

BAB II

TINJAUAN PUSTAKA

2.1. *Nurse Call*

Nurse Call System adalah alat yang berfungsi untuk memanggil Suster , alat ini sebagai komunikasi khusus antar pasien dan perawat dalam area rumah sakit , fungsi tujuan dari alat ini untuk mendukung dan mengoptimalkan tugas para perawat dalam melayani pasien yang sedang dirawat dalam ruang rawat inap.

Sistem Nurse call ini di design berdasarkan tujuan komunikasi atau skala prioritas yang dapat diketahui melalui warna lampu yang menyala serta jenis nada panggilnya. Dengan Begitu Perawat dapat mengambil keputusan mana yang harus di dahului berdasarkan prioritas pasien yang memanggil yang masuk ke meja nurse station atau meja counter perawat.

2.2. **Microphone**

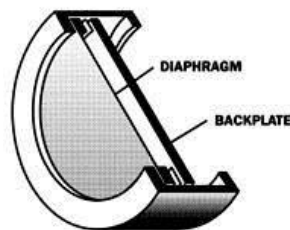
Microphone adalah suatu alat atau komponen Elektronika yang dapat mengubah atau mengkonversikan energi akustik (gelombang suara) ke energi listrik (Sinyal Audio). Ada beberapa jenis microphone diantaranya yang digunakan pada tugas akhir ini adalah condenser microphone. Semua jenis Mikrofon memiliki suatu bagian utama yang disebut dengan Diafragma. Condenser mic ditunjukkan pada Gambar 2.1 (Krezt, 2015) yang berfungsi sebagai alat untuk perekaman suara pada tugas akhir ini .



Gambar 2.1 Electret Microphone Condenser-1002

Condenser mic bekerja berdasarkan diafragma atau susunan blackplate yang harus tercatu oleh listrik membentuk sound-sensitive capacitor. Gelombang suara yang masuk ke microphone akan menggetarkan komponen diafragma ini. Letak dari diafragma ditempatkan di depan sebuah blackplate. Susunan dari

elemen ini membentuk sebuah kapasitor yang biasa disebut juga kondenser. Kapasitor memiliki kemampuan untuk menyimpan muatan maupun tegangan. Ketika elemen tersebut terisi dengan muatan, medan listrik akan terbentuk di antara diafragma dan blackplate, yang dimana besarnya itu proporsional terhadap ruang yang terbentuk diantaranya. Variasi akan lebar space antara diafragma dengan blackplate terjadi dikarenakan adanya pergerakan diafragma relative terhadap blackplate yang disebabkan oleh adanya tekanan suara yang mengenai diafragma. Hal ini menghasilkan sinyal elektrik dari gelombang suara yang masuk ke condenser microphone. Gambar 2.2 (Robby, 2017) menunjukkan gambar bagian dari condenser microphone



Gambar 2.2 Bagian dari Condenser mic

2.3. Module Bluetooth HC-05

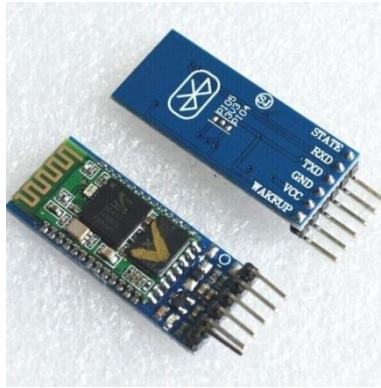
Module Bluetooth pada dasarnya mempunyai dua model yaitu *master device* dan *slave device*. Nama *device* setelah nomer yang genap menunjukkan *master* atau *slave* dan itu tidak dapat diganti. Contohnya :

Master device: HC-04 M, M = *Master Slave Device:*

HC-04-S, S = *Slave*

Tetapi apabila keluarannya adalah nomor ganjil HC-03, HC-05 untuk mengatur *slave* dan *master*nya cukup dengan mengubah pada *AT Command* (Maryanto, 2016). Komunikasi antara dua *module bluetooth* dapat terjadi paling tidak terdapat kondisi sebagai berikut.

