekatan sintaks dan statistik. Pendekatan melalui pola-pola ini

meniru cara kerja otak manusia, pada pola ini sistem membuat *rule-rule* tertentu disertai dengan menggunakan data statistik sebagai dasar untuk pengambilan keputusan.

Suatu sistem pengenalan pola pada dasarnya terdiri atas tiga tahap, yaitu penerimaan data, pengenalan data dan pengenalan obyek atau pembuatan keputusan. Adapun pengenalan utama dalam pengenalan pola adalah pendekatan geometrik dan pendekatan struktural. Pendekatan struktural dilakukan dengan penentuan dasar yang mendeskripsikan obyek yang akan dikenali.

Untuk pengenalan pola dengan pendekatan Jaringan Saraf Tiruan kita seolah-olah membuat sebuah sistem yang kinerjanya sama dengan otak kita. Agar sistem tersebut bisa menjadi cerdas, kita harus memberikan pelatihan terhadap sistem tersebut selama rentang waktu yang kita tentukan. Karena dengan melatih sistem tersebut maka akan menambah *rule-rule* serta data statistik yang di gunakan oleh sistem untuk mengambil keputusan.

2.2. Pengenalan Suara

Pengenalan suara adalah proses yang dilakukan oleh komputer untuk mengidentifikasi kata-kata yang diucapkan kepadanya. Penelitian pengenalan suara oleh mesin telah dilakukan hampir selama lima dekade, dimana percobaan awal oleh mesin dibuat tahun 1950-an, yaitu pada saat berbagai penelitian mencoba untuk mengeksploitasi ide fundamental dari *acoustic-phonetic*.

Sejarah penelitian dibidang pengenalan suara antara lain:

- a. Tahun 1952 di laboratorium, Bell, Davis, Bidullph, Balashek membuat suatu sistem pengenalan digit terisolali untuk seorang pembicara. Sistem tersebut sangat tergantung kepada pengukuran resonansi *spectral* di daerah vokal dari setiap digit.

- b. Tahun 1956, sebuah usaha independen pada laboratorium RCA, Olson dan Belar berusaha untuk mengenali sepuluh suku kata yang berbeda dari setiap pembicara yang juga bergantung pada pengukuran *spectral* pada area vokal.
- c. Tahun 1959, Universitas Collage di Inggris, Fry dan Denes mencoba untuk membuat suatu pengenalan fenom untuk mengenali empat vokal dan sembilan konsonan. Mereka menggunakan keputusan dari pengenalan.
- d. Usaha lain pada periode ini adalah mengenalkan vokal oleh Forgie dan Forgi, dikonstruksikan di laboratorium Lincon MIT pada tahun 1959, dimana sepuluh vokal disisipkan dalam format a/b/-vokal-/t dapat dikenali.
- e. Penelitian speech pada tahun 1980an bercirikan pada pergeseran teknologi dari pendekatan berbasis *template* menjadi statistik *modeling*, terutama pendekatan model *Hidden Markov Model*. Ide lain di perkenalkan pada akhir 1980-an adalah penerapan *Neural network* pada *Speech Recognition*. *Neural network* pertama kali dikenalkan pada tahun 1950, tetapi tidak terbukti berguna pada awalnya karena terlalu banyak masalah praktikal.

2.3. Sinyal Percakapan

Sinyal dapat didefinisikan sebagai kuantitas fisik yang bervariasi seiring waktu atau variabel bebas lainnya yang menyimpan suatu informasi. Contoh sinyal adalah: suara manusia, kode morse, tegangan listrik di kabel telepon, variasi intensitas cahaya pada sebuah serat optik yang digunakan pada telepon atau jaringan komputer, dan lain-lainnya.

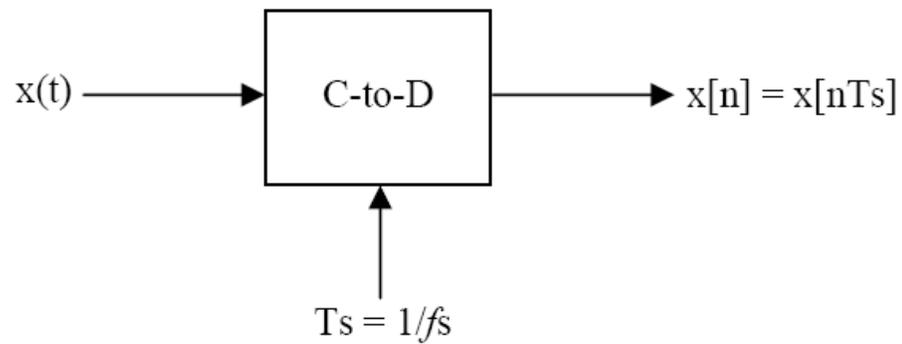
Sinyal dapat diklasifikasikan menjadi beberapa jenis yaitu: sinyal waktu kontinu, sinyal waktu diskrit, sinyal nilai kontinu, sinyal nilai diskrit, sinyal *random*, dan sinyal

nonrandom. Sinyal waktu kontinu dengan nama lain sinyal analog adalah sinyal yang belum melalui proses apapun. Sedangkan sinyal nilai diskrit atau sinyal digital adalah sinyal analog yang telah melalui proses *sampling*, *quantization*, dan *encoding*.

