BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis

Analisis adalah proses mengurai konsep ke dalam bagian-bagian yang lebih sederhana, sedemikian rupa sehingga struktur logisnya menjadi jelas. Analisis merupakan metode untuk menguji, menilai, dan memahami sistem pemikiran yang kompleks dengan memecahnya ke dalam unsur-unsur yang lebih sederhana sehingga hubungan antar unsur-unsur itu menjadi jelas.

3.1.1 Analisis Masalah

Kecerdasan buatan bisa digunakan untuk menentukan pola maupun pendektisian dan pengenalan terhadap jamur yang siap panen maupun belum siap panen. **Jamur** merupakan bagian dari tumbuhan yang bisa diketahui secara langsung dengan melihat fisik jamur tersebut apakah jamur ini siap panen atau belum siap panen yaitu dengan menggunakan kecerdasan buatan atau Artificial Intelligence .

Untuk menyelesaikan masalah pendeteksi jamur dalam JST, akan digunakan Jaringan Syaraf Tiruan Learning Vector Quantization sebagai solusi alternatif untuk menyelesaikan masalah tersebut, karena JST LVQ merupakan contoh metode pembelajaran dengan supervised (terawasi) yang biasa dipakai untuk pengenalan pola secara tipikal pengelompokan pola-pola ke dalam kelas-kelas pola, sehingga tepat jika digunakan JST LVQ dalam pendeteksi jamur karena algoritma LVQ dapat belajar secara otomatis untuk mengklasifikasikan vektor-vektor input

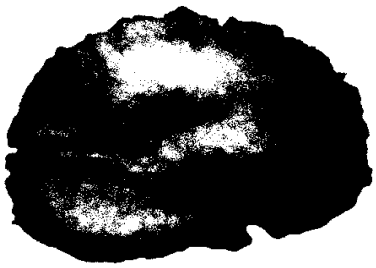
3.1.2 Analisis Data Masukan

Kondisi jamur yang akan di jadikan data masukan yaitu jamur tiram putih, jamur tiram coklat dimana dari jamur tersebut memiliki parameter yang sama untuk menentukan jamur siap panen yaitu parameternya berdasarkan ukuran . Citra inputan

dari sistem merupakan citra yang dihasilkan melalui *capture* objek yang diambil menggunakan kamera handphone oleh pengguna dengan ukuran 480 x 640 piksel.

Setiap proses pada proses pengolahan citra menggunakan nilai pixel dari citra untuk perhitungannya. Setiap pixel pada citra terdiri dari 3 unsur warna dasar yaitu Red (R), Green (G) dan Blue (B), yang sering disingkat RGB, dan setiap unsur memiliki nilai antara 0-255 yang menjadi kode kedalaman warna dari masing-masing unsur Sebagai contoh perhitungan pada proses-proses pengolahan citra, digunakan gambar "citra jamur" dengan ukuran lebar 12 pixel dan panjang 9 pixel seperti terlihat pada Gambar 3.1.

x/y	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4												
5												
6												
7												
8												

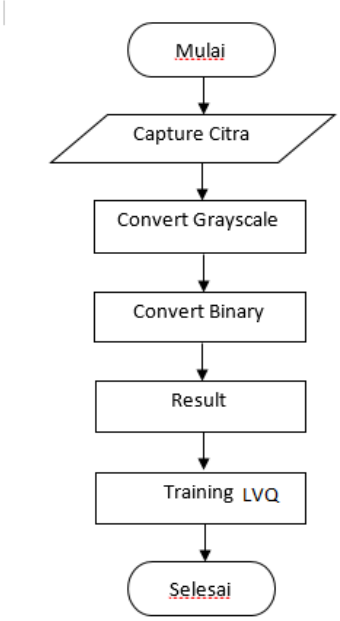


Gambar 3.1. Gambar citra jamur dengan ukuran lebar 12 pixel dan panjang 9 pixel

3.1.3 Analisis Sistem

Sistem yang akan dibangun adalah sebuah sistem pendeteksian pertumbuhan jamur yang bertujuan untuk mengimplementasikan Learning Vektor Quantization dalam mengenal pertumbuhan jamur dari suatu objek citra yang didapatkan dengan melakukan *capture* foto melalui kamera perangkat handphone android .

Citra inputan yang telah didapatkan oleh sistem handphone dengan ukuran 480 x 640 piksel. kemudian sistem mengubah citra jamur jadi grayscale. Setelah warna citra berhasil di rubah ke dalam citra grayscale , kemudian dilakukan pendeteksian tepi. Pendeteksian tepi dilakukan untuk memperjelas informasi warna yaitu batas-batas tepian objek berwarna yang dibutuhkan oleh pengguna sistem. Setelah melakukan tahap deteksi tepi dengan menggunaka operator canny yang berfungsi untuk binerisasi dan nilai biner tersebut di gunakan untuk pelatihan LVQ , lalu nilai nilai tersebut di training untuk mendapatkan hasil dari inputan citra jamur . Urutan proses pendeteksian jamur dapat dilihat pada gambar 3.2.



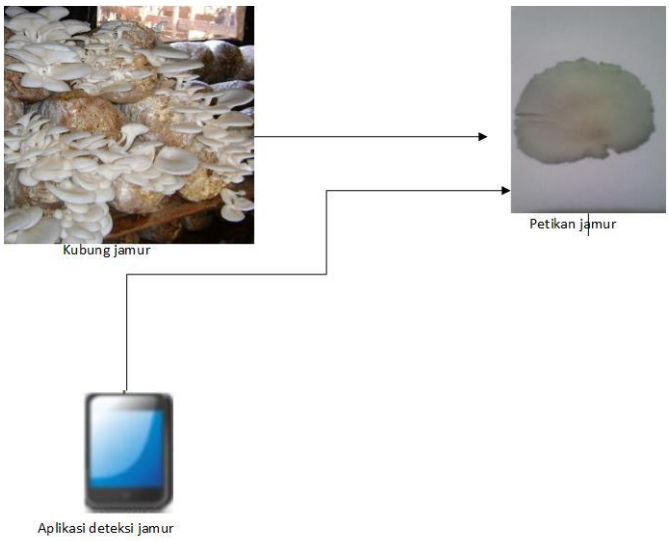
Gambar 3.2 Urutan proses pendeteksian jamur

3.1.4 Analisis Proses

Secara umum sistem ini dibagi menjadi beberapa proses. Setiap proses ini memiliki peran masing-masing dalam menjalankan proses untuk mengenali gambar. Proses-proses tersebut dapat dilihat pada Gambar 3.2 , Untuk mengetahui lebih detail dari setiap proses yang ada maka akan dijelaskan sebagai berikut::

A. Capture Citra

Proses pengambilan citra dilakukan yaitu dengan mengambil sehelai jamur lalu di letakkan di atas kertas buram , disini proses yang paling penting dikarenakan proses yang akan menghasilkan gambar yang sebagai media utama untuk penelitian selanjutnya



Gambar 3.3 proses pengambilan gambar jamur

Citra inputan dari sistem merupakan *file* citra yang dihasilkan melalui *capture* objek yang diambil menggunakan kamera perangkat hanphone terhadap citra yang telah terinstal oleh aplikasi pendeteksi jamur. Setelah citra didapatkan oleh sistem, citra tersebut akan dilakukan penskalaan atau *scaling* menjadi 480 x 640 piksel. *Scaling* ini dimaksudkan untuk mengurangi beban proses algoritma terhadap citra inputan.

B. Konversi citra asli ke 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 [3]

Proses yang pertama kali dilakukan merubah citra RGB menjadi citra grayscale. citra asli yang dikonversi ke citra grayscale dapat dilihat pada gambar

Color	Li	Lu	Avg	R	G	B	Li	Lu	Avg
				255	0	0	128	54	85
				255	153	0	128	162	136
				255	255	0	128	235	170
				0	255	0	128	181	85
				0	0	255	128	18	85
				255	0	255	128	71	170
				0	255	255	128	199	170
				128	128	128	128	127	128
				0	0	0	0	0	0

Gambar 3.4 Matrik pixel perubahan warna

dari color Red kita bisa dapat value : li (lightness) : Lu (luminosity) : dan avg (average)

Kemudian ketiga warna tersebut diolah berdasarkan persamaan berikut:

Grayscale = $\frac{R+G+B}{3}$

ini pseducode :Algorithm :

For Each Pixel in Image

{

Red = Pixel.

