BAB III

PERANCANGAN

3.1 Deskripsi Komponen yang Digunakan

Dalam penerapan jaringan syaraf tiruan untuk mendeteksi bentuk pola gambar pada permukaan batuan ini, diperlukan perangkat atau peralatan yang mendukung dalam desain dan pemrogramannya. Adapun perangkat yang digunakan adalah :

3.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam perancangan untuk mendeteksi bentuk pola gambar pada permukaan batuan ini berupa komputer laptop sebagai alat pemrosesan dan pemprograman jaringan syaraftiruan. Penulis menggunakan komputer dengan *motherboard Intel GMA* 4500M *(Aser Aspire 4732Z)*, *proccessor intel pentium* T4400 (2.2 GHz, 800 MHz FSB), *harddisk* 160 GB, *Virtual Graphics Accelereation* (VGA) 1 GB *memory*.

3.1.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam penerapan jaringan syaraf tiruan (JST) ini yaitu untuk mendeteksi bentuk pola pada permukaan batuan adalah aplikasi matlab 7.1, matlab disini adalah sistem interaktif yang menggunakan *element* data dasarnya adalah *array* yang tidak membutuhkan dimensi. Hal ini mempermudah untuk menyelesaikan masalah komputasi terutama menyangkut matriks dan *vector*. Nama matlab merupakan singkatan dari *matrix laboratory*. Pada awalnya matlab dibuat untuk mempermudah pengembangan perangkat lunak berbasis matriks oleh proyek *LINPACK* dan *EISPACK*.

Matlab memiliki beberapa keunggulan dan kemudahan dalam desain dan analisis sistem kendali dan sistem *linear*. Fungsi-fungsi pendukung terangkum dalam *Control System Toolbox* dan *Signal Proccesing Toolbox*. Kebanyakan operasi dalam sistem kendali dan sistem linear berhubungan dengan operasi

matriks, aritmatika kompleks, menghitung akar, model konversi, dan penggambaran fungsi-fungsi yang kompleks. Seperti telah kita ketahui sebelumnya, matlab telah di desain untuk memudahkan pengerjaan operasi-operasi tersebut diatas dan analisa fungsi. Pelatihan ini lebih ditekankan pada pemrograman dan menggunakan fungsi-fungsi dasar dalam bahasa matlab sehinga teori mengenai ilmu sistem itu sendiri hanya akan disinggung sedikit pada topik-topik yang relevan.

3.2 Proses Kerja *Learning Vector Quantization* (LVQ)

Pada metode *Learning Vector Quantization* (LVQ) terdiri atas dua lapisan tersembunyi. Lapisan tersembunyi pertama merupakan lapisan kompetitif yang terdiri S1 *neuron*. Bobot-bobot input akan diset sama dengan nilai tengah dari vector-vektor *input*. Sedangkan lapisan tersembunyi kedua teraktivasi dengan fungsi *linear (Purelin)* yang terdiri dari S2 *neuron* sesuai dengan jumlah kelas yang diharapkan, dengan catatan S1 > S2. Misalkan *neuron* ke satu, dua dan ke tiga pada lapisan kompetitif termasuk dalam kelas satu, maka ketiga *neuron* pertama pada lapisan tersembunyi kedua, dan akan memiliki bobot sama dengan nol menuju ke *neuron* yang lain (selain *neuron* pertama) pada lapisan tersembunyi kedua. *Learning Vector Quantization* (LVQ) sendiri dapat dibangun dengan menggunakan fungsi seperti berikut : *Newlvq*

3.2.1 Langkah-Langkah Membangun *Learning Vector Quantization* (LVQ)

Berikut ini adalah pelatihan jaringan LVQ jika diberikan *(n)* buah data dengan *(m)* buah *variable* input. Dan data tersebut dibagi dalam sebuah K kelas. Proses Pengambilan Citra.

1. Tetapkan:

   a) Bobot awal *variable input* ke-j menuju ke kelas ke-i yaitu Wij, dengan i = 1,2, …….., K dan J = 1,2, …., m.

   b) Maksimum *epoch* = *Maxepoch*.

   c) Parameter *learning rate* = α.

d) Pengurungan *learning rate* = *Dec* α.

e) Minimal *learning rate* yang diperbolehkan = Min α.

2. Masukkan:

Data input Xij dengan i = 1,2, …,n dan j = 1,2, .., m.

