BAB 2
LANDASAN TEORI

2.1 Intelegent transformation system

Intelligent Transportation Sistem (ITS) adalah kesatuan teknologi yang dibangun untuk mengatasi masalah-masalah yang terjadi pada transportasi [**7**]. Lebih jelasnya, intelligent transportation sistem adalah sistem yang mengintegrasikan transportasi dengan waktu yang akurat dan efisien dan di dalamnya diterapkan teknologi informasi, komputer, sensor, kontrol elektron, kontrol teori, sertai kecerdasan buatan [**8**]. Menurut Wewengkang, Intelligent Transportation Sistem umumnya membutuhkan teknologi semisal Automatic Visual Traffic Surveillance yang menggunakan kamera sebagai alat utama sebagai penerima sensor. Zhou di dalam penelitiannya yang berjudul, "Bus Passenger Recognitionand Track of Video Sequence", mengatakan bahwa pengenalan deteksi danpelacakan dari obyek-obyek yang bergerak menjadi salah satu fokus dalam penelitian dalam dunia komputer yang yang berhubungan dengan computer vision. Pada saat ini, penelitian mengenai pelacakan obyek menggunakan computer vision tersebut, umumnya digunakan pada traffic surv illance, pelayanan perbankan, homing guidance dan penerbangan militer. Dari sinilah, studi mengenai deteksi dan tracking pergerakan obyek menjadi hal yang realistik dan signifikan .

2.2 Kendaraan Bermotor

Menurut Peraturan Pemerintah Republik Indonesia Nomor 44 Tahun 1993 tentang “Kendaraan dan Pengemudi”, Kendaraan Bermotor adalah kendaraan yang digerakkan oleh peralatan teknik yang berada pada kendaraan itu. Umumnya kendaraan bermotor menggunakan mesin pembakaran dalam (perkakas atau alat untuk menggerakkan atau membuat sesuatu yg dijalankan dengan roda, digerakkan oleh tenaga manusia atau motor penggerak, menggunakan bahan bakar minyak atau tenaga alam).

Kendaraan bermotor memiliki roda, dan biasanya berjalan di atas jalanan. Berdasarkan UU No. 14 tahun 1992 , yang dimaksud dengan peralatan teknik

dapat berupa motor atau peralatan lainnya yang berfungsi untuk mengubah suatu sumber daya energi tertentu menjadi tenaga gerak kendaraan bermotor yang bersangkutan. Ada beberapa jenis kendaraan bermotor diantaranya mobil dan sepeda motor. Mobil adalah kendaraan darat yang digerakkan oleh tenaga mesin, beroda empat atau lebih (selalu genap), dengan menggunakan bahan bakar minyak seperti bensin atau solar untuk menghidupkan mesinnya. Contoh mobil dapat dilihat pada gambar 2.1



Gambar 2.1 Gambar Mobil

2.3 Citra

Dalam penelitian ini terdapat istilah “citra” yang dalam Bahasa Inggris disebut image. Rinaldi [**9**] berpendapat bahwa secara harafiah, citra (image) adalahgambar pada bidang dwimatra (dua dimensi). Berdasarkan keterangan tersebut, dapat disimpulkan bahwa contoh citra antara lain adalah foto dan lukisan. Citra ada dua macam, yaitu citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog, misalnya mata manusia dan kamera analog. Citra diskrit dihasilkan melalui proses digitalisasi terhadap citra kontinu. Beberapa sistem optik dilengkapi dengan fungsi digitalisasi sehingga ia mampu menghasilkan citra diskrit, misalnya kamera digital dan scanner. Citra diskrit disebut juga citra digital. Komputer digital yang umum dipakai saat ini hanya dapat mengolah citra digital [**9**].

2.3.1 Citra Digital

Agar dapat diolah dengan menggunakan komputer digital, maka suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Proses

representasi citra kontinu menjadi nilai-nilai diskrit disebut digitalisasi. Citra digital adalah hasil proses digitalisasi.

Suatu citra digital mempunyai fungsi dua dimensi f(x,y) yang memiliki ukuran M baris dan N kolom dimana x dan y adalah koordinat pada bidang dwimatra dan f(x,y) adalah intensitas cahaya (brightness) atau amplitudo atau derajat keabuan (grey level). Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar atau pixel) menyatakan nilai derajat keabuan pada titik tersebut. Citra digital berukuran N x M (tinggi = N, lebar = M) dinyatakan dengan matriks N x M

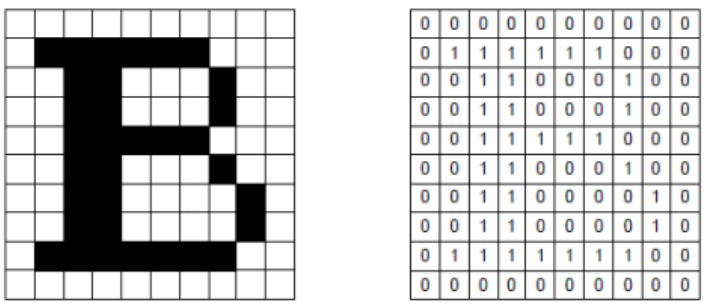
2.3.2 Tipe Tipe Citra Digital

Berdasar format penyimpanan nilai citra dibagi 3 tipe yaitu :

1. Citra biner (binary image) adalah citra yang hanya mempunyai dua nilai derajat keabuan: hitam dan putih. Pixel yang bernilai 1 melambangkan warna hitam, sedangkan pixel yang bernilai 0 adalah warna putih. Dinyatakan dalam suatu fungsi:

$$f(x,y) \sum\{0,1\} \qquad \text{..... (2.1)}$$

Berikut ini adalah contoh citra biner dan representasi nilai tiap pixel seperti pada gambar 2.2 [**9**].



Gambar 2.2 Representase Citra Biner

2. Citra yang terdiri dari satu layer warna dengan derajat keabuan tertentu. Untuk kebanyakan citra digital 8-bit, maka sistem Grey-scale diukur berdasarkan skala intensitas kecerahan, yang bernilai 0 – 255, dimana yang hitam pekat adalah 0 dan yang terputih adalah 255. Dinyatakan dalam suatu fungsi :

$$f(x, y) \sum[0 \ldots 255] \quad \ldots\ldots (2.2)$$

Gambar 2.3 menunjukkan contoh citra skala keabuan 8-bit dengan ukuran 512 x 512 pixel [**9**].



