BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Masalah

Pengembangan penelitian mengenai *Content Based Image Retrieval* sudah pernah dilakukan sebelumnya. Sistem yang dikembangkan sebelumnya lebih banyak berfokus pada fitur warna juga tanpa menggunakan metode pengelompokan khusus.

Masalah muncul karena dengan menggunakan fitur warna, seringkali dipengaruhi oleh seberapa intensitas cahaya yang dimiliki suatu citra. Selain itu, tanpa metode pengelompokan gambar, sistem CBIR belum dapat menghasilkan hasil yang baik.

Dalam penelitian ini, metode yang diajukan untuk sistem CBIR adalah metode ekstraksi ciri statistik. Metode ini dipilih dengan pertimbangan bahwa metode ini cukup mudah diimplementasikan dan dapat menghasilkan hasil ekstraksi yang cukup baik [15].

Selain itu, dengan harapan untuk dapat meningkatkan akurasi sistem, maka akan diterapkan metode klasifikasi dengan algoritma *Levenberg-Marquardt*. Alasan penggunaan metode ini adalah karena metode ini merupakan pengembangan metode *backpropagation* yang mampu melakukan klasifikasi citra dengan baik [16].

3.2 Analisis Data Masukan

Data yang digunakan sebagai data masukan dalam sistem *Content Based Image Retrieval* (CBIR) ini adalah gambar dengan format .bmp, .jpg, .jpeg, dan .png. Resolusi gambar yang dimasukkan tidak dibatasi karena sistem akan mengkonversi resolusi gambar sebelum diolah. Resolusi citra yang digunakan masih dapat merepresentasikan citra asli yakni 150 x 150 piksel [3]. Gambar yang akan diolah merupakan citra 8 bit dengan skala keabuan 0-255 (256 skala). Citra 8 bit dengan 256 skala keabuan merupakan citra standar yang dapat digunakan untuk pengolahan citra [4]. Untuk penelitian ini

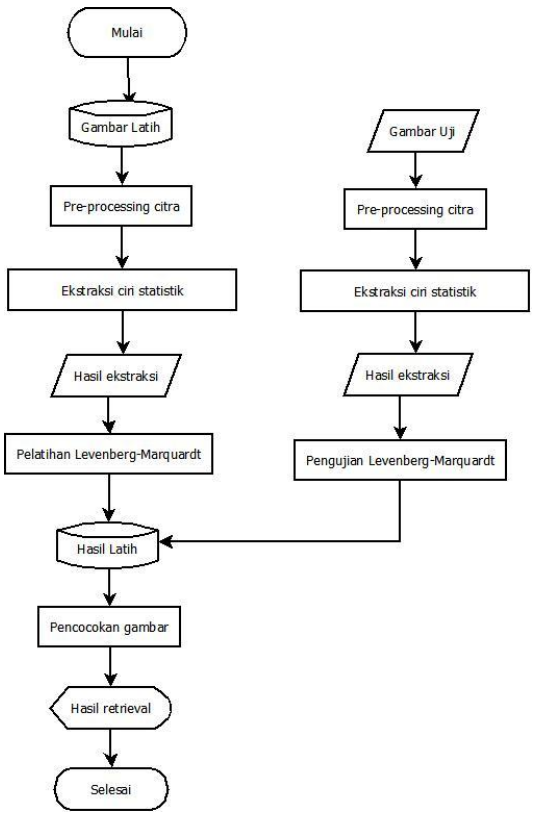
gambar masukan yang akan digunakan 120 data yang berasal dari 8 kategori klasifikasi yang berbeda. Setiap kategori menggunakan 15 data. Adapun kategori yang digunakan yakni bunga matahari, bunga mawar, harimau, daun, bis, burung, anjing, dan bumi. Berikut merupakan kategori gambar yang akan digunakan beserta dengan nilai target *output*nya :

Tabel 3.1 Target Klasifikasi

Nilai Target (Y1,Y2,Y3)	**Klasifikasi**
000	Harimau
001	Bunga mawar
010	Daun
011	Bunga matahari
100	Bis
101	Burung
110	Anjing
111	Bumi

3.3 Analisis Sistem

Sistem *Content Based Image Retrieval* (CBIR) dengan algoritma *Levenberg-Marquardt* secara umum dapat dilihat dari skema berikut :



Gambar 3.1 *Flowchart* Sistem CBIR

Berikut penjelasan umum sistem :

1) Gambar masukan diambil dari 120 data gambar yang sudah disiapkan untuk penelitian. Gambar yang masuk sistem akan melalui tahap *pre-processing* yakni proses *resize* dan *grayscale*.
2) Gambar yang telah melalui tahap *pre-processing* tersebut kemudian digunakan untuk proses ekstraksi ciri statistik yang akan menghasilkan nilai *mean, variance, skewness, kurtosis,* dan *entrophy.*
3) Nilai – nilai tersebut akan digunakan sebagai variabel dalam *input layer* pada jaringan *Levenberg-Marquardt.*
4) Dilakukan proses pelatihan *Levenberg-Marquardt* yang akan menghasilkan bobot-bobot baru untuk proses pengujian dan hasil klasifikasi citra latih yang akan digunakan untuk pencocokan gambar. Hasil klasifikasi ini akan disimpan dalam *database.*
5) Dengan menggunakan beberapa citra uji, dilakukan proses pengujian *Levenberg-Marquardt*. Pengujian ini akan menghasilkan klasifikasi citra yang akan dicocokkan dengan data hasil latih dalam *database* untuk mendapatkan hasil *retrieval*.
6) Hasil retrieval yang didapat akan ditampilkan secara urut dengan melihat nilai *error* yang paling kecil.

3.4 Analisis Metode

Dalam penelitian ini metode yang digunakan adalah metode pengolahan citra dan metode kecerdasan buatan yang meliputi tahap *pre-processing* ekstraksi ciri, dan algoritma *Levenberg-Marquardt*.

3.4.1 *Pre-processing*

Pre-processing yang akan dilakukan pada penelitian ini adalah proses *resize*gambar masukan menjadi ukuran 150 x 150 piksel. Hal ini ditujukan untuk mempercepat proses komputasi.

Berikut merupakan contoh dengan menggunakan citra bumi :



Gambar 3.2 Citra Bumi (1)

Citra masukan di atas memiliki ukuran asli 750 x 750 piksel, yang kemudian di*resize* menjadi citra 150 x 150 piksel. Tahap selanjutnya adalah mengubah citra berwarna menjadi citra *grayscale*. Adapun persamaan yang digunakan untuk mengubah citra berwarna menjadi grayscale yang digunakan adalah :

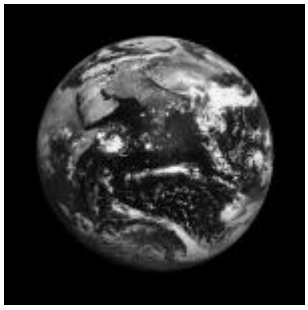
$$\boldsymbol{Grayscale} = \frac{\boldsymbol{R+G+B}}{\boldsymbol{3}} \qquad (3.1)$$

dengan :

R : nilai *Red* piksel

G : nilai *Green* piksel

B : nilai *Blue* piksel

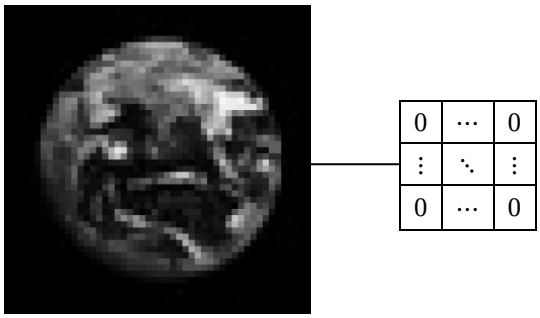


Gambar 3.3 Citra Bumi *Grayscale* 150 x 150 piksel

3.4.2 Ekstraksi Ciri

Metode ekstraksi ciri yang digunakan dalam penelitian ini adalah metode ekstraksi ciri statistik orde 1. Metode ekstraksi ciri statistik orde 1 dilakukan untuk mendapatkan nilai *mean, variance, skewness, kurtosis*, dan *entrophy*.

Dengan citra yang sudah diolah menjadi citra *grayscale* berukuran 150 x 150 piksel, maka kita lakukan proses ekstraksi ciri. Proses ekstraksi ciri dimulai dengan mengambil nilai keabuan dari setiap piksel citra.



Gambar 3.4 Nilai Keabuan Citra Bumi

Selanjutnya, hitung total dan probabilitas kemunculan piksel dengan skala keabuan yang sama. Probabilitas atau nilai histogram dihitung dengan persamaan :

$$\boldsymbol{p(f_n) = \frac{n_i}{n}} \qquad (3.2)$$

dengan

n_i : jumlah piksel yang memiliki skala keabuan i

n : jumlah seluruh piksel citra

Tabel yang menunjukan skala keabuan beserta jumlah piksel dan probabilitas atau nilai histogramnya dapat dilihat pada lampiran E.

