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.

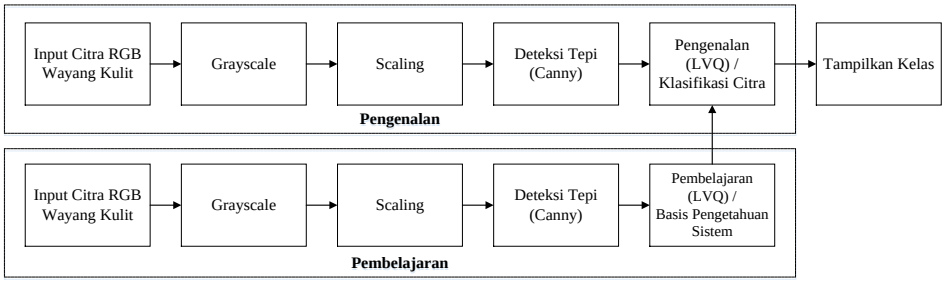
3.1.1. Analisis Masalah

Analisis masalah dilakukan untuk mengetahui masalah apa saja yang terjadi pada pembangunan program. Saat ini sudah banyak pengenalan pola citra yang dilakukan baik dari pengenalan pola citra tanda tangan, huruf, batik dan wayang kulit. Meskipun demikian, masih terdapat masalah dalam melakukan identifikasi citra. Masalah yang terjadi dalam kasus ini adalah kemiripan pola citra, pada faktor citra itu sendiri terdapat beberapa hal yang mempengaruhi baik buruknya hasil dari identifikasi citra, diantaranya adalah kualitas citra yang kurang baik, pola citra yang mirip dan ukuran citra yang tidak sama. Untuk kualitas citra dan ukuran citra berbeda, dengan menggunakan *Scaling*, apakah akurasi dan waktunya akan menurun atau jauh lebih baik. Sedangkan untuk pola yang sama atau kemiripan citra, penelitian ini menggunakan klasifikasi LVQ untuk memperoleh nilai error yang kecil dan mencari hasil akhir melalui perbandingan perubahan ukuran atau skala citra untuk setiap tahap pengujiannya, sehingga akan diketahui apakah LVQ dapat mengidentifikasi citra wayang kulit pandawa 5 dengan baik atau tidak.

3.1.2. Deskripsi Umum Sistem

Metode *Learning Vector Quantization* memiliki langkah-langkah yang sama dengan metode lainnya dalam sistem pengenalan. Data *learning* (data yang

digunakan untuk proses pembelajaran) terlebih dahulu harus dipelajari untuk mendapatkan bobot yang kemudian disimpan kedalam *file* penyimpanan dan dapat digunakan pada proses pengenalan atau uji (data yang akan dikenali). Gambaran sistem secara umum dapat dilihat pada Gambar 3.1.



Gambar 3.1 Gambaran Umum Sistem[8]

3.1.3. Proses Pengolahan Citra

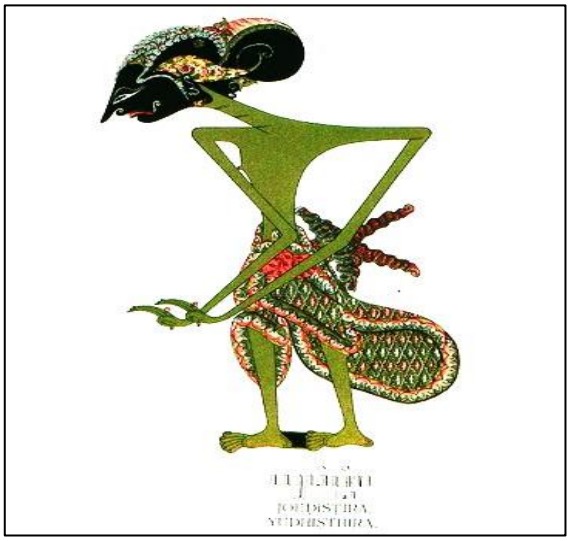
Proses pengolahan citra ini merupakan suatu proses awal yang dijalankan ketika melakukan penambahan data. Proses akan dimulai dari *Input* citra RGB, *Grayscale*, *Scaling* dan *edgedetection* terlebih dahulu sebelum dimasukkan kedalam penyimpanan. Proses pengolahan citra ini akan dilakukan sebelum masuk pada proses pembelajaran dan pengenalan. Sehingga untuk proses pengolahan citra ini akan diwakili oleh salah satu citra masukkan, karena pada dasarnya proeses pengolahan citra memiliki tahapan yang sama. Sebagai contoh perhitungannya akan digunakan proses pengolahan citra untuk tahap pembelajaran.