Target berupa kelas, yaitu Tk dengan k = 1, 2, ..,n.

3. Tetapkan kondisi awal *epoch*, yaitu *epoch*=0

4. Kerjakan jika (*epoch*<*Maxepoch*) dan (α > Min α)

epoch =*epoch*+1

b) Kerjakan untuk i = 1 sampai n

- Tentukan J sedemikian sehingga | Xi - Wj| minimum dengan j= 1, 2, , K.

- Perbaiki Wj dengan ketentuan:

- Jika T = Cj makaWj + α (Xj –Wj)
- Jika T ≠ Cj makaWj + α (Xj –Wj)

c) Setelah nilai α (pengurangan nilai α bias dilakukan dengan α = α- Dec α atau dengan cara α = α*Dec α) .

Setelah dilakukan pelatihan akan diperoleh bobot-bobot akhir. Bobot-bobot inilah yang nantinya akan digunakan untuk melakukan pengujian atau simulasi dengan data yang berbeda (tidak sama dengan data yang digunakan untuk pelatihan).

Adapun *algoritma* Pengujian LVQ sebagai berikut :

1. Masukkan data yang akan diuji, misalnya Xij dengan i = 1, 2, …,np dan j = 1, 2, …, m.

2. Kerjakan untuk i = 1 sampai np

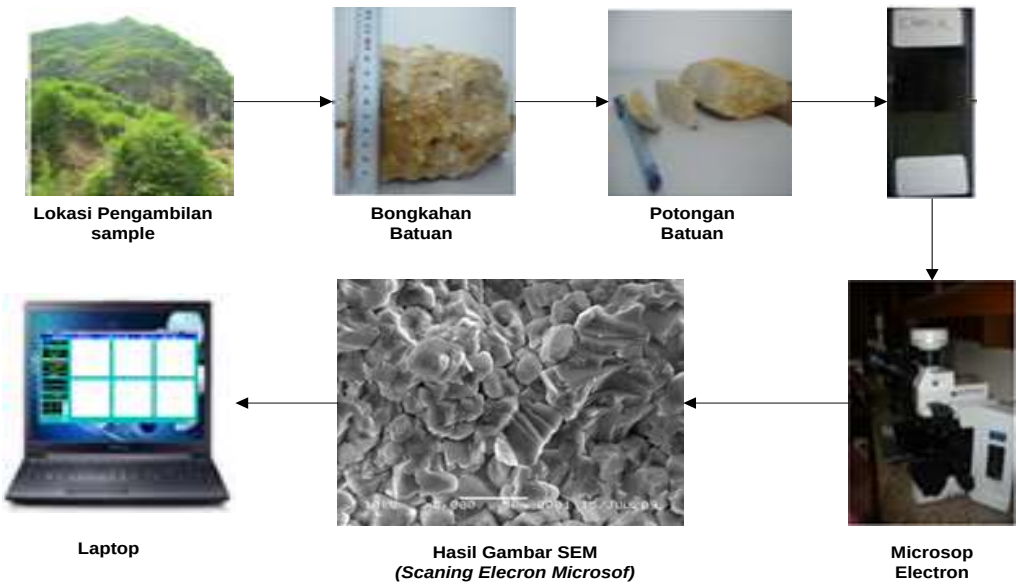
a) Tentukan J sedemikian sehingga | Xi - Wj| minimum dengan j= 1, 2, , K.

b) J adalah kelas untuk Xi.

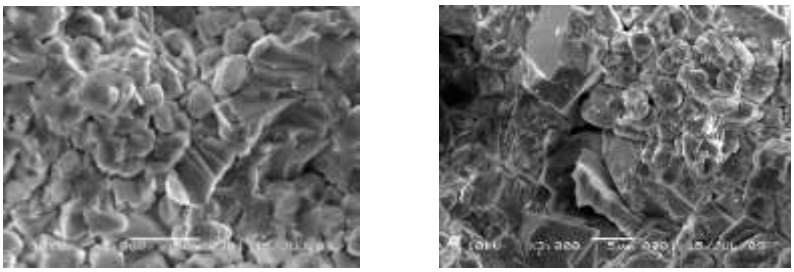
3.3 Proses Pengambilan Citra

Proses pengambilan citra dilakukan turun langsung ke lapangan didaerah pegunungan padalarang yang diambil sebuah sampel bungkahan batuan yaitu batuan sedimen berjenis karbonat atau gamping dan lalu proses selanjutnya

dengan tahapan seperti dibawah seperti melalui proses SEM *(Scanning Electron Microscope),* disini proses yang paling penting dikarenakan proses ini proses yang akan menghasilkan gambar yang sebagai media utama untuk penelitian selanjutnya. Dengan menghasilkan gambar seperti (*sem500x, sem1000x, sem3000x, sem5000x).* Gambar yang diambil disimpan dalam format **.Png*. atau **.Bmp.* Pengambilan citra digambarkan seperti gambar berikut ini,



