

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Tinjauan Pustaka**

Sebagai tinjauan pustaka berikut ini beberapa contoh penelitian yang sudah dilakukan oleh para peneliti yang dapat digunakan sebagai acuan dan pengetahuan.

Penelitian dengan judul “ *Identifikasi tekstur citra bubuk susu menggunakan alihragam gelombang singkat untuk mendeteksi keaslian produk susu*”, dalam penelitian ini yang berjudul identifikasi citra susububuk menggunakan alihragam gelombang singkat untuk mendeteksi keaslian produk susu . Aplikasi ini dirancang untuk dapat mengidentifikasi produk susu bubuk yang dipandang sebagai tekstur dalam citra digital. Dalam penelitian ini ,citra tekstur butiran bubuk susu diekstrak menggunakan alihragam gelombang singkat (*wavelet*) untuk mendapatkan ciri-ciri khusus dalam bentuk *feature vector*. *Feature vector* ini diterapkan sebagai vector masukan proses pengenalan pada *learning vector quantization* (LVQ) melalui aturan dan proses pembelajaran (Supatman, 2008).

Penelitian dengan judul “*Sistem intelijen prediksi dan penilaian kualitas susu pasteurisasi dengan menggunakan logika fuzzy dan jaringan syaraf tiruan*”, dalam penelitian ini sistem kontrol kualitas dioperasikan dalam dua bagian yaitu dimensi kualitas produk seperti performansi, feature, keandalan, konformasi, durability, estetika, kualitas yang dirasakan oleh konsumen dan menentukan suhu pasteurisasi. Sistem diimplementasikan dengan *fuzzy inference system* (FIS) dan jaringan syaraf tiruan (Winnie Septani &Marimin, 2005).

Penelitian dengan judul “*Identifikasi dan klasifikasi kemurnian susu sapi berdasarkan pemrosesan sinyal video menggunakan metode discrete cosine transform (dct) dan support vector machine (svm)*” dalam penelitian ini penulis bertujuan untuk mengidentifikasi kemurnian pada susu sapi dengan cara melihat dari hasil rekaman video tetesan bulir-bulir susu sapi, dengan cara melihat *frame by frame* agar dapat menentukan jumlah bulir susu sapi per tetesan. Deteksi yang akan

dilakukan menggunakan metode *Discrete Cosine Transform* (DCT) dan klasifikasi dengan *Support Vector Machine* (SVM). *Discrete Cosine Transform* adalah sebuah teknik yang digunakan untuk merubah sinyal menjadi komponen frekuensi dasar sedangkan *Support Vector Machine* (SVM) adalah metode klasifikasi dengan tujuan menentukan hyperplane terbaik yang memisahkan kelas-kelas yang ada (Kinanti Balqis Maharani & Dr. Ir.Bambang Hidayat, 2017).

Penelitian dengan judul ***“Implementasi Metode Segmentasi dan LVQ untuk Identifikasi Citra Daging Sapi Dan Babi”***, penelitian ini akan mengembangkan sistem pengenalan daging sapi dan babi menggunakan penggabungan metode segmentasi *SpatialFuzzyC-Means* (S-FCM), ekstraksi ciri warna HSV (*Hue, Saturation, Value*) ekstraksi ciri tekstur *Gray level co-occurrence matrix* (GLCM) dan klasifikasi *Learning Vector Quantization* (LVQ). Pada tahap awal sistem menjalankan proses segmentasi citra menggunakan metode *spatial fuzzy c-means* untuk mendapatkan citra daging yang terpisah dari *background*-nya. Setelah itu baru dilakukan proses pengenalan ciri menggunakan HSV, pengenalan tekstur dengan GLCM dan terakhir proses klasifikasi menggunakan LVQ. Pengujian pada penelitian ini terdapat dua skenario yang dilakukan yaitu pertama jumlah data latih yang digunakan 30 dan data uji 15. Sedangkan yang kedua jumlah data latih 50 dengan data uji 15 dan dapat mengenali citra daging sapi dan citra daging babi dengan persentase nilai akurasi tertinggi 80 % dengan nilai *learning rate* ( $\alpha$ ) 0,1 dan jumlah data latih 30 dan 30. Nilai minimal *learning rate* ( $M_{\alpha}$ ) yang digunakan adalah 0,01 dan nilai pengurangan  $\alpha$  adalah 0,1 (Jasril & dkk,2017).