3.1.3.1. *Input* Citra RGB

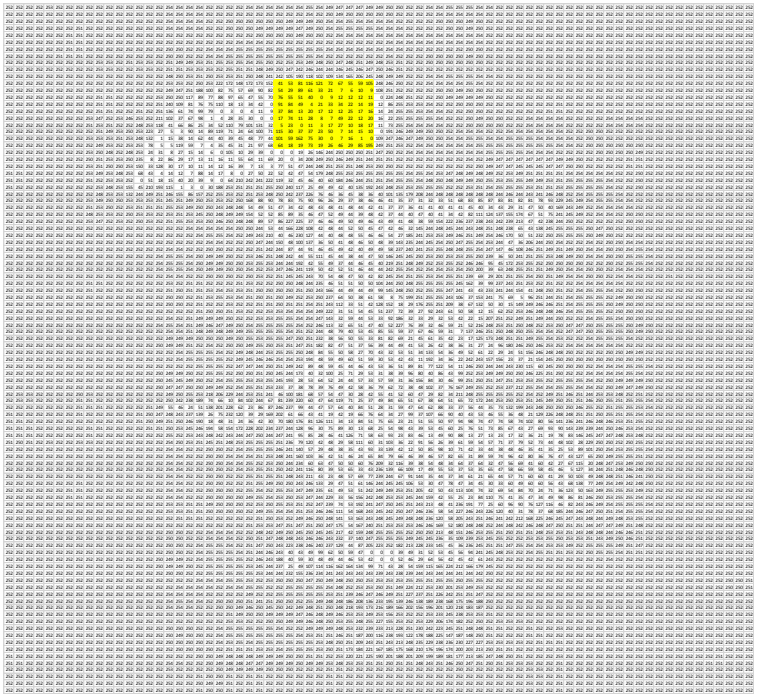
Setiap proses pada pengolahan citra menggunakan nilai *Pixel* dari citra untuk dilakukan perhitungan. Setiap *Pixel* pada citra terdiri dari 3 unsur warna dasar yaitu *Red* (R), *Green* (G) dan *Blue* (B), yang sering disingkat RGB dan setiap unsur memiliki nilai antara 0-255 yang menjadi kode kedalaman warna dari masing-masing unsur. Citra masukkan di sistem merupakan citra wayang kulit Yudistira yang akan digunakan untuk proses pembelajaran dan pengenalan. Citra *Input*-an berasal dari sistem direktori komputer pengguna, seperti pada gambar 3.2 – 3.5

a. *Input* Citra Pembelajaran

Pada gambar dibawah ini merupakan *Input* citra RGB untuk proses pembelajaran.



Gambar 3.2 Citra RGB Yudistira Pembelajaran



Gambar 3.3 Nilai *Pixel* Citra Yudistira Pembelajaran

Citra awal yang dimasukan kedalan analisis ini adalah gambar dengan ukuran 250 x 250 *Pixel* yang artinya terdiri dari 250 kolom dan 250 baris *Pixel*. Sehingga total keseluruhannya adalah 62500 *Pixel*, maka dari itu diambil sample matriks 10x10 untuk memudahkan perhitungan dari matriks citra dibawah ini, dimana nilai-nilai *Pixel* ini diambil menggunakan aplikasi *Matlab*. seperti yang terlihat pada Tabel 3.1.

Tabel 3.1 Nilai RGB Sampel Citra Pembelajaran Wayang Dengan *Pixel* 10x10

x/y	0	1	2	3	4	5	6	7	8	9
0	67	88	124	164	164	98	85	70	73	117
1	76	55	120	90	59	35	15	14	16	15
2	91	72	68	58	10	15	15	15	15	12
3	92	88	57	12	27	35	35	24	15	18
4	33	80	13	24	23	15	11	26	17	14
5	16	70	11	35	17	12	50	21	10	18
6	18	27	5	25	17	23	28	11	19	17
7	151	55	59	58	40	57	8	16	19	14
8	140	88	187	99	48	7	9	20	7	0
9	86	37	37	95	38	37	50	34	90	199

x/y	0	1	2	3	4	5	6	7	8	9
0	41	53	81	116	121	72	67	55	59	105
1	54	29	89	61	33	21	7	6	10	9
2	76	55	51	40	0	9	12	12	12	11
3	91	84	49	4	21	33	34	22	14	19
4	37	84	13	20	17	12	12	25	17	16
5	17	74	11	28	8	7	49	22	12	20
6	5	23	0	11	3	17	27	10	18	17
7	115	30	37	37	23	50	7	14	15	10
8	101	59	162	75	30	0	7	16	1	0
9	64	18	19	73	19	26	46	29	85	195