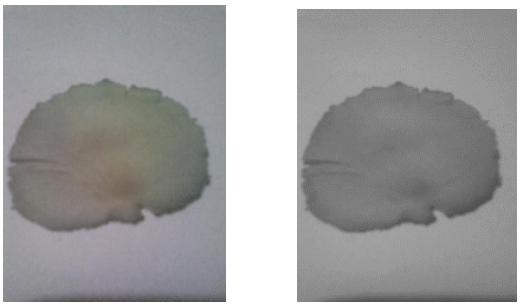
Red Green = Pixel.

```
 Green Blue = Pixel.
 Blue Gray = (Red + Green + Blue) / 3
 Pixel.Red = Gray
 Pixel.Green = Gray
  Pixel.Blue = Gray
}
```

implement algorithm : /*Inisialisasi*/

```
GS_RED = 0.299;
GS_GREEN = 0.587;
GS_BLUE = 0.114;
for (int x = 0; x < width; ++x)
        for (int y = 0; y < height; ++y)
pixel = bmpOriginal.get
Pixel(x, y) A = Color.alpha(pixel);
R = Color.red(pixel);
G = Color.green(pixel);
B = Color.blue(pixel);
R = G = B = (int)
(GS_RED * R + GS_GREEN * G + GS_BLUE * B);
```

Dari proses algoritma di atas maka hasilnya seperti gambar 3.5



a. Citra asli b. Grayscale

Gambar 3.5 Konversi citra asli ke grayscale

C. Deteksi Tepi

Citra yang telah dirubah menjadi bentuk grayscale kemudian diubah menjadi citra yang menghasilkan tepi objek yang jelas menggunkan deteksi tepi canny dan menghasilkan binerisasi dan nilai biner tersebut di gunakan untuk pelatihan LVQ .

ini pseducode :Algorithm :

Note : inputan gambar hasil yg telah GRAYCYCLE

For Each Pixel in Image

{

threshold = 127;

{nilai yang disesuaikan standar smooth}

int red = Color.red(pixel);

if (red < threshold)

cannyColor = hitam (R + G + B);

else cannyColor = hitam (R + G + B);

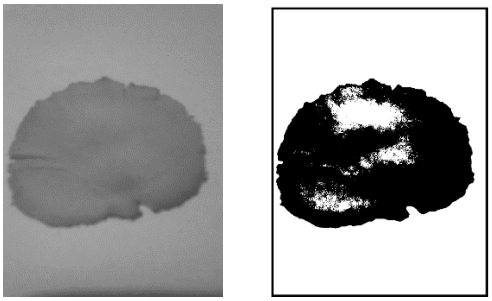
}

implement algorithm :

threshold = 127;

```
for (int x = 0; x < width; ++x)
        for (int y = 0; y < height; ++y)
pixel = bmpOriginal.getPixel(x, y);
red = Color.red(pixel);
        if (red < threshold) bmpBinary.setPixel(x, y, 0xFF000000);
        else
        bmpBinary.setPixel(x, y, 0xFFFFFFFF);
```

pada gambar 3.6 di bawah ini hasil citra grayscale yang telah difilter untuk mendapatkan deteksi tepi



a. Citra grayscale b. Deteksi Tepi Canny

Gambar 3.6 Konversi grayscale ke filter canny (deteksi tepi)

D. Proses Pembelajaran

Untuk analisis pembelajaran Learning Vector Quantization maka sebelumnya harus menentukan parameter-parameternya. Parameter-parameternya yaitu :

1. Prosedur Penentuan Masukan

   Jumlah masukkan ditentukan berdasarkan ukuran citra yang diinputkan ke dalam sistem, dalam simulasi ini citra berukuran 480 x 640 pixel
2. Prosedur Penentuan Parameter Pembelajaran

   Parameter pembelajaran yaitu maksimum epoch, Pengurangan Pembelajaran dan laju pembelajaran (alpha). Parameter ini akan berpengaruh pada kinerja sistem dalam mengenali citra.
3. Prosedur Penentuan Fungsi Bobot

   Penentuan fungsi bobot yang akan digunakan berdasarkan fungsi aktifasi. Penentuan nilai bobot minimum dan maksimum ditentukan atas dasar $0< n < 1$ yang merupakan jangkauan dari fungsi bipolar biner. Fungsi bipolar biner yaitu fungsi yang memiliki jarak atau range antara 1 dan 0. Fungsi ini akan membatasi pemberian nilai untuk setiap pembelajaran.
4. Menentukan keluaran

   Secara umum alur pelatihan dapat digambarkan sebagai berikut:

   a. Gambar mula-mula diinisialisasi

   b. Gambar diubah menjadi bentuk biner

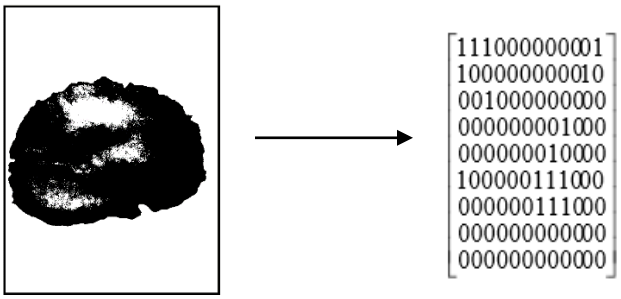
   c. Lakukan proses ekstraksi ciri

   Proses ekstraksi ciri yaitu proses dimana setiap pixel dari citra akan dirubah ke dalam biner 1 dan 0 sehingga terjadi keseragaman dalam menjalankan pembelajaran learning vector quantization.

   d. Lakukan pelatihan learning vector quantization pada gambar. Pelatihan learning vector quantization pada sistem ini, secara alur dapat digambarkan sebagai berikut :

1) Inisialisasikan vektor referensi dan learning rate(α).
2) Selama kondisi berhenti bernilai salah, kerjakan :
    a. Untuk masing-masing pelatihan vektor input
    b. Temukan J sehingga $\|X - W_j\|$bernilai minimum
    c. Perbaiki Wj dengan :
        i. Jika T=Cj maka
        Wj(baru) = Wj(lama) +α [x - Wj(lama)]
        ii. Jika T≠Cj maka
        Wj(baru) = Wj(lama) +α [x - Wj(lama)]
3) Kurangi learning rate
4) Tes kondisi berhenti.

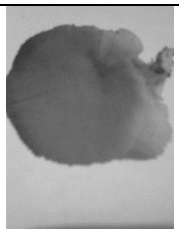
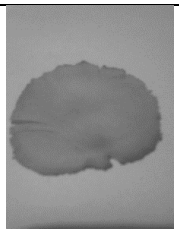
Secara proses, gambar yang ada akan dimasukan untuk pelatihan dijadikan biner agar gambar dapat dipelajari. Berikut ini ilustrasi atau gambaran jika gambar dijadikan biner dengan ukuran diperkecil 12x9 pixel. Gambar 3.7 adalah gambar yang dijadikan target pelatihan

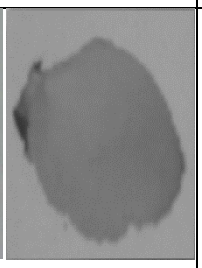


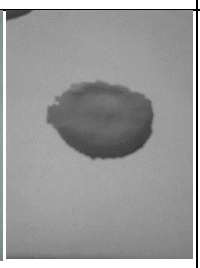
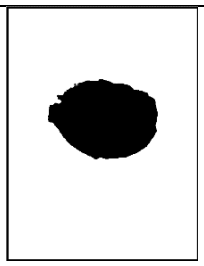
Gambar 3.7 Gambar pelatihan (Gambar ke biner)

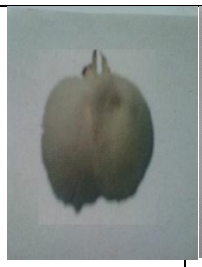
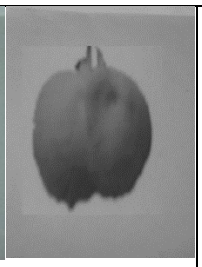
Berikut merupakan contoh pembelajaran LVQ pada gambar. Gambar yang akan dilakukan pembelajaran yaitu gambar-gambar yang memiliki kesamaan dengan gambar-gambar yang akan dikenali. Gambar-gambar yang akan dikenali sebagai bobot yaitu gambar yang memiliki parameter ukuran diameter jamur berbeda beda. pada matriks berukuran 12 x 9 seperti terlihat pada gambar 3.8 terdapat 10 inputan vector dengan 2 kelas dan jenis jamur siap panen dan belum siap panen.

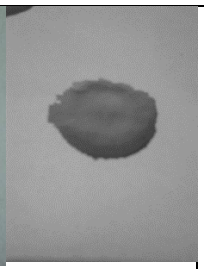
Tabel 3.1 Gambar yang akan di latih

No	RGB	Grayscale	Binary	Vektor	Jenis
1				0000000000100 0001000000011 0010010110010 0101100100111 1000001111000 0000110000000 1100000001100 1001010000000 0000	Siap Panen
2				0000000000111 1000001111000 0011100000010 1000000100000 0001000000001 0000000010000 0000101010000 1100001000000 0000	Siap Panen
3				0000000000000 0000000000000 0010000000000 0000000000000 0001000000000 0000000010000 0000111001100 1110001000000 0000	Siap Panen

4				0000000000111 1001001111000 0000010000010 0000000000000 0000000000000 0010000000000 0000100000000 0000000000000 0000	Siap Panen
5				0000000000100 0001000000011 0010010110010 0101100100111 1000001111000 0000110000000 1100000001100 1001010000000 0000	Siap Panen

No	RGB	Grayscale	Binary	Vektor	Jenis
1				0000000000000 0000000000000 0010000000000 0000000000000 0001000000000 0000000010000 0000111001100 1110001000000 0000	Belum siap panen

2				0000000000111 1001001111000 0000010000010 0000000001111 1000000000000 0010000000000 0000100000000 0000000000000 0000	Belum siap panen
3				0000000000111 0000001001000 0010000000010 0011000110000 0001110000001 1100000011100 0000101000000 0000011000000 0000	Belum siap panen
4				0000000000111 0000000000000 0001000010010 0010000100000 0001000000001 1000000011100 0000111000000 1110000000000 0000	Belum siap panen

5				0000000000111 1001001111000 0000010000010 1110000000000 0000000000000 0010000000000 0000100011000 0000000000000 0000	Belum siap panen

maka setiap gambar yang akan dilatih akan diubah menjadi satu array. Maka hasil yang ditampilkan dalam array yaitu:

P=[0000000000100000100000001100100101100100101100100111100000111100
0000011000000011000000011001001010000000000;

0000000000111100000111100000111000000101000000100000000010000000100
0000001000000001010100001100001000000000;

0000000000000000000000000000100000000000000000000000000001000000000000
0000001000000001110011001110001000000000;