Gambar 2.3 Citra Kapal

3. Citra Berwarna

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang-nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak 256 x 256 x 256 = 16.777.216 jenis warna.

Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y

dan komponen-z. Misalkan sebuah vektor dituliskan sebagai r = (x,y,z). Untuk warna, komponen-komponen tersebut digantikan oleh komponen R (Red), G (Green), B (Blue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB(xxx, xxx, xxx). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB(0,0,0). Gambar 2.4 menunjukkan contoh citra berwarna 8-bit. Representasi dalam citra digital dinyatakan dalam persamaan:

$$f_R(x,y) \sum[0 \dots 255]$$
$$f_G(x,y) \sum[0 \dots 255] \quad \text{.....} \quad (2.3)$$
$$f_B(x,y) \sum[0 \dots 255]$$



Gambar 2.3 Citra Pepper

2.4 Computer Vision

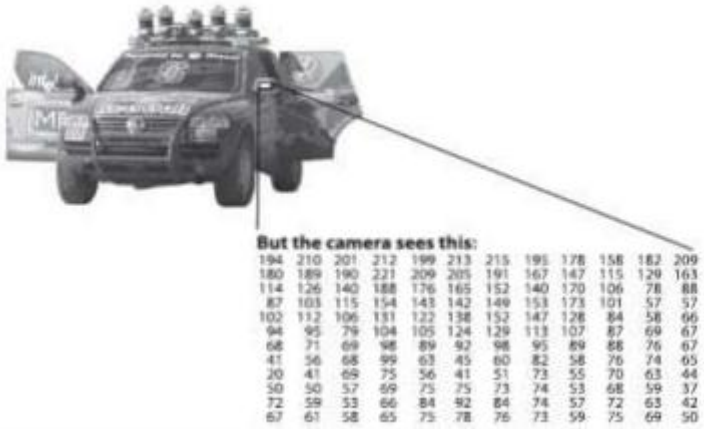
Pada hakikatnya, computer vision mencoba meniru cara kerja sistem visualmanusia. Dalam proses penglihatan manusia, manusia melihat objek dengan menggunakan indera penglihatan yang berupa mata, lalu citra objek diteruskan ke otak untuk diinterpretasikan sehingga manusia mengerti objek apa yang tampak. Hasil interpretasi ini kemudian digunakan untuk pengambilan keputusan [**9**] .

Tabel 2.1 Tabel Human Vision dan Computer Vision

Human Vision	Computer Vision
Menggunakan mata dan visual cortex	Menggunakan kamera-kamera yang

di dalam otak.	terhubung pada sistem komputer
Menemukan dari gambar objek apa yang ada dalam penglihatan, dimana posisinya, bagaimana mereka bergerak, dan apa bentuknya.	Secara otomatis menginterpretasi gambar-gambar dan mencoba untuk mengerti isinya seperti pada human vision

Secara garis besar, computer vision adalah sebuah teknologi mesin yang mampu mengenali objek yang diamati. Kemampuan untuk mengenali inimerupakan kombinasi dari pengolahan citra dan pengenalan pola. Pengolahan citra adalah proses awal dalam computer vision untuk menghasilkan citra yang lebih baik atau lebih mudah diinterpretasikan, sedangkan pengenalan pola adalah proses identifikasi objek pada citra. Dalam sistem penglihatan mesin (machine vision) komputer hanya mampu mendapatkan informasi berupa kumpulan angka-angka dari media input data seperti kamera atau disket. Berikut contoh gambar untuk mengilustrasikannya.



Gambar 2.4 Ilustrasi Penglihatan Komputer

2.5 Pengolahan Citra Digital

Pengolahan citra adalah pemrosesan citra dengan menggunakan komputer agar kualitasnya lebih baik. Tujuannya agar citra dapat dengan mudah diinterpretasi oleh manusia atau komputer. Pada gambar 2.5 digambarkan bahwa teknik pengolahan citra memanipulasi citra menjadi citra lain. Jadi, masukannya

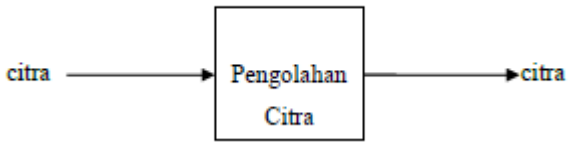
adalah citra dan keluarannya juga citra. Namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan.

Pada umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila [**10**]:

a. Perbaikan atau modifikasi citra perlu dilakukan untuk meningkatkan kualitas

   penampakan atau menonjolkan beberapa aspek informasi yang terkandung di dalam citra,

b. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
c. Sebagian citra perlu digabung dengan citra yang lain.



Gambar 2.5 Pengolahan Citra

2.5.1 Metode Ubah Nilai RGB ke Grayscale

Untuk memudahkan pengolahan citra, biasanya citra RGB akan dikonversi menjadi citra skala keabuan (grayscale). Karena pada citra RGB pengolahan citra akan dilakukan dengan menghitung pada 3 channel warna, sedangkan pada citra skala keabuan cukup 1 channel warna sehingga memudahkan proses komputasi. Format citra skala keabuan memakai warna hitam sebagai warna minimal (0) dan warna putih (255) sebagai warna maksimalnya, sehingga warna antaranya adalah abu-abu seperti ditunjukkan pada gambar.

Terdapat tiga macam metode algoritma untuk mengubah nilai R G B menjadi grayscale antara lain:

1. Lightness

Algoritmanya adalah mencari nilai tertinggi dan terendah dari nilai R G B, kemudian nilai tertinggi dan terendah tersebut dijumlahkan lantas dikalikan dengan 0.5 (dibagi 2). Secara matematis dapat dirumuskan :

Grayscale = (max(R,G,B) + min(R,G,B)) * 0.5

….. (2.4)

2. Average

Algoritmanya adalah dengan menjumlahkan seluruh nilai R G B, kemudian dibagi 3, sehingga diperoleh nilai rata-rata dari R G B, nilai rata-rata itulah yang dapat dikatakan sebagai grayscale. Rumus matematisnya adalah :

Grayscale = (R + G + B) / 3

….. (2.5)

3. Luminosity

Algoritmanya adalah dengan mengalikan setiap nilai R G B dengan konstanta tertentu yang sudah ditetapkan nilainya, kemudian hasil perkalian seluruh nilai R G B dijumlahkan satu sama lain. Rumus matematisnya adalah :

Grayscale = (0.21 * R) + (0.72 * G) + (0.07 * B) ….. (2.6)

Untuk konversi RGB ke grayscale dengan metode Luminosity, konstanta yang digunakan lebih dititikberatkan pada channel warna hijau (green) karena mata manusia lebih sensitif pada warna hijau daripada warna lainnya. Rekomendasi nilai konstanta Luminosity yang umum digunakan pada perangkat pengolahan citra saat ini ada 2 tipe, yaitu BT.709 dan BT.601 yang direkomendasikan oleh International Telecomuncation Union (ITU). BT.709 adalah standar format untuk high-definition television (HDTV), sedangkan BT.601 adalah standar format untuk standar-definition television (SDTV).