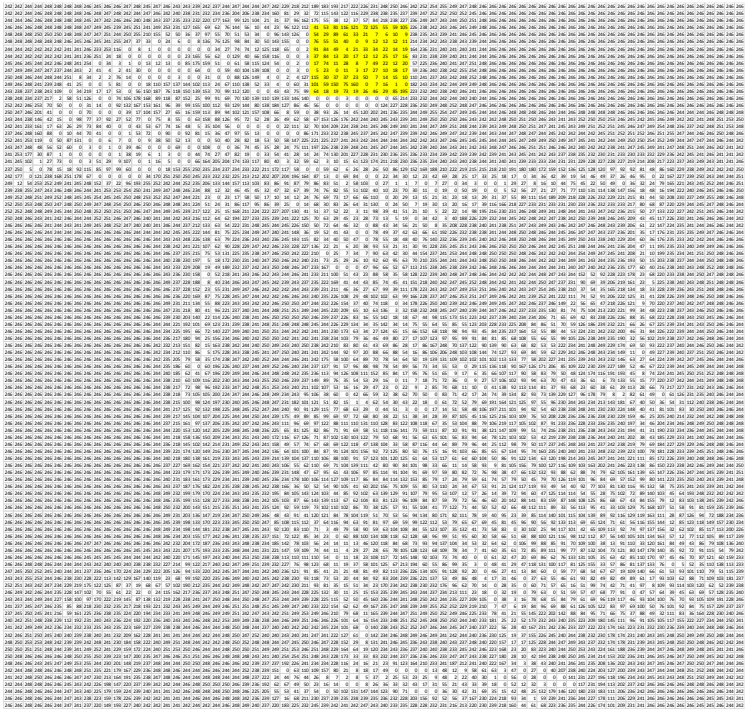
x/y	0	1	2	3	4	5	6	7	8	9
0	55	66	95	128	135	86	82	68	71	114
1	66	46	109	82	52	33	15	14	15	14
2	87	70	69	61	14	18	17	17	17	14
3	97	94	60	15	33	40	42	29	20	23
4	42	89	20	27	26	23	21	33	24	20
5	21	79	18	36	19	20	60	31	20	24
6	19	33	8	23	15	29	38	18	26	24
7	148	53	56	50	33	58	15	21	25	20
8	136	87	185	94	45	8	14	23	10	3
9	88	39	40	97	40	39	53	35	91	200

b. *Input* Citra Pengenalan

Pada gambar dibawah ini merupakan *Input* citra RGB untuk proses Pengenalan.



Gambar 3.4 Citra RGB Yudistira Pengenalan



Gambar 3.5 Nilai *Pixel* Citra Yudistira Pengenalan

Berikut sampel nilai citra RGB yang akan digunakan untuk proses pengenalan yang terdiri dari *Pixel* 10 x 10, dapat dilihat pada table 3.2 berikut.

Tabel 3.2 Nilai RGB Sampel Citra Pengenalan Wayang Dengan *Pixel* 10 x 10

x/y	0	1	2	3	4	5	6	7	8	9
0	67	88	124	164	164	98	85	70	73	117
1	76	55	120	90	59	35	15	14	16	15
2	91	72	68	58	10	15	15	15	15	12
3	92	88	57	12	27	35	35	24	15	18
4	33	80	13	24	23	15	11	26	17	14
5	16	70	11	35	17	12	50	21	10	18
6	18	27	5	25	17	23	28	11	19	17
7	151	55	59	58	40	57	8	16	19	14
8	140	88	187	99	48	7	9	20	7	0
9	86	37	37	95	38	37	50	34	90	199

x/y	0	1	2	3	4	5	6	7	8	9
0	41	53	81	116	121	72	67	55	59	105
1	54	29	89	61	33	21	7	6	10	9
2	76	55	51	40	0	9	12	12	12	11
3	91	84	49	4	21	33	34	22	14	19
4	37	84	13	20	17	12	12	25	17	16
5	17	74	11	28	8	7	49	22	12	20
6	5	23	0	11	3	17	27	10	18	17
7	115	30	37	37	23	50	7	14	15	10
8	101	59	162	75	30	0	7	16	1	0
9	64	18	19	73	19	26	46	29	85	195