- *Sampling* adalah proses mengambil nilai-nilai sinyal pada titik-titik diskrit sepanjang variabel waktu dari sinyal waktu kontinu, sehingga didapatkan sinyal waktu diskrit. Jumlah titik-titik yang diambil setiap detik dinamakan sebagai *sampling rate*. Dalam melakukan *sampling*, perlu diperhatikan kriteria *Nyquist* yang menyatakan bahwa sebuah sinyal harus memiliki *sampling rate* yang lebih besar dari $2f_m$, dengan f_m adalah frekuensi paling tinggi yang muncul di sebuah sinyal.
 - Proses *sampling* perhatikan sinyal sinus berikut ini:

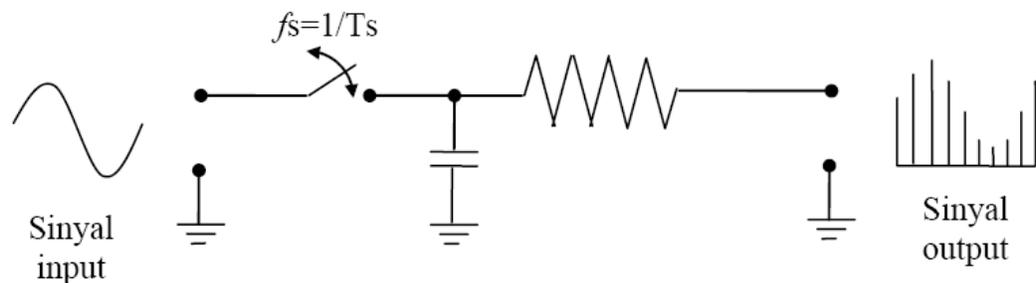
$$x(t) = A \cos(\omega t + \phi) \dots\dots\dots(1)$$

Sinyal tersebut merupakan contoh sinyal waktu kontinu. Kita juga seringkali menggunakan terminologi sinyal analog untuk menyebutnya. Untuk proses komputasi, sinyal waktu kontinu harus dirubah menjadi bentuk waktu diskrit dan dilanjutkan dengan proses digitalisasi. Untuk memperoleh bentuk sinyal waktu diskrit, sinyal waktu kontinu harus di-sampel.



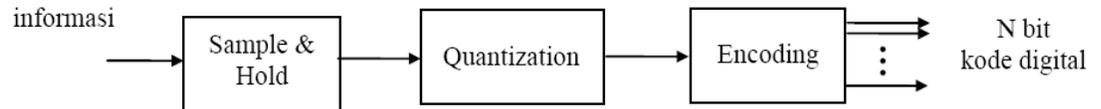
Gambar 2.1 Blok Diagram Konversi Sinyal Kontinyu Menjadi Sinyal Diskrit.

Sekuen $x[n]$ di dapatkan setelah proses perubahan dari *continues to discrete* (C-to-D). Kondisi realnya secara hardware adalah menggunakan rangkaian sampling seperti **Gambar 2.2** berikut ini.



Gambar 2.2 Rangkaian Sampling.

Rangkaian sampling diatas merupakan sebuah ujung tombak dari sebuah *analog to digital conversion* (ADC).



Gambar 2.3 Block Diagram Rangkaian ADC.

Persyaratan frekuensi sampling menurut teorema *Shannon* harus sama dengan atau melebihi 2 kali frekuensi sinyal yang di sample.

$$f_s \geq 2 \times f_i \dots\dots\dots(2)$$

Jika sinyal informasi yang kita sample memiliki komponen frekuensi beragam, misalnya untuk sinyal wiacara memungkinkan untuk memiliki frekuensi dari 20 sampai 4000 Hz, maka sinyal informasi tersebut bisa dituliskan sebagai:

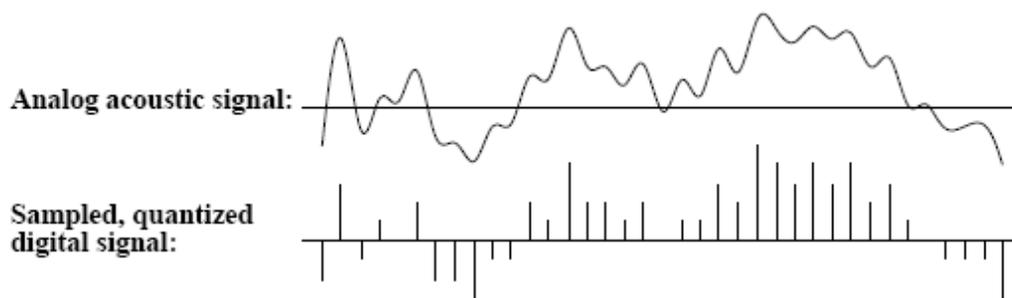
$$x(t) = \sum_{i=1}^{i \max} \sin(2\pi f_i t) \dots\dots\dots(3)$$

Dan persyaratan untuk frekuensi smpling menjadi:

$$f_s \geq 2 \times f_{i \max} \dots\dots\dots(4)$$

Frekuensi sampling seringkali dikatakan dengan terminology *sampling rate*, yaitu jumlah sample yang diambil setiap detik, $f_s=1/T_s$ yang juga dikenal sebagai *Nyquist rate*.