Penelitian dengan judul ***“Risk based Estimate Of Effect Of Foodborne Diseases On Public Health”*** Kontaminasi satu koloni *sakazakii* memiliki peluang hidup maksimum sebesar 6,5% untuk dapat berkembang hingga mencapai jumlah yang signifikan 1 juta sel/g produk dalam waktu maksimal 100 jam per suhu 18-37<sup>0</sup>C. Artinya satu sel hidup *sakazakii* mengkontaminasi produk susu formula pada proses produksi. Hanya dalam dua jam produk tersebut sudah sangat berbahaya bagi bayi. Dalam waktu dua jam dalam susu formula sudah mengandung berjuta- juta bakteri (Havelaar & Zweitering,2004).

## **2.2 Landasan Teori**

### **2.2.2.1 Susu Formula**

Susu formula adalah susu cair atau bubuk dengan komposisi tertentu untuk bayi atau anak yang berfungsi sebagai pengganti air susu ibu (ASI). Susu formula biasanya diberikan karena alasan tertentu kondisi ibu tidak bisa memenuhi kebutuhan susu bayi, misalnya karena bekerja, karena air susu ibu (ASI) yang tidak bisa keluar. Susu formula adalah susu yang dibuat dari susu sapi atau susu buatan yang diubah komposisinya hingga dapat dipakai sebagai pengganti (ASI). Alasan dipakai susu sapi sebagai bahan dasar karena banyaknya susu yang dapat dihasilkan oleh peternak (Pudjadi, 2000).

### **2.2.2.2 Citra Digital**

Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data yang bersifat optik. Citra terbagi dua yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinyu seperti gambar pada monitor televisi, foto sinar X, hasil CT scan. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer (Sutoyo, 2009).

Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang dipresentasikan dengan deretan *bit* tertentu. Suatu citra dapat didefinisikan sebagai fungsi  $f(x,y)$ , berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f dititik koordinat (x,y), dinamakan intensitas atau tingkat kebutuhan dari citra pada titik tersebut. Apabila nilai x, y, dan nilai amplitudo f secara keseluruhan berhingga (*finite*), dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital .

Citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan piksel dalam *array* dua dimensi. Citra jenis ini disebut citra bitmap (*bitmap image*) atau citra raster (*raster image*). Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor (*vector graphics*). Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom dan N baris dimana perpotongan antara baris dan kolom piksel. Piksel sama dengan *picture*

*element*, yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu kordinat dan intensitas atau warna. Nilai yang terdapat pada kordinat  $(x,y)$  adalah  $f(x, y)$  yaitu besar intensitas atau warna dari piksel dititik itu. Matrik yang dinyatakan citra digital yaitu dengan matriks berukuran N (baris/tinggi) x M (kolom/lebar) seperti pada Persamaan 2.1.

$$F(x,y) = \begin{bmatrix} f(0.0) & f(0.1) & \dots & f(0.M-1) \\ f(1.0) & f(1.1) & \dots & f(1.M-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1.0) & f(N-1.1) & \dots & f(N-1.M-1) \end{bmatrix} \dots\dots\dots(2.1)$$

Keterangan :

N = jumlah baris  $0 = y = N - 1$ .

M = jumlah kolom  $0 = x = M - 1$ .