x/y	0	1	2	3	4	5	6	7	8	9
0	55	66	95	128	135	86	82	68	71	114
1	66	46	109	82	52	33	15	14	15	14
2	87	70	69	61	14	18	17	17	17	14
3	97	94	60	15	33	40	42	29	20	23
4	42	89	20	27	26	23	21	33	24	20
5	21	79	18	36	19	20	60	31	20	24
6	19	33	8	23	15	29	38	18	26	24
7	148	53	56	50	33	58	15	21	25	20
8	136	87	185	94	45	8	14	23	10	3
9	88	39	40	97	40	39	53	35	91	200

3.1.3.2.*Scaling*

Scaling merupakan proses mengubah ukuran gambar, hal ini perlu dilakukan agar semua gambar digital data masukan memiliki ukuran yang sama. Berikut merupakan langkah-langkah proses *Scaling* :

a. Pertama tentukan tujuan ukuran yang di inginkan, disini sebgai contohnya akan dilakukan proses *Scaling* dengan ukuran panjang 5 *Pixel* dan lebar 5 *Pixel*.

b. Hitung skala perubahan panjang dan lebar dengan cara membagi panjang atau lebar awal dengan panjang atau lebar tujuan.

$$nwidthfactor = \frac{width}{nwidth} = \frac{10}{5} = 2$$

$$nheightfactor = \frac{height}{nheight} = \frac{10}{5} = 2$$

c. Untuk setiap matriks gambar hasil, dilakukan perhitungan d sampai g

d. Tentukan kordinat *Pixel* disekitar matriks yang sedang diulang, x dan y menunjukkan kordinat yang sedang dihitung. Berikut ini perhitungan untuk kordinat 0,0.

$$fr_x = x * nwidthfactor = 0 * 2 = 0$$

$$fr_y = y * nheightfactor = 0 * 2 = 0$$

$$C_x = fr_x + 1 = 0 + 1 = 1$$

$$C_y = fr_y + 1 = 0 + 1 = 1$$

$$f_x = x * nwidthfactor - fr_x = 0 - 0 = 0$$

$$f_y = y * nheightfactor - fr_y = 0 - 0 = 0$$

$$n_x = 1{,}0 * f_x = 1{,}0 - 0 = 1{,}0$$

$$n_y = 1{,}0 * f_y = 1{,}0 - 0 = 1{,}0$$

e. Mengambil nilai *Pixel* pada 4 koordinat dengan menggunakan persamaan berikut :

$$C1 = pixel(fr_x, fr_y) = pixel(0{,}0)$$

$$C2 = pixel(C_x, fr_y) = pixel(1{,}0)$$

$$C3 = pixel(fr_x, C_y) = pixel(0{,}1)$$

$$C4 = pixel(C_x, C_y) = pixel(1{,}1)$$

Nilai RGB untuk *Pixel* pada C1, C2, C3 dan C4 adalah sebagai berikut:

	Red	***Green***	***Blue***
C1	54	54	54
C2	69	69	69
C3	65	65	65
C4	43	43	43

f. Menghitung nilai RGB baru dengan menggunakan nilai RGB dari keempat koordinat yang sudah diambil dengan menggunakan persamaan berikut :

Red :

$b1 = (n_x * C1.R + f_x * C2.R) = (1{,}0 * 54 + 0 * 69) = 54$

$b2 = (n_x * C3.R + f_x * C4.R) = (1{,}0 * 65 + 0 * 43) = 65$

$R = (n_y * b1 + f_y * b2) = (1{,}0 * 54 + 0 * 65) = 54$

***Green* :**

$b1 = (n_x * C1.G + f_x * C2.G) = (1{,}0 * 54 + 0 * 69) = 54$

$b2 = (n_x * C3.G + f_x * C4.G) = (1{,}0 * 65 + 0 * 43) = 65$

$G = (n_y * b1 + f_y * b2) = (1{,}0 * 54 + 0 * 65) = 54$

***Blue* :**

$b1 = (n_x * C1.B + f_x * C2.B) = (1{,}0 * 54 + 0 * 69) = 54$

$b2 = (n_x * C3.B + f_x * C4.B) = (1{,}0 * 65 + 0 * 43) = 65$

$G = (n_y * b1 + f_y * b2) = (1{,}0 * 54 + 0 * 65) = 54$

g. Set nilai *Pixel* pada kordinat yang sedang diulang dengan nilai RGB hasil dari perhitungan yang diperoleh. Sehingga *Pixel*(0,0) memiliki nilai R=54, G=54 dan B=54