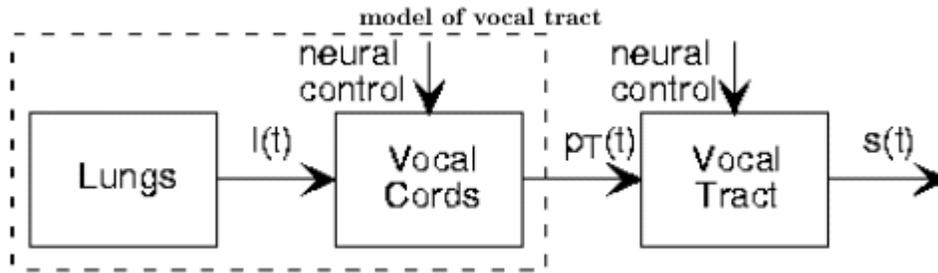
- *Quantization* adalah proses memetakan nilai-nilai dari sinyal nilai kontinu menjadi nilai-nilai yang diskrit, sehingga didapatkan sinyal nilai diskrit.
- *Encoding* adalah proses mengubah nilai-nilai sinyal ke menjadi bilangan biner. Pada **gambar 2.4** dapat dilihat perbedaan antara sinyal analog dengan sinyal digital.



Gambar 2.4 Diagram Sinyal Analog dan Sinyal Digital.

Sinyal yang berbentuk digital dapat disimpan dalam media penyimpanan di komputer. WAV file (berasal dari kata *wave*) merupakan format umum yang paling sederhana untuk menyimpan data sinyal audio. WAV file terdiri dari 3 potongan informasi yaitu: RIFF *chunk*, FORMAT *chunk*, dan DATA *chunk*. RIFF *chunk* berisi informasi yang menandakan bahwa file berbentuk WAV. FORMAT *chunk* berisi parameter-parameter seperti jumlah channel, *sample rate*, resolusi. DATA *chunk* yang berisi data aktual sinyal digital.

Sinyal yang dihasilkan dari suara manusia sewaktu melakukan percakapan disebut sebagai sinyal percakapan. Sinyal percakapan merupakan kombinasi kompleks dari variasi tekanan udara yang melewati pita suara dan *vocal tract*, yaitu mulut, lidah, gigi, bibir, dan langit-langit. Sistem produksi sinyal percakapan dapat dilihat pada **gambar 2.5**.



Gambar 2.5 Diagram Model Sistem Produksi Suara.

Sinyal percakapan terdiri dari serangkaian suara yang masing-masing menyimpan sepotong informasi. Berdasarkan cara menghasilkannya, suara tersebut terbagi menjadi *voiced* dan *unvoiced*. Suara *voiced* dihasilkan dari getaran pita suara, sedangkan suara *unvoiced* dihasilkan dari gesekan antara udara dengan *vocal tract*.

Sinyal percakapan memiliki beberapa karakteristik, misalnya: *formant*, *pitch*, dan intensitas. *Formant* adalah variasi resonansi yang dihasilkan oleh *vocal tract*. *Pitch* adalah frekuensi dari sinyal atau yang sering disebut sebagai intonasi. Sedangkan intensitas adalah kekuatan suara. Karakteristik-karakteristik tersebut berguna dalam melakukan analisis sinyal.

2.4. Transformasi Fourier

Transformasi fourier (dinamakan atas Joseph Fourier) adalah sebuah transformasi integral yang menyatakan-kembali sebuah fungsi dalam fungsi basis sinusoidal(fungsi trigonometri), yaitu sebuah fungsi sinusoidal penjumlahan atau integral dikalikan oleh beberapa koefisien ("amplitudo").

Cara kerja transformasi fourier adalah dengan mendekomposisi sinyal ke bentuk fungsi eksponensial dari frekuensi yang berbeda-beda. Caranya adalah dengan didefinisikan ke dalam dua persamaan berikut:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-2\pi ft} dt \dots\dots\dots(5)$$

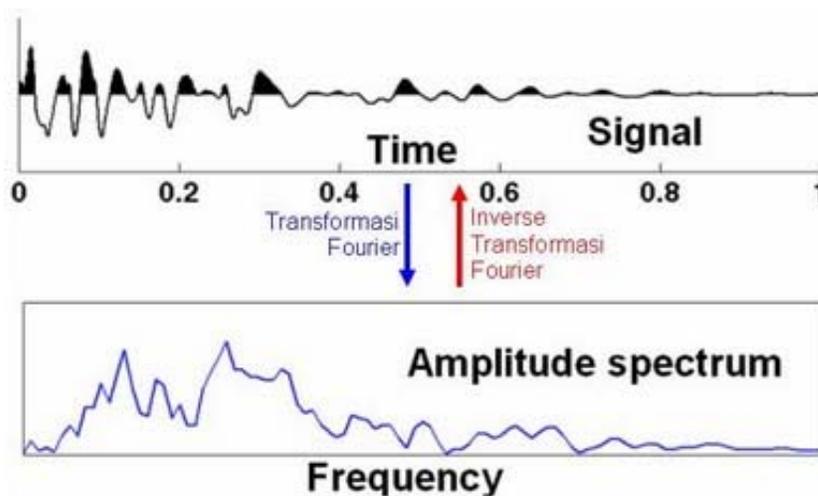
$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{-2\pi ft} df \dots\dots\dots(6)$$

Dalam persamaan tersebut, t adalah waktu dan f adalah frekuensi. x merupakan notasi sinyal dalam ruang waktu dan X adalah notasi untuk sinyal dalam domain frekuensi. Persamaan (1) disebut Transformasi Fourier dari x(t) sedangkan persamaan (2) disebut Invers Transformasi Fourier dari X(f), yakni x(t). Persamaan (1) dapat juga ditulis sebagai :

$$\text{Cos}(2\pi ft)+j\text{Sin}(2\pi ft)\dots\dots\dots(7)$$

2.4.1. Transformasi Fourier Diskrit

Transformasi fourier diskrit adalah metoda untuk mengubah gelombang seismik dalam domain waktu menjadi domain frekuensi. Proses perubahan dari domain waktu menjadi domain frekuensi dapat dilihat pada **gambar 2.6**