Setelah itu dilakukan perhitungan ekstraksi ciri statistik orde 1 untuk semua skala keabuan.

1) *Mean*

Mean dihitung dengan persamaan 2.1 yakni :

$$\boldsymbol{\mu = \sum_{n} f_n p(f_n)}$$

dengan :

fn : skala keabuan

P(fn) : probabilitas skala keabuan (lampiran E)

maka,

$$\boldsymbol{\mu = (0 \times 0,5) + (1 \times 0,006) + \cdots + (255 \times 0) = 39,823}$$

2) *Variance*

Variance dihitung dengan persamaan 2.2 yakni :

$$\boldsymbol{\sigma^2 = \sum_n (f_n - \mu)^2 p(f_n)}$$

dengan :

$\boldsymbol{\mu}$: nilai *mean*

maka,

$$\boldsymbol{\sigma^2 = ((0 - 39,823)^2 \times 0,5) + ((1 - 39,823)^2 \times 0,006) + \cdots +}$$
$$\boldsymbol{((255 - 39,823)^2 \times 0) = 3325,973}$$

3) *Skewness*

Skewness dihitung dengan persamaan 2.3 yakni :

$$\boldsymbol{\alpha_3 = \frac{1}{\sigma^3} \sum_n (f_n - \mu)^3 p(f_n)}$$

dengan :

$\boldsymbol{\sigma^3}$: standar deviasi skala keabuan

maka,

$$\boldsymbol{\alpha_3 = \frac{1}{191813,052}\Big(((0 - 39,823)^3 \times 0,5) + ((1 - 39,823)^3 \times 0,006)}$$
$$\boldsymbol{+ \cdots + ((255 - 39,823)^3 \times 0)\Big) = 1,532}$$

4) *Kurtosis*

Kurtosis dihitung dengan persamaan 2.4 yakni :

$$\boldsymbol{\alpha_4 = \frac{1}{\sigma^4} \sum_n (f_n - \mu)^4 p(f_n) - 3}$$

$$\boldsymbol{\alpha_4 = \frac{1}{11062098,972} \times \Big(((0 - 39,823)^4 \times 0,5 - 3) + ((0 - 39,823)^4 \times}$$
$$\boldsymbol{0,006 - 3) + \cdots + ((0 - 39,823)^4 \times 0 - 3)\Big)} = 4,513$$

5) *Entrophy*

Entrophy dihitung dengan persamaan 2.5 yakni :

$$\boldsymbol{H = -\sum_n p(f_n) \cdot {}^2\log p(f_n)}$$

$$\boldsymbol{H = -((0,5 \times {}^2\log 0,5) + (0,006 \times {}^2\log 0,015) +}$$
$$\boldsymbol{\ldots + (0 \times {}^2\log 0)) = 4,776}$$

Nilai *mean*, *variance*, *skewness*, *kurtosis*, dan *entrophy* inilah yang akan menjadi *input layer* pada algoritma *Levenberg-Marquardt*. Untuk perhitungan ekstraksi ciri statistik semua data dapat dilihat pada lampiran F.

3.4.3 Algoritma *Levenberg-Marquardt*

Algoritma *Levenberg-Marquardt* yang digunakan terdiri dari proses pelatihan dan proses pengujian. Berikut merupakan beberapa parameter yang harus ditentukan sebelum proses pelatihan :

1) *Maximum epoch*

   Epoch merupakan perulangan atau iterasi dari proses yang dilakukan untuk mencapai target yang ditentukan. *Maximum epoch* adalah jumlah maksimal yang akan dilakukan selama proses pelatihan. Iterasi akan dihentikan bila nilai *epoch* sudah melebihi *maximum epoch*

2) *Maximum error*

   Maximum error merupakan nilai toleransi *error* yang juga akan digunakan untuk kondisi penghentian iterasi. Iterasi akan dihentikan bila nilai *error* lebih kecil atau sama dengan *maximum error.*

3) *Learning rate* / parameter *Marquardt*

   Learning rate atau disebut juga parameter *Marquardt* (μ) merupakan laju pembelajaran yang berupa perkalian negatif dari gradien untuk menentukan perubahan nilai bobot dan bias. Semakin besarnya *learning rate* akan berpengaruh pada semakin besarnya langkah pembelajaran. Penentuan nilai *learning rate* perlu diperhatikan karena jika nilai *learning rate* diatur terlalu besar, maka algoritma akan menjadi tidak stabil. Namun jika nilai *learning rate* diatur terlalu kecil, maka algoritma akan mencapai target dalam jangka waktu yang lama.

4) Faktor beta

   Faktor beta (β) merupakan parameter yang digunakan untuk memperbaiki nilai *learning rate.*

Dengan 5 input yang dimasukkan antara lain *mean, variance, skewness, kurtosis, dan entrophy*, dan 3 output yang ingin dihasilkan, maka jumlah *hidden layer* ditentukan dengan :

$$\boldsymbol{N_H = \sqrt{N_I \times N_O}} \quad (3.3)$$

dengan :

N_H : jumlah node *hidden*

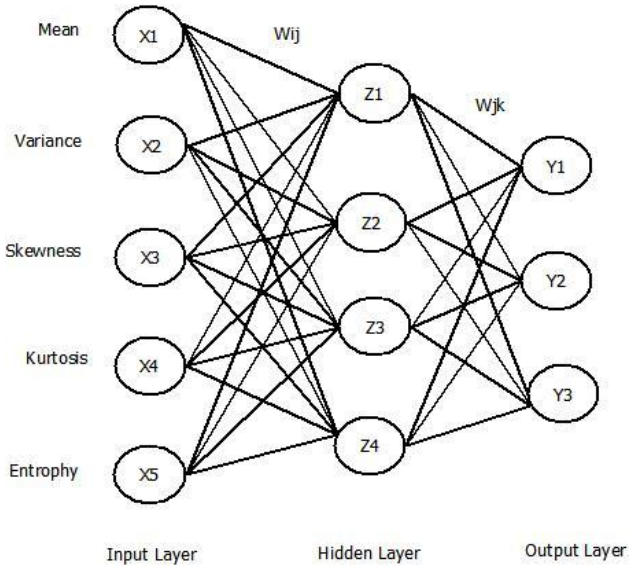
N_I : jumlah node input

N_O : jumlah node output

maka,

$$N_H = \sqrt{5 \times 3}$$
$$= 3{,}873 \approx 4$$

Sesuai dengan perhitungan di atas, maka ditentukan bahwa hidden layer yang digunakan adalah 2. Berikut merupakan gambaran rancangan arsitektur yang digunakan :



Gambar 3.5 Arsitektur *Levenberg-Marquardt*

Rancangan arsitektur di atas, jaringan yang akan dibangun terdiri dari 3 lapisan / *layer* yakni *input layer* yaitu X1, X2, X3, X4, dan X5, *hidden layer* Z1, Z2, Z3, dan Z4 dan 3 *output layer* Y1, Y2, dan Y3.

Berikut merupakan tahap – tahap dalam proses pelatihan *Levenberg-Marquardt* (diterapkan pada contoh citra bumi yang sudah diekstraksi ciri) :

1) Inisialisasi parameter awal yakni *maximum epoch*, *maximum error*, *learning rate*, dan faktor beta. Untuk simulasi ini, dipilih *maximum epoch* = 10, *maximum error* = 0,01, μ = 0,01, dan β = 2.

   Inisialisasi bobot awal. Bobot awal merupakan bilangan acak antara 0 sampai 1. Berikut merupakan bobot yang digunakan sebagai contoh :

$$Wij = \begin{bmatrix} 0,5 & 0,4 & 0,25 & 0,3 & 0,12 \\ 0,1 & 0,6 & 0,3 & 0,3 & 0,73 \\ 0,24 & 0,3 & 0,3 & 0,6 & 0,1 \\ 0,12 & 0,3 & 0,3 & 0,4 & 0,51 \end{bmatrix}$$

$$Wjk = \begin{bmatrix} 0,35 & 0,1 & 0,23 & 0,38 \\ 0,2 & 0,4 & 0,18 & 0,4 \\ 0,27 & 0,6 & 0,4 & 0,31 \end{bmatrix}$$

2) Hitung *input* setiap node pada *hidden layer* dengan persamaan 2.15

$$\boldsymbol{z_{-in_j} = \sum_{i=1}^{n} x_i \times w_{ij}}$$

   dengan :

   x_i : *input* sistem berupa nilai *mean*, *variance*, *skewness*, kurtosis, dan *entrophy* yang dinormalisasi dengan *range* yang didapat dari perhitungan pada lampiran H, seperti berikut :

Tabel 3.2 *Range Mean*

Mean	**Representasi**
19,334 – 41,716	0,125
41,717 – 64,097	0,250
64,098 – 86,479	0,375
86,480 – 108,86	0,5
108,861 – 131,242	0,625
131,243 – 153,623	0,75
153,624 – 176,004	0,875
176,005 – 198,386	1