1. Komunikasi harus berasal dari *master* dengan *slave*, apabila keduanya adalah *master* atau *slave* maka komunikasi tidak akan terjalin.
2. *Password* antara dua *modul bluetooth* harus benar. Gambar 2.3 menunjukkan gambar dari modul Bluetooth HC-05



Gambar 2.3 Module Bluetooth HC-05

2.4. Arduino Nano

Arduino merupakan sebuah platform dari physical computing yang bersifat open source. Arduino tidak hanya sekedar sebuah alat pengembang, tetapi merupakan kombinasi dari hardware, bahasa pemrograman dan Integrated Development Environment (IDE) yang canggih IDE adalah sebuah software yang berperan untuk menulis program, meng-compile menjadi kode biner dan meng-upload ke dalam memory microcontroller.

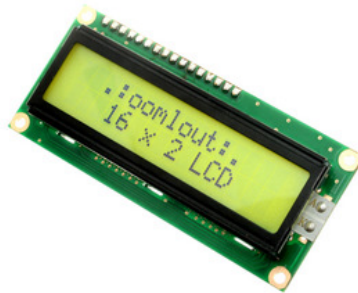


Gambar 2.4 Arduino Nano

Arduino Nano adalah salah satu board mikrokontroler yang berukuran kecil, lengkap dan mendukung penggunaan breadboard. Arduino Nano diciptakan dengan basis microcontroller ATmega328 (untuk Arduino Nano versi 3.x) atau Atmega 16 (untuk Arduino versi 2.x). Arduino Nano kurang lebih memiliki fungsi yang sama dengan Arduino Duemilanove, tetapi dalam paket yang berbeda. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, dan dihubungkan ke komputer menggunakan port USB Mini-B. Arduino Nano dirancang dan diproduksi oleh perusahaan Graviteth.

2.5. Liquid Crystal Display (LCD) 16 x 2

LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan seven-segment dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan sandwich memiliki polarizer cahaya vertikal depan dan polarizer cahaya horisontal belakang yang diikuti dengan lapisan reflektor. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan.



Gambar 2.5 Gambar LCD 16 x 2

2.6. Buzzer

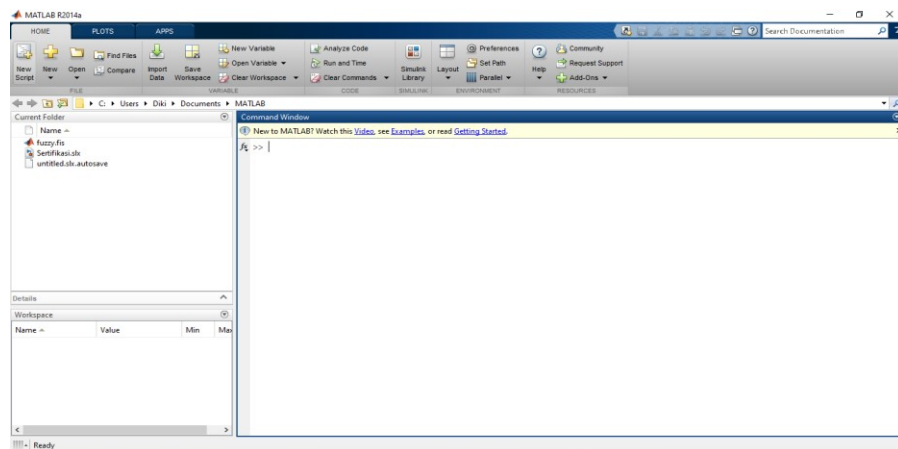
Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).



Gambar 2.6 Gambar Buzzer

2.7. MATLAB

MATLAB adalah bahasa komputasi teknis yang sangat terkenal. MATLAB sangat tangguh untuk operasi-operasi yang melibatkan matriks, seperti pemrosesan gambar, pemrosesan suara, persamaan polinomial, dan sebagainya. MATLAB sangat tangguh dalam hal simulasi dan display data, baik dalam bentuk gambar maupun grafik. Tidak hanya itu, dengan berbagai Toolbox yang dimilikinya, MATLAB menjadi perangkat lunak yang paling sering digunakan oleh akademisi, praktisi dan pihak terkait lainnya. Gambar 2.7 merupakan tampilan dari lembar kerja MATLAB.