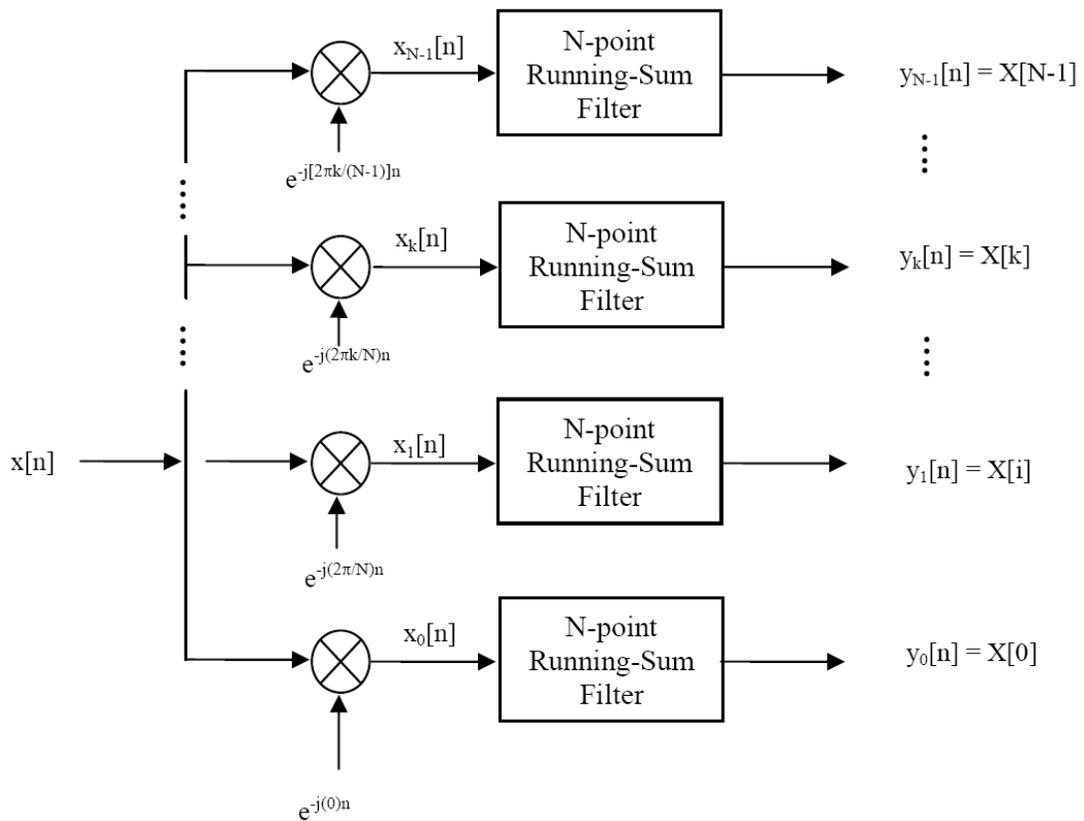


Gambar 2.6 Cara Kerja Transformasi Fourier.

Salah satu cara mentransformasi sinyal dari domain waktu ke dalam domain frekuensi adalah dengan menggunakan *discrete fourier transform* (DFT).

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi / N)kn} \dots\dots\dots(8)$$

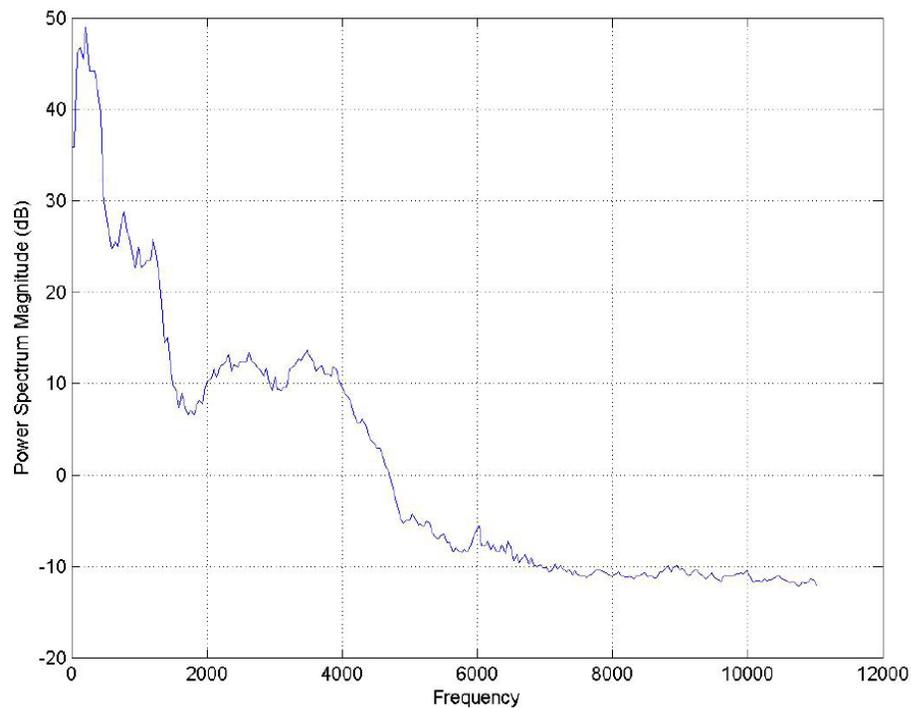
Persamaan (1) diatas menyatakan bahwa sinyal akan periodic pada setiap nilai N. Implementasi DFT dapat diwujudkan dengan sebuah Bank Filter seperti **Gambar 2.8** berikut ini.