0000000000111100100111100000000100000100000000000000000000000000000000
1000000000000001000000000000000000000000;

0000000000100000100000001100100101100100101100100111100000111100000
0011000000011000000011001001010000000000;

0000000000000000000000000000100000000000000000000000000001000000000000
0000001000000001110011001110001000000000;

0000000000111100100111100000000100000100000000001111100000000000000
1000000000000001000000000000000000000000;

000000000011100000010010000010000000010001100011000000011100000011100000010100000000000110000000000;

000000000011100000000000000010000100100010000100000000100000000110000001110000001110000001110000000000000;

000000000011110010011110000000010000010111000000000000000000000000010000000000000100011000000000000000000];

Dua input awal akan dijadikan sebagai inisialisasi bobot:

Table 3.2 inisialisasi bobot

Vektor (w)	**Jenis**	**Kelas**
1. 0000000000100000100000001100100101100100101100100111100000011110000000110000000011000000011000100100000	Siap Panen	1
2. 0000000000111100000111100000111000000101000000100000000010000000010000000010000000010101000011100000100000000	Belum siap panen	2

Sedangkan delapan input sisanya:

Table 3.3 inputan

Vektor (x)	**Jenis**	**Kelas**
1. 0000000000000000000000000000010000000000000000000000000010000000000000000010000000011100110011100100000000	Siap Panen	1

2.	0000000000111100100111100000000100000100000000000000000000000100000000000000010000000000000000000000000000	Belum siap panen	2
3.	0000000000100000100000001100100101100100101100100111100000011110000000110000000011000000011001001010000000000	Siap Panen	1
4.	0000000000000000000000000000100000000000000000000000000010000000000000000100000000111001100111000010000000000	Belum siap panen	2
5.	0000000000111100100111100000000100000100000000001111100000000000000010000000000000000100000000000000000000000	Siap Panen	1
6.	0000000000111000000100100000100000000100011000110000000111000000111000000111000000101000000000000110000000000	Belum siap panen	2
7.	0000000000111000000000000000010000100100010000010000000100000001100000001110000001110000001110000001100000000	Siap Panen	1
8.	0000000000111100100111100000000100000101110000000000000000000010000000000000001000110000000000000000000000	Belum siap panen	2

Sebagai nilai awal dipilih *learning rate* (α = 0,05), dengan pengurangan sebesar 0,1 * α dan maksimum epoh (MaxEpoh = 10). Berikut ini merupakan penghitungannya

Epoh ke-1:

Data ke-1

00000000000000000000000000001000000000000000000000000001000000000000
000 00100000000111001100111000100000000000

Jarak pada :

Bobot ke – 1

$$= \sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +$$

$$\sqrt{(0-0)^2 + (0-1)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 +$$

$$\sqrt{(0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +$$

$$\sqrt{(0-0)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 +$$

$$\sqrt{(0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 +$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (1-0)^2 + (1-0)^2 +}$$

$$\sqrt{(1-0)^2 + (0-0)^2 + (0-0)^2 + (1-1)^2 + (1-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$= 6$$

Bobot ke -2

$$= \sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 +}$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 +}$$

$$\sqrt{(0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-1)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-1)^2 + (1-0)^2 +}$$

$$\sqrt{(1-1)^2 + (0-0)^2 + (0-1)^2 + (1-0)^2 + (1-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-1)^2 + (1-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$= 4{,}47$

Jarak terkecil pada bobot ke – 2

Target data ke – 1 = 1

Bobot ke – 2 yang baru :