Hasil akhir dari perhitungan *Scaling* untuk proses pengenalan dan pembelajaran pada citra wayang kulit Yudistira dengan ukuran lebar 10 *Pixel* dan panjang 10 *Pixel* akan berubah menjadi citra berukuran lebar 5 *Pixel* dan panjang 5 *Pixel* seperti terlihat pada Tabel 3.5 - 3.6 dan gambar 3.8 – 3.9.

a. Hasil *Scaling* Citra Pembelajaran

Pada gambar 3.5 dibawah ini merupakan citra *Scaling* wayang kulit Yudistira yang akan digunakan dalam proses pembelajaran,

Tabel 3.3 Citra Pembelajaran Wayang Kulit Yudistira Dengan Nilai *Pixel* 5x5

x/y	0	1	2	3	4
0	54	100	140	78	68
1	85	63	8	15	15
2	37	15	22	15	19
3	14	4	12	31	21
4	126	178	41	10	6



Gambar 3.6 Hasil *Scaling* Citra Wayang Kulit Pembelajaran

b. Hasil *Scaling* Citra Pengenalan

Pada gambar 3.9 dibawah ini merupakan citra *Scaling* wayang kulit Yudistira yang akan digunakan dalam proses pengenalan.

Tabel 3.4 Citra Pengenalan Wayang Kulit Yudistira Dengan Nilai *Pixel* 5x5

x/y	0	1	2	3	4
0	46	97	139	171	48
2	86	63	8	15	15
4	37	15	22	15	14
6	12	3	12	31	21
8	119	170	151	10	4



Gambar 3.7 Hasil *Scaling* Citra Wayang Kulit Pengenalan

3.1.3.3.Grayscale

Grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap *Pixel*-nya, dengan kata lain nilai bagian *RED* = *GREEN* = *BLUE*. Dimana yang sering dikenal dengan citra yang memiliki warna keabu-abuan. Pada contoh kasus ini yang akan menjadi data masukannya adalah gambar citra RGB yang sebelumnya sudah di masukkan (Tabel 3.1). Berikut ini langkah-langkah proses mengubah citra RGB menjadi citra *Grayscale*.

Nilai RGB untuk setiap *Pixel* pada citra wayang kulit Yudistira yang akan diubah menjadi keabu-abuan ditunjukkan pada nilai-nilai *Pixel* berikut ini. Berikut adalah tahapan proeses *Grayscale* dari citra pembelajaran.

x/y	0	1	2	3	4	5	6	7	8	9
0	67	88	124	164	164	98	85	70	73	117
1	76	55	120	90	59	35	15	14	16	15
2	91	72	68	58	10	15	15	15	15	12
3	92	88	57	12	27	35	35	24	15	18
4	33	80	13	24	23	15	11	26	17	14
5	16	70	11	35	17	12	50	21	10	18
6	18	27	5	25	17	23	28	11	19	17
7	151	55	59	58	40	57	8	16	19	14
8	140	88	187	99	48	7	9	20	7	0
9	86	37	37	95	38	37	50	34	90	199

x/y	0	1	2	3	4	5	6	7	8	9
0	41	53	81	116	121	72	67	55	59	105
1	54	29	89	61	33	21	7	6	10	9
2	76	55	51	40	0	9	12	12	12	11
3	91	84	49	4	21	33	34	22	14	19
4	37	84	13	20	17	12	12	25	17	16
5	17	74	11	28	8	7	49	22	12	20
6	5	23	0	11	3	17	27	10	18	17
7	115	30	37	37	23	50	7	14	15	10
8	101	59	162	75	30	0	7	16	1	0
9	64	18	19	73	19	26	46	29	85	195

x/y	0	1	2	3	4	5	6	7	8	9
0	55	66	95	128	135	86	82	68	71	114
1	66	46	109	82	52	33	15	14	15	14
2	87	70	69	61	14	18	17	17	17	14
3	97	94	60	15	33	40	42	29	20	23
4	42	89	20	27	26	23	21	33	24	20
5	21	79	18	36	19	20	60	31	20	24
6	19	33	8	23	15	29	38	18	26	24
7	148	53	56	50	33	58	15	21	25	20
8	136	87	185	94	45	8	14	23	10	3
9	88	39	40	97	40	39	53	35	91	200