L = maksimal warna intensitas  $0 = f(x,y) = L - 1$ . (*gray level/ derajat keabuan*)

Berdasarkan Persamaan 2.1. tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi *intensitas*  $f(x,y)$ , dimana harga  $(x)$  baris dan  $(y)$  kolom merupakan koordinat posisi dan  $f(x,y)$  adalah nilai fungsi pada setiap titik  $(x,y)$  yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel dititik tersebut. Pada proses digitalisasi sampling dan kuantitas diperoleh besar baris M dan kolom N hingga citra membentuk matrik  $M \times N$  dan jumlah keabuan piksel G.

Pengolahan citra digital adalah sebuah disiplin ilmu yang mempelajari hal – hal yang berkaitan dengan perbaikan kualitas gambar ( peningkatan kontras, transformasi warna, restorasi citra ), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan citra ciri (*feature images*) yang optimal untuk

tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. *Input* dari pengolahan citra adalah citra, sedangkan *output* adalah citra hasil pengolahan.

Berdasarkan jenis dari nilai terkuantisasinya, citra digital dapat diklasifikasikan seperti pada Sub Bab 2.2.2.1 sampai dengan 2.2.2.3.

### **2.2.2.1 Citra Biner**

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan citra piksel yaitu hitam dan putih. Citra biner juga disebut dengan citra (*black and white*). Hanya dibutuhkan satu *bit* untuk mewakili setiap piksel citra biner.

### **2.2.2.2 Citra Grayscale**

Citra *grayscale* adalah citra digital yang hanya memiliki satu nilai kanal pada setiap piksel nya. Citra ini hanya terbentuk dari warna hitam, keabuan, dan putih pada tingkatan yang berbeda-beda. Mulai dari warna hitam pada tingkat intensitas terendah hingga warna putih hingga intensitas tertinggi. Citra ini disebut juga citra hitam putih atau citra monokromatik. Citra *grayscale* berikut memiliki kedalaman warna 8 *bit* ( 256 kombinasi warna keabuan).

### **2.2.2.3 Citra Warna**

Citra warna adalah citra digital yang setiap pikselnya mengandung informasi warna. Informasi warna ini biasanya dibentuk dari paling sedikit 3 sampel (saluran warna). Saluran warna yang umum dipakai dalam komputer adalah RGB (*Red Green Blue*), tetapi dalam konteks lain sering juga digunakan saluran warna lain seperti CMYK atau YcbCr.

### **2.2.2.4 Pra-Proses**

*Preprocessing* adalah proses pengolahan data citra data asli sebelum data tersebut di proses berikutnya. Pengolahan citra merupakan suatu system dimana proses dilakukan dengan input berupa citra (*image*) dan *output* – nya juga berupa

perbaikan kualitas citra atau penyajian citra. *Output* citra mempunyai kualitas lebih baik dibandingkan input citra. Pengolahan citra mempunyai dua tujuan utama, yaitu sebagai berikut:

Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi secara jelas atau dengan kata lain manusia dapat melihat informasi yang di harapkan. Dalam hal ini interpretasi terhadap informasi yang ada dilakukan oleh manusia.

Mengekstraksi informasi ciri yang menonjol pada suatu citra, dimana output berupa informasi citra sehingga manusia mendapatkan informasi ciri dari citra secara numerik atau dengan kata lain komputer (mesin) melakukan interpretasi terhadap informasi yang ada pada citra melalui besaran – besaran.

Pra - proses mengambil informasi dari piksel yang ada pada gambar, setiap komponen pada sistem pengenalan karakter dirancang untuk mengurangi jumlah data. Langkah pertama adalah *image preprocessing* yang bertujuan untuk mengubah intensitas pixel gambar agar mudah digunakan pada proses selanjutnya. Beberapa pra-proses yang sering digunakan adalah proses *cropping* dan proses *grayscale* (arah keabuan).

*Cropping* adalah proses pemotongan citra pada koordinat tertentu pada area citra. Proses ini dilakukan untuk mengambil bagian yang dirasa penting atau bagian yang mempunyai paling banyak informasi untuk diolah menggunakan jaringan syaraf tiruan. Selain itu proses ini juga dapat mengubah ukuran citra menjadi lebih kecil, sehingga akan mempercepat proses komputasi. Selain dengan melakukan *cropping*, untuk mempercepat proses komputasi dapat melakukan *grayscale*.