W21(baru) = w21(lama) + α [x11-w21(lama)]=0+0,05 * (0-0) = 0

W22(baru) = w22(lama) + α [x12-w22(lama)]=0+0,05 * (0-0) = 0

W23(baru) = w23(lama) + α [x13-w23(lama)]=0+0,05 * (0-0) = 0

W24(baru) = w24(lama) + α [x14-w24(lama)]=0+0,05 * (0-0) = 0

W25(baru) = w25(lama) + α [x15-w25(lama)]=0+0,05 * (0-0) = 0

W26(baru) = w26(lama) + α [x16-w26(lama)]=0+0,05 * (0-0) = 0

W27(baru) = w27(lama) + α [x17-w27(lama)]=0+0,05 * (0-0) = 0

W28(baru) = w28(lama) + α [x18-w28(lama)]=0+0,05 * (0-0) = 0

W29(baru) = w29(lama) + α [x19-w29(lama)]=0+0,05 * (0-0) = 0

W210(baru) = w210(lama) + α [x110-w210(lama)]=0+0,05 * (0-0) = 0

W211(baru) = w211(lama) + α [x111-w211(lama)]=1+0,05 * (0-1) = 0

W212(baru) = w212(lama) + α [x112-w212(lama)]=1+0,05 * (0-1) = 0

W213(baru) = w213(lama) + α [x113-w213(lama)]=1+0,05 * (0-1) = 0

W214(baru) = w214(lama) + α [x114-w214(lama)]=1+0,05 * (0-1) = 0

W215(baru) = w215(lama) + α [x115-w215(lama)]=0+0,05 * (0-0) = 0

W216(baru) = w216(lama) + α [x116-w216(lama)]=0+0,05 * (0-0) = 0

W217(baru) = w217(lama) + α [x117-w217(lama)]=0+0,05 * (0-0) = 0

W218(baru) = w218(lama) + α [x118-w218(lama)]=0+0,05 * (0-0) = 0

W219(baru) = w219(lama) + α [x119-w219(lama)]=0+0,05 * (0-0) = 0

W220(baru) = w20(lama) + α [x120-w220(lama)]=1+0,05 * (0-1) = - 1,05

W221(baru) = w221(lama) + α [x121-w221(lama)]=1+0,05 * (0-1) = - 1,05

W222(baru) = w22(lama) + α [x122-w222(lama)]=1+0,05 * (0-1) = - 1,05

W223(baru) = w223(lama) + α [x123-w223(lama)]=1+0,05 * (0-1) = - 1,05

W224(baru) = w224(lama) + α [x124-w224(lama)]=0+0,05 * (0-0) = 0

W225(baru) = w225(lama) + α [x125-w225(lama)]=0+0,05 * (0-0) = 0

W226(baru) = w226(lama) + α [x126-w226(lama)]=0+0,05 * (0-0) = 0

W227(baru) = w227(lama) + α [x127-w227(lama)]=0+0,05 * (0-0) = 0

W228(baru) = w228(lama) + α [x128-w228(lama)]=0+0,05 * (0-0) = 0

W229(baru) = w229(lama) + α [x129-w229(lama)]=1+0,05 * (1-1) = 0

W230(baru) = w230(lama) + α [x130-w230(lama)]=1+0,05 * (0-1) = 0

W231(baru) = w231(lama) + α [x131-w231(lama)]=1+0,05 * (0-1) = - 1,05

W232(baru) = w232(lama) + α [x132-w232(lama)]=0+0,05 * (0-0) = 0

W233(baru) = w233(lama) + α [x133-w233(lama)]=0+0,05 * (0-0) = 0

W234(baru) = w234(lama) + α [x134-w234(lama)]=0+0,05 * (0-0) = 0

W235(baru) = w235(lama) + α [x135-w235(lama)]=0+0,05 * (0-0) = 0

W236(baru) = w236(lama) + α [x136-w236(lama)]=0+0,05 * (0-0) = 0

W237(baru) = w237(lama) + α [x137-w237(lama)]=0+0,05 * (0-0) = 0

W238(baru) = w238(lama) + α [x138-w238(lama)]=1+0,05 * (0-1) = - 1,05

W239(baru) = w239(lama) + α [x139-w239(lama)]=0+0,05 * (0-0) = 0

W240(baru) = w240(lama) + α [x140-w240(lama)]=1+0,05 * (0-1) = - 1,05

W241(baru) = w241(lama) + α [x141-w241(lama)]=0+0,05 * (0-0) = 0

W242(baru) = w242(lama) + α [x142-w242(lama)]=0+0,05 * (0-0) = 0

W243(baru) = w243(lama) + α [x143-w243(lama)]=0+0,05 * (0-0) = 0

W244(baru) = w244(lama) + α [x144-w244(lama)]=0+0,05 * (0-0) = 0

W245(baru) = w245(lama) + α [x145-w245(lama)]=0+0,05 * (0-0) = 0

W246(baru) = w246(lama) + α [x146-w246(lama)]=0+0,05 * (0-0) = 0

W247(baru) = w247(lama) + α [x147-w247(lama)]=1+0,05 * (0-1) = - 1,05

W248(baru) = w248(lama) + α [x148-w248(lama)]=0+0,05 * (0-0) = 0

W249(baru) = w249(lama) + α [x149-w249(lama)]=0+0,05 * (0-0) = 0

W250(baru) = w250(lama) + α [x150-w250(lama)]=0+0,05 * (0-0) = 0

W251(baru) = w251(lama) + α [x151-w251(lama)]=0+0,05 * (0-0) = 0

W252(baru) = w252(lama) + α [x152-w252(lama)]=0+0,05 * (0-0) = 0

W253(baru) = w253(lama) + α [x153-w253(lama)]=0+0,05 * (0-0) = 0

W254(baru) = w254(lama) + α [x154-w254(lama)]=0+0,05 * (0-0) = 0

W255(baru) = w255(lama) + α [x155-w255(lama)]=0+0,05 * (0-0) = 0

W256(baru) = w256(lama) + α [x156-w256(lama)]=1+0,05 * (1-1) = 0

W257(baru) = w57(lama) + α [x157-w257(lama)]=0+0,05 * (0-0) = 0

W258(baru) = w58(lama) + α [x158-w258(lama)]=0+0,05 * (0-0) = 0

W259(baru) = w59(lama) + α [x159-w259(lama)]=0+0,05 * (0-0) = 0

W260(baru) = w260(lama) + α [x160-w260(lama)]=0+0,05 * (0-0) = 0

W261(baru) = w261(lama) + α [x161-w261(lama)]=0+0,05 * (0-0) = 0

W262(baru) = w262(lama) + α [x162-w262(lama)]=0+0,05 * (0-0) = 0

W263(baru) = w263(lama) + α [x163-w263(lama)]=0+0,05 * (0-0) = 0

W264(baru) = w264(lama) + α [x164-w264(lama)]=0+0,05 * (0-0) = 0

W265(baru) = w265(lama) + α [x165-w265(lama)]=1+0,05 * (0-1) = - 1,05

W266(baru) = w266(lama) + α [x166-w266(lama)]=0+0,05 * (0-0) = 0

W267(baru) = w267(lama) + α [x167-w267(lama)]=0+0,05 * (0-0) = 0

W268(baru) = w268(lama) + α [x168-w268(lama)]=0+0,05 * (0-0) = 0

W269(baru) = w269(lama) + α [x169-w269(lama)]=0+0,05 * (0-0) = 0

W270(baru) = w270(lama) + α [x170-w270(lama)]=0+0,05 * (0-0) = 0

W271(baru) = w271(lama) + α [x171-w271(lama)]=0+0,05 * (0-0) = 0

W272(baru) = w272(lama) + α [x172-w272(lama)]=0+0,05 * (0-0) = 0

W273(baru) = w273(lama) + α [x173-w273(lama)]=0+0,05 * (0-0) = 0

W274(baru) = w274(lama) + α [x174-w274(lama)]=1+0,05 * (1-1) = 0

W275(baru) = w275(lama) + α [x175-w275(lama)]=0+0,05 * (0-0) = 0

W276(baru) = w276(lama) + α [x176-w276(lama)]=0+0,05 * (0-0) = 0

W277(baru) = w277(lama) + α [x177-w277(lama)]=0+0,05 * (0-0) = 0

W278(baru) = w278(lama) + α [x178-w278(lama)]=0+0,05 * (0-0) = 0

W279(baru) = w279(lama) + α [x179-w279(lama)]=0+0,05 * (0-0) = 0

W280(baru) = w280(lama) + α [x180-w280(lama)]=0+0,05 * (0-0) = 0

W281(baru) = w281(lama) + α [x181-w281(lama)]=0+0,05 * (0-0) = 0

W282(baru) = w282(lama) + α [x182-w282(lama)]=0+0,05 * (0-0) = 0

W283(baru) = w283(lama) + α [x183-w283(lama)]=1+0,05 * (1-1) = 0

W284(baru) = w284(lama) + α [x184-w284(lama)]=0+0,05 * (1-0) = 0,05

W285(baru) = w285(lama) + α [x185-w285(lama)]=1+0,05 * (1-1) = 0

W286(baru) = w286(lama) + α [x186-w286(lama)]=0+0,05 * (0-0) = 0

W287(baru) = w287(lama) + α [x187-w287(lama)]=1+0,05 * (0-1) = - 1,05

W288(baru) = w288(lama) + α [x188-w288(lama)]=0+0,05 * (1-0) = 0,05

W289(baru) = w289(lama) + α [x189-w289(lama)]=0+0,05 * (1-0) = 0,05

W290(baru) = w290(lama) + α [x190-w290(lama)]=0+0,05 * (0-0) = 0

W291(baru) = w291(lama) + α [x191-w291(lama)]=0+0,05 * (0-0) = 0

W292(baru) = w292(lama) + α [x192-w292(lama)]=1+0,05 * (1-1) = 0

W293(baru) = w293(lama) + α [x193-w293(lama)]=1+0,05 * (1-1) = 0

W294(baru) = w294(lama) + α [x194-w294(lama)]=0+0,05 * (1-0) = 0,05

W295(baru) = w295(lama) + α [x195-w295(lama)]=0+0,05 * (0-0) = 0

W296(baru) = w296(lama) + α [x196-w296(lama)]=0+0,05 * (0-0) = 0

W297(baru) = w297(lama) + α [x197-w297(lama)]=0+0,05 * (0-0) = 0

W298(baru) = w298(lama) + α [x198-w298(lama)]=1+0,05 * (1-1) = 0

W299(baru) = w299(lama) + α [x199-w299(lama)]=0+0,05 * (0-0) = 0

W2100(baru) = w2100(lama) + α [x1100-w2100(lama)]=0+0,05 * (0-0) = 0

W2101(baru) = w2101(lama) + α [x1101-w2101(lama)]=0+0,05 * (0-0) = 0

W2102(baru) = w2102(lama) + α [x1102-w2102(lama)]=0+0,05 * (0-0) = 0

W2103(baru) = w2103(lama) + α [x1103-w2103(lama)]=0+0,05 * (0-0) = 0

W2104(baru) = w2104(lama) + α [x1104-w2104(lama)]=0+0,05 * (0-0) = 0

W2105(baru) = w2105(lama) + α [x1105-w2105(lama)]=0+0,05 * (0-0) = 0

W2106(baru) = w2106(lama) + α [x106-w2106(lama)]=0+0,05 * (0-0) = 0

W2107(baru) = w2107(lama) + α [x1107-w2107(lama)]=0+0,05 * (0-0) = 0

W2108(baru) = w2108(lama) + α [x1108-w2108(lama)]=0+0,05 * (0-0) = 0

Sehingga W2 = (0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1,05 -1,05 -1,05 -1,05 0 0 0 0 0 0 0 -1,05 0 0 0 0 0 0 -1,05 0 -1,05 0 0 0 0 0 0 -1,05 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1,05 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0,05 0 0 -1,05 0,05 0,05 0 0 0 0 0,05 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)

W1 = (0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 1 0 0 1 0 1 1 0 0 1 0 0 1 1 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0)

Data ke – 2

0000000000111100100111100000000100000100000000000000000000000000001000 0000000000010000000000000000000000000

Jarak pada bobot ke – 1

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-1)^2 + (1-0)^2 +}$$

$$\sqrt{(1-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (1-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 + (0-1)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 +}$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (1-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 +}$$

$$\sqrt{(0-1)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$= 6$

Bobot ke – 2

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-0)^2 + (1-0)^2 +}$$

$$\sqrt{(1-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (1-0)^2 + (0-0)^2 +}$$

$$\sqrt{\begin{matrix}(0-0)^2 + \left(1-(-1{,}05)\right)^2 + \left(1-(-1{,}05)\right)^2 + \left(1-(-1{,}05)\right)^2 + \left(1-(-1{,}05)\right)^2 \\ +(0-0)^2 +\end{matrix}}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-(-1{,}05)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-(-1{,}05)^2 + (0-0)^2 + (0-(-1{,}05)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-(-1{,}05))^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0))^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-(-1{,}05))^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-0)^2 + (0-0{,}05)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-(-1{,}05))^2 + (0-0{,}05)^2 + (0-0{,}05)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0{,}05)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$\sqrt{(0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 +}$$

$$= 5{,}50$$

Jarak terkecil pada bobot ke – 2

Target data ke – 2 = 2

Bobot ke – 2 yang baru :