a. Setiap *Pixel* dihitung nilai *Grayscale* nya dengan rumus sebagai berikut:

$$Grayscale(x,y) = \frac{red(x,y) + green(x,y) + blue(x,y)}{3}$$

Contoh untuk kasus ini akan digunakan citra pada *Pixel* (0.0) dan (0.1) :

$$Grayscale(0,0) = \frac{67 + 55 + 41}{3} = 54$$

$$Grayscale(0,1) = \frac{76 + 66 + 54}{3} = 65$$

Nilai RGB setiap *Pixel* akan diatur ulang dengan nilai *Grayscale* yang dihasilkan pada perhitungan langkah a). Hasilnya untuk proses *Grayscale* pada citra pembelajaran dan pengenalan ditunjukkan pada Gambar 3.6 – 3.7 dan Tabel 3.3 – 3.4 berikut.

a. Hasil *Grayscale* Citra Pembelajaran

   Pada gambar 3.6 dibawah ini merupakan citra *Grayscale* wayang kulit Yudistira yang akan digunakan dalam proses pembelajaran.



Gambar 3.8 Citra Wayang Kulit Pembelajaran Hasil *Grayscale*

Tabel 3.5 Sampel Pembelajaran Citra Yudistira Hasil Proses *Grayscale*

x/y	0	1	2	3	4	5	6	7	8	9
0	54	69	100	136	140	85	78	64	68	112
1	65	43	106	78	48	30	12	11	14	13
2	85	66	63	53	8	14	15	15	15	12
3	93	89	55	10	27	36	37	25	16	20
4	37	84	15	24	22	17	15	28	19	17
5	18	74	13	33	15	13	53	25	14	21
6	14	28	4	20	12	23	31	13	21	19
7	138	46	51	48	32	55	10	17	20	15
8	126	78	178	89	41	5	10	20	6	1
9	79	31	32	88	32	34	50	33	89	198

b. Hasil *Grayscale* Citra Pengenalan

Pada gambar 3.7 dibawah ini merupakan citra *Grayscale* wayang kulit Yudistira yang akan digunakan dalam proses Pengenalan.



Gambar 3.9 Citra Wayang Kulit Pengenalan Hasil *Grayscale*

Tabel 3.6 Sampel Pengenalan Citra Yudistira Hasil Proses *Grayscale*

x/y	0	1	2	3	4	5	6	7	8	9
0	46	61	97	136	139	85	171	63	48	105
1	61	43	106	78	48	30	12	11	14	8
2	86	66	63	44	8	9	15	10	15	7
3	80	61	55	10	27	36	37	17	16	17
4	37	56	15	17	22	17	15	28	14	17
5	10	74	13	24	15	13	53	22	14	30
6	12	28	3	20	12	23	31	13	21	19
7	117	46	51	48	32	55	10	17	20	15
8	119	78	170	89	151	3	10	20	4	0
9	77	31	32	88	32	34	50	33	89	156

3.1.3.4. Edgedetection (Canny)

Dalam analisis proses deteksi tepi *Canny* ini terdapat langkah-langkah atau alur dari deteksi tepi *Canny* yang harus dilalui, seperti yang bisa dilihat berikut ini. deteksi tepi menggunakan metode *Canny*. Tahapan pertama yang dilakukan adalah *Gaussian Filter*, kemudian *Intensity Gradient*, *Double Threshold* dan tahapan terakhir adalah *Edge Tracking by Hysteresis*.

Algoritma deteksi tepi menggunakan *Canny :*

1. Langkah pertama pada deteksi tepi *Canny* adalah mengkonvolusikan hasil matriks *Grayscale* dengan tapis *Gaussian*, dengan rumus tapis *Gaussian*.

$$\frac{1}{115}\begin{pmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{pmatrix}$$

x/y	0	1	2	3	4
0	54	100	140	78	68
1	84	63	8	15	15
2	37	15	22	15	20
3	14	4	12	31	21
4	126	178	41	10	6

2. Kemudian kita ubah terlebih dahulu tapis *gaussian* kedalam bentuk desimal, agar memudahkan untuk dihitung, setelah itu dapat dikalikan dengan matrik B atau di sini dengan matrik sampel dari *Grayscale*, dapat dilihat pada tabel di bawah ini.