Gambar 2.7 Bank Filter.

Untuk mengoptimalkan proses komputasi, DFT bisa dimodifikasi dengan satu algorithma yang seringkali kita kenal sebagai *fast fourier transform* (FFT). Dengan FFT proses komputasi bias direduksi dari N^2 menjadi $N \log_2 N$. Misalnya dengan

menggunakan DFT kita akan melakukan transformasi sebanyak $N=1024$ titik, maka kita memerlukan perkalian sebanyak $N^2 = 1.048.567$. Sedangkan dengan menggunakan FFT perkalian yang diperlukan sebanyak $N \log_2 N = 5120$ perkalian. Sebuah contoh hasil menggunakan algoritma FFT untuk system yang lebih kompleks adalah untuk mengolah sinyal bicara. Pada **gambar 2.8** ditunjukkan sebuah hasil proses FFT untuk kalimat “a-i-u-e-o”.



Gambar 2.8 Grafik Hasil Proses FFT.

2.5. Jaringan Saraf Tiruan (*Neural Network*)

2.5.1. Inspirasi Biologi

Jaringan Saraf Tiruan (JST) terinspirasi dari penelitian kecerdasan buatan, terutama percobaan untuk menirukan *fault-tolerance* dan kemampuan untuk belajar dari sistem saraf biologi dengan model struktur *low-level* otak.

Otak manusia memiliki sekitar 10.000.000.000 sel saraf yang saling berhubungan. Sel saraf mempunyai cabang struktur *input* (dendrites), yaitu sebuah inti sel dan percabangan struktur *output* (axon). Axon dari sebuah sel terhubung dengan dendrites yang lain melalui sebuah *synapse*. Ketika sebuah sel saraf aktif, kemudian menimbulkan suatu signal *electrochemical* pada axon. Signal ini melewati *synapses* menuju ke sel saraf yang lain. Sebuah sel saraf yang lain akan mendapatkan signal jika memenuhi biasan tertentu yang sering disebut dengan nilai ambang (threshold).

Bila dibandingkan dengan komputer modern, kecepatan dari *neuron* pada otak manusia jauh lebih lambat, namun karena jumlah *neuron* yang sangat banyak pada otak manusia membuat otak manusia dapat melakukan banyak pekerjaan yang sangat cepat serta menggunakan energi atau tenaga yang sangat sedikit dan efisien. Menurut Faggin (1991) sebagai contoh, pengenalan obyek seperti wajah seseorang dapat dilakukan hanya dalam waktu sepersepuluh detik, sementara dengan *database* yang sama dengan komputer modern saat ini dibutuhkan waktu yang jauh lebih lama setidaknya beberapa menit lebih lama dari otak manusia.

2.5.2. Pengertian Jaringan Saraf Tiruan (*Neural Network*)

Neural network merupakan suatu metode *Artificial Intelligence* yang konsepnya meniru sistem jaringan saraf yang ada pada tubuh manusia, dimana dibangun *node-node* yang saling berhubungan satu dengan yang lainnya. *Node-node* tersebut terhubung melalui suatu *link* yang biasa disebut dengan istilah *weight*. Ide dasarnya adalah mengadopsi cara kerja otak manusia yang memiliki ciri-ciri *paralel processing*, *processing element* dalam jumlah besar dan *fault tolerance*.

Suatu jaringan saraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi, hal ini terinspirasi oleh model kerja otak biologis. Beberapa definisi tentang jaringan saraf tiruan adalah sebagai berikut di bawah ini.

- Hecht-Nielsen (1988) mendefinisikan sistem saraf buatan sebagai berikut:

"Suatu neural network (NN), adalah suatu struktur pemroses informasi yang terdistribusi dan bekerja secara paralel, yang terdiri atas elemen pemroses (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang diinterkoneksi bersama dengan alur sinyal searah yang disebut koneksi. Setiap elemen pemroses memiliki koneksi keluaran tunggal yang bercabang (fan out) ke sejumlah koneksi kolateral yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemroses tersebut). Keluaran dari elemen pemroses tersebut dapat merupakan sebarang jenis persamaan matematis yang diinginkan. Seluruh proses yang berlangsung pada setiap elemen pemroses harus benar-benar dilakukan secara lokal, yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan dalam memori lokal".

- Menurut Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY, Macmillan, mendefinisikan jaringan saraf sebagai berikut:

"Sebuah jaringan saraf adalah sebuah prosesor yang terdistribusi paralel dan mempunyai kecenderungan untuk menyimpan pengetahuan yang didapatkannya dari pengalaman dan membuatnya tetap tersedia untuk digunakan. Hal ini menyerupai kerja otak dalam dua hal yaitu: 1. Pengetahuan diperoleh oleh jaringan melalui suatu proses belajar. 2. Kekuatan hubungan antar sel saraf yang dikenal dengan bobot sinapsis digunakan untuk menyimpan pengetahuan".

- Dan menurut Zurada, J.M. (1992), *Introduction To Artificial Neural Systems*, Boston: PWS Publishing Company, mendefinisikan sebagai berikut:

“Sistem saraf tiruan atau jaringan saraf tiruan adalah sistem selular fisik yang dapat memperoleh, menyimpan dan menggunakan pengetahuan yang didapatkan dari pengalaman”.