Gambar 2.7 Tampilan Lembar Kerja MATLAB

Didalam MATLAB juga terdapat GUI (*Graphical User Interface*) yang berfungsi untuk membuat *design interface*. *Interface* yang dibangun adalah obyek grafik seperti tombol, kotak, panel, teks, slider, dan menu. Pada Tugas Akhir ini MATLAB digunakan sebagai media untuk klasifikasi data dan memproses data dari sinyal suara menjadi data yang dapat dijalankan

melalui Arduino serta pembuatan *interface*.

2.8. Arduino IDE

Arduino diciptakan untuk para pemula bahkan yang tidak memiliki basic bahasa pemrograman sama sekali karena menggunakan bahasa C++ yang telah dipermudah melalui *library*. Arduino menggunakan *Software Processing* yang digunakan untuk menulis program kedalam Arduino. *Processing* sendiri merupakan penggabungan antara bahasa C++ (Rusmala Dewi, 2012) dan Java. *Software* Arduino ini dapat di-install di berbagai *operating system* (OS) seperti: LINUX, Mac OS, Windows. Arduino tidak hanya sekedar sebuah alat pengembangan, tetapi kombinasi dari *hardware*, bahasa pemrograman dan *Integrated Development Environment* (IDE) yang canggih (Zheng, T., et al, 2016). IDE adalah sebuah *software* yang sangat berperan untuk menulis program, meng-*compile* menjadi kode biner dan meng-*upload* ke dalam memori mikrokontroler. *Software* IDE Arduino terdiri dari 3 (tiga) bagian:

1. Editor program, untuk menulis dan mengedit program dalam bahasa *processing*. *Listing* program pada Arduino disebut *sketch*.
2. *Compiler*, modul yang berfungsi mengubah bahasa *processing* (kode program) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh mikrokontroler.
3. *Uploader*, modul yang berfungsi memasukkan kode biner kedalam memori mikrokontroler (Arifin, Zulita, & Hermawansyah, 2016)

2.9. Mel-Frequency Cepstrum Coefficient (MFCC)

Sinyal suara merupakan gelombang longitudinal yang merambat melalui media (zat perantara). Batas frekuensi bunyi yang dapat didengar oleh manusia berkisar antara frekuensi 20 Hz sampai dengan 20 KHz, dan frekuensi yang dapat didengar dengan baik dan jelas oleh telinga manusia yaitu di atas 10.000 Hz (Pelton 1993).

Berdasarkan pada peubah bebas waktu (t) sinyal dibedakan menjadi dua jenis yaitu:

a. Sinyal Analog

Sinyal analog adalah suatu besaran yang berubah dalam waktu atau dalam ruang dan yang memiliki semua nilai untuk setiap nilai waktu (atau setiap nilai ruang). Sinyal analog sering disebut sinyal kontinu untuk menggambarkan bahwa besaran itu memiliki nilai yang kontinu (tak terputus). Sinyal analog dapat dilihat pada Gambar 2.8.



Gambar 2.8 Sinyal analog. (Ikra Dewantara, 2013)