0.02	0.03	0.04	0.03	0.02
0.03	0.08	0.1	0.08	0.03
0.04	0.1	0.13	0.1	0.04
0.03	0.08	0.1	0.08	0.03
0.02	0.03	0.04	0.03	0.02

A

x/y	0	1	2	3	4
0	54	100	140	78	68
1	84	63	8	15	15
2	37	15	22	15	20
3	14	4	12	31	21
4	126	178	41	10	6

B

Setelah itu hitung seperti perhitungan matriks seperti biasanya (matriks **A** * matriks **B**), sebagai salah satu perhitungan *konvolusi gaussian* :

$= (0.02 * 54) + (0.03 * 84) + (0.04 * 37) + (0.03 * 14) + (0.02 * 126)$

$= 1.08 + 2.52 + 1.48 + 0.42 + 2.52$

$= 8.02 \approx 8$

Setelah itu, diteruskan sehingga menjadi tabel seperti tabel 3.7

Tabel 3.7 Hasil *Gaussian*

8	8.2	5.1	3.7	3.4
16.9	15.2	9.2	7.8	7.1
21.8	19.8	12.1	10.1	9.2
16.9	15.2	9.2	7.8	7.1
8	8.2	5.1	3.7	3.4

3. Kemudian tahap selanjutnya menggunakan metode *Sobel*, yaitu mengkonvolusikan dengan daerah kernel kearah *horizontal* (Gx) dan *Vertical* (Gy) yang didapatkan dari hasil *Gaussian* tabel 3.7 dikalikan dengan operator *Sobel*, maka hasilnya bisa dilihat pada gambar 3.10 dan 3.11 dibawah ini. Contoh salah satu perhitungan untuk operator *Sobel* Gx.

8	8.2	5.1	3.7	3.4
16.9	15.2	9.2	7.8	7.1
21.8	19.8	12.1	10.1	9.2
16.9	15.2	9.2	7.8	7.1
8	8.2	5.1	3.7	3.4

-1	0	1
-2	0	2
-1	0	1

= (8*(-1)) + (8.2*(-2)) + (5.1*(-1)) + (16.9*0) + (15.2*0) + (9.2*0) + (21.8*1) + (19.8*2) + (12.1*1) = 44

44	32	25.6
-24.6	-7.4	0
-39	-32	-25.6

Jika, hasil Konvolusi menghasilkan nilai *Pixel* negatif, maka nilai tersebut dijadikan 0, sebaliknya jika hasil Konvolusi menghasilkan nilai *Pixel* lebih besar dari nilai keabuan maksimum, maka nilai tersebut dijadikan nilai keabuan maksimum. Maka nilai konvolusinya seperti pada tabel 3.8 dan gambar 3.10 berikut.

Tabel 3.8 Hasil *Sobel* Horizontal (Gx)

44	32	25.6
0	0	0
0	0	0



Gambar 3.10 Citra Wayang Kulit Hasil Konvulasi *Sobel* Gx

Setelah perhitungan Gx selesai, selanjutnya dihitung Gy dengan cara hasil *Gaussian* dikalikan dengan operator *Sobel* Gy untuk vertikal, Maka nilai konvolusinya seperti pada tabel 3.9 dan gambar 3.11 berikut:

8	8.2	5.1	3.7	3.4
16.9	15.2	9.2	7.8	7.1
21.8	19.8	12.1	10.1	9.2
7.1	7.8	9.2	7.8	7.1
8	8.2	5.1	3.7	3.4

1	2	1
0	0	0
-1	-2	-1

= (8*1) + (8.2*0) + (5.1*(-1)) + (16.9*2) + (15.2*0) + (9.2*(-2)) + (21.8*1) + (19.8*0) + (12.1*(-1))

= 28

Tabel 3.9 Hasil *Sobel* Vertikal (Gy)

28	29	8.8
25	26.8	10
8.4	14.2	8.8



Gambar 3.11 Citra Wayang Kulit Hasil Konvulasi *Sobel* Gy

4. Kemudian di *thresholding*, dengan cara mengubahnya menjadi nilai *biner* atau nilai 1 atau 0. Cara menentukan *threshold* dengan cara

$$T = \frac{\text{nilai Max} + \text{nilai Min}}{2}$$

$$T = \frac{0+255}{2} = 127.5 \approx 127$$

Maka nilai ambangnya 127 maka jika diatas nilai 127 maka nilai =1 dan jika kurang dari 127 maka nilai = 0. Hasil dari nilai dan citra deteksi tepi

Canny untuk proses pembelajaran dan pengenalan dapat dilihat pada gambar 3.12 dan Tabel 3.10

a. Citra Hasil Deteksi Tepi Pembelajaran

Pada gambar 3.12 dibawah ini merupakan citra hasil deteksi tepi *Canny* wayang kulit Yudistira yang akan digunakan dalam proses pembelajaran.