Adapun nilai konstanta Luminosity yang diberikan pada rekomendasi BT.709 adalah seperti pada persamaan 2.7, sedangkan untuk rekomendasi BT.601 dapat dilihat pada rumus matematis berikut:

Grayscale = (0.299 * R) + (0.587 * G) + (0.114 * B) ….. (2.7)

2.5.2 Histogram citra

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas pixel dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (relative) dari intensitas

pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (brightness) dan kontras (contrast) dari sebuah gambar [**9**].

Umumnya setiap pixel memiliki nilai derajad keabuan. Misalkan citra dengan L derajat keabuan, yaitu dari nilai 0 sampai L-1. Secara sistematis histogram citra dihitung dengan rumus

$$h_i = \frac{n_i}{n} \quad , i = 0,1,\ldots, L-1$$

…..(2.8)

Sebagai contoh penghitungan histogram citra, misalkan matriks citra digital berukuran 8 x 8 pixel dengan derajat keabuan dari 0 sampai 15 (ada 16 buah derajat keabuan)

$$\begin{bmatrix} 3 & 7 & 7 & 8 & 10 & 12 & 14 & 10 \\ 2 & 0 & 0 & 0 & 1 & 8 & 15 & 15 \\ 14 & 6 & 5 & 9 & 8 & 10 & 9 & 12 \\ 12 & 12 & 11 & 8 & 8 & 10 & 11 & 1 \\ 0 & 2 & 3 & 4 & 5 & 13 & 10 & 14 \\ 4 & 5 & 0 & 0 & 1 & 0 & 2 & 2 \\ 15 & 13 & 11 & 10 & 9 & 9 & 8 & 7 \\ 2 & 1 & 0 & 10 & 11 & 14 & 13 & 12 \end{bmatrix}$$

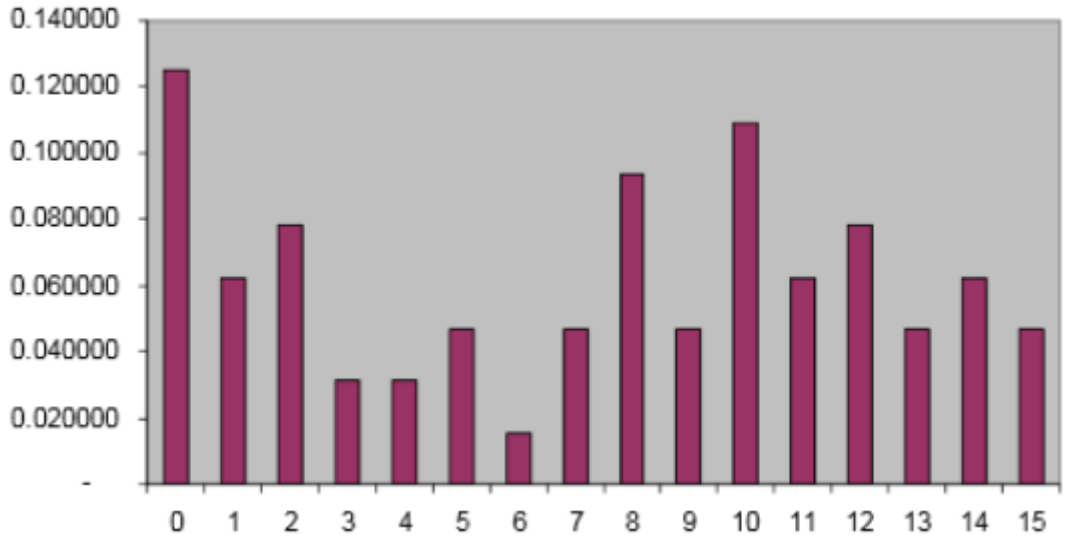
Gambar 2.6 Matrix Citra Digital

Tabulasi perhitungan histogramnya dapat dilihat pada tabel 2.2 Dapat dilihat bahwa semakin besar nilai ni maka semakin besar pula nilai hi [**9**].

Table 2.2 Perhitungan Histogram

i	n_i	$h_i = n_i / n$ (n=64)
0	8	0.125
1	4	0.0625
2	5	0.078125
3	2	0.03125
4	2	0.03125
5	3	0.046875
6	1	0.015625
7	3	0.0468875
8	6	0.09375
9	3	0.046875
10	7	0.109375
11	4	0.0625
12	5	0.078125
13	3	0.046875
14	4	0.0625
15	3	0.046875

Histogram citra digital dari matriks citra digital diatas dapat dilihat pada Gambar 2.7 berikut ini:

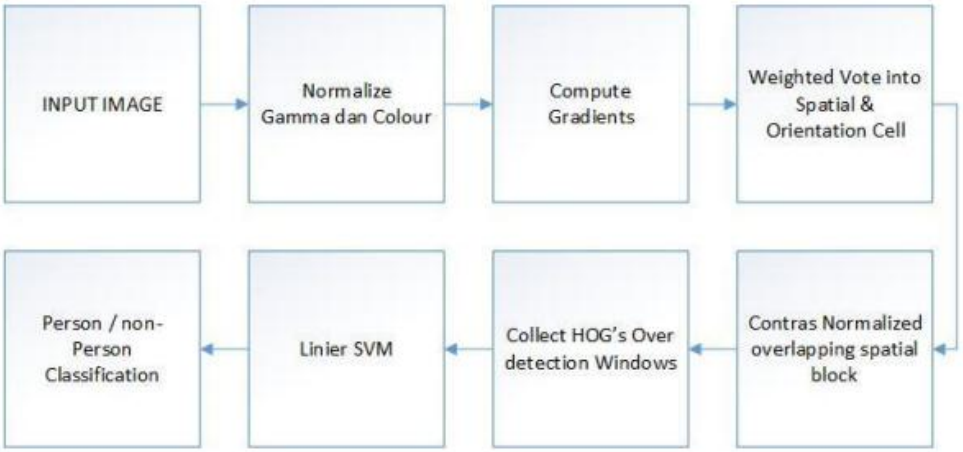


Gambar 2.7 Histogram Citra Digital

2.6 Histogram of Oriented Gradients

Tujuan dari proses analisis citra adalah mengidentifikasi parameterparameter yang diasosiasikan dengan ciri (feature) dari objek di dalam citra, untuk selanjutnya parameter tersebut digunakan dalam menginterpretasi citra. Tahappertama analisis citra yaitu ekstraksi ciri (feature extraction).

Histograms of Oriented Gradients (HOG) adalah pendeskripsi ciri (feature descriptor) yang digunakan dalam computer vision dan image processing untuk pendeteksian objek. Metode HOG diperkenalkan pertama kali oleh Dalal dan Triggs pada tahun 2005 . Metode HOG dalam penelitian ini akan digunakan untuk menguji apakah dalam citra uji terdapat manusia atau tidak. Gambar 2.8 menunjukkan langkah kerja proses ekstraksi fitur dari metode HOG.



Gambar 2.8 Proses Ekstraksi fitur HOG

Pertama, proses normalize gamma and colour atau normalisasi gamma dan warna pada citra. Tahap ini adalah tahap opsional yang berfungsi untuk mengurangi pengaruh efek pencahayaan pada citra.

Proses kedua adalah compute gradients atau perhitungan gradien pada frame citra. Pada tahap kedua ini, citra pertama kali dikonversi kedalam bentuk citra skala keabuan. Kemudian perhitungan gradien dilakukan dengan menerapkan metode sobel pada citra secara vertikal dan horisontal dengan filter kernel:

$$D_2(x)\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \quad dan\ D_2(y)\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}$$

…... (2.9)

Dengan melambangkan image sebagai I, kita akan menghitung nilai x dan y menggunakan operasi konvolusi :

$$I_x = I \times D_x \ dan \ I_y = I \times D_y$$

……(2.10)

Nilai magnitude dan orintasi gradien dapat dihitung setelah mendapatkan nilai x, y derivatives (I_x dan I_y) dengan persamaan:

$$|G| = \sqrt{I_x^2 + I_y^2} \ dan \ \theta = \arctan\frac{I_y}{I_x}$$

……(2.11)

Langkah berikutnya yaitu weighted vote into spatial and orientation cells. Dalam tahap ini, gambar dibagi dalam ruang-ruang kecil yang disebut cell. Cell adalah sebuah ruang yang berukuran 4x4 pixel. Setiap pixel dalam cell dikelompokkan dalam bin berdasarkan nilai orientasi yang didapat dari perhitungan gradien. Tiap-tiap cell tersebut akan dibuat sebuah histogram dengan 9 channel histogram yang didistribusikan kedalam orientasi sudut 0° - 180°.

Selanjutnya tahap normalisasi block. Pada saat proses compute gradient di peroleh nilai gradien yang berbeda-beda. Sehingga perlu pengelompokkan tiap cells. Cells perlu dikelompokkan lagi menjadi kelompok yang lebih besar yangdisebut block. Performa terbaik yang dihasilkan dalam deteksi manusia adalah dengan menggunakan 3x3 cell per block [1]. Sedangkan setelah pengelompokkan menjadi block, block ini biasanya terjadi overlap atau tumpang tindih sehingga harus dilakukan normalisasi. Teknik normalisasi yang sering digunakan antara lain:

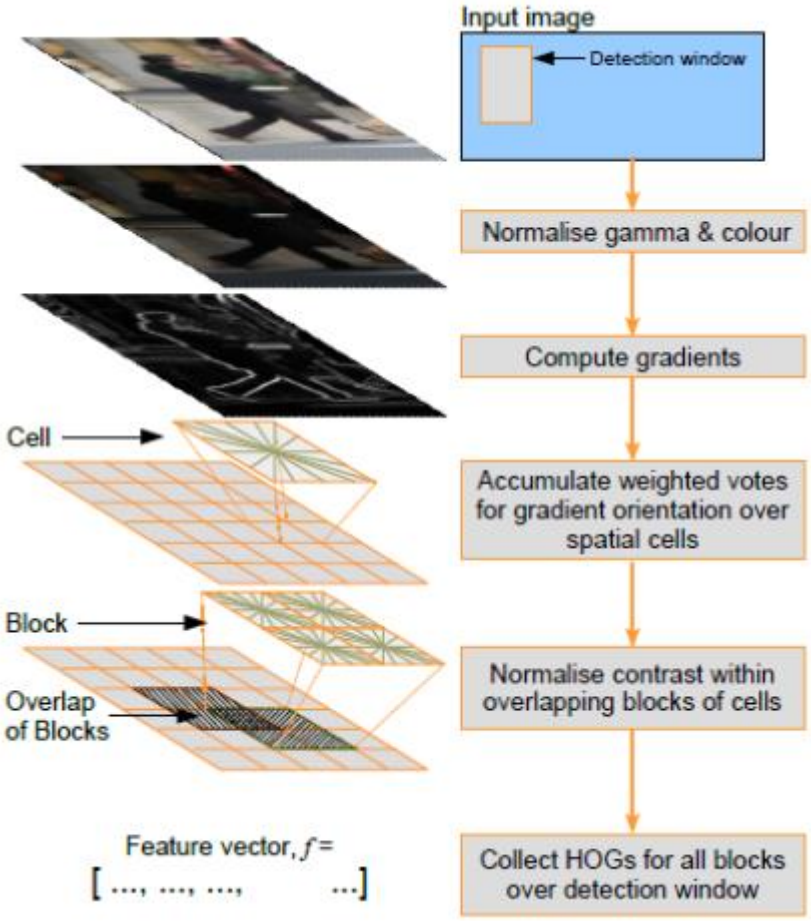
$$L2-norm: \ v \rightarrow \frac{v}{\sqrt{\left(||v||_2^2 + \varepsilon^2\right)}} \qquad \text{……(2.12)}$$

$$\text{L1-norm:} \qquad v \rightarrow \frac{v}{\sqrt{(\|v\|_1 + \varepsilon)}} \qquad \text{……(2.13)}$$

$$\text{L1-sqrt:} \qquad v \rightarrow \frac{v}{(\|v\|_1 + \varepsilon)} \qquad \text{……(2.14)}$$

Langkah kelima yaitu collect HOG over detection window. Semua histogram untuk setiap block akan digabungkan untuk menghasilkan sebuah feature vector. Feature vector ini akan digunakan dalam Support Vector Machine untuk proses pendeteksian manusia.