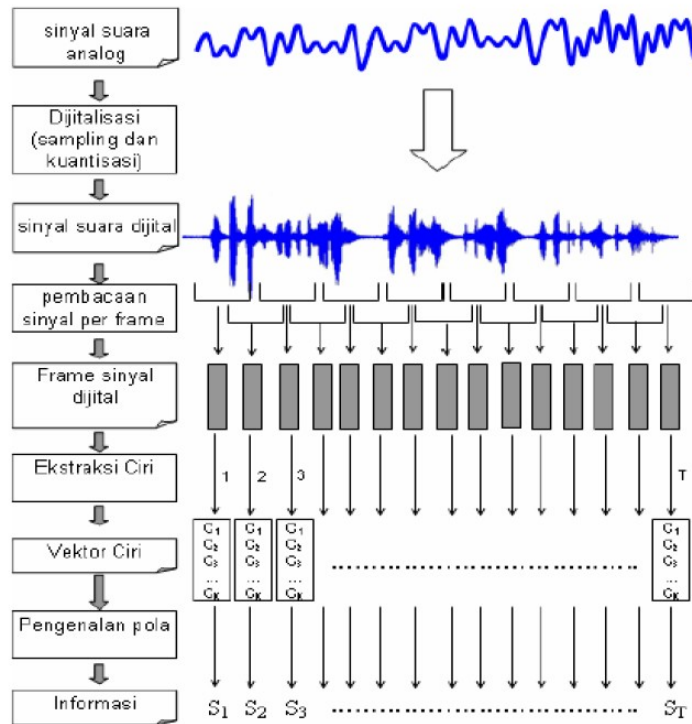
b. Sinyal Diskrit

Sinyal Diskrit merupakan suatu besaran yang berubah dalam waktu atau dalam ruang dan yang memiliki nilai pada suatu titik-titik waktu tertentu. Jarak setiap titik waktu bias saja berbeda-beda namun untuk kemudahan penurunan sifat matematikanya biasanya jarak antar titik waktu adalah sama. Sinyal diskret dapat dilihat pada Gambar 2.9.



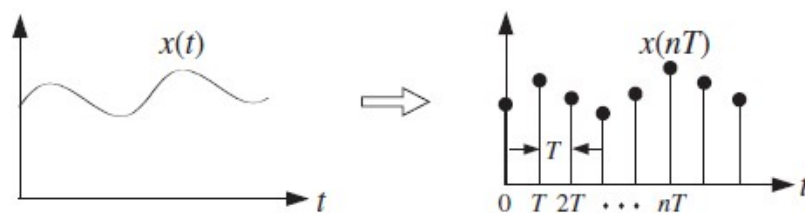
Gambar 2.9. Sinyal Diskrit (Ikra Dewantara, 2013)

Pemrosesan suara digital dapat dilakukan jika suara dalam bentuk sinyal analog yang bersifat kontinyu ditransformasi kedalam bentuk digital yang bersifat diskret. Menurut Buono *et al.* (2009), pemrosesan suara merupakan teknik transformasi sinyal suara menjadi informasi yang berarti sesuai dengan yang diinginkan. Secara umum proses transformasi diawali dari digitalisasi sinyal ekstraksi cirri dan diakhiri dengan pengenalan pola untuk klasifikasi. Tahapan transformasi sinyal analog sampai menghasilkan informasi disajikan pada gambar 2.10.



Gambar2.10.Tahapan transformasi sinyal menjadi informasi (Buono *et al.* 2009)

Menurut Orfanidis (2010) pemrosesan sinyal analog menjadi sinyal digital harus melalui proses *sampling* dan kuantisasi. *Sampling* merepresentasikan sinyal analog $x(t)$ yang diukur secara berkala setiap t detik sehingga waktu didiskritisasi dalam satuan *sampling interval* T . Ilustrasi *sampling* disajikan pada Gambar 2.11.



Gambar 2.11 Ilustrasi *sampling* (Orfanidis 2010)

Hasil dari *sampling* adalah representasi sebuah vektor yang terdiri atas nilai-nilai amplitudo terhadap waktu. Amplitudo merepresentasikan besar/kecilnya volume suara. Panjang suatu vektor ini ditentukan oleh durasi suara tersebut dan *sampling rate* yang digunakan. *Sampling rate* adalah banyaknya *sampling* yang diambil dalam waktu tertentu (t). Hubungan antara panjang vektor dan *sampling rate* dapat dinyatakan dalam Persamaan 1.

$$S = Fs \times t \quad (2.1)$$

S = Panjang vektor

Fs = *Sampling rate* yang digunakan (Hz)

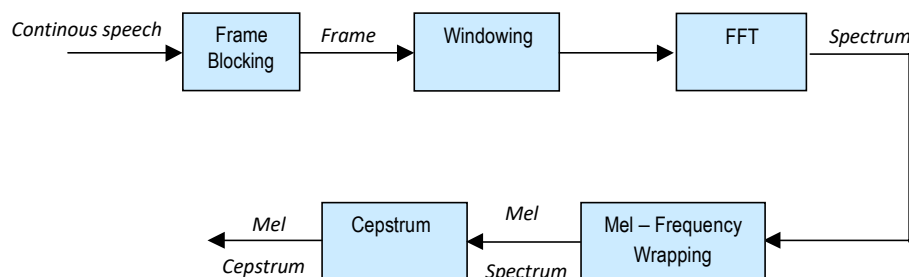
t = waktu(detik)

Tahapan selanjutnya ialah kuantisasi. Kuantisasi adalah pemetaan nilai-nilai amplitude kedalam representasi nilai 8 bit atau 16 bit.