Tabel 3.3 *Range Variance*

Variance	**Representasi**
148,157 – 1222,579	0,125
1222,580 – 2297,001	0,250
2297,002 – 3371,423	0,375
3371,424 – 4445,846	0,5
4445,847 – 5520,268	0,625

5520,269 – 6594,69	0,75
6594,691 – 7669,112	0,875
7669,113 – 8743,534	1

Tabel 3.4 *Range Skewness*

Skewness	**Representasi**
-1,324 – -0,748	0,125
-0,749 – -0,173	0,250
-0,174 – 0,403	0,375
0,404 – 0,979	0,5
0,980 – 1,555	0,625
1,556 – 2,131	0,75
2,132 – 2,707	0,875
2,708 – 3,282	1

Tabel 3.5 *Range* Kurtosis

Kurtosis	**Representasi**
1,283 – 1,926	0,125
1,927 – 2,570	0,250
2,571 – 3,213	0,375
3,214 – 3,856	0,5
3,857 – 4,499	0,625
4,500 – 5,143	0,75
5,144 – 5,786	0,875
5,787 – 6,429	1

Tabel 3.6 *Range Entrophy*

Entrophy	**Representasi**
1,815 – 2,45	0,125
2,451 – 3,086	0,250
3,087 – 3,721	0,375
3,722 – 4,356	0,5
4,357 – 4,991	0,625
4,992 – 5,627	0,75
5,628 – 6,262	0,875
6,263 – 6,897	1

w_{ij} : bobot acak awal yang diinisialisasi pada langkah 1

maka,

$$z_{_in_1} = (0{,}125 \times 0{,}5) + (0{,}375 \times 0{,}4) + (0{,}625 \times 0{,}25) + (0{,}625 \times 0{,}3) + (0{,}625 \times 0{,}12) = 0{,}669$$

$$z_{_in_2} = (0{,}125 \times 0{,}1) + (0{,}375 \times 0{,}6) + (0{,}625 \times 0{,}3) + (0{,}625 \times 0{,}3) + (0{,}625 \times 0{,}73) = 1{,}106$$

$$z_{-in_3} = (0{,}125 \times 0{,}24) + (0{,}375 \times 0{,}3) + (0{,}625 \times 0{,}3) + (0{,}625 \times 0{,}6) + (0{,}625 \times 0{,}1) = 0{,}843$$

$$z_{-in_4} = (0{,}125 \times 0{,}12) + (0{,}375 \times 0{,}3) + (0{,}625 \times 0{,}25) + (0{,}625 \times 0{,}4) + (0{,}625 \times 0{,}51) = 0{,}903$$

Hitung *output* setiap node pada *hidden layer* dengan persamaan 2.16 dan 2.17

$$\boldsymbol{z_j = f(z_in_j)}$$

$$\boldsymbol{z_j = \frac{1}{1 + \exp(-z_in_j)}}$$

$$z_1 = \frac{1}{1 + \exp(-0{,}669)} = 0{,}661$$

$$z_2 = \frac{1}{1 + \exp(-1{,}106)} = 0{,}751$$

$$z_3 = \frac{1}{1 + \exp(-0{,}843)} = 0{,}699$$

$$z_4 = \frac{1}{1 + \exp(-0{,}903)} = 0{,}711$$

3) Hitung *input* setiap node *output layer* dengan persamaan 2.18

$$\boldsymbol{y_{-in_k} = \sum_{j=1}^{n} z_j \times w_{jk}}$$

$$y_{-in_1} = (0{,}661 \times 0{,}35) + (0{,}751 \times 0{,}1) + (0{,}699 \times 0{,}23) + (0{,}711 \times 0{,}38) = 0{,}738$$

$$y_{-in_2} = (0{,}661 \times 0{,}2) + (0{,}751 \times 0{,}4) + (0{,}699 \times 0{,}18) + (0{,}711 \times 0{,}4) = 0{,}843$$

$$y_{-in_3} = (0{,}661 \times 0{,}27) + (0{,}751 \times 0{,}6) + (0{,}699 \times 0{,}4) + (0{,}711 \times 0{,}31) = 1{,}130$$

Hitung *output* setiap node *output layer* dengan persamaan 2.19

$$\boldsymbol{y_k = f(y_in_k)}$$

$$\boldsymbol{y_k = \frac{1}{1 + \exp(-y_in_k)}}$$

$$y_1 = \frac{1}{1 + \exp(-0{,}738)}$$
$$= 0{,}676$$
$$y_2 = \frac{1}{1 + \exp(-0{,}843)}$$
$$= 0{,}699$$
$$y_3 = \frac{1}{1 + \exp(-1{,}130)}$$
$$= 0{,}756$$

4) Hitung *error* setiap node *output layer* dengan persamaan 2.20. Tahap ini merupakan awal tahap propagasi balik.

$$\boldsymbol{e = [t_k - y_k]^T}$$

dengan :

t : vektor target yang diasumsikan adalah $[1 \quad 1 \quad 1]$ sebagai target untuk citra bumi (Tabel 3.1)

maka,

$$e = [1 - 0{,}676 \quad 1 - 0{,}699 \quad 1 - 0{,}756]$$
$$e = \begin{bmatrix} 0{,}324 \\ 0{,}301 \\ 0{,}244 \end{bmatrix}$$

5) Hitung matriks Jacobian dengan persamaan 2.13

$$\boldsymbol{J = \left[\frac{\partial e}{\partial w}\right]}$$

$$= \begin{bmatrix} \frac{0{,}324}{0{,}5} & \frac{0{,}324}{0{,}4} & \frac{0{,}324}{0{,}25} & \frac{0{,}324}{0{,}3} & \frac{0{,}324}{0{,}12} & \frac{0{,}324}{0{,}1} & \cdots & \frac{0{,}324}{0{,}4} & \frac{0{,}324}{0{,}31} \\ \frac{0{,}301}{0{,}5} & \frac{0{,}301}{0{,}4} & \frac{0{,}301}{0{,}25} & \frac{0{,}301}{0{,}3} & \frac{0{,}301}{0{,}12} & \frac{0{,}301}{0{,}1} & \cdots & \frac{0{,}301}{0{,}4} & \frac{0{,}301}{0{,}31} \\ \frac{0{,}244}{0{,}5} & \frac{0{,}244}{0{,}4} & \frac{0{,}244}{0{,}25} & \frac{0{,}244}{0{,}3} & \frac{0{,}244}{0{,}12} & \frac{0{,}244}{0{,}1} & \cdots & \frac{0{,}244}{0{,}4} & \frac{0{,}244}{0{,}31} \end{bmatrix}$$

$$= \begin{bmatrix} 0{,}647 & 0{,}809 & 1{,}294 & 1{,}078 & 2{,}699 & 3{,}235 & \dots & 0{,}809 & 1{,}044 \\ 0{,}602 & 0{,}752 & 1{,}203 & 1{,}003 & 2{,}507 & 3{,}009 & \dots & 0{,}752 & 0{,}971 \\ 0{,}488 & 0{,}611 & 0{,}977 & 0{,}814 & 2{,}035 & 2{,}443 & \dots & 0{,}611 & 0{,}788 \end{bmatrix}$$

6) Hitung selisih bobot dengan persamaan 2.14

$$\Delta \boldsymbol{W} = [\boldsymbol{J^T J} + \boldsymbol{\mu I}]^{-} \boldsymbol{J^T e}$$

Berikut merupakan hasil selisih bobot yang didapat :

$$\Delta W = \begin{bmatrix} 0{,}003 \\ 0{,}003 \\ 0{,}006 \\ 0{,}005 \\ 0{,}012 \\ 0{,}014 \\ 0{,}002 \\ 0{,}005 \\ 0{,}005 \\ 0{,}002 \\ 0{,}006 \\ 0{,}005 \\ 0{,}005 \\ 0{,}002 \\ 0{,}014 \\ 0{,}012 \\ 0{,}005 \\ 0{,}006 \\ 0{,}004 \\ 0{,}003 \\ 0{,}004 \\ 0{,}014 \\ 0{,}007 \\ 0{,}004 \\ 0{,}007 \\ 0{,}003 \\ 0{,}008 \\ 0{,}003 \\ 0{,}005 \\ 0{,}002 \\ 0{,}003 \\ 0{,}004 \end{bmatrix}$$

7) *Update error* dengan persamaan 2.21

$$\boldsymbol{e(new)} = \frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{e^T e}$$

$$e(new) = \frac{1}{2} [0{,}326 \quad 0{,}303 \quad 0{,}247] \times \begin{bmatrix} 0{,}326 \\ 0{,}303 \\ 0{,}247 \end{bmatrix}$$

$$= 0{,}130$$

8) *Update* bobot setiap node *output layer* dengan persamaan 2.22

$$\boldsymbol{w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}}$$

dengan Δw_{jk} didapat dari 12 isi terakhir matriks ΔW, maka

$$Wjk = \begin{bmatrix} 0{,}355 & 0{,}106 & 0{,}233 & 0{,}382 \\ 0{,}204 & 0{,}414 & 0{,}186 & 0{,}404 \\ 0{,}277 & 0{,}603 & 0{,}408 & 0{,}313 \end{bmatrix}$$