*Grayscale* adalah suatu citra dimana nilai dari setiap pixel merupakan sampel tunggal. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas dua warna saja yaitu "hitam" dan "putih" saja. Citra *grayscale* disimpan dalam format delapan *bit* untuk setiap sampel piksel, yang memungkinkan sebanyak 256 intensitas. Nilai

intensitas paling rendah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih.

Untuk mengubah citra berwarna yang mempunyai nilai matriks masing - masing RGB (*Red, Green, dan Blue*) menjadi citra *grayscale* dengan nilai *s*, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai RGB (*Red, Green, dan Blue*) sehingga dapat dituliskan pada Persamaan 2.2.

$$S = \frac{R+G+B}{3} \dots\dots\dots(2.2)$$

### 2.2.2.3 Histogram Citra

Histogram merupakan representasi grafis untuk distribusi warna dari citra digital atau menggunakan penyebaran nilai-nilai intensitas piksel dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan relatif dari intensitas sebuah citra, kecerahan dan kontras dari sebuah gambar.

Histogram citra adalah suatu grafik yang menyatakan hubungan antara BV (*brightness value*) dengan kemunculan setiap nilai frekuensi gradasi warna. Dari histogram pada suatu citra yang ada dapat menentukan berapa buah objek yang terdapat pada suatu citra tersebut. Jika terdapat semakin banyak kurva yang terdapat pada suatu histogram maka semakin bervariasi pula objek pada citra tersebut. Untuk itu tiap band yang berada pada suatu citra, memiliki histogram dengan kurva yang berbeda – beda.

Histogram citra merupakan salah satu bentuk representasi grafis karakteristik spektral citra yang bersangkutan. Dengan histogram seseorang dapat memahami citra yang dipelajari baik dari kecerahan dan ketajaman. Histogram juga dapat diduga jenis saluran spectral citra yang digunakan. Perubahan atas distribusi nilai pada citra secara langsung akan berakibat pada perubahan tampilan histogram. Sebaliknya, dengan memainkan bentuk histogram nya banyak program pengolah citra secara intensif mampu merubah tampilan citra. Dengan kata lain,

preangkat lunak pengolahan citra kadang-kadang menggunakan histogram sebagai jembatan pengguna data citra (Sutoyo, 2009).

Kuantitasi dalam hal warna histogram mengacu kepada proses mengurangi jumlah bit dengan mengambil warna-warna yang sangat mirip satu sama lain dan menempatkan kedalam *bit* yang sama. Secara *default* jumlah maksimum *bit* yang dapat diperoleh dengan menggunakan fungsi histogram di matlab adalah 256. Agar histogram memberikan gambaran yang akurat tentang kondisi hasil produksi, perlu dilakukan pengolahan data yang akurat terlebih dulu. Pengolahan data pada histogram menjadi sangat penting, terutama dalam menentukan besaran nilai tengah (standar) dan seberapa banyak kelas-kelas data yang akan menggambarkan penyebaran data yang tercipta (Suharsimi, 2009) .

#### **2.2.2.4 Jaringan Syaraf Tiruan**

Jaringan syaraf tiruan (*artificial neural network*) adalah sebuah alat pemedoman data statistik nonlinier, *Neural network* dapat digunakan untuk memodelkan hubungan yang kompleks antara *input* dan *output* untuk menentukan pola – pola pada data (Widodo, 2005).

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

- a. Pemrosesan informasi terjadi pada elemen - elemen yang disebut (*neuron*).
- b. Sinyal dikirimkan di antara neuron-neuron melalui penghubung-penghubung .
- c. Penghubung antara neuron memilikibobot yang akan memperkuat atau memperlemah sinyal.
- d. Untuk mendapatkan *output*, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linear) yang dikenakan pada jumlah *input* yang diterima. Biasanya *output*, ini selanjutnya di dibandingkan dengan suatu batas ambang.