W21(baru) = w21(lama) + α [x21-w21(lama)]=0+0,05 * (0-0) = 0

W22(baru) = w22(lama) + α [x22-w22(lama)]=0+0,05 * (0-0) = 0

W23(baru) = w23(lama) + α [x23-w23(lama)]=0+0,05 * (0-0) = 0

W24(baru) = w24(lama) + α [x24-w24(lama)]=0+0,05 * (0-0) = 0

W25(baru) = w25(lama) + α [x25-w25(lama)]=0+0,05 * (0-0) = 0

W26(baru) = w26(lama) + α [x26-w26(lama)]=0+0,05 * (0-0) = 0

W27(baru) = w27(lama) + α [x27-w27(lama)]=0+0,05 * (0-0) = 0

W28(baru) = w28(lama) + α [x28-w28(lama)]=0+0,05 * (0-0) = 0

W29(baru) = w29(lama) + α [x29-w29(lama)]=0+0,05 * (0-0) = 0

W210(baru) = w210(lama) + α [x210-w210(lama)]=0+0,05 * (0-0) = 0

W211(baru) = w211(lama) + α [x211-w211(lama)]=0+0,05 * (1-0) = 0,05

W212(baru) = w212(lama) + α [x212-w212(lama)]=0+0,05 * (1-0) = 0,05

W213(baru) = w213(lama) + α [x213-w213(lama)]=0+0,05 * (1-0) = 0,05

W214(baru) = w214(lama) + α [x214-w214(lama)]=0+0,05 * (1-0) = 0,05

W215(baru) = w215(lama) + α [x215-w215(lama)]=0+0,05 * (0-0) = 0

W216(baru) = w216(lama) + α [x216-w216(lama)]=0+0,05 * (0-0) = 0

W217(baru) = w217(lama) + α [x217-w217(lama)]=0,05 +0,05 * (1-0,05) = 0,095

W218(baru) = w218(lama) + α [x218-w218(lama)]=0+0,05 * (0-0) = 0

W219(baru) = w219(lama) + α [x219-w219(lama)]=0+0,05 * (0-0) = 0

W220(baru) = w20(lama) + α [x220-w220(lama)]=0+0,05 * (1-0) = 0,05

W221(baru) = w221(lama) + α [x221-w221(lama)]=0+0,05 * (1-0) = 0,05

W222(baru) = w22(lama) + α [x222-w222(lama)]=0+0,05 * (1-0) = 0,05

W223(baru) = w223(lama) + α [x223-w223(lama)]=0+0,05 * (1-0) = 0,05

W224(baru) = w224(lama) + α [x224-w224(lama)]=0+0,05 * (0-0) = 0

W225(baru) = w225(lama) + α [x225-w225(lama)]=0+0,05 * (0-0) = 0

W226(baru) = w226(lama) + α [x226-w226(lama)]=0+0,05 * (0-0) = 0

W227(baru) = w227(lama) + α [x227-w227(lama)]=0+0,05 * (0-0) = 0

W228(baru) = w228(lama) + α [x228-w228(lama)]=0+0,05 * (0-0) = 0

W229(baru) = w229(lama) + α [x229-w229(lama)]= -1,05 +0,05 * (0- (-1,05)) = 1,05

W230(baru) = w230(lama) + α [x230-w230(lama)]= -1,05+0,05 * (0-(-1,05)) = 1,05

W231(baru) = w231(lama) + α [x231-w231(lama)]= -1,05+0,05 * (0-(-1,05)) = 1,05

W232(baru) = w232(lama) + α [x232-w232(lama)]=0,05 +0,05 * (1-0,05) = 0,095

W233(baru) = w233(lama) + α [x233-w233(lama)]=0+0,05 * (0-0) = 0

W234(baru) = w234(lama) + α [x234-w234(lama)]=0+0,05 * (0-0) = 0

W235(baru) = w235(lama) + α [x235-w235(lama)]=0+0,05 * (0-0) = 0

W236(baru) = w236(lama) + α [x236-w236(lama)]=0+0,05 * (0-0) = 0

W237(baru) = w237(lama) + α [x237-w237(lama)]=0+0,05 * (0-0) = 0

W238(baru) = w238(lama) + α [x238-w238(lama)]=0+0,05 * (1-0) = 0,05

W239(baru) = w239(lama) + α [x239-w239(lama)]=0+0,05 * (0-0) = 0

W240(baru) = w240(lama) + α [x240-w240(lama)]= -1,05 +0,05 * (0-(-1,05)) = 1,05

W241(baru) = w241(lama) + α [x241-w241(lama)]=0+0,05 * (0-0) = 0

W242(baru) = w242(lama) + α [x242-w242(lama)]=0+0,05 * (0-0) = 0

W243(baru) = w243(lama) + α [x243-w243(lama)]=0+0,05 * (0-0) = 0

W244(baru) = w244(lama) + α [x244-w244(lama)]=0+0,05 * (0-0) = 0

W245(baru) = w245(lama) + α [x245-w245(lama)]=0+0,05 * (0-0) = 0

W246(baru) = w246(lama) + α [x246-w246(lama)]=0+0,05 * (0-0) = 0

W247(baru) = w247(lama) + α [x247-w247(lama)]= -1,05 +0,05 * (0-(-1,05)) = 1,05

W248(baru) = w248(lama) + α [x248-w248(lama)]=0+0,05 * (0-0) = 0

W249(baru) = w249(lama) + α [x249-w249(lama)]=0+0,05 * (0-0) = 0

W250(baru) = w250(lama) + α [x250-w250(lama)]=0+0,05 * (0-0) = 0

W251(baru) = w251(lama) + α [x251-w251(lama)]=0+0,05 * (0-0) = 0

W252(baru) = w252(lama) + α [x252-w252(lama)]=0+0,05 * (0-0) = 0

W253(baru) = w253(lama) + α [x253-w253(lama)]=0+0,05 * (0-0) = 0

W254(baru) = w254(lama) + α [x254-w254(lama)]=0+0,05 * (0-0) = 0

W255(baru) = w255(lama) + α [x255-w255(lama)]=0+0,05 * (0-0) = 0

W256(baru) = w256(lama) + α [x256-w256(lama)]= - 1,05 +0,05 * (0-(-1,05)) = 1,05

W257(baru) = w57(lama) + α [x257-w257(lama)]=0+0,05 * (0-0) = 0

W258(baru) = w58(lama) + α [x258-w258(lama)]=0+0,05 * (0-0) = 0

W259(baru) = w59(lama) + α [x259-w259(lama)]=0+0,05 * (0-0) = 0

W260(baru) = w260(lama) + α [x260-w260(lama)]=0+0,05 * (0-0) = 0

W261(baru) = w261(lama) + α [x261-w261(lama)]=0+0,05 * (0-0) = 0

W262(baru) = w262(lama) + α [x262-w262(lama)]=0+0,05 * (0-0) = 0

W263(baru) = w263(lama) + α [x263-w263(lama)]=0+0,05 * (0-0) = 0

W264(baru) = w264(lama) + α [x264-w264(lama)]=0+0,05 * (0-0) = 0

W265(baru) = w265(lama) + α [x265-w265(lama)]= -1,05 +0,05 * (0-(-1,05)) = 1,05

W266(baru) = w266(lama) + α [x266-w266(lama)]=0+0,05 * (0-0) = 0

W267(baru) = w267(lama) + α [x267-w267(lama)]=0+0,05 * (0-0) = 0

W268(baru) = w268(lama) + α [x268-w268(lama)]=0,05+0,05 * (1-0,05) = 0,095

W269(baru) = w269(lama) + α [x269-w269(lama)]=0+0,05 * (0-0) = 0

W270(baru) = w270(lama) + α [x270-w270(lama)]=0+0,05 * (0-0) = 0

W271(baru) = w271(lama) + α [x271-w271(lama)]=0+0,05 * (0-0) = 0

W272(baru) = w272(lama) + α [x272-w272(lama)]=0+0,05 * (0-0) = 0

W273(baru) = w273(lama) + α [x273-w273(lama)]=0+0,05 * (0-0) = 0

W274(baru) = w274(lama) + α [x274-w274(lama)]= -1,05 +0,05 * (0-(-1,05)) = 1,05

W275(baru) = w275(lama) + α [x275-w275(lama)]=0+0,05 * (0-0) = 0

W276(baru) = w276(lama) + α [x276-w276(lama)]=0+0,05 * (0-0) = 0

W277(baru) = w277(lama) + α [x277-w277(lama)]=0+0,05 * (0-0) = 0

W278(baru) = w278(lama) + α [x278-w278(lama)]=0+0,05 * (0-0) = 0

W279(baru) = w279(lama) + α [x279-w279(lama)]=0+0,05 * (0-0) = 0

W280(baru) = w280(lama) + α [x280-w280(lama)]=0+0,05 * (0-0) = 0

W281(baru) = w281(lama) + α [x281-w281(lama)]=0+0,05 * (0-0) = 0

W282(baru) = w282(lama) + α [x282-w282(lama)]=0+0,05 * (0-0) = 0

W283(baru) = w283(lama) + α [x283-w283(lama)]=0+0,05 * (1-0) = 0,05

W284(baru) = w284(lama) + α [x284-w284(lama)]=0+0,05 * (0-0) = 0

W285(baru) = w285(lama) + α [x285-w285(lama)]= -1,05 +0,05 * (0-(-1,05)) = 1,05

W286(baru) = w286(lama) + α [x286-w286(lama)]=0+0,05 * (0-0) = 0

W287(baru) = w287(lama) + α [x287-w287(lama)]= -1,05 +0,05 * (0-(-1,05)) = 1,05

W288(baru) = w288(lama) + α [x288-w288(lama)]=0+0,05 * (0-0) = 0

W289(baru) = w289(lama) + α [x289-w289(lama)]=0+0,05 * (0-0) = 0

W290(baru) = w290(lama) + α [x290-w290(lama)]=0+0,05 * (0-0) = 0

W291(baru) = w291(lama) + α [x291-w291(lama)]=0+0,05 * (0-0) = 0

W292(baru) = w292(lama) + α [x292-w292(lama)]= -1,05+0,05 * (0-(-1,05)) = 1,05

W293(baru) = w293(lama) + α [x293-w293(lama)]= -1,05+0,05 * (0-(-1,05)) = 1,05

W294(baru) = w294(lama) + α [x294-w294(lama)]=0+0,05 * (0-0) = 0

W295(baru) = w295(lama) + α [x295-w295(lama)]=0+0,05 * (0-0) = 0

W296(baru) = w296(lama) + α [x296-w296(lama)]=0+0,05 * (0-0) = 0

W297(baru) = w297(lama) + α [x297-w297(lama)]=0+0,05 * (0-0) = 0

W298(baru) = w298(lama) + α [x298-w298(lama)]= -1,05 +0,05 * (0-(-1,05)) = 1,05

W299(baru) = w299(lama) + α [x299-w299(lama)]=0+0,05 * (0-0) = 0

W2100(baru) = w2100(lama) + α [x2100-w2100(lama)]=0+0,05 * (0-0) = 0

W2101(baru) = w2101(lama) + α [x2101-w2101(lama)]=0+0,05 * (0-0) = 0

W2102(baru) = w2102(lama) + α [x2102-w2102(lama)]=0+0,05 * (0-0) = 0

W2103(baru) = w2103(lama) + α [x2103-w2103(lama)]=0+0,05 * (0-0) = 0

W2104(baru) = w2104(lama) + α [x2104-w2104(lama)]=0+0,05 * (0-0) = 0

W2105(baru) = w2105(lama) + α [x2105-w2105(lama)]=0+0,05 * (0-0) = 0

W2106(baru) = w2106(lama) + α [x2106-w2106(lama)]=0+0,05 * (0-0) = 0

W2107(baru) = w2107(lama) + α [x2107-w2107(lama)]=0+0,05 * (0-0) = 0

W2108(baru) = w2108(lama) + α [x2108-w2108(lama)]=0+0,05 * (0-0) = 0

Sehingga W2 : (0 0 0 0 0 0 0 0 0 0 0,05 0,05 0,05 0,05 0 0 0,095 0 0 0,05 0,05 0,05 0,05 0 0 0 0 0 1,05 1,05 1,05 0,095 0 0 0 0 0 0,05 0 1,05 0 0 0 0 0 0 1,05 0 0 0 0 0 0 0 0 1,05 0 0 0,095 0 0 0 0 0 1,05 0 0 0 0 0 0 0 0 0,05 0 1,05 0 1,05 0 0 0 0 1,05 1,05 0 0 0 0 1,05 0 0 0 0 0 0 0 0 0 0)

Proses tersebut diteruskan untuk epoh ke-2 sampai ke-4, untuk setiap data dengan menggunakan cara yang sama. Setelah mencapai epoh yang ke-4 diperoleh bobot akhir:

W1 = (0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 ..... 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 .....