Update bobot setiap node *hidden layer* dengan persamaan 2.23

$$\boldsymbol{w_{ij}(new) = w_{ij}(old) + \Delta w_{ij}}$$

dengan Δw_{ij} didapat dari 20 isi pertama matriks ΔW, maka

$$Wij = \begin{bmatrix} 0{,}503 & 0{,}403 & 0{,}256 & 0{,}305 & 0{,}132 \\ 0{,}114 & 0{,}602 & 0{,}305 & 0{,}305 & 0{,}732 \\ 0{,}246 & 0{,}305 & 0{,}305 & 0{,}602 & 0{,}114 \\ 0{,}12 & 0{,}305 & 0{,}256 & 0{,}403 & 0{,}513 \end{bmatrix}$$

Iterasi masih dilanjutkan karena nilai *error* masih lebih besar dari *maximum error*. Iterasi baru dihentikan setelah mencapai *maximum epoch* yakni 1000. Nilai bobot baru yang didapat adalah :

$$Wij = \begin{bmatrix} 0{,}503 & 0{,}403 & 0{,}256 & 0{,}305 & 0{,}132 \\ 0{,}114 & 0{,}602 & 0{,}305 & 0{,}305 & 0{,}732 \\ 0{,}246 & 0{,}305 & 0{,}305 & 0{,}602 & 0{,}114 \\ 0{,}12 & 0{,}305 & 0{,}256 & 0{,}403 & 0{,}513 \end{bmatrix}$$

$$Wjk = \begin{bmatrix} 0{,}355 & 0{,}106 & 0{,}233 & 0{,}382 \\ 0{,}204 & 0{,}414 & 0{,}186 & 0{,}404 \\ 0{,}277 & 0{,}603 & 0{,}408 & 0{,}313 \end{bmatrix}$$

Selanjutnya akan dilakukan proses pengujian. Pengujian algoritma *Levenberg-Marquardt* yang menggunakan tahap yang sama dengan pelatihan namun tanpa proses propagasi balik. Bobot yang digunakan bukan bilangan acak melainkan bobot baru dari proses pelatihan.

1) Inisialisasi bobot. Bobot yang digunakan diambil dari proses pelatihan.

$$Wij = \begin{bmatrix} 0{,}503 & 0{,}403 & 0{,}256 & 0{,}305 & 0{,}132 \\ 0{,}114 & 0{,}602 & 0{,}305 & 0{,}305 & 0{,}732 \\ 0{,}246 & 0{,}305 & 0{,}305 & 0{,}602 & 0{,}114 \\ 0{,}12 & 0{,}305 & 0{,}256 & 0{,}403 & 0{,}513 \end{bmatrix}$$

$$Wjk = \begin{bmatrix} 0{,}355 & 0{,}106 & 0{,}233 & 0{,}382 \\ 0{,}204 & 0{,}414 & 0{,}186 & 0{,}404 \\ 0{,}277 & 0{,}603 & 0{,}408 & 0{,}313 \end{bmatrix}$$

2) Hitung *input* setiap node pada *hidden layer*

$$\boldsymbol{z_{_in_j}} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{x_i} \times \boldsymbol{w_{ij}}$$

$$z_{_in_1} = (0{,}125 \times 0{,}503) + (0{,}375 \times 0{,}403) + (0{,}625 \times 0{,}256) + (0{,}625 \times 0{,}305) + (0{,}625 \times 0{,}132) = 0{,}136$$

$$z_{_in_2} = (0{,}125 \times 0{,}114) + (0{,}375 \times 0{,}602) + (0{,}625 \times 0{,}305) + (0{,}625 \times 0{,}305) + (0{,}625 \times 0{,}732) = 0{,}201$$

$$z_{_in_3} = (0{,}125 \times 0{,}246) + (0{,}375 \times 0{,}305) + (0{,}625 \times 0{,}305) + (0{,}625 \times 0{,}602) + (0{,}625 \times 0{,}114) = 0{,}102$$

$$z_{_in_4} = (0{,}125 \times 0{,}132) + (0{,}375 \times 0{,}305) + (0{,}625 \times 0{,}256) + (0{,}625 \times 0{,}403) + (0{,}625 \times 0{,}513) = 0{,}102$$

Hitung *output* setiap node pada *hidden layer*

$$\boldsymbol{z_j} = \boldsymbol{f}(\boldsymbol{z_in_j})$$

$$\boldsymbol{z_j} = \frac{\mathbf{1}}{\mathbf{1} + \exp(-\boldsymbol{z_in_j})}$$

$$z_1 = \frac{1}{1 + \exp(-0{,}136)} = 0{,}534$$

$$z_2 = \frac{1}{1 + \exp(-0{,}201)} = 0{,}550$$

$$z_3 = \frac{1}{1 + \exp(-0{,}102)} = 0{,}526$$

$$z_4 = \frac{1}{1 + \exp(-0{,}102)} = 0{,}525$$

3) Hitung *input* setiap node *output layer*

$$\boldsymbol{y_{_in_k}} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{z_j} \times \boldsymbol{w_{jk}}$$

$$y_{_in_1} = (0{,}534 \times 0{,}353) + (0{,}550 \times 0{,}104) + (0{,}526 \times 0{,}233) + (0{,}525 \times 0{,}382) = 0{,}570$$

$$y_{_in_2} = (0{,}534 \times 0{,}203) + (0{,}550 \times 0{,}411) + (0{,}526 \times 0{,}185) + (0{,}525 \times 0{,}403) = 0{,}644$$

$$y_{_in_3} = (0{,}534 \times 0{,}276) + (0{,}550 \times 0{,}603) + (0{,}526 \times 0{,}406) + (0{,}525 \times 0{,}313) = 0{,}857$$

Hitung *output* setiap node *output layer*

$$\boldsymbol{y_k = f(y_in_k)}$$

$$\boldsymbol{y_k = \frac{1}{1 + \exp(-y_in_k)}}$$

$$y_1 = \frac{1}{1 + \exp(-0{,}570)} = 0{,}639$$

$$y_2 = \frac{1}{1 + \exp(-0{,}644)} = 0{,}656$$

$$y_3 = \frac{1}{1 + \exp(-0{,}857)} = 0{,}702$$

Untuk penentuan hasil klasifikasi gambar, dilihat dari nilai Y1, Y2, Y3. Jika $y_k \geq 0{,}5$ maka $Y = 1$ selain itu 0. Hasil klasifikasi output yang diharapkan dapat dilihat kembali pada tabel 3.1.

Dengan contoh perhitungan di atas, hasil Y1, Y2, Y3 yang didapat sebesar 0,639, 0,656, dan 0,702

Tabel 3.7 Hasil Klasifikasi Citra Latih

	Y1	**Y2**	**Y3**
	0,639	0,656	0,702
Nilai Target	1	1	1

Karena nilai target yang didapat adalah 1 1 1, maka sesuai dengan tabel klasifikasi, gambar tersebut diklasifikasikan sebagai gambar bumi. Untuk menghasilkan *retrieval*, gambar yang diujikan dapat lebih dari 1. Hasil klasifikasi untuk 120 data yang digunakan, dapat dilihat pada lampiran G.

Sebagai contoh, pengujian akan dilakukan pada citra bumi baru seperti berikut :



Gambar 3.6 Citra Bumi (2)

Citra di atas memiliki nilai ekstraksi ciri statistik *mean* : 45,661, *variance* : 3444,463, *skewness* : 1,140, kurtosis : 3,369, dan *entrophy* : 3,708. Maka, berikut proses pengujian *Levenberg-Marquardt* untuk mengetahui hasil klasifikasi citra tersebut :

1) Inisialisasi bobot. Bobot yang digunakan diambil dari proses pelatihan citra bumi sebelumnya.

$$Wij = \begin{bmatrix} 0{,}503 & 0{,}403 & 0{,}256 & 0{,}305 & 0{,}132 \\ 0{,}114 & 0{,}602 & 0{,}305 & 0{,}305 & 0{,}732 \\ 0{,}246 & 0{,}305 & 0{,}305 & 0{,}602 & 0{,}114 \\ 0{,}12 & 0{,}305 & 0{,}256 & 0{,}403 & 0{,}513 \end{bmatrix}$$

$$Wjk = \begin{bmatrix} 0{,}355 & 0{,}106 & 0{,}233 & 0{,}382 \\ 0{,}204 & 0{,}414 & 0{,}186 & 0{,}404 \\ 0{,}277 & 0{,}603 & 0{,}408 & 0{,}313 \end{bmatrix}$$