- DARPA Neural Network Study (1988, AFCEA International Press, p. 60) mendefinisikan jaringan syaraf buatan sebagai berikut :

Sebuah jaringan syaraf adalah sebuah sistem yang dibentuk dari sejumlah elemen pemroses sederhana yang bekerja secara paralel dimana fungsinya ditentukan oleh struktur jaringan, kekuatan hubungan, dan pengolahan dilakukan pada komputasi elemen atau nodes.

Jaringan saraf tiruan menyerupai otak manusia dengan dua cara :

- a. Pengetahuan yang diperoleh jaringan dari lingkungannya melalui proses pembelajaran.
- b. Kekuatan hubungan antar *neuron*, dikenal dengan istilah *synaptic weight*, dan digunakan untuk menyimpan pengetahuan yang diperoleh.

Neural network sendiri pun dibagi-bagi kembali menjadi beberapa bagian yang lebih kecil, dimana masing-masing metode mempunyai karakteristik sendiri-sendiri, serta memiliki keunggulan dan kelemahan dalam mengenali suatu pola. Metode-metode tersebut diantaranya adalah *Bidirectional associate memory* atau lebih dikenal dengan istilah BAM, *Hopfield Network*, *Counter Propagation Network*, *Backpropagation* dan masih banyak metode-metode lainnya yang sudah atau sedang dikembangkan dengan para ahli.

Pada umumnya *neural network* dibagi berdasarkan *layer-layer* yaitu *input layer*, *hidden layer*, *output layer*. Setiap *node* pada masing-masing *layer* memiliki suatu *error rate*, yang akan digunakan untuk proses *training*.

Pada kenyataannya (kebiasaannya), kebanyakan *neural system* harus diajari (*training*) terlebih dahulu. Mereka akan mempelajari asosiasi, *pattern*, dan fungsi yang baru. Pemakai-pemakai *neural network* tidak menspesifikasikan sebuah algoritma untuk dieksekusi dalam setiap perhitungan. Mereka akan memilih arsitektur tertentu dengan pandangan mereka, dengan karakteristik *neuron*, *weight*, dan memilih model training sendiri. Sehingga dari hasil tersebut, informasi *network* dapat diubah oleh para pemakai. *Artificial neural system* juga dapat mengkalkulasi teknik matematik, seperti meminimalisasi kesalahan suatu perhitungan.

Neural network sangat berperan dalam teknologi dan beberapa disiplin ilmu, yang membantu dalam menentukan model-model *neural network* dan *system non linear dynamic*. Salah satunya adalah matematika. Matematika adalah model neural yang paling berpotensi karena kekompleksannya. Elektronika dan ilmu komputer juga menggunakan metode ini, karena berperan dalam pengiriman sinyal data.

2.5.3. Sejarah Perkembangan Jaringan Saraf Tiruan

Sejarah Perkembangan Jaringan Saraf Tiruan dimulai pada tahun 1949, Donald Hebb melakukan algoritma *Learning* atau *training* yang mengatakan bahwa jika dua *neuron* aktif secara simultan, maka kekuatan hubungan antar *neuron* akan meningkat. Kemudian pada tahun 1950an dan 1960an, sebuah perkumpulan yang bernama Group of Researchers berhasil menemukan konsep *perceptrons*. Pada tahun berikutnya yaitu tahun 1969, Marvin Minsky secara matematis menunjukkan kelemahan *perceptrons*.

Penelitian Jaringan Saraf Tiruan pada tahun 1970an. Tahun 1987, Group of Researchers menggelar *International Conference on Neural network*, San Diego, AS.

2.5.4. Sistem Neural

a. *Node*

Neuron adalah unit pemrosesan informasi yang merupakan dasar dari operasi jaringan saraf tiruan. Sel-sel saraf tiruan ini dirancang berdasarkan sifat-sifat dari *neuron* biologis. Sel saraf tiruan ini biasa disebut sebagai *processing elements*, unit atau *node*.

Node adalah sebuah sel *neuron* yang disetiap *node*-nya memiliki *output*, *error*, dan *weight*. Jadi di setiap *node*, dimanapun itu pasti memiliki ketiga unsur tersebut. Hubungan antar *node* diasosiasikan dengan suatu nilai yang disebut dengan bobot atau *weight*. Setiap *node* pasti memiliki *output*, *error* dan bobotnya masing-masing.

Output merupakan keluaran dari suatu *node*. *Error* merupakan tingkat kesalahan yang terdapat dalam suatu *node* dari proses yang dilakukan. *Weight* merupakan bobot dari *node* tersebut ke *node* yang lain pada *layer* yang berbeda. Nilai *weight* berkisar antara -1 dan 1.

Bobot-bobot atau *weight* yang tersimpan di dalam jaringan saraf tiruan ini disebut sebagai bobot interkoneksi. Nilai bobot yang baik akan memberikan keluaran yang sesuai, dalam arti mendekati keluaran yang diharapkan (*target output*) untuk suatu *input* yang diberikan.

Menurut Mitchell (1997) bobot awal dalam suatu jaringan saraf tiruan biasanya diperoleh secara *random* dan sebaiknya diinisialisasi dengan nilai yang relative kecil, yaitu berkisar antara -0,1 sampai 0,1. Memasuki tahap pelatihan, bobot tersebut akan

mengalami penyesuaian melalui suatu proses perhitungan matematik agar tercapai nilai bobot yang sesuai.

b. *Input, Hidden, dan Output Layer*

Input layer merupakan *layer* tempat sebuah *input* dimasukkan (inisialisasi *input*), dan *layer* ini dilakukan proses-proses selanjutnya. Menurut Michell (1997) *hidden layer* berfungsi untuk membantu proses. Semakin banyak *hidden layer* yang digunakan, maka semakin bagus dan semakin cepat pula *output* yang diinginkan didapat. Akan tetapi waktu *training* akan berlangsung semakin lama.