*Neuron* biologi merupakan sistem yang “*fault tolerant*” dalam dua hal. Pertama, manusia dapat mengenali sinyal *input* yang agak berbeda dari yang di terima sebelumnya. Sebagai contoh manusia sering dapat mengenali seseorang yang wajahnya agak berbeda karena sudah lama tidak berjumpa. Kedua, tetap mampu bekerja dengan baik. Jika sebuah *neuron* rusak, maka *neuron* yang lain di latih untuk menggantikan fungsi *neuron* yang rusak (Sing, 2004).

Hal yang ingin dicapai dengan melatih jaringan syaraf tiruan adalah untuk mencapai keseimbangan antara kemampuan memorisasi dan generalisasi. Yang dimaksud dengan memorisasi adalah kemampuan jaringan syaraf tiruan untuk mengambil kembali secara sempurna sebuah pola yang telah dipelajari. Kemampuan generalisasi adalah kemampuan jaringan syaraf tiruan untuk menghasilkan respon yang bisa diterima terhadap pola-pola yang sebelumnya telah di pelajari. Hal ini sangat bermanfaat bila pada suatu saat ke dalam jaringan syaraf tiruan itu di inputkan informasi baru yang belum pernah dipelajari, maka jaringan syaraf tiruan itu masih akan tetap memberikan tanggapan yang baik memberikan keluaran yang mendekati (Puspaningrum, 2006).

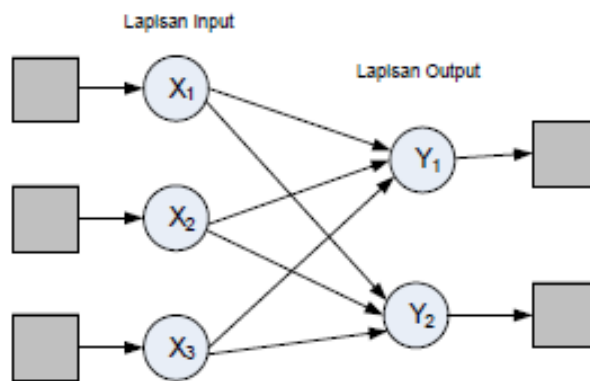
Konsep dasar jaringan syaraf tiruan setiap pola-pola *input* dan *output* yang di berikan ke dalam jaringan syaraf tiruan diproses oleh *neuron*. *Neuron-neuron* tersebut terkumpul di dalam lapisan-lapisan yang disebut *neuron layers*.

Lapisan *Input* adalah unit-unit di dalam lapisan *input* disebut unit-unit *input* . Unit – unit *input* tersebut menerima pola inputan data dari luar yang menggambarkan suatu permasalahan. Lapisan Tersembunyi dimana unit-unit di dalam lapisan tersembunyi disebut unit-unit tersembunyi. Dimana *output* tidak dapat secara langsung diamati. Sedangkan lapisan *Output* unit-unit di dalam lapisan *output* disebut unit – unit *output*. *Output* dari lapisan ini merupakan solusi jaringan syaraf tiruan terhadap suatu permasalahan,

Jaringan syaraf tiruan memiliki beberapa asitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur jaringan syaraf tiruan antara lain (Kusumadewi, 2003)