2) Hitung *input* setiap node pada *hidden layer*

$$\boldsymbol{z_{-in_j} = \sum_{i=1}^{n} x_i \times w_{ij}}$$

$$z_{-in_1} = (0{,}25 \times 0{,}503) + (0{,}5 \times 0{,}403) + (0{,}625 \times 0{,}256) + (0{,}5 \times 0{,}305) + (0{,}375 \times 0{,}132) = 0{,}687$$

$$z_{-in_2} = (0{,}25 \times 0{,}114) + (0{,}5 \times 0{,}602) + (0{,}625 \times 0{,}305) + (0{,}5 \times 0{,}305) + (0{,}375 \times 0{,}732) = 0{,}945$$

$$z_{-in_3} = (0{,}25 \times 0{,}246) + (0{,}5 \times 0{,}305) + (0{,}625 \times 0{,}305) + (0{,}5 \times 0{,}602) + (0{,}375 \times 0{,}114) = 0{,}746$$

$$z_{_in_4} = (0{,}25 \times 0{,}132) + (0{,}5 \times 0{,}305) + (0{,}625 \times 0{,}256) + (0{,}5 \times 0{,}403) + (0{,}375 \times 0{,}513) = 0{,}737$$

Hitung *output* setiap node pada *hidden layer*

$$\boldsymbol{z_j = f(z_in_j)}$$

$$\boldsymbol{z_j = \frac{1}{1 + \exp(-z_in_j)}}$$

$$z_1 = \frac{1}{1 + \exp(-0{,}687)} = 0{,}665$$

$$z_2 = \frac{1}{1 + \exp(-0{,}945)} = 0{,}720$$

$$z_3 = \frac{1}{1 + \exp(-0{,}746)} = 0{,}678$$

$$z_4 = \frac{1}{1 + \exp(-0{,}737)} = 0{,}676$$

3) Hitung *input* setiap node *output layer*

$$\boldsymbol{y_{_in_k} = \sum_{j=1}^{n} z_j \times w_{jk}}$$

$$y_{_in_1} = (0{,}665 \times 0{,}353) + (0{,}720 \times 0{,}104) + (0{,}678 \times 0{,}233) + (0{,}676 \times 0{,}382) = 0{,}727$$

$$y_{_in_2} = (0{,}665 \times 0{,}203) + (0{,}720 \times 0{,}411) + (0{,}678 \times 0{,}185) + (0{,}676 \times 0{,}403) = 0{,}830$$

$$y_{_in_3} = (0{,}665 \times 0{,}276) + (0{,}720 \times 0{,}603) + (0{,}678 \times 0{,}406) + (0{,}676 \times 0{,}313) = 1{,}105$$

Hitung *output* setiap node *output layer*

$$\boldsymbol{y_k = f(y_in_k)}$$

$$\boldsymbol{y_k = \frac{1}{1 + \exp(-y_in_k)}}$$

$$y_1 = \frac{1}{1 + \exp(-0{,}727)}$$

$$= 0{,}674$$

$$y_2 = \frac{1}{1 + \exp(-0{,}803)}$$
$$= 0{,}696$$
$$y_3 = \frac{1}{1 + \exp(-1{,}105)}$$
$$= 0{,}751$$

Melihat dari hasil perhitungan di atas, nilai Y1, Y2, Y3 yang didapat sebesar 0,674, 0,696, dan 0,751

Tabel 3.8 Hasil Klasifikasi Citra Uji

	Y1	**Y2**	**Y3**
	0,674	0,696	0,751
Nilai Target	1	1	1

Nilai target yang didapat adalah 1 1 1, maka gambar tersebut masuk klasifikasi citra bumi.

Dengan menggunakan contoh perhitungan sebelumnya, setiap citra dengan hasil target 1 1 1 akan disimpan ke *database* dengan hasil klasifikasi citra bumi, kemudian ketika pengguna memasukan citra bumi baru ke sistem *retrieval*, sistem akan mencocokan semua citra dengan nilai yang sama dalam database, yang kemudian akan ditampilkan sebagai hasil *retrieval* citra bumi. Begitupun berlaku untuk semua citra lain.

Sebagai contoh, ketika mengujikan citra bumi kedua, jika pengguna ingin mendapatkan hasil *retrieval*, maka sistem akan menampilkan citra bumi pertama yang sudah dilatih dan diujikan sebagai citra yang sudah terdapat dalam *database.*

3.5 Analisis Kebutuhan Non-Fungsional

Untuk dapat membangun dan mengimplementasikan sistem, terdapat beberapa spesifikasi komponen-komponen yang dibutuhkan. Berikut analisis kebutuhan non-fungsional sistem :

3.5.1 Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang dibutuhkan untuk membangun sistem antara lain :

1) *Monitor* : 14 inch 64-bit

2) *Processor* : Intel Core i3 CPU @ 2,10 GHz
3) RAM : 2 GB
4) *Keyboard*
5) *Mouse*

3.5.2 Analisis Kebutuhan Perangkat Lunak

Berikut perangkat lunak yang digunakan :

1) Sistem operasi : Windows 7 Ultimate
2) Microsoft Visual Studio 2010
3) XAMPP Server
4) SQLYog Community

3.5.3 Analisis Kebutuhan Pengguna

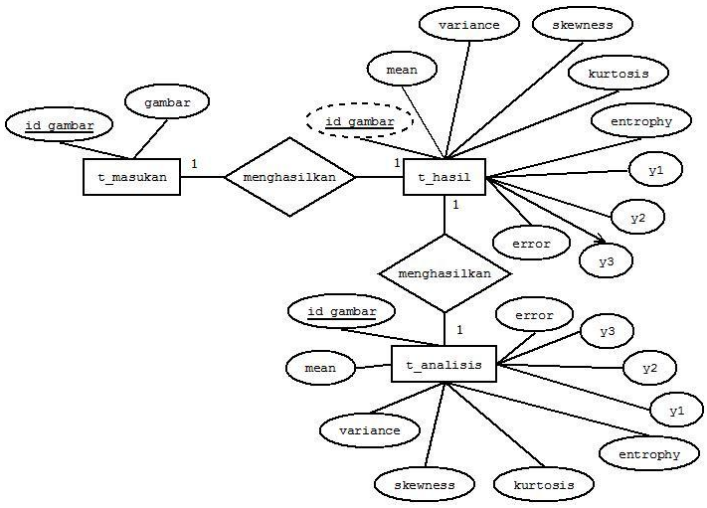
Adapun kriteria pengguna yang dapat menggunakan sistem ini adalah :

1) Pengguna memahami cara mengoperasikan komputer.
2) Pengguna memahami cara kerja sistem CBIR dengan proses klasifikasi

3.5.4 Analisis Basis Data

1) *Entity Relationship Diagram* (ERD)

Untuk dapat melihat hubungan antar entitas dalam sistem, dapat dilihat dari diagram berikut :



Gambar 3.7 ERD

2) Kamus Data

Secara lebih rinci, atribut setiap entitas dapat dilihat dalam tabel berikut :

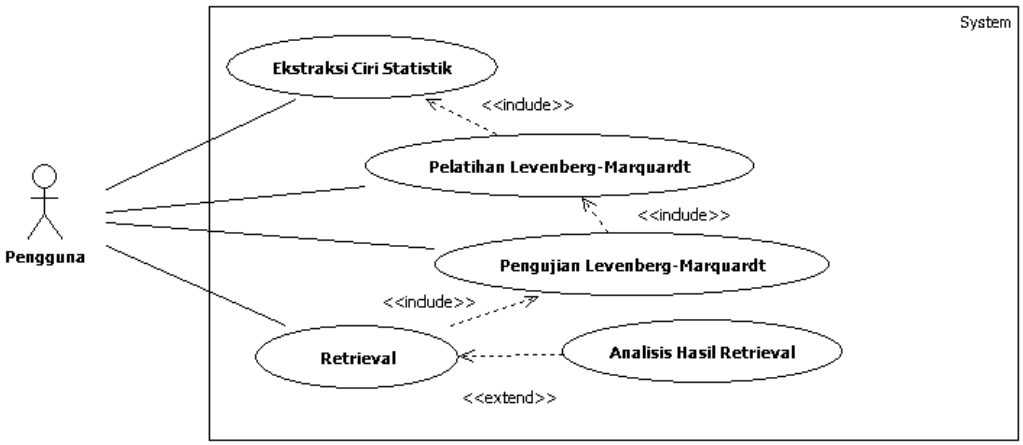
Tabel 3.9 Kamus Data

Entitas	Atribut	Keterangan
t_masukan	id_gambar int (4) not null, gambar longblob	Primary key
t_hasil	mean double, variance double, skewness double, kurtosis double, entrophy double, y1 int (2), y2 int (2), y3 int (2), error double, id_gambar int (4) not null	 Foreign key, primary key
t_analisis	id_gambar int (4) not null, mean double, variance double, skewness double, kurtosis double, entrophy double, y1 int (2), y2 int (2), y3 int (2), error double	Primary key

3.6 Pemodelan Sistem

3.6.1 *Use Case Diagram*

Untuk mengetahui interaksi yang dapat dilakukan pengguna dalam sistem, dapat dilihat dari diagram berikut :



Gambar 3.8 *Use Case Diagram*

3.6.1.1 Definisi Aktor

Tabel 3.10 Definisi Aktor

Aktor	Deskripsi
Pengguna	Aktor ini memiliki hak untuk memasukkan data, melakukan pengolahan citra, melakukan pelatihan algoritma *Levenberg-Marquardt*, melakukan pengujian, dan mendapatkan hasil retrieval citra dan akurasi algoritma.