Output layer adalah *layer* yang menampung hasil proses dari suatu *neural network*. *Forward propagation* bertujuan untuk menentukan *output* dari suatu *node*. *output* yang dimaksud disini adalah *output* dari *output layer*. Karena masing-masing *node* tersebut memiliki *output*.

c. Training

Proses belajar suatu *neural network* terdiri dari proses *Forward*, *Backward*, dan *Update Weight*, sekali melewati 3 tahap itu disebut dengan 1 kali *training* (1 cycle). Semakin banyak *training* yang dilakukan maka akan semakin kecil pula tingkat *error* yang dihasilkan di *output layer*-nya. Dengan demikian semakin kecil juga *error* suatu sistem.

Menurut Rao (1995), ada dua metode *Learning* dalam *neural network*, yaitu :

1. *Supervised Learning* adalah suatu metode dimana *neural network* belajar dari pasangan data *input* dan *target*, pasangan ini disebut *training pair*. Biasanya jaringan dilatih dengan sejumlah *training pair*, dimana suatu *input* vektor di aplikasikan, menghasilkan nilai di *output*, lalu hasil pada *output* tersebut akan dibandingkan

dengan target *output*. Selisihnya akan dikembalikan ke jaringan, kemudian dihitung *error*-nya, melalui *error* ini akan didapatkan selisih yang terdapat di dalam *weight*. Oleh karena itu terdapat *weight* baru yang cenderung memiliki *error* yang lebih kecil, sehingga akan didapat *error* yang lebih minimum dari *error* yang pertama. Vektor-vektor dalam *training set* diaplikasikan seluruhnya secara berurutan. Pertama-tama *error* dihitung, kemudian *weight* disesuaikan sampai seluruh *training set* menghasilkan *error* yang sekecil-kecilnya. Pada dasarnya konsep ini berawal dari konsep *human brain*.

Model *Neural network* yang menggunakan metode *Supervised Learning* diantaranya adalah sebagai berikut :

- a. Model *Backpropagation*
- b. Model *Bidirectional Associative Memory*
- c. *Hopfield Network*

2. *Non-Supervised (Unsupervised) Learning*

Unsupervised Learning dianggap sebagai model dalam konsep sistem biologis. Teori ini dikembangkan oleh Kohonen (1984) dan beberapa ilmuwan lainnya. Dalam *Unsupervised Learning* tidak diperlukan target *output*. Training hanya terdiri dari vektor-vektor *input*, tanpa memiliki pasangan target. Algoritma *training* mengubah *weight* jaringan untuk menghasilkan *output* yang konsisten. Aplikasi dari vektor-vektor yang cukup serupa menghasilkan pola *output* yang sama. Dengan demikian proses *training* akan menghasilkan sifat-sifat statistik dalam bentuk pengelompokan vektor-vektor dalam beberapa kelas. Dengan mengaplikasikan suatu vektor dari suatu kelas sebagai *input*, maka akan menghasilkan vektor *output* yang spesifik.

Model *Neural network* yang menggunakan metode *Non-Supervised (Unsupervised) Learning* diantaranya adalah sebagai berikut :

- a. *Adaptive Resonance Theory (ART)*
- b. *Compotitive Learning*
- c. *Kohonen's self organizing maps.*

2.5.5. Arsitektur Jaringan Saraf Tiruan

Arsitektur jaringan adalah lapisan yang menggambarkan pola keterhubungan antara *neuron*, baik didalam lapisan yang sama maupun antara lapisan yang berbeda. Menurut Kusumadewi (2003) ada beberapa arsitektur jaringan saraf, antara lain :

1. *Single Layer Neural Network*

Single Layer Neural Network adalah *Neural network* yang hanya memiliki satu lapisan dengan bobot terhubung. Jaringan ini hanya menerima *input* kemudian secara langsung mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi. *Neuron-neuron* pada lapisan ini saling berhubungan. Semua unit *input* akan dihubungkan dengan setiap unit *output*.

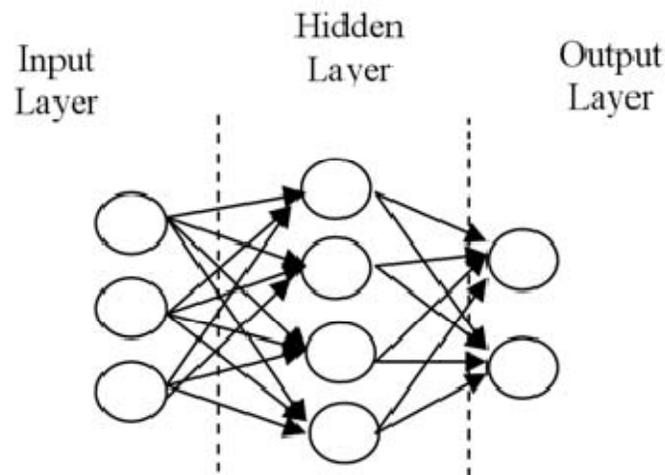
2. *Multi Layer Neural Network*

Multi Layer Neural Network adalah *neural network* yang memiliki karakteristik *multi layer* dimana setiap *node* pada suatu *layer* terhubung dengan setiap *node* pada *layer* didepannya. Berarsitektur umpan maju atau lebih dikenal dengan *feed forward network* yang menggunakan metode *supervised Learning*.