Untuk lebih jelasnya mengenai istilah-istilah seperti cells, blocks, overlap block, dan lain-lain dapat dilihat pada gambar 2.9



Gambar 2.9 Gambaran Umum Metode HOG

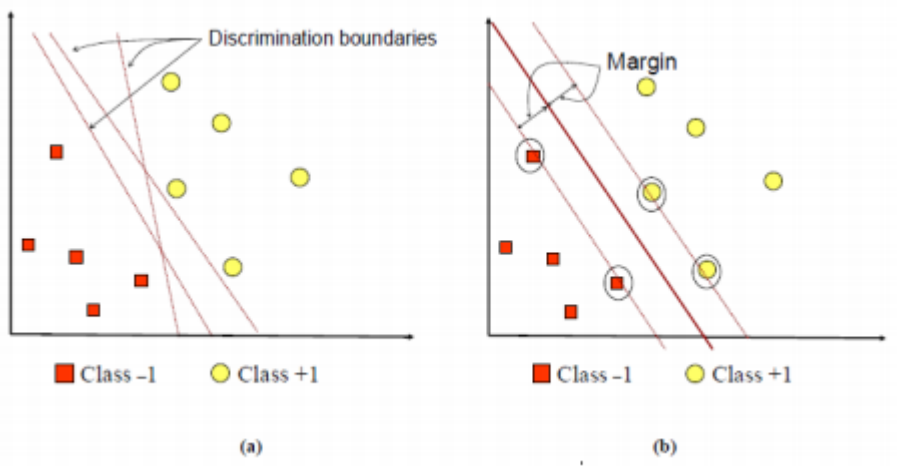
2.7 Support vector machine

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang pattern recognition. SVM adalah metode learning machine yang bekerja atas prinsip Structural Risk Minimization (SRM) dengan tujuan menemukan hyperplane terbaik yang memisahkan dua buah class pada input space [**11**].

SVM dalam penelitian ini termasuk dalam rangkaian metode HOG yang merupakan alat untuk menetapkan apakah dalam detection window terdapat sosok manusia atau tidak. SVM classifier berfungsi untuk memisahkan class manusia dan nonmanusia.

Pada gambar 2.10a terdapat beberapa pattern yang terbagi dalam dua buah class-1 dan class+1. Pattern yang tergabung dalam class-1 dilambangkan dengan kotak warna merah, sedangkan pattern yang ada dalam class+1 dilambangkan dengan lingkaran warna kuning. Masalah klasifikasi dapat diterjemahkan dengan usaha mencari garis (hyperlane) yang memisahkan antara kedua kelompok tersebut. Beberapa alternatif garis pemisah ditampilkan dalam gambar 2.10a.

Gambar 2.10b menunjukkan hyperlane yang memisahkan class-1 dengan class+1 yang memiliki margin tertentu. Margin adalah jarak antara hyperlane dengan pattern terdekat dari masing2 class. Pattern terdekat ini disebut sebagai support vector



Gambar 2.10 SVM Mencari Hyperlane Terbaik

Data yang tersedia dinotasikan sebagai sedangkan label masing masing dinotasikan $y_i \in \{-1,+1\}$ untuk i = 1,2,… ,l, yang mana l adalah banyaknya data. Diasumsikan kedua class-1 dan class+1 dapat terpisah secara sempurna oleh hyperplane berdimensi d , yang didefinisikan

$$\vec{w}.\vec{x_i} + b = 0 \qquad \text{…..(2.15)}$$

Maka untuk pattern $\overrightarrow{x_i}$ yang termasuk kelas-1 dirumuskan sebagai pattern yang memenuhi pertidaksamaan:

$$\vec{w}.\overrightarrow{x_i} + b \leq -1 \quad \text{.....(2.16)}$$

Dari persamaan diatas kita mendapatkan persamaan baru yang equal yaitu:

$$y_i\,(\overrightarrow{x_i}\,\vec{w} + b) \leq 1 \quad \text{.....(2.17)}$$

sedangkan pattern $\overrightarrow{x_i}$ yang termasuk kelas+1 dirumuskan sebagai pattern yang memenuhi pertidaksamaan:

$$\vec{w}.\overrightarrow{x_i} + b \geq +1 \quad \text{....(2.18)}$$

Dari persamaan diatas kita mendapatkan persamaan baru yang equal yaitu:

$$y_i(\overrightarrow{x_i}\,\vec{w} + b) \leq 1 \quad \text{....(2.19)}$$

dari persamaan diatas dapat disimpulkan bahwa nilai dari support vector akan selalu sesuai dengan persamaan:

$$y_i(\overrightarrow{x_i}\,\vec{w} + b) - 1 = 0 \quad \text{....(2.20)}$$

Untuk menemukan margin dari kelas+1 dan kelas-1 dapat dihitung dengan cara menghitung jarak dari x+ dan x- sebagai berikut:

$$width = (x_+ - x_-).\frac{\vec{w}}{\|w\|} \quad \text{....(2.21)}$$

nilai x+ dan - x– masing-masing adalah:

$$x_+ = 1 - b \quad \text{....(2.22)}$$

$$-x_- = 1 + b \quad \text{....(2.21)}$$

maka didapat persamaan :

$$width = (x_+ - x_-).\frac{\vec{w}}{\|w\|} = \frac{2}{\|w\|} \quad \text{....(2.23)}$$

Untuk memaksimalkan margin maka dilakukan dengan memaksimalkan nilai width:

$$max\,\frac{2}{\|w\|} \quad \text{....(2.24)}$$

Karena nilai $\max\frac{2}{\|w\|}$ adalah untuk jarak antar support vector, maka nilai dari hyperlane