3.6.1.2 Definisi Use Case

Tabel 3.11 Definisi *Use Case*

Use Case	Deskripsi
Ekstraksi ciri statistik	Fungsi ini digunakan untuk melakukan proses perhitungan ekstraksi ciri statistik
Pelatihan *Levenberg-Marquardt*	Fungsi ini dapat digunakan oleh aktor untuk melakukan proses pelatihan algoritma *Levenberg-Marquardt*
Pengujian *Levenberg-Marquardt*	Fungsi ini dapat digunakan oleh aktor untuk melakukan proses pengujian algoritma *Levenberg-Marquardt* untuk mengetahui hasil klasifikasi citra masukan.
Retrieval	Fungsi ini dapat digunakan oleh aktor untuk mendapatkan hasil *retrieval* dari sistem berdasarkan citra masukan dan mendapatkan hasil analisisnya

3.6.2 *Use Case Scenario*

Untuk lebih memahami setiap *use case* dapat dilihat dari skenario di bawah :

Tabel 3.12 *Use Case Scenario* Ekstraksi Ciri Statistik

Use Case Name	Ekstraksi Ciri Statistik		
Related Requirements	-		
Goal in Context	Pengguna dapat melakukan perhitungan ekstraksi ciri statistik		
Preconditions	Pengguna memiliki data berupa gambar		
Successful End Condition	Didapatkan nilai-nilai keluaran ekstraksi ciri statistik		
Failed End Condition	Tidak berhasil didapatkan nilai-nilai keluaran ekstraksi ciri statistik		
Primary Actor	Pengguna		
	Trigger	Sistem mendapatkan hasil ekstraksi ciri statistik	
Main Flow	Step	Action	
		Pengguna	Sistem
	1	*Input* data gambar	
	2	Memilih menu ekstraksi	
	3		Melakukan *pre-processing*
	4		Melakukan kalkulasi ekstraksi ciri statistik
	5		Menampilkan hasil ekstraksi
Extensions	Step	Branching Action	

		-

Tabel 3.13 *Use Case Scenario* Pelatihan *Levenberg-Marquardt*

Use Case Name	Pelatihan *Levenberg-Marquardt*		
Related Requirements	-		
Goal in Context	Pengguna dapat menggunakan proses pelatihan algoritma *Levenberg-Marquardt*		
Preconditions	Pengguna memiliki data latih di dalam *database*		
Successful End Condition	Proses kalkulasi pelatihan menghasilkan hasil klasifikasi yang akan disimpan dalam *database*		
Failed End Condition	Proses kalkulasi tidak berhasil diselesaikan		
Primary Actor	Pengguna		
	Trigger	Sistem menyimpan hasil klasifikasi di database	
Main Flow	Step	Action	
		Pengguna	Sistem
	1	Memilih menu pelatihan	
	2		Melakukan *pre-processing*
	3		Melakukan ekstraksi ciri statistik
	4		Melakukan kalkulasi pelatihan *Levenberg-Marquardt*
	5		Menyimpan hasil klasifikasi ke database
Extensions	Step	Branching Action	
		-	

Tabel 3.14 *Use Case Scenario* Pengujian *Levenberg-Marquardt*

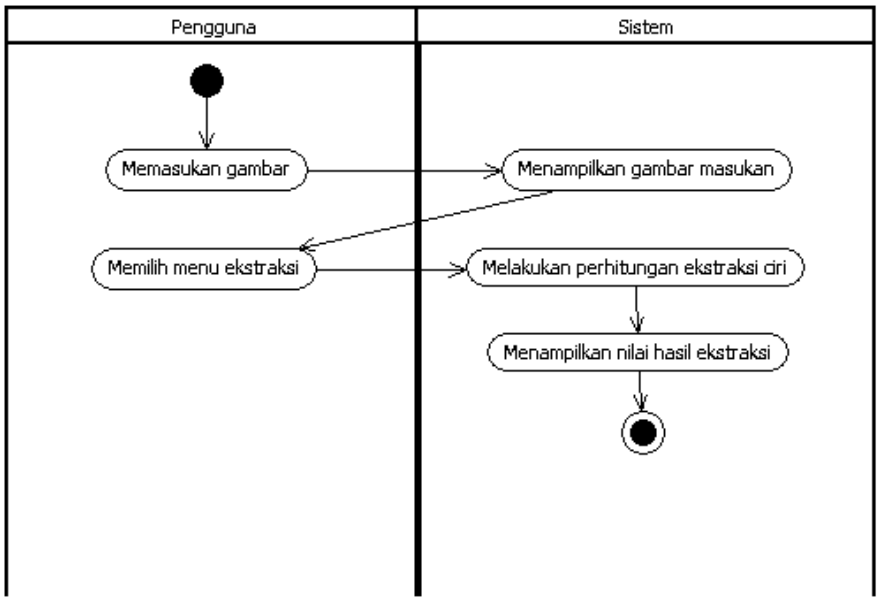
Use Case Name	Pengujian *Levenberg-Marquardt*		
Related Requirements	Pelatihan *Levenberg-Marquardt*, ekstraksi ciri statistik		
Goal in Context	Pengguna dapat proses pengujian algoritma *Levenberg-Marquardt* dan mendapatkan hasil klasifikasi citra		
Preconditions	Sudah dilakukan proses pelatihan sebelumnya dan terdapat hasil klasifikasi data latih dalam database		
Successful End Condition	Citra masukan dapat diklasifikasikan		
Failed End Condition	Citra masukan tidak dapat diklasifikasikan		
Primary Actor	Pengguna		
	Trigger	Sistem menampilkan hasil klasifikasi citra masukan	
Main Flow	Step	Action	
		Pengguna	Sistem
	1	Memilih menu uji	
	2		Melakukan kalkulasi pengujian *Levenberg-Marquardt*
	3		Menampilkan hasil klasifikasi citra uji
Extensions	Step	Branching Action	
		-	

Tabel 3.15 *Use Case Scenario Retrieval*

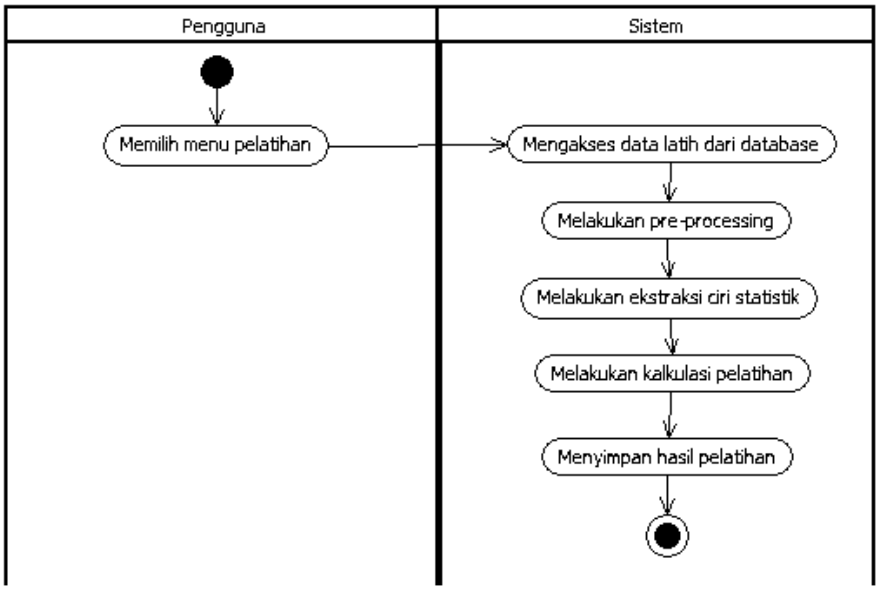
Use Case Name	*Retrieval*		
Related Requirements	Pengujian *Levenberg-Marquardt*		
Goal in Context	Pengguna bisa mendapatkan gambar-gambar serupa dengan citra masukan		
Preconditions	Proses pelatihan dan pengujian sudah dilakukan sebelumnya dan sudah terdapat hasil klasifikasi di database		
Successful End Condition	Dapat menampilkan beberapa citra dengan hasil klasifikasi yang sama		
Failed End Condition	Tidak dapat menampilkan citra		
Primary Actor	Pengguna		
	Trigger	Sistem menampilkan hasil *retrieval* berupa gambar-gambar dengan hasil klasifikasi yang sama dengan masukan	
Main Flow	Step	Action	
		Pengguna	Sistem
	1	Memilih menu *retrieve*	
	2		Mencocokan hasil klasifikasi ke *database*
	3		Menampilkan hasil *retrieval*
Extensions	Step	Branching Action	
	3.1	Menampilkan hasil analisis data	