Model ini merupakan model yang paling sering dipakai dalam pengembangan sistem neural dan memiliki kinerja yang sangat baik dalam sisi keakuratan. Model ini mempunyai dua fase dalam pelatihannya yaitu fase *forward* dan *backward*. Cara kerja

jaringan ini adalah setelah *input* masuk ke *input layer* maka data akan diolah dan diteruskan ke masing-masing bagian di depannya sampai pada *output layer*. Nilai di

Output layer akan dibandingkan dengan nilai target, lalu akan dihasilkan sinyal *error* bagi masing *node* di *output layer*. Kemudian sinyal ini ditransmisikan kembali atau lebih dikenal dengan *Backpropagation* yang berasal dari lapisan keluaran ke masing-masing sel pada lapisan sebelumnya.



Gambar 2.9 Konsep Backpropagation.

Umumnya operasi model jaringan ini terdapat dua mekanisme kerja yaitu :

1. Mekanisme latihan atau belajar (*Training mode / Learning mode*). Pada mekanisme ini, jaringan akan dilatih untuk dapat menghasilkan data sesuai dengan target yang diharapkan melalui satu atau lebih pasangan-pasangan data (data *input* dan data target). Semakin lama waktu latihan maka kinerja jaringan akan semakin baik. Demikian juga dengan semakin banyak pasangan data yang digunakan dalam pelatihan maka kinerja akan semakin baik.
2. Mekanisme produksi (*Production Mode*) atau biasa disebut dengan mekanisme pengujian (*Try Out Mode*), pada mekanisme ini, jaringan diuji apakah dapat

mengenali sesuai dengan apa yang diharapkan, setelah melalui proses pelatihan terlebih dahulu.

2.6. Metode *Backpropagation*

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyinya. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*feedforward*) harus dikerjakan terlebih dahulu.

Inti dari *Backpropagation* adalah untuk mencari *error* suatu *node*. Dari hasil *forward phase* akan dihasilkan suatu *output*, dari *output* tersebut, pastilah tidak sesuai dengan target yang diinginkan. Perbandingan kesalahan dari target yang diinginkan dengan *output* yang dihasilkan disebut dengan *error*.

Menurut Haykin (1999) *Backpropagation* merupakan suatu teknik untuk meminimalisasi *gradient* pada dimensi *weight* dalam jaringan saraf tiruan lapis banyak (*multi layer neural network*), proses pelatihan akan dilakukan berulang-ulang sampai nilai *error* lebih kecil dari yang ditentukan.

Dalam proses pelatihan jaringan *Backpropagation* ini, digunakan fungsi nilai ambang batas *binary sigmoid*. Menurut Fausett (1994) sebelum melakukan proses pelatihan, terdapat beberapa parameter jaringan yang harus ditentukan dahulu, yaitu :

1. Tingkat pelatihan (*Learning rate*) yang dilambangkan dengan parameter, harus diberikan dan mempunyai nilai positif kurang dari satu. Semakin tinggi nilainya, maka semakin cepat kemampuan jaringan untuk belajar. Akan tetapi hal ini kurang

baik. Karena *error* yang dihasilkan tidak merata, tidak merata disini maksudnya adalah error untuk pembelajarannya tidak tentu, bisa lebih banyak di sisi tertentu.

2. Toleransi kesalahan (*error tolerance*), semakin kecil kesalahan, maka jaringan akan memiliki nilai bobot yang lebih akurat, tetapi akan memperpanjang waktu pelatihan.
3. Jumlah maksimal proses pelatihan yang dilakukan (*maximum epoch*), biasanya bernilai besar dan diberikan untuk mencegah terjadi perulangan tanpa akhir.
4. Nilai ambang batas atau bias (*threshold value*), dilambangkan θ . Parameter ini tidak harus diberikan (optional). Apabila tidak diberikan, maka nilainya sama dengan nol.

2.6.1. Penjelasan *Backpropagation*

Inti dari *Backpropagation* adalah untuk mencari *error* suatu *node*. Dari hasil *forward phase* akan dihasilkan suatu *output*, dari *output* tersebut, pastilah tidak sesuai target yang diinginkan. Perbandingan kesalahan dari target yang diinginkan dengan *output* yang dihasilkan disebut dengan *error*.

Dalam *Backpropagation* juga dikenal istilah yang disebut inisialisasi *output*. Inisialisasi *output* pada dasarnya adalah menentukan *error* di suatu *node* dengan sebuah target yang diinginkan. Karakteristik *Backpropagation* dapat diuraikan sebagai berikut:

1. *Node / processing element* dan fungsi aktivasi
 - a) Kontinu.
 - b) Dapat dideferensiasikan / diteruskan.
 - c) Turunan fungsi mudah dihitung.
 - d) Fungsi aktivasi yang biasa digunakan adalah fungsi *sigmoid*.

2. *Topology*

Terdiri dari satu lapisan masukan (*input layer*), satu atau lebih lapisan tersembunyi (*hidden layer*), dan satu lapisan keluaran (*output layer*). Setiap *neuron / processing element* pada suatu lapisan mendapat sinyal masukan dari semua *neuron* pada lapisan sebelumnya (beserta sinyal bias).

3. *Learning Rule*

Menggunakan *delta rule* atau *error connection Learning rule*.