2.9.1. Ekstraksi Ciri dengan MFCC

MFCC merupakan teknik ekstraksi ciri yang telah luas dipakai pada pemrosesan sinyal suara, terutama pada pengenalan pembicara. Penggunaan teknik ini pada system pemrosesan sinyal memberikan pengenalan yang lebih baik dibandingkan dengan metode lain yang sudah ada (Buono *et al.*2009).

Menurut Do (1994) MFCC adalah teknik ekstraksi ciri yang populer dan paling banyak digunakan. Blok diagram teknik MFCC disajikan pada Gambar 2.12.



Gambar 2.12 Diagram Block Teknik MFCC (Do, 1994)

Frame blocking dan overlapping: Sinyal suara dibaca per blok (*frame*), terdiri dari sejumlah N titik *samples*, dan antara dua *frame* yang bersebelahan terdapat *overlap*. *Overlap* ini dipisahkan oleh M ($M < N$). *Frame* pertama diawali dengan titik *sample* N , sedangkan *frame* kedua diawali dengan M *samples* setelah *frame* pertama. *Overlap* terjadi pada $N - M$ atau *frame* pertama yang saling tumpang tindih terhadap *frame* kedua.

Windowing: Proses *windowing* dilakukan pada setiap *frame* dengan tujuan untuk meminimumkan diskontinuitas antar dua *frame* yang bersebelahan, khususnya pada bagian awal dan akhir. Jika *window* didefinisikan sebagai $w(n)$, $0 \leq n \leq N - 1$, *windowing* dapat dihitung menggunakan perkalian vektor seperti pada Persamaan 2.

$$y(n) = x(n)w(n), 0 \leq n \leq N-1 \quad (2.2)$$

Fungsi window yang akan digunakan adalah *Hamming window* atau $w(n)$ yang disajikan pada Persamaan 3

$$w(n) = 0,54 - 0,46\cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N - 1 \quad (2.3)$$

dengan:

N = banyaknya *samples* dalam setiap *frame*

n = *frame* ke- n

$y(n)$ = output sinyal

$x(n)$ = Input Sinyal

$w(n)$ = fungsi *Hamming window*

FFT: *Fast fourier transform* (FFT) bertujuan melakukan konversi pada setiap *frame* yang terdiri atas N sampel dari domain waktu ke domain frekuensi. FFT adalah algoritma yang cepat untuk mengimplementasikan *Discrete Fourier Transform* (DFT) yang didefinisikan pada N sampel $\{x_n\}$. Fungsi FFT disajikan pada Persamaan :

$$F(k) = \sum_{n=0}^{N-1} f(n)W^{n.k}, 0 \leq n \leq N - 1 \quad (2.4)$$

Dimana $W_N = e^{-j\frac{2\pi}{N}}$

Hasil perhitungan DFT berjumlah N^2 , contoh terdapat 4 data maka dibutuhkan 16 kali perhitungan. Agar lebih mudah dipahami perhitungan tersebut dibentuk kedalam sebuah matriks seperti Persamaan 2.5

$$\begin{bmatrix} W_N^{0,0} & W_N^{1,0} & W_N^{2,0} & W_N^{3,0} \\ W_N^{0,1} & W_N^{1,1} & W_N^{2,1} & W_N^{3,1} \\ W_N^{0,2} & W_N^{1,2} & W_N^{2,2} & W_N^{3,2} \\ W_N^{0,3} & W_N^{1,3} & W_N^{2,3} & W_N^{3,3} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} \quad (2.5)$$

FFT mempercepat perhitungan DFT dengan setengah kali perhitungan DFT. DFT membutuhkan n^2 kali perhitungan, namun FFT hanya memerlukan $(n/2 + 1) \times n + n/2$ kali perhitungan (Adhitya, et al., 2016). Misalkan jumlah data $n = 50$ dengan menggunakan FFT cukup dilakukan $((26 \times 50) + 25) = 1.325$ kali perhitungan, sedangkan jika menggunakan DFT dibutuhkan sejumlah 2.500 kali perhitungan.