3.6.3 *Activity Diagram*

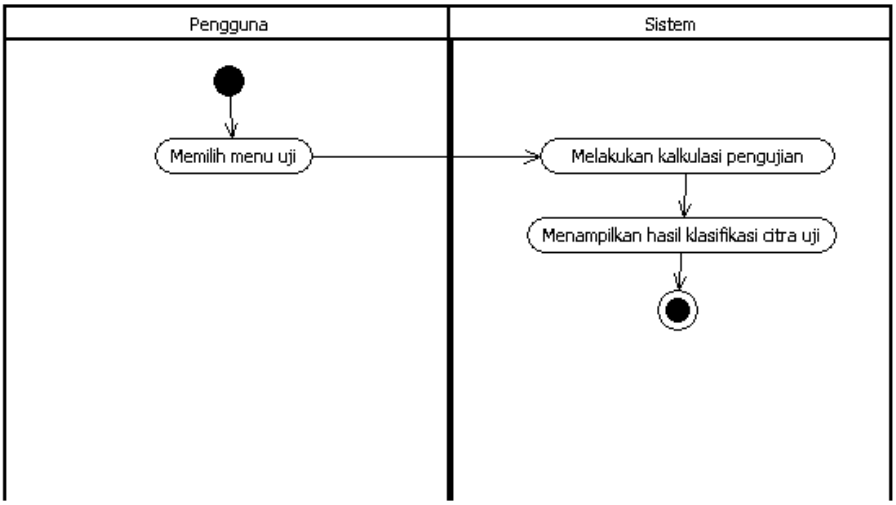
Berikut merupakan aktivitas dari setiap *use case* yang dapat lebih jelas dilihat pada *activity diagram* di bawah :



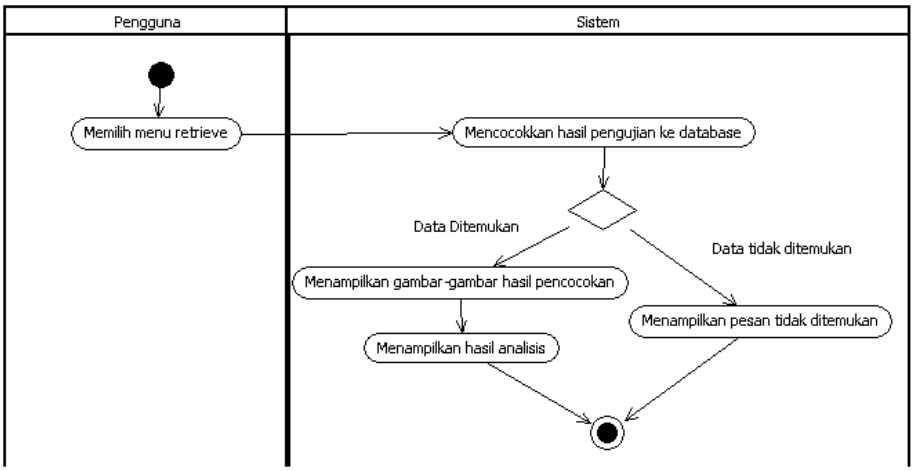
Gambar 3.10 *Activity Diagram* Ekstraksi Ciri Statistik



Gambar 3.10 *Activity Diagram* Pelatihan *Levenberg-Marquardt*



Gambar 3.11 *Activity Diagram* Pengujian *Levenberg-Marquardt*



Gambar 3.12 *Activity Diagram Retrieval*

3.6.4 *Class Diagram*

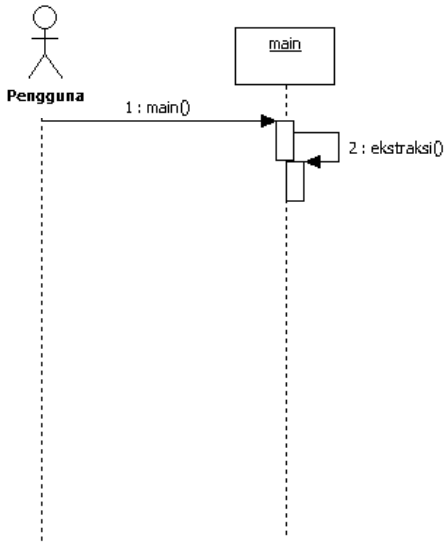
Untuk mendefinisikan kelas-kelas yang terdapat di dalam sistem, dapat dilihat dari diagram berikut :



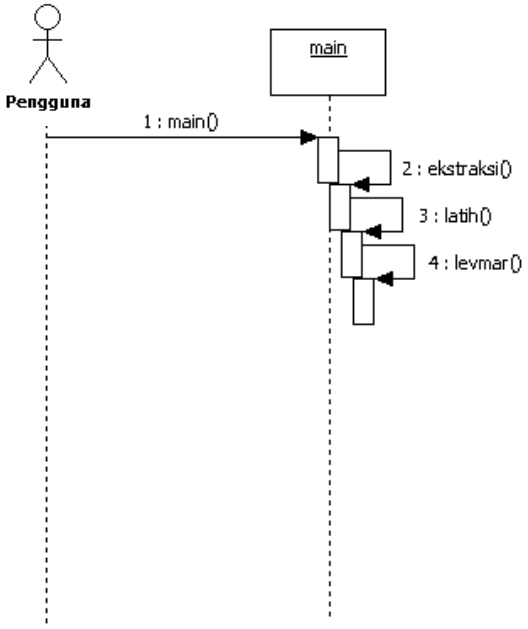
Gambar 3.13 *Class Diagram*

3.6.5 *Sequence Diagram*

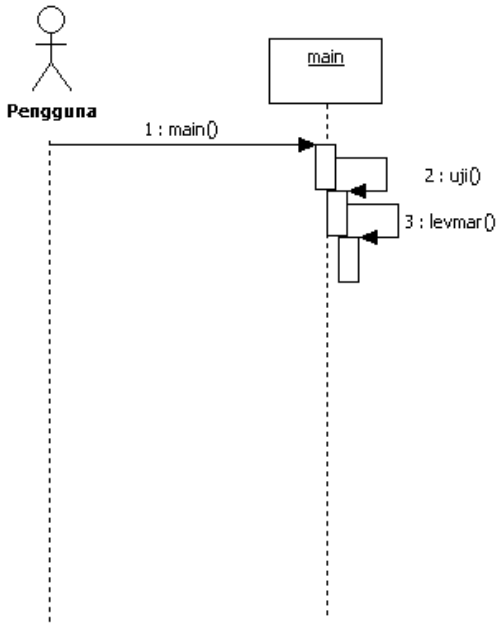
Sequence diagram menggambarkan perilaku objek pada setiap use case dengan mendeskripsikan waktu hidup dan pesan objek. *Sequence diagram* sistem yang dibangun dapat dilihat di bawah ini :



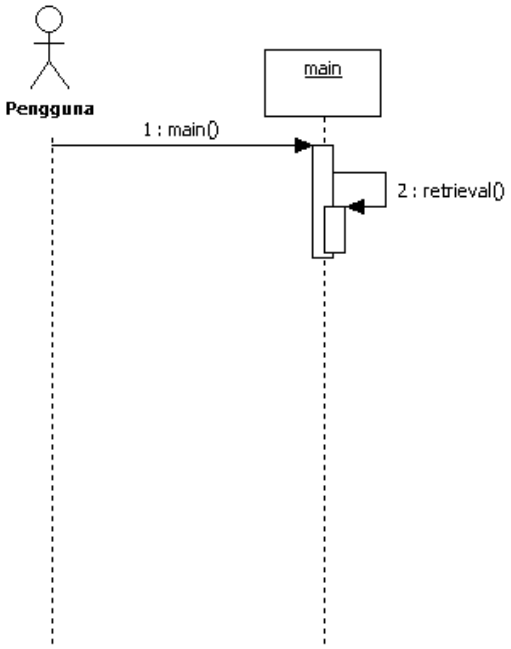
Gambar 3.14 *Sequence Diagram* Ekstraksi Ciri Statistik



Gambar 3.15 *Sequence Diagram* Pelatihan *Levenberg-Marquardt*



Gambar 3.16 *Sequence Diagram* Pengujian *Levenberg-Marquardt*

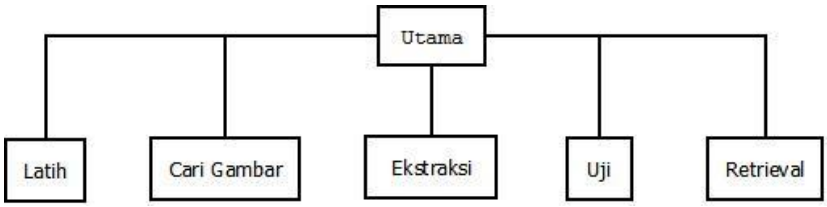


Gambar 3.17 *Sequence Diagram Retrieval*

3.7 Perancangan Sistem

3.7.1 Perancangan Struktur Menu

Untuk lebih memahami menu yang akan diimplementasikan dalam sistem, berikut merupakan perancangan struktur menu yang digunakan :