0.9999 0.9999 -0.0328 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999.....

0.9999 -0.0328 -0.0328 -0.0328 -0.0328 -0.0328 -0.0328 -0.0328 -0.0328...

-0.0328 -0.0328 -0.0328 0.9999 -0.1186 -0.0328 0.9999 0.0000 0.0000 0.0000..... .....)

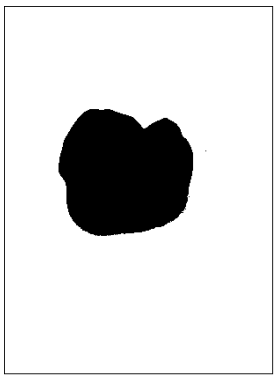
W2 = (0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 ..... 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 .....

0.9999 0.9999 -0.0328 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999 0.9999.....

0.9999 -0.2251 -0.0328 -0.0328 -0.0328 -0.0328 -0.6298 -0.2836 -0.0328...

-0.0328 -0.0328 -0.4294 0.4452 -0.1186 -0.0328 0.9999 0.0000 0.0000 0.0000..... .....)

Apabila kita ingin mensimulasikan



(000000000011100000010010000010000000010000000011000000011100000011100 000 001110000001010000000000011000000000) maka kita cari terlebih dahulu jarak input tersebut dengan kedua bobot. Nomor dari bobot dengan jarak yang terkecil akan menjadi kelasnya.

Jarak pada:

Bobot ke – 1 = 18,5

Bobot ke – 2 = 10

Jarak terkecil pada bobot ke-2 Sehingga input tersebut termasuk dalam kelas belum siap panen

3.2 Analisis Kebutuhan Data Non Fungsional

Analisis kebutuhan non fungsional ini merupakan pendukung sistem yang akan dijalankan. Adapun kebutuhan-kebutuhannya yaitu:

3.2.1 Kebutuhan Perangkat Keras

Komputer dan mobile phone terdiri dari perangkat keras dan perangkat lunak yang saling berinteraksi. Perangkat lunak memberikan instruksi-instruksi kepada perangkat keras untuk melakukan suatu tugas tertentu, sehingga dapat menjalankan suatu sistem di dalamnya. Adapun spesifikasi yang digunakan dalam penelitian ini yaitu:

1. Perangkat Keras

a. Processor : Intel Pentium Core 3 Duo @ 2.2 GHz

b. Memory : 512 MB

c. VGA : 256 MB

d. Hardisk : 180 GB

e. Monitor : 17 inch

f. Mouse dan Keyboard

3.2.2 Kebutuhan Perangkat Pembangun

Perangkat lunak digunakan dalam sebuah sistem merupakan perintah-perintah yang diberikan kepada perangkat keras agar bisa saling berinteraksi diantara keduanya. Perangkat lunak yang dibutuhkan untuk membangun aplikasi ini adalah sebagai berikut :

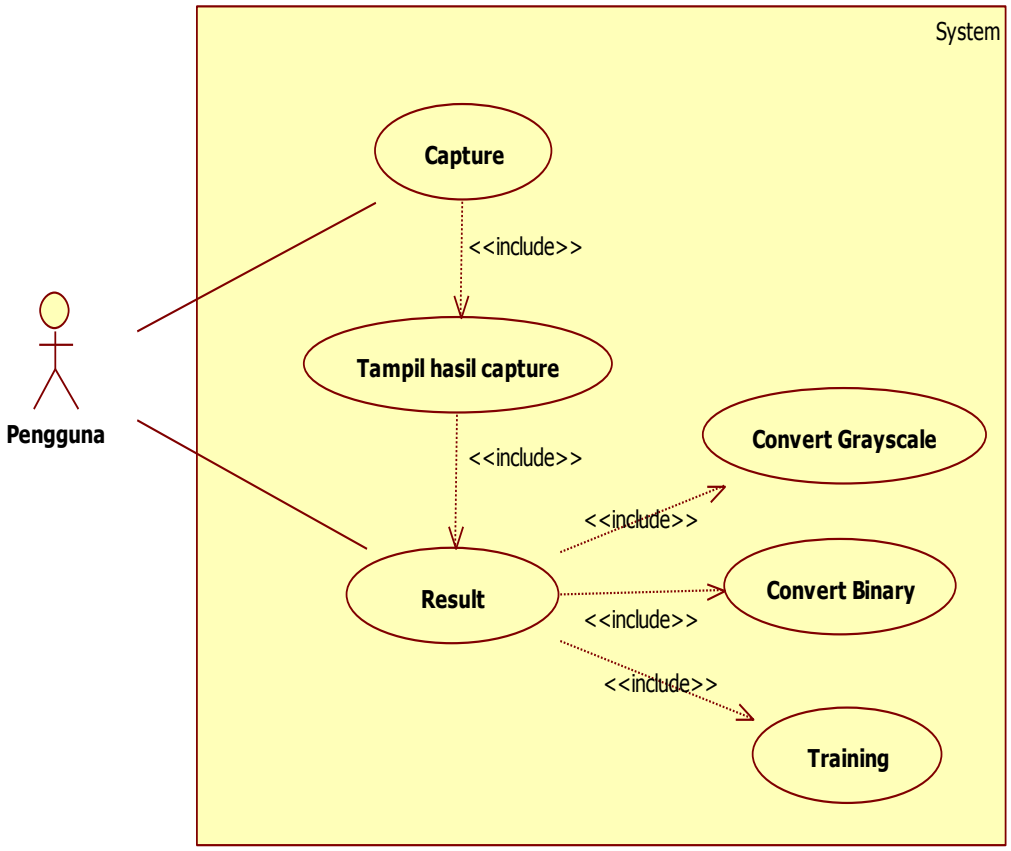
1. Sistem Operasi Windows 8
2. Eclips
3. Android sdk

3.3 Pemodelan Sistem

Pemodelan sistem berfungsi untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem pada aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

A. Use Case Diagram

Use case adalah konstruksi untuk mendeskripsikan bagaimana sistem terlihat di mata pengguna. Sasaran pemodelan use case diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan .



Gambar 3.8 UseCase Diagram

A. Identifikasi Use Case

Berikut tabel identifikasi *use case* yang terdapat dalam sistem :

Tabel 3.4 Identifikasi *Use Case*

***Use Case* Pendeteksi jamur**		
No	*Use Case*	Deskripsi
UC-01	Capture	Fungsionalitas untuk mengambil citra jamur yang dilakukan oleh user
UC-02	Tampil hasil capture	Fungsionalitas untuk menampilkan hasil dari capture
UC-03	Result	Fungsional untuk pengolahan citra
UC-04	Convert Grayscale	Fungsionalitas untuk merubah citra RGB menjadi hitam putih
UC-05	Convert Binary	Fungsionalitas dari citra grayscale ke deteksi tepi dan akan di rubah menjadi biner yang akan di jadikan pelatihan LVQ
UC-06	Training	Hasil pembelajaran LVQ

B. Skenario Use Case

Skenario Use Case berfungsi untuk menjelaskan lebih detail mengenai modul-modul yang terdapat dalam Aplikasi deteksi jamur ini. Adapun penjelasan mengenai keterangan tabel Skenario Use Case adalah sebagai berikut :

Tabel 3.5 Skenario Use Case Capture

Nama Use Case	Capture
Deskripsi	Untuk memulai mengambil objek
Aktor	Pengguna
Kondisi Awal	Kamera handphone siap
Kondisi Akhir	System menampilkan hasil pengolahan citra
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengguna berada pada tampilan utama sistem pendeteksi jamur	
2. Pengguna memilih menu Capture	3. System menjalankan camera yang ada pada handphone android

Table 3.6 Skenario Use case Tampil Hasil Capture

Nama Use Case	Tampil Hasil Capture
Deskripsi	Menampilkan hasil capture
Aktor	Pengguna
Kondisi Awal	Hasil capture belum ada
Kondisi Akhir	System menampilkan hasil capture
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pengguna berada pada tampilan utama sistem pendeteksi jamur	
2. Pengguna memilih menu Capture	3. System menjalankan camera yang ada pada handphone android
	4. Tampil hasil capture

Table 3.7 Skenario Use case Result

Nama Use Case	Result
Deskripsi	Untuk proses pengolahan citra
Aktor	Pengguna
Kondisi Awal	citra jamur
Kondisi Akhir	System menampilkan hasil pembelajaran LVQ
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Tampil hasil capture
2. Pilih Tombol Result	3. Proses Pengolahan Citra

Tabel 3.8 Skenario Use Case Convert Grayscale

Nama Use Case	Convert Grayscale
Deskripsi	Untuk proses pengolahan citra Grayscale
Aktor	Sistem
Kondisi Awal	citra jamur sudah ada
Kondisi Akhir	System menampilkan citra jamur Grayscale
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1.Sistem memproses citra asli
	2. citra jamur asli atau RGB akan diolah menjadi grayscale
	3. Tampil Citra jamur grayscale