Perhitungan FFT dapat dilihat pada Persamaan 2.6 dibawah ini

$$F(k) = \sum_{n=0}^{N-1} f(n) \cos\left(\frac{2\pi nkT}{N}\right) - j \sum_{n=0}^{N-1} f(n) \sin\left(\frac{2\pi nkT}{N}\right) \quad (2.6)$$

Perhitungan DFT dapat disederhanakan dengan cukup menghitung setengah periode saja, setengah periode berikutnya dapat dihitung dengan menggunakan Persamaan 2.7 (Adhitya, et al., 2016):

$$F\left(x + \frac{T}{2}\right) = \text{Real}\left\{F\left(\frac{T}{2} - x\right)\right\} - j \text{Im}\{F(x)\} \quad (2.7)$$

Langkah berikutnya setelah didapatkan data FFT maka dilakukan perhitungan *Mel-Frequency*. Untuk setiap nada dengan frekuensi f , diukur dalam Hz, *pitch* subjektif diukur pada skala yang disebut "mel" skala. Skala mel frekuensi adalah frekuensi linier berada dibawah 1000 Hz dan logaritmik diatas 1000Hz (Muldayani, Purwanto, & Arief, 2015) .

Untuk menghitung mels untuk f frekuensi dalam Hz, dapat menggunakan rumus Persamaan 2.8 berikut

$$F(\text{mel}) = 1127 \log \left[1 + \frac{f}{700} \right] \quad (2.8)$$

langkah terakhir spektrum log mel dirubah kembali ke domain waktu menggunakan *Discrete cosine transform* (DCT) . Karena koefisien spektrum mel adalah bilangan real, maka perlu konversi ke domain waktu. Hasil dari perhitungan tersebut disebut *Mel Frequency Cepstrum Coefisien*

(MFCC). Perhitungan *Discrete cosine transform* (DCT) ditunjukkan pada Persamaan 2.9 berikut ini :

$$X[k] = \sum_{n=0}^{N-1} \left(\sqrt{\frac{2}{N}} \right) \cos \left(\left(n + \frac{1}{2} \right) k \cdot \frac{\pi}{N} \right) \quad (2.9)$$

Setiap sinyal suara akan menghasilkan pola MFCC yang berbeda- beda. Pola dari MFCC ini disimpan dan dijadikan *datasheet* yang nantinya digunakan sebagai *input* dari BP-NN.

2.10. Back Propagation Neural Network (BPNN)

Jaringan syaraf tiruan adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. JST dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa :

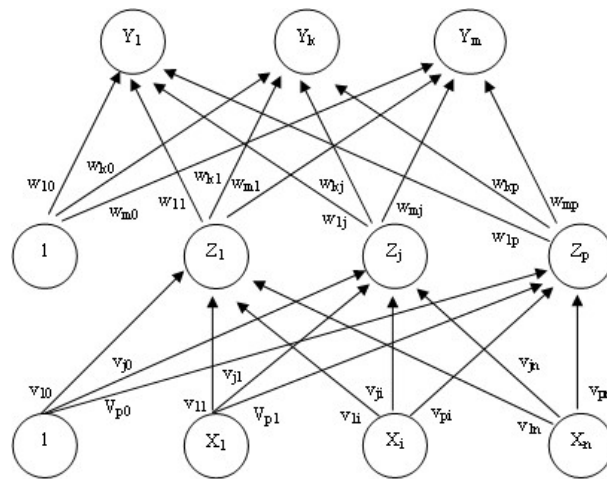
- a) Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron).
- b) Sinyal dikirimkan diantara neuron - neuron melalui penghubung-penghubung.
- c) Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
- d) Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

Sejarah JST pertama kali diperkenalkan pada tahun 1943 oleh McCulloch dan Pitts yang menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. Model Jaringan Syaraf Tiruan yang dikenalkan oleh McCulloch dan Pitts adalah model HEBB. Disusul oleh Resonblatt (1958) yang memperkenalkan perceptron kemudian Widrow dan Hoff (1960) yang mengembangkan perceptron dengan aturan pelatihan jaringan yang mana dengan aturan ini akan mengubah bobot perceptron apabila keluaran yang dihasilkan tidak sesuai dengan target yang diinginkan.