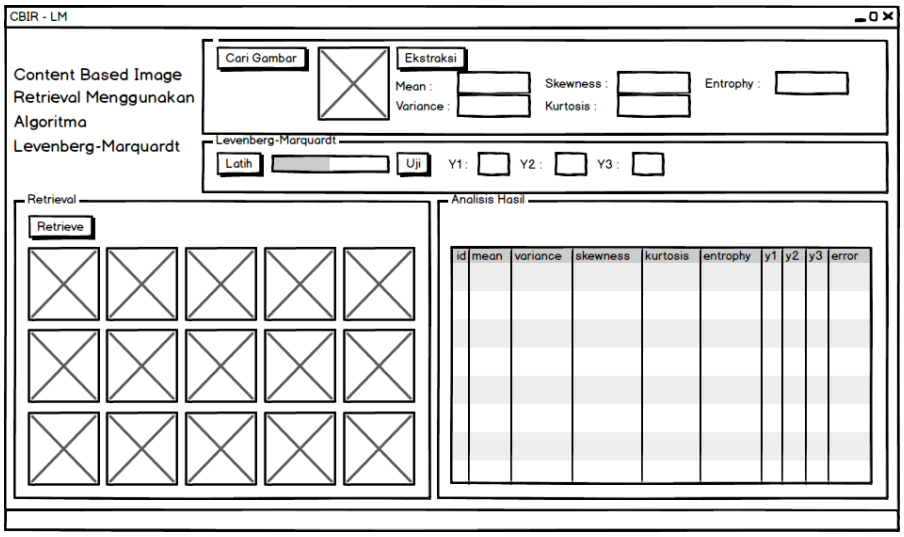


Gambar 3.18 Perancangan Struktur Menu

3.7.2 Perancangan Antarmuka

Perancangan antarmuka sistem CBIR yang dibuat dapat dilihat berikut ini :

Tabel 3.16 Perancangan Antarmuka Utama

T01	
	Klik “Cari Gambar” untuk menuju memasukkan gambar Bila belum melakukan pelatihan, muncul P01 Klik “Ekstraksi” untuk melakukan perhitungan ekstraksi ciri Bila belum melakukan pelatihan, muncul P01 Bila belum memasukkan gambar, muncul P02 Klik “Latih” untuk melakukan pelatihan *Levenberg-Marquardt* Bila pelatihan selesai, muncul P05 Klik “Uji” untuk melakukan pengujian *Levenberg-Marquardt* Bila belum melakukan pelatihan, muncul P01 Bila belum memasukkan gambar, muncul P02 Bila belum melakukan ekstraksi, muncul P03 Klik “Retrieve” untuk

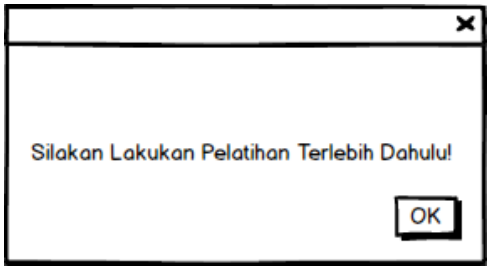
	menampilkan hasil *retrieval* Bila belum melakukan pelatihan, muncul P01 Bila belum memasukkan gambar, muncul P02 Bila belum melakukan ekstraksi, muncul P03 Bila belum melakukan pengujian, muncul P04 Bila hasil *retrieve* tidak ditemukan, muncul P06 Klik "X" untuk keluar

3.7.3 Perancangan Pesan

Berikut merupakan beberapa pesan yang akan muncul sebagai informasi atas aksi yang dilakukan pengguna :

1) Pesan Lakukan Pelatihan (P01)

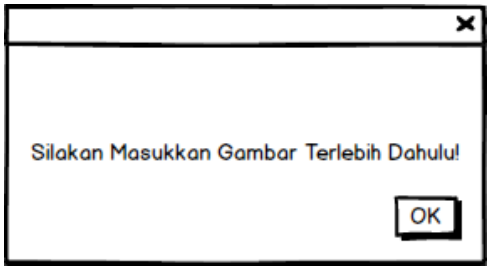
   Pesan akan muncul ketika pengguna memilih tombol "Cari Gambar" atau "Ekstraksi" atau "Uji" atau "Retrieve" sebelum melakukan pelatihan *Levenberg-Marquardt*.



Gambar 3.19 Pesan Lakukan Pelatihan (P01)

2) Pesan Masukkan Gambar (P02)

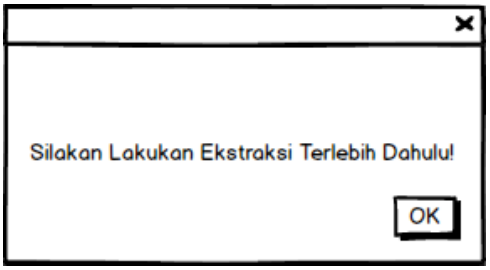
   Pesan akan muncul ketika pengguna memilih tombol "Ekstraksi" atau "Uji" atau "Retrieve" sebelum memasukkan gambar.



Gambar 3.20 Pesan Masukkan Gambar (P02)

3) Pesan Lakukan Ekstraksi (P03)

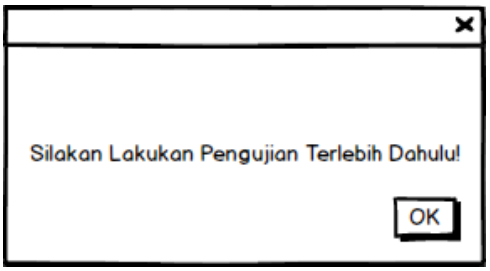
Pesan akan muncul ketika pengguna memilih tombol “Uji” atau “Retrieve” sebelum melakukan ekstraksi ciri.



Gambar 3.21 Pesan Lakukan Ekstraksi (P03)

4) Pesan Lakukan Pengujian (P04)

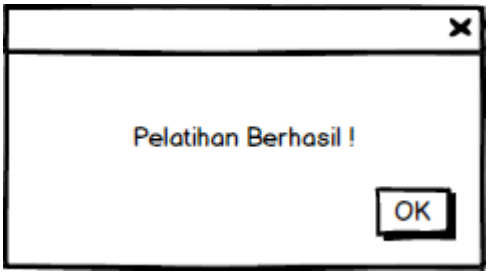
Pesan akan muncul ketika pengguna memilih tombol “Retrieve” sebelum melakukan pengujian *Levenberg-Marquardt.*



Gambar 3.22 Pesan Lakukan Pengujian (P04)

5) Pesan Pelatihan Berhasil (P05)

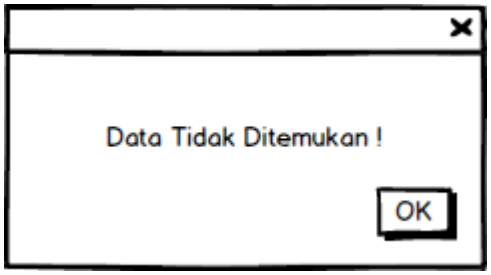
Pesan akan muncul ketika proses pelatihan *Levenberg-Marquardt* selesai dilakukan.



Gambar 3.23 Pesan Pelatihan Berhasil (P05)

6) Pesan Data Tidak Ditemukan (P06)

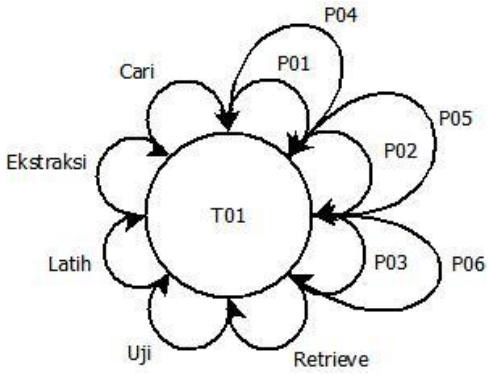
Pesan akan muncul ketika pengguna memasukkan gambar uji, namun tidak dapat ditemukan hasil serupa.



Gambar 3.24 Pesan Data Tidak Ditemukan (P06)

3.7.4 Jaringan Semantik

Berikut merupakan jaringan semantik sistem :



Gambar 3.25 Jaringan Semantik

Keterangan :

T01	: Antarmuka utama
P01	: Pesan lakukan pelatihan
P02	: Pesan masukkan gambar
P03	: Pesan lakukan ekstraksi
P04	: Pesan lakukan pelatihan
P05	: Pesan pelatihan berhasil
P02	: Pesan data tidak ditemukan
Cari	: Tombol “Cari Gambar”
Ekstraksi	: Tombol “Ekstraksi”
Latih	: Tombol “Latih”
Uji	: Tombol “Uji”
Retrieve	: Tombol “Retrieve”