Gambar 3.1 Proses pengambilan citra

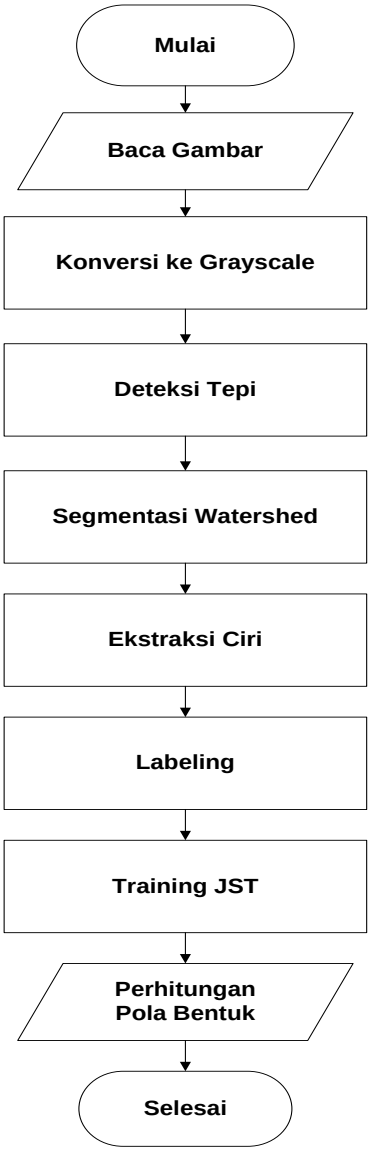


(a) (b)

Gambar 3.2 Struktur mineral gamping dengan perbesaran (a) 5000x (b) 3000x yang akan dilatih

3.4 Perancangan Sistem (*flowchart* diagram sistem)

Secara garis besar sistem program yang akan dirancang untuk mendeteksi dan menghitung pola bentuk wajik dan bukan wajik dapat dilihat pada *flowchart* 3.3 dibawah ini,



Gambar 3.3 *Flowchart* perancangan sistem

3.4.1 Baca Citra

Citra digital diperoleh dari foto *thin slice* dalam ukuran *micropis*, batuan *gamping* berwarna dengan penyimpanan format **Bmp* dan **Png*. Citra yang dipilih adalah citra 24 bit sehingga dikenali sebagai citra RGB.

Sebagai contoh proses tersebut seperti pada langkah dibawah ini:

```
[FileName,PathName] = uigetfile('*.png','Select the png-file');
```

Atau

```
[FileName,PathName] = uigetfile('*.Bmp','Select the Bmp-file');
```

3.4.2 *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.

Sebagai contoh proses tersebut seperti pada langkah dibawah ini:

```
gambar =imread(FileName);
gambar=rgb2gray(gambar)
```

3.4.3 Deteksi Tepi

Deteksi tepi digunakan sebagai proses awal sebelum dilakukan segmentasi dengan tujuan saat dilakukan pemisahan objek yang satu dengan yang lain maka tepi setiap objek telah diketahui sebelumnya.

Sebagai contoh proses tersebut seperti pada langkah dibawah ini:

```
gambar1 = medfilt2(gambar);
handles.sobel = edge(gambar1,'sobel');
gambar=fspecial('sobel');
```

3.4.4 Segmentasi *Watershed*

Proses selanjutnya yaitu proses segmentasi *watershed* yang dilakukan untuk memisahkan latar depan dan latar belakang, sehingga diperoleh citra hitam-putih. Proses ini dilakukan dengan memberikan nilai ambang sehingga piksel dengan nilai keabuan di atas nilai ambang akan menjadi satu (putih) sedangkan yang nilainya di bawah nilai ambang akan menjadi nol (hitam).