Adalah $\max \frac{1}{\|w\|}$atau dapat dengan meminimalkan nilai$\|w\|$. Hal ini dapat dirumuskan

dengan Qudratic Programming Problem dengan mencari :

$$\min \frac{1}{2} \|w\|^2 \quad \text{....(2.25)}$$

Problem ini dapat diselesaikan dengan Langrange Multiplier:

$$L = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{i} \alpha_i (y_i ((\overrightarrow{x_i}.\vec{w} + b) - 1)) \quad \text{....(2.26)}$$

α_i adalah Langrange Multiplier, yang bernilai nol atau positif ($\alpha_i \geq 0$). Nilai optimal dari persamaan dapat dihitung dengan meminimalkan L terhadap $\vec{w}$ dan b, dan $\vec{x}_i \vec{x}_j$ memaksimalkan L terhadap α_i Dengan memperhatikan sifat bahwa titik optimal gradien L=0, persamaan dapat dimodifikasi sebagai maksimalisasi problem yang hanya mengandung α_i saja, sebagaimana persamaan dibawah:

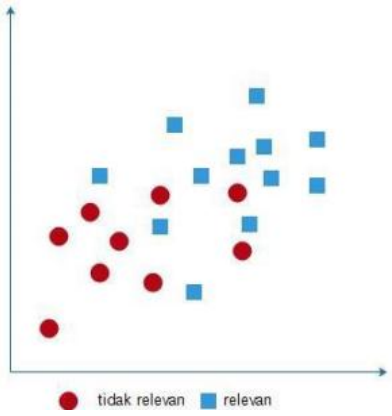
$$L = \sum_{i=1}^{i} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{i} \alpha_i \alpha_j \, y_i y_j \;\; \vec{x}_i \vec{x}_j \quad \text{....(2.27)}$$

Subject to:

$$\alpha_i \geq 0 (i: 1,2, \ldots . , l) \;\; \sum_{i=1}^{i} \alpha_x \alpha_y \quad \text{....(2.28)}$$

Dari hasil perhitungan ini diperoleh α_i yang kebanyakan bernilai positiv. Data yang berkolerasi dengan α_i yang positiv inilah yang disebut support vector.

Dot product $\overrightarrow{x_i}\,\overrightarrow{x_j}$ dari persamaan 2.27 sering diganti dengan simbol K. K adalah matrik kernel. Matrik kernel ini digunakan untuk membuat data set yang bersifat non-linear menjadi linear. Contoh dataset yang bersifat non linear dapat dilihat pada Gambar 2.11



Gambar 2.11 DataSet Non Linier

Oleh karena itu dalam SVM terdapat fungsi kernel yang dapat mengubah data set yang tidak linear menjadi linier dalam space baru.

Table 2.3 Kernel Trik pada SVM

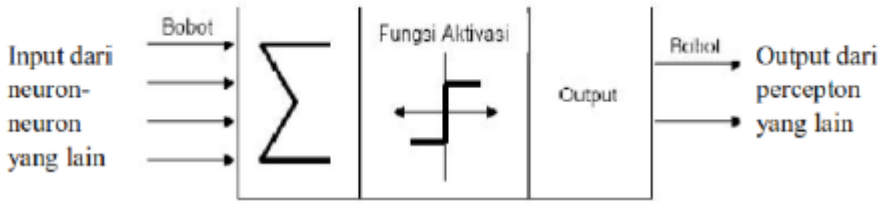
Kernel type	definisi
Linear	$K(x_i, x_j) = x^T\, x$
polynomial	$K(x_i, x_j) = (x^T\, x_i + 1)^p$
gaussian	$K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$

Pemilihan fungsi kernel yang tepat adalah hal yang sangat penting, karena fungsi kernel ini akan menentukan feature space dimana classifier akan dicari.

2.8 Neural Network

Neural Network atau Jaringan syaraf tiruan adalah salah satu kajian pada kecerdasan buatan. Neural Network digunakan untuk menyelesaikan masalah-masalah yang kompleks dan sulit dipahami, dimanan sejumlah besar data mengenai masalah tersebut telah dikumpulkan. Neural Network mencari pola dan hubungan dalam data yang sangat besar yang terlalu rumit dan sulit dianalisis manusia menggunakan perangkat keras dan piranti lunak yang menyerupai pola-pola pemrosesan dalam otak manusia.

Komponen Neural Network terdiri dari neuron-neuron yang saling berhubungan. Neuron – neuron ini akan mentransformasikan informasi yang diterima melalui sambungan keluarannya menuju neuron-neuron yang lain. Pada Neural Network hubungan neuron-neuron tersebut dikenal dengan nama bobot [**12**] . Komponen Neural Network seperti terlihar pada gambar 2.1 terdiri dari input (informasi) bobot (nilai- nilai tertentu), fungsi aktivasi berfungsi apabila input yang dimasukan sesuai dengan nilai ambang (threshold) yang ditentukan jika tidak sesuai maka fungsi aktivasi tidak diaktifkan, dan apabila neuron – neuron tersebut diaktifkan maka neuron akan mengirimkan output melalui bobot – bobot.

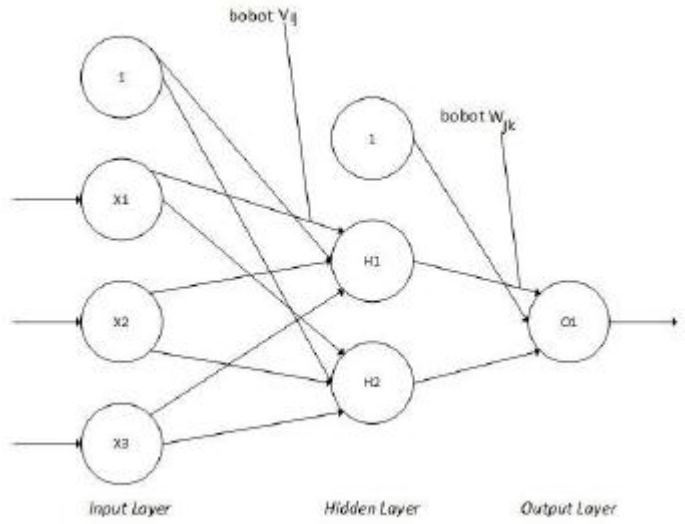


Gambar 2.11 Komponen Neural Network

Karakterisitik yang dimiliki Neural Network adalah sebagai berikut:

a. Arsitektur Jaringan

Merupakan gambaran pola hubungan antara neuron. Pada algoritma Neural Network neuron-neuron disusun pada suatu lapisan (layer), umumnya lapisan-lapisan tersebut terdiri dari lapisan masukan (input layer), lapisan tersembunyi (hidden layer) dan lapisan keluaran (output layer). Neuron-neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama. Factor terpenting dalam menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya. Pada setiap lapisan yang sama, neuron-neuron akan memiliki fungsi aktivasi yang sama. Arsitektur Jaringan Neural Network yang digunakan pada penelitian ini adalah arsitektur jaringan dengan banyak lapisan (multilayer net) yang memiliki satu lapisan tersembunyi (hidden layer) diantara lapisan masukan dan lapisan keluaran. Arsitektur yang akan digunakan dapat dilihat pada gambar 2.12 berikut:



Gambar 2.12 Arsitektur Neural Network

Dimana:

X = input neuron, merupakan variabel lapisan masukan

H = hidden neuron, merupakan output dari lapisan masukan

O = output neuron, merupakan output dari lapisan tersembunyi

V_{ij}= bobot yang menghubungkan lapisan masukan menuju lapisan tersembunyi

W_{jk}= bobot yang menghubungkan lapisan tersembunyi menuju lapisan Keluaran

b. Pembelajaran

Algoritma Neural Network mempuyai dua macam proses pembelajaran yaitu pembelajaran terawasi (supervised learning) dan pembelajaran tidak terawasi (unsupervised learning).

1. Pembelajaran terawasi (supervised learning)

   Pada Pembelajaran terawasi output yang diharapakan telah diketahui sebelumnya. Pada proses pembelajaran, satu pola input akan diberikan ke satu neuron pada lapisan input. Pola ini akan dirambatkan di sepanjang jaringan syaraf hingga sampai ke neuron pada lapisan output. Lapisan output ini akan membangkitkan pola output yang nantinya akan dicocokkan

dengan pola output targetnya. Apabila terjadi perbedaan antara pola output hasil pembelajaran dengan pola target, maka disini akan muncul error. Apabila nilai error ini masih cukup besar, mengindikasikan bahwa masih perlu dilakukan lebih banyak pembelajaran lagi. Metode pembelajaran terawasi diantaranya Hebb Rule, Perceptron, Delta Rule, Backpropagation, Learning Vektor Quantization, Heteroassociative Memory [**12**].

2. Pembelajaran tak terawasi (*unsupervised learning*)

   Pada pembelajaran tak terawasi tidak memerlukan target output. Pada pembelajaran ini, tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran iniadalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu [**12**].

3. Fungsi aktivasi

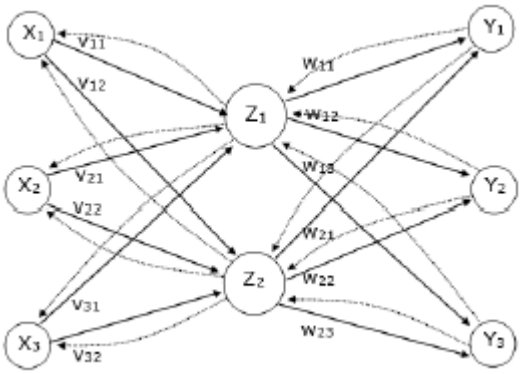
   Fungsi aktivasi yang mentranformasikan total input Neural Network pada suatu neuron untuk menghasilkan sinyal keluaran (outgoing activity). Fungsi aktivasi yang digunakan pada penelitian ini adalah fungsi sigmoid. Fungsi ini digunakan untuk Neural Network yang dilatih dengan menggunakan metode backpropagation. Fungsi sigmoid memiliki nilai pada range 0 sampai 1. Fungsi ini sering digunakan untuk jaringan yang membutuhkan nilai output yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai outputnya 0 atau 1. Fungsi sigmoid terlihat pada persamaan 2.9 berikut [**12**].

$$f(x) = \frac{1}{1+e^{-x}} \qquad \ldots.(2.29)$$

2.8.1 Backpropagation Neural Network

Backpropagation neural network merupakan tipe jaringan saraf tiruan yang menggunakan metode pembelajaran terawasi (supervised learning). Pada supervised learning terdapat pasangan data input dan output yang dipakai untuk melatih Neural Network hingga diperoleh bobot yang diinginkan. Metode Backpropagation menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (backward). Untuk mendapatkan error ini, tahap perambatan maju (forward) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid, seperti terlihat pada persamaan 2.29.

Arsitektur Backpropagation Neural Network terdiri atas tiga lapis (layer) yaitu lapis masukan (input layer) x_i, lapis tersembunyi (hidden layer) z_j, dan lapis keluaran (output keluaran) y_k. Lapis masukan dan lapis tersembunyi dihubungkan dengan bobot vij dan antara lapis tersembunyi dan lapis keluaran dihubungkan oleh bobot w_{jk}.. Arsitektur dapat dilihat pada gambar 2.13



Gambar 2.13 Arsitektur Backpropagation

Jumlah lapis tersembunyi dan jumlah neuron tiap lapisan tidak ada rumus tertentu untuk menghitungnya, namun terdapat aturan yang sering dipakai baik untuk menentukan jumlah lapis tersembunyi maupun jumlah neuron tersembunyi didalamnya. Permasalahan yang membutuhkan dua lapis tersembunyi jarang ditemukan, meskipun neural network yang terdiri

dari dua lapis tersembunyi dapat merepresentasikan segala jenis bentuk fungsi. Tidak terdapat alasan teoritis untuk menggunakan neural network lebih dari dua lapis tersembunyi. Faktanya untuk berbagai macam permasalahan tidak terdapat alasan untuk menggunakan lebih dari satu lapis tersembunyi [**13**]. Kemudian untuk menentukan jumlah neuron pada lapis tersembunyi sangat penting dalam membangun arsitektur jaringan. Menggunakan terlalu sedikit neuron pada lapis tersembunyi kemungkinan akan menghasilkan underfittin, sebaliknya menggunakan terlalu banyak neuron pada lapis tersembunyi mungkin akan menghasilkan overfitting dan apabila data latih sangat banyak akan sangat lama untuk melakukan pelatihan [**13**].