Jaringan Syaraf Tiruan ditentukan oleh 3 hal :

- Pola hubungan antar neuron (disebut arsitektur jaringan).
- Metode untuk menentukan bobot penghubung (disebut metode *training / learning / algoritma*).
- Fungsi aktivasi.

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi. Gambar 2.13 (Wirda Ayu U.,2005) adalah arsitektur backpropagation dengan n buah masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m buah keluaran.



Gambar 2.13 Arsitektur Backpropagation

v_{ji} merupakan bobot garis dari unit masukan x_i ke unit layer tersembunyi z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layer tersembunyi z_j). w_{kj} merupakan bobot dari unit layer tersembunyi z_j ke unit keluaran y_k (w_{k0} merupakan bobot dari bias di layer tersembunyi ke unit keluaran z_k).

Algoritma backpropagation menggunakan error output untuk mengubah nilai bobot- bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensiasikan, seperti sigmoid, tansig atau purelin.

2.10.1. ALGORITMA BACKPROPAGATION

Langkah 1 : Inisialisasi bobot dengan bilangan nilai acak kecil
Langkah 2 : Selama kondisi berhenti salah, kerjakan langkah 3 s.d. 8

Umpan Maju (Feedforward)

Langkah 3 : Tiap unit masukan ($x_i, i=1, \dots, n$) menerima isyarat masukan x_i dan diteruskan ke unit-unit tersembunyi (hidden layer)

Langkah 4 : Tiap unit tersembunyi ($z_j, z=1, \dots, p$) menjumlahkan bobot sinyal input.

$$Z_{injk} = v_{0j} + \sum_{i=1}^n x_i v_{ij}$$

dengan menerapkan fungsi aktivasi hitung :

$$Z_j = f(Z_{injk})$$

misal, fungsi aktivasi yang digunakan adalah sigmoid :

$$y = f(x) = \frac{1}{1+e^{-x}}$$

Langkah 5 : Tiap unit keluaran ($y_k, k=1, \dots, m$) menjumlahkan isyarat masukan berbobot

$$Y_{ink} = w_{0k} + \sum_{j=1}^p z_j v_{jk}$$

dengan menerapkan fungsi aktivasi hitung :

$$Y_j = f(Y_{ink})$$

Perambatan Galat Mundur (Backpropagation)

Langkah 6 : Tiap unit keluaran ($y_k, k=1, \dots, m$) menerima pola pelatihan masukannya.

Hitung galat (error) informasinya :

$$\delta_k = (t_k - y_k) f'(y_{ink})$$

Hitung koreksi bobot dan biasnya :

$$\Delta w_{jk} = \alpha \delta_k x_j$$

$$\Delta w_{0k} = \alpha \delta_k$$

Langkah 7 : Tiap unit tersembunyi ($z_j, z=1, \dots, p$) menjumlahkan delta masukannya (dari unit-unit yang berada pada lapisan atasnya).

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk}$$

Hitung galat (error) informasinya :

$$\delta_j = \delta_{in_j} f'(x_{in_j})$$

Hitung koreksi bobot dan biasnya :

$$\Delta v_{ij} = \alpha \delta_j x_i$$

Perbaiki bobot dan bias

Langkah 8 : Tiap unit keluaran (y_k , $k=1, \dots, m$) memperbaharui bobot dan bias ($j=0, 1, \dots, p$)

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$$

Tiap unit tersembunyi (z_j , $z=1, \dots, p$) memperbaharui bobot dan biasnya ($i=0, 1, \dots, n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$$

Langkah 9 : Uji syarat berhenti