Jenis batuan gamping *(Karbonat)* memiliki bentuk umum yaitu bentuk wajik dan bukan wajik dengan tengah yang mempunyai sudut tertentu yang telah tertera. Efek pencahayaan menyebabkan beberapa bentuk batuan karbonat terlihat terang pada bagian tengah. Pada beberapa batuan, bagian tengahnya yang terang terlihat sangat luas dan melingkupi sebagian besar batuan, bahkan terlihat pecah. Hal ini dapat menimbulkan kesulitan dalam pengolahan. Oleh karena itu perlu dilakukan rekonstruksi citra berdasarkan proses operasi *morfologi* sehingga objek *(karbonat)* menjadi bentuk yang diinginkan, yaitu betuk sudut dengan bagian terang yang tidak terlalu luas, atau bentuk sudut yang benar-benar penuh. Proses selanjutnya yaitu Segmentasi *watershed* sebagai bagian dari matematika *morfologi* yaitu pemisahan objek yang satu dengan objek yang lain dalam suatu gambar dengan member jarak pada masing masing objek. Tujuan dari *segmentasi* ini yaitu untuk memisahkan dan member jarak apabila terdapat pada batuan yang saling bertumbukan.

Sebagai contoh proses tersebut seperti pada langkah dibawah ini:

```
g=handles;
loadinput2
handles.tresh = im2bw(gambar1,graythresh(gambar1));
handles.tresh = imcomplement(handles.tresh);
nilai3=get(handles.edit1,'string');
nilai3=str2num(nilai3);
nilai=get(handles.edit2,'String');
nilai= str2num(nilai);
se =strel('square',nilai);
handles.tresh = imerode(handles.tresh,se);
handles.tresh = bwareaopen(handles.tresh,nilai3);
nilai2 = get(handles.edit3,'String');
nilai2 = str2num(nilai2);
se2 =strel('square',nilai2);
handles.dilation=imdilate(handles.tresh, se2);
handles.fgm = imfill(handles.dilation,'holes');
handles.fgm = imregionalmax(handles.dilation);
handles.fgm = imfill(handles.fgm,'holes');
handles.D = bwdist(handles.tresh,'euclidean');
handles.DL = watershed(handles.D);
handles.bgm = handles.DL == 0;
gambar2 = imimposemin(handles.DL, handles.bgm | handles.fgm);
gambar2 = imcomplement(gambar2);
handles.L = watershed(gambar2);
handles.Lrgb = label2rgb(handles.L, 'jet', 'w', 'shuffle'); %garis
label warnaputihSimbolkan "W"
guidata(hObject,handles);
saveinput3gambar2
axes(g.axes3)
imshow(gambar2),title('PROSES WATERSHED');
```

3.4.5 Ekstrasi Ciri

Kemudian dilakukan proses ekstrasi ciri objek berdasarkan bentuk, dengan menggunakan metode ini dapat dibedakan bentuk pola wajik dengan bukan wajik, proses pencarian bentuk pada permukaan batuan yang berbentuk wajik dan bukan wajik ini dilakukan satu persatu berdasarkan label pada suatu citra, jika diketahui terdapat pola wajik pada suatu citra batuan karbonat maka bentuk tersebut disimpan dalam suatu *variable*, dan bentuk lain dari *variable* yang memiliki pola bentuk bukan wajik dianggap sebagai batuan yang berbentuk bukan wajik dan disimpan pada *variable* yang berbeda, kemudian setelah proses pencarian selesai maka seluruh hasil pencarian batuannyang berbentuk wajik dan bukan wajik tersebut ditampilkan. Proses pengolahan citra digital berakhir dengan tampilan deskripsi atas hasil pengolahan dalam bentuk tekstual dan tampilan pengolahan citra batuan karbonat.