Tabel 3.9 Skenario Use Case Convert Binary

Nama Use Case	Convert Binary
Deskripsi	Proses deteksi tepi yang hasilnya berupa binerisasi dari citra

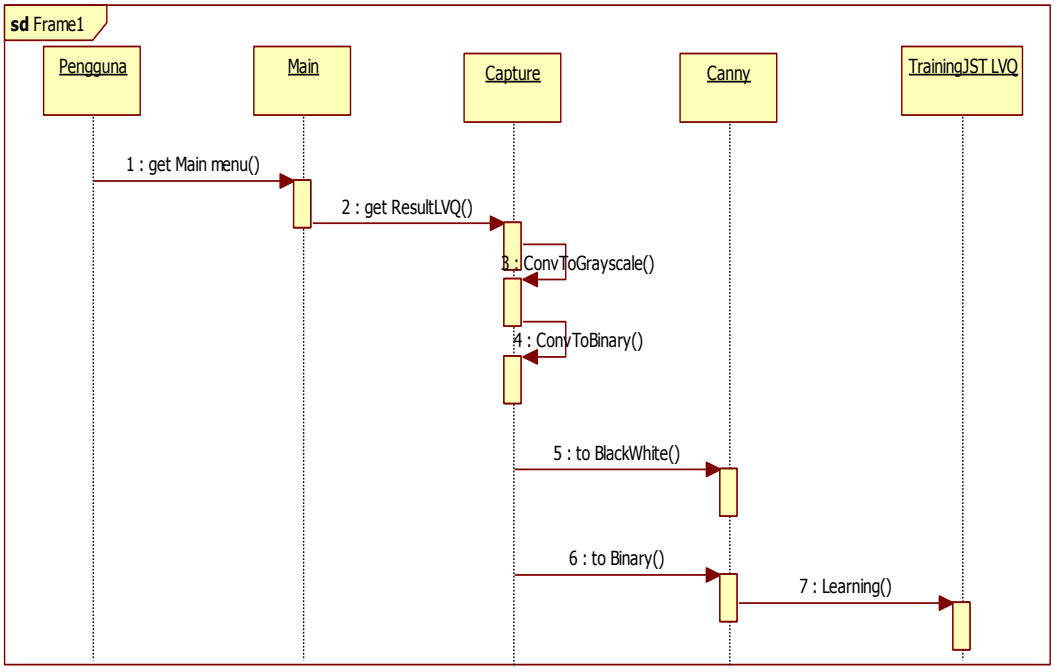
Aktor	Sistem
Kondisi Awal	citra jamur grayscale
Kondisi Akhir	System menampilkan citra jamur yang sudah di deteksi tepi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1.Sistem memproses gambar
	2. citra jamur Grayscale akan diolah menggunakan deteksi tepi canny
	3. Tampil Citra Jamur dengan hasil deteksi tepi

Tabel 3.10 Skenario Use Case Training

Nama Use Case	Training
Deskripsi	Hasil dari proses pengolahan citra yang akan di jadikan pelatihan
Aktor	Sistem
Kondisi Awal	
Kondisi Akhir	System menampilkan hasil pembelajaran
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Sistem memproses gambar
	2. citra jamur Grayscale akan diolah menggunakan deteksi tepi canny
	3. Tampil Citra Jamur dengan hasil deteksi tepi binerisasi yang akan di jadi kan pembelajaran LVQ
	4. Hasil pembelajaran LVQ

C. Sequance Diagram

Sebuah diagram yang menggambarkan interaksi antar objek di dalam sebuah system. Interaksi tersebut berupa message yang digambarkan terhadap waktu. Sequance diagram terdiri dari dimensi horizontal (objek-objek) dan dimensi vertical (waktu) . dimensi vertikal adalah poros waktu, dimana waktu berjalan ke arah bawah, sedangkan dimensi horizontal merepresentasikan objek-objek individual.Tiap objek (termasuk *actor*) tersebut mempunyai waktu aktif yang direpresentasikan dengan kolom vertikal yang disebut dengan *lifeline.*

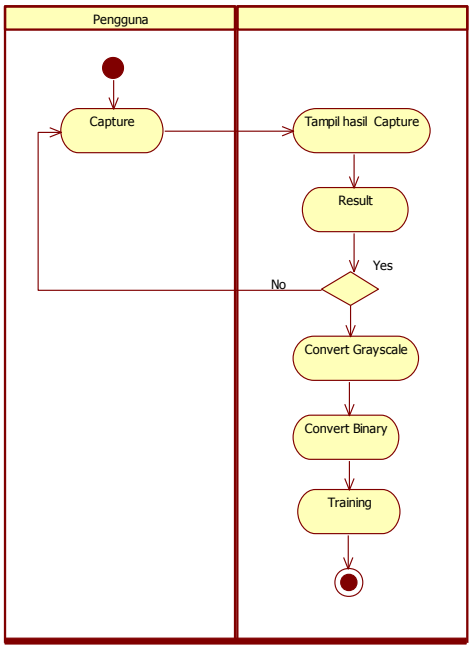


Gambar 3.9 Squance Diagram

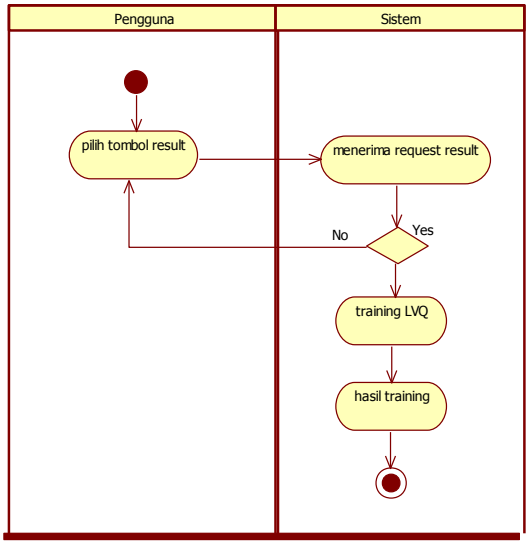
D. Activity Diagram

Diagram activity menggambarkan berbagai alir aktivitas dalam sistem yang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Penggambaran *activity* diagram memiliki kemiripan dengan *flowchart* diagram*. *Activity diagram* memodelkan *event-event*

yang terjadi pada *Use Case* dan digunakan untuk pemodelan aspek dinamis dari system



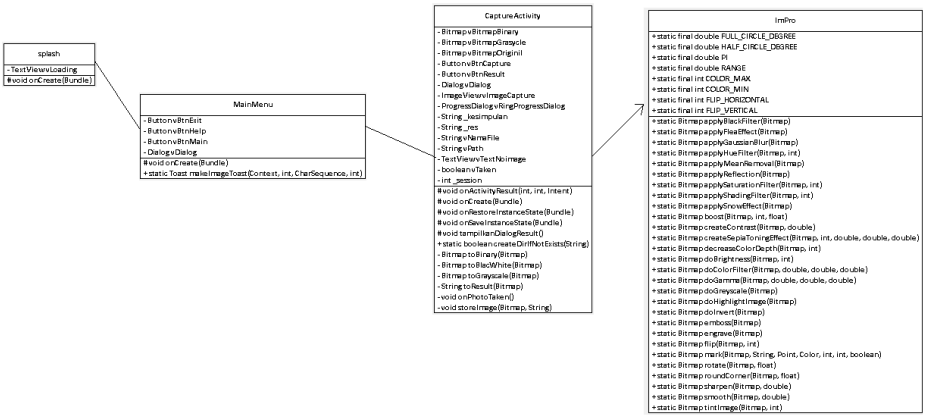
Gambar 3.10 Activity Diagram



Gambar 3.11 Activity Diagram Result

E. Class Diagram

Diagram ini ditujukan untuk mendefinisikan proses yang berjalan pada aplikasi pendeteksian jamur ini.



Gambar 3.12 Class Diagram tahapan implementasi

Tabel 3.11 keterangan *class diagram*

Name	*Type*	Fungsi
Class Splash	*Boundary*	*Class interface* ini memiliki fungsionalitas *preview screen* awal
Class Main	*Boundary*	Class interface ini memiliki fungsionalitas untuk menampilkan menu utama dengan *method-method* getMessageHelp(), getCaptureMenu() dan getExit
Class Capture	*Boundary*	Merupakan *class interface* inti yang memiliki fungsionalitas

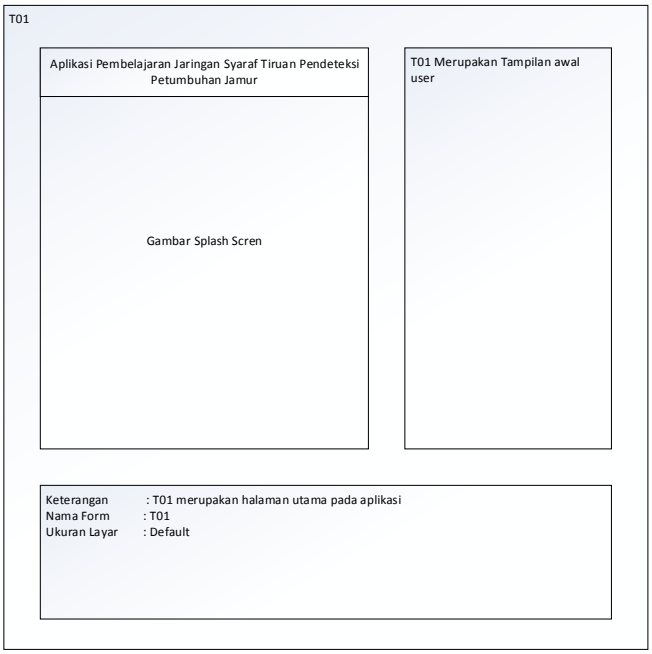
		utama mengambil gambar dari camera yang kemudian merealisasikan algoritma control *canny detection*
Class Cany	*Control*	*Class control* yang berfungsi sebagai trigger image processing kedalam sub-proses yang lebih rinci