## 1. Jaringan layar tunggal (single layer network)

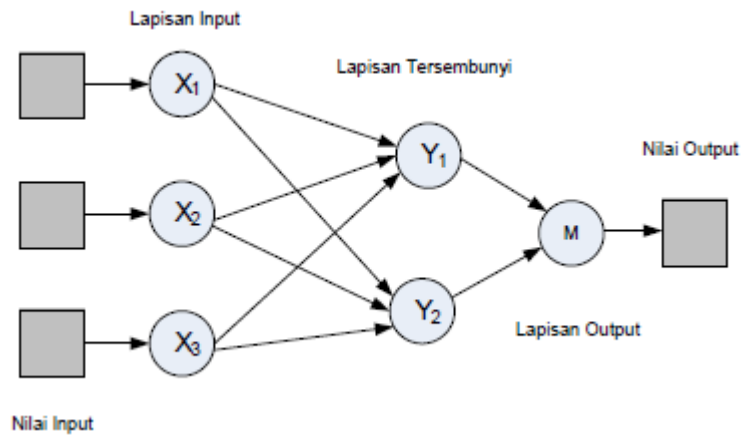
Jaringan dengan lapisan tunggal terdiri dari satu layer *input* dan satu layer *output*. Setiap *neuron* yang terdapat di dalam lapisan *input* selalu terhubung dengan setiap *neural* yang terdapat pada layar *output*. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi. Adapun arsitektur layer tunggal dapat dilihat pada Gambar 2. 1 Arsitektur Layer Tunggal



Gambar 2. 1 Arsitektur Layer Tunggal (Hermawan, 2006).

## 2. Jaringan layar jamak (multi layer network).

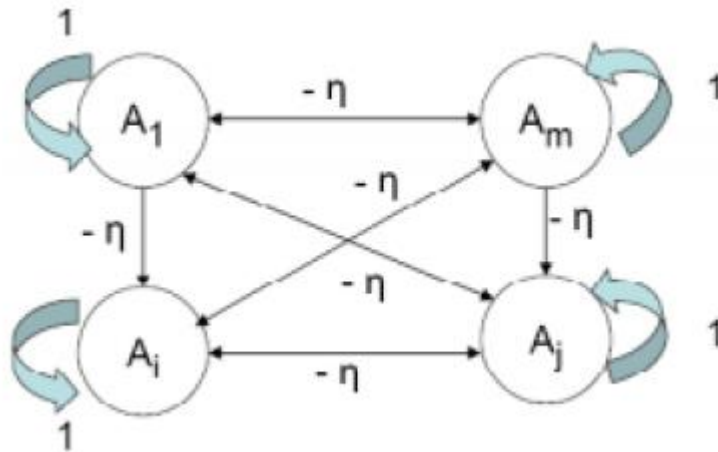
Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yakni *layer input*, *layer output*, *layer* tersembunyi. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang kompleks dibandingkan jaringan dengan lapisan tunggal. Namun proses pelatihan sering membutuhkan waktu yang cenderung lama. Adapun arsitektur layer jamak dapat dilihat pada Gambar 2.2 Arsitektur Layer Jamak



Gambar 2. 2 Arsitektur Layer Jamak (Hermawan, 2006).

### 3. Jaringan dengan lapisan kompetitif.

Pada jaringan ini sekumpulan *neuron* bersaing untuk mendapatkan hak menjadi aktif. Adapun arsitektur layer kompetitif dapat dilihat pada Gambar 2 . 3 Arsitektur Layer Kompetitif



Gambar 2. 3 Arsitektur Layer Kompetitif (Hermawan, 2006).