Aturan yang sering dipakai dalam menentukan jumlah neuron pada lapis tersembunyi adalah sebagai berikut:

1. Jumlah neuron pada lapis tersembunyi adalah antara jumlah neuron lapis masukan dan neuron lapis keluaran [**13**].
2. Jumlah neuron pada lapis tersembunyi adalah 2/3 jumlah neuron lapis masukan, ditambah jumlah neuron lapis keluaran [**13**].
3. Jumlah neuron pada lapis tersembunyi harus lebih sedikit dari dua kali lipat jumlah neuron lapis tersembunyi [**13**]
4. Jumlah neuron pada lapis tersembunyi dipengaruhi jumlah data latih [**14**].

   Persamaan yang digunakan untuk aturan ini adalah sebagai berikut :

$$N_h = \frac{Jumlah\ Data\ Latih}{\alpha(N_i+N_0)} \qquad \ldots.(2.30)$$

Keterangan:

N_h= Jumlah Neuron Lapisan Tersembunyi

N_i = Jumlah Neuron Lapisan Masukan

N_o = Jumlah Neuron Lapisan Keluaran

α = skala nomor antara 2 – 10

Untuk melatih Backpropagation, Neural Network diberi pola masukan sebagai pola pelatihan kemudian pola tersebut akan menuju ke unit pada lapis tersembunyi untuk diterusan pada unit yang berada pada lapis keluaran. Keluaran sementara pada lapis tersembunyi akan diteruskan pada lapis keluaran dan lapis keluaran akan memberikan tanggapan yang disebut sebagai keluaran sementara yk. ketika yk ≠ tk dimana tk adalah keluaran yang diharapkan, maka selisih (error) keluaran sementara u'k akan disebarkan mundur (backward) pada lapis tersembunyi dan diteruskan ke unit pada lapis masukan. Oleh karena itu proses tersebut disebut propagasi balik (backpropagation) [**15**].

Pada metode Backpropagation terdapat dua tahapan proses, yaitu :

1. Tahapan pelatihan, dimana pada tahap ini metode backpropagation diberikan sejumlah data pelatihan dan target.
2. Tahap pengujian atau penggunaan, pengujian dilakukan setelah melakukan tahap belajar atau tahap pelatihan.

Kondisi pelatihan berhenti atau stopping condition dapat menggunakan dua cara yang biasa dipakai, yaitu:

1. Membatasi iterasi yang ingin dilakukan.
   Misalnya jaringan akan dilatih sampai iterasi yang ke-500.
2. Membatasi error.
   Misalnya menentukan besar Mean Square Error.

Secara umum langkah dalam pelatihan Neural Network menggunakan Backpropagation adalah sebagai berikut :

1. Menentukan jumlah input (pola masukan), hidden layer, dan output (target pelatihan).
2. Memberi nilai awal secara random bagi seluruh bobot antara input-hidden layer dan hidden-output layer.
3. Melakukan feedforward
   a. Tiap unit input (x_i) menerima sinyal input dan sinyal tersebut dikirimkan pada seluruh unit pada hidden layer.

b. Tiap unit pada hidden layer (Z_in_j) menjumlahkan sinyal-sinyal input terbobot:

$$z_in_j = V_{0j} + \sum_{i=1}^{n} x_i v_{ij} \quad \text{....(2.31)}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$z_j = f(z_{in_j}) \quad \text{....(2.31)}$$

kirimkan sinyal tersebut ke semua unit di output layer

c. Tiap unit pada output layer (Y_in_k) menjumlahkan sinyal-sinya input terbobot:

$$y_in_k = W_{0j} + \sum_{i=1}^{n} z_i w_{jk} \quad \text{....(2.32)}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya:

$$y_k = f(y_{in_k}) \quad \text{....(2.33)}$$

4. Melakukan Backpropagation

a. Tiap unit pada output layer (y_k) menerima target pola yang dengan berhubungan dengan pola input pembelajaran, hitung informasi errornya :

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad \text{....(2.34)}$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}):

$$\Delta w_{jk} = \alpha\, \delta_k\, z_j \quad \text{....(2.35)}$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai w_{0k})

$$\Delta w_{0k} = \alpha\, \delta_k \quad \text{....(2.36)}$$

Kirimkan δ_k ke unit yang berada dilapisan bawahnya

b. Tiap unit pada hidden layer (Z_in_j) menjumlahkan delta inputnya (dari unit yang berada pada lapisan atasnya):

$$\delta_{in_j} = \sum_{k=1}^{m} \delta_k w_{jk} \quad \text{....(2.37)}$$

kalikan nilai ini dengan turunan dari fungsi aktivasinya untu menghitung informasi error:

$$\delta_j = \delta_in_j f'(z_{in_j}) \quad \text{....(2.38)}$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}):

$$\Delta v_{ij} = \alpha\ \delta_j\ X_i \quad \text{....(2.39)}$$

Hitung juga koreksi bias (yang nantinya akan digunakan untuk memperbaiki nilai v_{0j}):

$$\Delta V_{0j} = \alpha\ \delta_j \quad \text{....(2.40)}$$

c. Tiap unit output (y_in_k) memperbaiki bias dan bobotnya:

$$W_{jk}(baru) = W_{jk}(lama) + \Delta W_{jk} \quad \text{....(2.41)}$$

tiap unit tersembunyi (z_in_j) memperbaiki bias dan bobotnya:

$$V_{jk}(baru) = V_{jk}(lama) + \Delta V \quad \text{....(2.42)}$$

5. Tes kondisi berhenti
   a. Apakah batas error minimum telah tercapai.
   b. Apakah batas iterasi maksimum telah tercapai.

Keterangan Persamaan :

z_in : Bobot sinyal unit tersembunyi

z : Keluaran unit tersembunyi

y_in : Bobot sinyal lapisan keluaran

y : Keluaran unit keluaran

δ : Kesalahan

Δw : Koreksi bobot unit keluaran

Δv : Koreksi bobot unit tersembunyi

x_i : Unit (neuron) pada lapisan input

n : Jumlah neuron (unit) pada lapisan input

v : Bobot awal masuk ke hidden

v_0 : Bobot bias yang menuju ke hidden

w : Bobot awal hidden ke output

w_0 : Bobot bias yang menuju ke output

α : Learning rate / rasio pembelajaran