F. Perancangan Antarmuka

Perancangan antarmuka dibagi menjadi dua bagian, yaitu perancangan antarmuka input dan perancangan antarmuka output.

a. Rancangan Antarmuka Halaman Utama

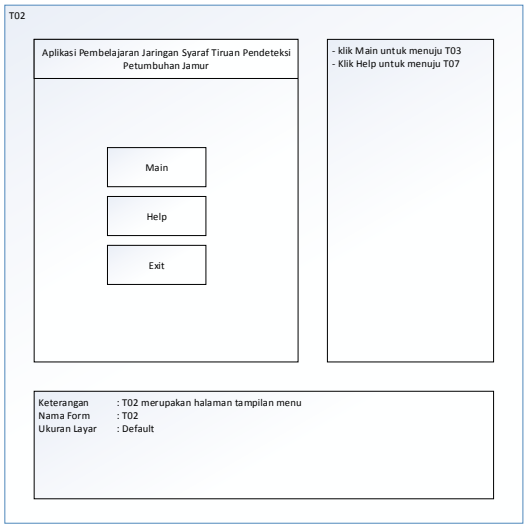
Pada rancangan antarmuka halaman utama terdapat *splash scren* dari aplikasi .Rancangan antarmuka dapat dilihat pada gambar 3.13



Gambar 3.13 Rancangan Halaman utama

b. Rancangan Antarmuka Tampilan Menu

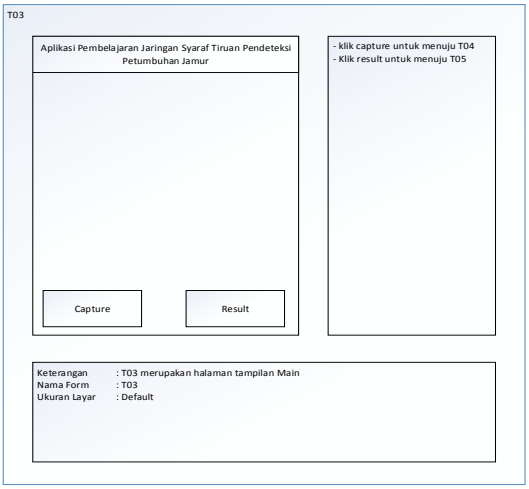
Rancangan antarmuka tampilan menu terdapat menu menu yang dapat di pilih .Rancangan antar muka dapat dilihat pada gambar 3.14



Gambar 3.14 Rancangan antar muka tampilan menu

c. Rancangan Antarmuka Main

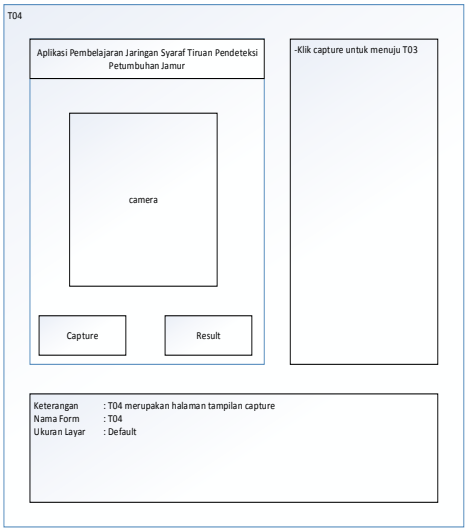
Rancangan antarmuka Main digunakan untuk memulai proses capture citra. Rancangan antarmuka dapat dilihat pada gambar 3.15



Gambar 3.15 Rancangan antarmuka Main

d. Rancangan Antarmuka Capture

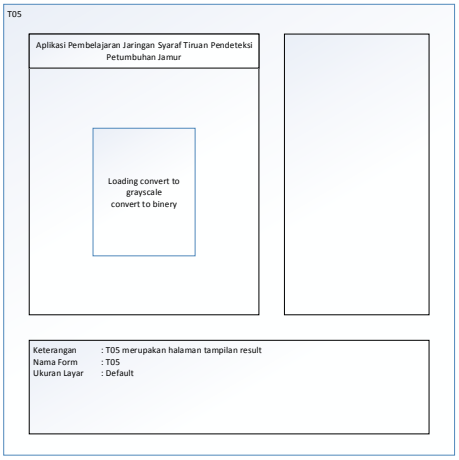
Rancangan antarmuka capture digunakan untuk mengcapture citra . Rancangan antarmuka dapat dilihat pada gambar 3.16



Gambar 3.16 Rancangan antarmuka Capture

e. Rancangan Antarmuka Result

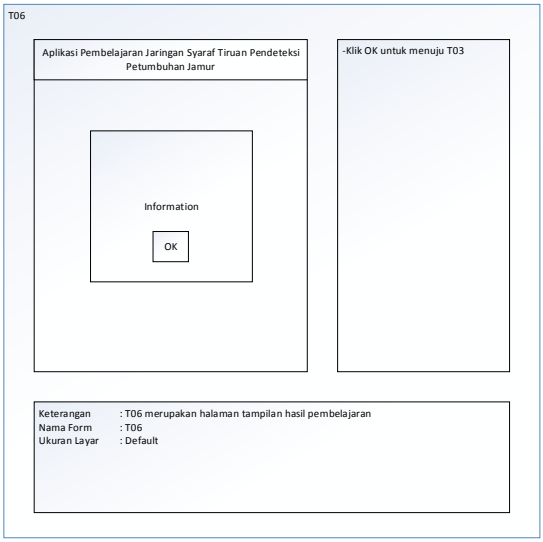
Rancangan antarmuka result digunakan untuk mengolah citra RGB ke convert sampai binery.Rancangan antarmuka dapat dilihat pada gambar 3.17



Gambar 3.17 Rancangan antarmuka Result

f. Rancangan Antarmuka Hasil Pembelajaran

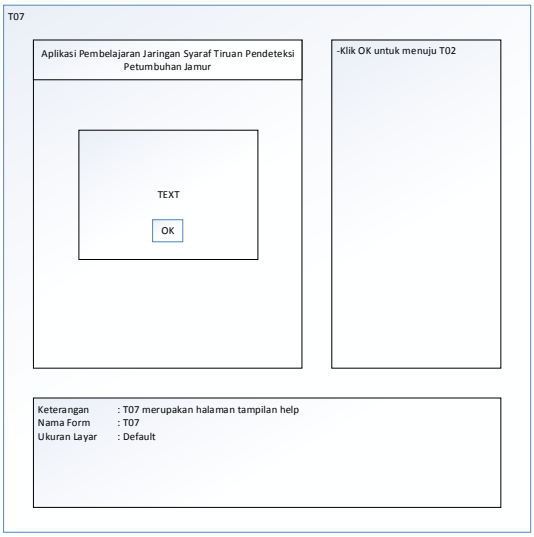
Rancangan Antarmuka Hasil pembelajaran dapat dilihat pada gambar 3.18



Gambar 3.18 Rancangan antarmuka Pembelajaran

g. Rancangan Antarmuka Help

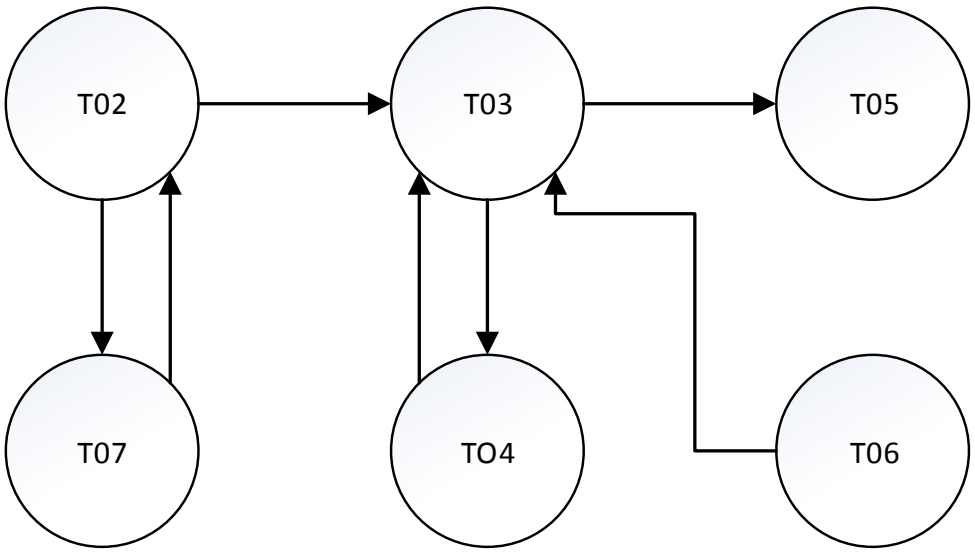
Rancangan antarmuka Help digunakan untuk menerima informasi tentang jamur. Rancnagan antarmuka dapat di lihat pada gambar 3.19



Gambar 3.19 Rancangan antarmuka Help

G. Jaringan Semantik

Jaringan semantik sistem yang akan dibangun dibedakan sesuai dengan jenis penggunanya. Jaringan semantiknya dapat dilihat pada gambar di bawah ini :



Gambar 3.19 Jaringan semantik