Berikut adalah sintak yang digunakan :

```
g=handles;
loadinput3
I = imread('sampel1.png');%citra asli
imshow(I)%tampilkan citra asli
%Proses BW(deteksi tepi)
BW = im2bw(I);
imshow(BW)
%pengenalan input BW
dim = size(BW)
data = round(dim(2)/2)-5;
data1 = round(dim(2)/1)-15;
data2 = round(dim(2)/2)-45;
data3 = round(dim(2)/1)-90;
data4 = round(dim(2)/2)-256;
%data input
wajik = min(find(BW(:,data)))
bukanwajik1= min(find(BW(:,data1)))
wajik2= min(find(BW(:,data2)))
bukanwajik3= min(find(BW(:,data3)))
wajik4= min(find(BW(:,data4)))
%membaca data input
boundary = bwtraceboundary(BW,[wajik, data],'N');
boundary = bwtraceboundary(BW,[bukanwajik1, data1],'N');
boundary = bwtraceboundary(BW,[wajik2, data2],'N');
boundary = bwtraceboundary(BW,[bukanwajik3, data3],'N');
boundary = bwtraceboundary(BW,[wajik4, data4],'N');
imshow(I)
hold on;
a= imfill(BW,'holes');
```

```
boundaries = bwboundaries(a);
%penyusuran pola
for k=1:2256
   b = boundaries{k};
%menentukan ketebalan ciri/ekstraksi ciri
plot(b(:,2),b(:,1),'b','LineWidth',1.5);
plot(boundary(:,2),boundary(:,1),'r','LineWidth',1.5);
end saveinput4 gambar3
axes(g.axes4)
imshow(gambar3),title('EKSTRAKSI CIRI');
```

3.5 Hitung Jumlah Batuan yang Memiliki Bentuk Wajik dan Bukan Wajik

A. Perhitungan Secara Sistem atau Otomatis

Perhitungan secara sistem maksudnya perhitungan yang mengambil datanya yang merujuk langsung kepada perhitungan pada sebuah sistem atau program yang sudah dibuat.

B. Perhitungan Secara Manual

Perhitungan secara manual yang maksudnya perhitungan dilakukan oleh penulisan atau yang mengamati program lalu perhitungan tersebut dilakukan perbandingan dengan perhitungan secara sistem tersebut lalu dihasilkan nilai selisihnya, setelah itu akan menghasilkan perhitungan yang akurat atau tidak. Pada proses perhitungan jumlah batuan memiliki bentuk wajik dan bukan wajik masing - masing objek pada citra batuan tersebut menggunakan metode *labeling* komponen *pixel* dimana setiap objek dalam masing-masing citra di beri label dan pengecekan terhadap objek pada citra sehingga jumlah objek dapat dihitung pada masing-masing citra batuan berbentuk Wajik dan Bukan Wajik.

3.5.1 Efektifitas Sistem

Akurasi sistem digunakan untuk menunjukan tingkat keakuratan kinerja dari sistem.

$$\boldsymbol{Akurasi = \frac{TP+TN}{TP+TN+FN+FP} x\ 100\%} \qquad \textbf{(3.1)}$$

Dengan :

Keterangan persamaan untuk pengujian I :

- **TP (***True Positive*) adalah menunjuk pada banyaknya pola bentuk wajik yang dikelompokkan dengan benar sebagai positif .
- **TN** (*True Negative*) adalah menunjuk pada banyaknya pola bentuk bukan wajik yang dikelompokkan dengan benar sebagai negatif .
- **FP** (*False Positif*) adalah menunjuk pada banyaknya pola bentuk wajik yang dikelompokkan sebagai positif (bentuk batuan berbentuk wajik).
- **FN** (*False Negatif*) adalah menunjuk pada banyaknya batuan karbonat yang berbentuk bukan wajik yang dikelompokkan sebagai negatif (bentuk batuan berbentuk bukan wajik).

Untuk memperoleh indeks prosentase yang menunjukkan kemampuan sistem ini dalam mengklasifikasi tingkat keakuratan bentuk maka harus dipertimbangkan seberapa besar *sensitivitas* dan *spesifisitas.*

a. ***Sensitivita****s* adalah indeks prosentase probabilitas hasil uji positif terhadap klasifikasi tingkat keakuratan pola bukan wajik.

$$\boldsymbol{Sensitivitas = \frac{TP}{TP+FN} x\ 100\%} \qquad \textbf{(3.2)}$$

b. ***Spesifisitas*** adalah indeks prosentase probabilitas uji dalam mengklasifikasi pola yang sempurna berbentuk wajik.

$$\boldsymbol{Spesifikasi = \frac{TN}{FP+FN} x\ 100\%} \qquad \textbf{(3.3)}$$

Dengan **TN** (*True Negatif*) adalah menunjuk pada banyaknya pola wajik yang sempurna dan menunjukkan hasil yang negatif.