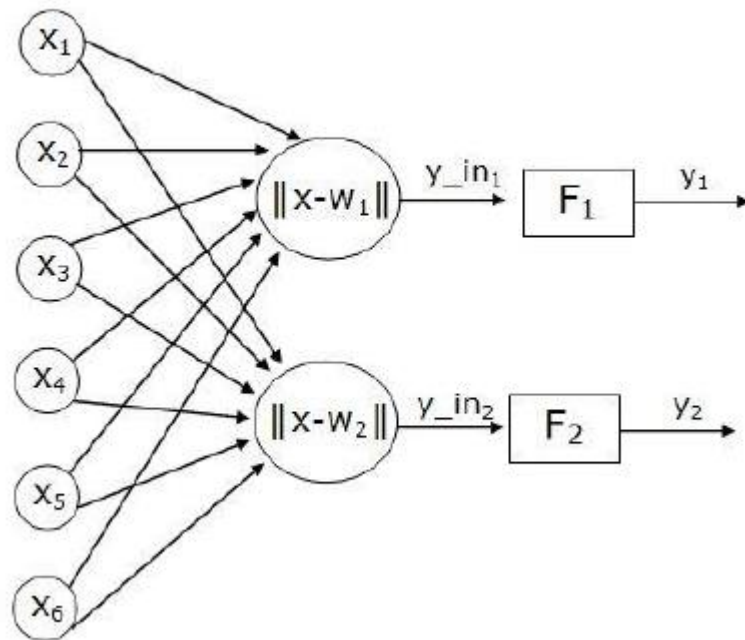
### **2.2.2.5 Metode Learning Vector Quantization**

*Learning Vector Quantization* (LVQ) merupakan suatu metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor *input*. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor-vektor *input*. Jika dua vektor *input* mendekati

sama, maka lapisan kompetitif akan meletakkan kedua vektor *input* tersebut ke dalam kelas yang sama. Tujuan dari algoritma ini adalah untuk mendekati distribusi kelas vektor untuk meminimalkan kesalahan dalam pengklasifikasian (Kusumadewi, 2003).

*Learning Vector Quantization* (LVQ) adalah metode untuk melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas yang di hasilkan berdasarkan jarak vektor tersebut. Jika ada dua vektor memiliki jarak yang cukup dekat atau mendekati sama maka kedua vektor tersebut dikelompokkan ke dalam kelas yang sama.

Adapun arsitektur algoritma *Learning Vector Quantization* dapat dilihat pada Gambar 2.4 Arsitektur *Learning Vector Quantization*.



Gambar 2. 4 Arsitektur *Learning Vector Quantization* (Kusumadewi, 2003)

Keterangan :

X = Vektor maksimum ( $X_1, X_2, \dots, X_n$ )

F = Lapisan Kompetitif

$y_{in}$  = Masukan lapisan kompetitif

y = Keluaran

W = Vector bobot unit keluaran

$\|X-W\|$  = Selisih nilai jarak *Euclidean* antara vector *input*

Algoritma diusulkan oleh Kohonen pada tahun 1986 sebagai perbaikan dari *Vector Quantization*. Model pembelajaran LVQ dilatih secara signifikan agar lebih cepat dibandingkan algoritma lain seperti *Back Propagation Neural Network*. Hal ini dapat meringkas atau mengurangi dataset besar untuk sejumlah kecil vektor. Berikut ini adalah algoritma dari *Learning Vector Quantization* (LVQ).

1. Tetapkan bobot ( $W$ ) dan maksimum iterasi *epoch*, *error* yang diharapkan ( $\epsilon$ ), *learning rate* ( $\alpha$ ).
2. Masukkan input yang terdiri dari data input  $x(m,n)$  dimana  $m$ = jumlah input dan  $n$ =jumlah data masukkan target  $T(1,n)$ .
3. Tetapkan kondisi awal : ( $epoch < \text{maksimum iterasi}$ ), atau ( $\alpha > \epsilon$ )*error*
  - $epoch = epoch + 1$
  - kerjakan untuk  $i=1$  sampai  $n$  tentukan  $j$  sedemikian rupa sehingga  $\|x-w_j\|$  adalah minimum, Perbaiki  $W_j$  dengan ketentuan
  - jika  $T=j$  maka  $w_j(\text{baru}) = w_j(\text{lama}) + \alpha(x - w_j(\text{lama}))$
  - jika  $T$  tidak sama dengan  $j$  maka  $W_j(\text{baru}) = w_j(\text{baru}) - \alpha(x - w_j(\text{lama}))$
  - Kurangi nilai  $\alpha(0.1 \times \alpha)$  pada iterasi berikutnya
- Diperoleh  $W$  baru sebagai bobot akhir.