Gambar 3.12 Citra Hasil Deteksi Tepi *Canny* Untuk Proses Pembelajaran

Tabel 3.10 Hasil *Threshold*ing Citra Pembelajaran

x/y	0	1	2	3	4
0	0	0	1	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	1	0	0	0

b. Citra Hasil Deteksi Tepi Pengenalan

Pada gambar 3.13 dibawah ini merupakan citra hasil deteksi tepi *Canny* wayang kulit Yudistira yang akan digunakan dalam proses Pengenalan.



Gambar 3.13 Citra Hasil Deteksi Tepi *Canny* Untuk Proses Pembelajaran

Tabel 3.11 Hasil *Threshold*ing Citra Pembelajaran

x/y	0	1	2	3	4
0	0	0	1	1	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	1	1	0	0

3.1.4. Proses Pembelajaran

Proses pembelajaran merupakan proses dimana komputer belajar dari sejumlah data yang ada dan menghasilkan nilai bobot. Nilai bobot ini nantinya akan digunakan untuk proses pembelajaran dan pengujian. Proses pembelajaran pada sistem ini akan menggunakan metode *Learning Vector Quantization*.

Data training yang digunakan sebagai contoh pada proses pembelajaran adalah *array* 1 dimensi dari citra wayang kulit Yudistira hasil dari proses *threshold*ing sebelumnya [a] atau terlihat pada tabel 3.11 ditambah dengan 1 buah *array* 1 dimensi citra wayang kulit Yudistira [b] atau terlihat pada tabel 3.12.

Serta 2 buah *array* 1 dimensi dari citra wayang kulit Arjuna ([c] dan [d]) dengan struktur yang berbeda yang telah melalui proses pengolahan citra yang sama seperti data (a). Data training ditunjukkan pada Tabel 3.12.

Tabel 3.12 Empat buah aray 1 dimensi data *Training*

[a]	[b]	[c]	[d]
0	0	1	0
1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	0
0	0	0	0
0	0	1	1
0	0	0	0
0	0	0	0
0	0	1	1
0	0	0	0
0	0	0	0
1	1	1	0
0	1	1	1
0	0	1	1

3.1.4.1.Pembelajaran *Learning Vector Quantization*

Proses pembelajaran dengan metode *learning vector quantization* menggunakan 1 alur perhitungan yaitu alur maju. Pada contoh perhitungan ini terdapat 2 buah kelas, yaitu kelas 1 untuk citra wayang kulit Yudistira dan kelas 2 untuk citra wayang kulit Arjuna.

Berikut perhitungan pembelajaran LVQ terhadap data *training* tersebut:

a. Setiap kelas (kelas 1 dan kelas 2) diambil salah satu *array* vektornya dan dijadikan sebagai inisialisasi bobot (w). Sedangkan *array* lainnya menjadi data masukan (x) Hasilnya terlihat pada Tabel 3.13.

Tabel 3.13 Inisialisasi Bobot dan Data Masukan Proses Pembelajaran LVQ

Wayang Kulit	Kelas	Vektor	
Yudistira	1	0 1 0 0 0 0 0 0 0 0 0 1 0 0	Bobot (w1)
Arjuna	2	1 1 1 1 0 0 1 0 0 1 0 0 1 1 1	Bobot (w2)
Yudistira	1	0 1 1 0 0 0 0 0 0 0 0 0 1 1 0	Masukan (x1)
Arjuna	2	0 1 1 1 0 0 1 0 0 1 0 0 0 1 1	Masukan (x2)

b. Tentukan nilai maksimum perulangan, eror minimum, rasio pembelajaran dan pengurangan rasio. Pada contoh ini digunakan batasan sebagai berikut:
   Maksimum perulangan = 50
   Eror minimum = 0,01
   Rasio pembelajaran = 0,05
   Pengurangan rasio = 0,1

c. Selama kondisi berhenti bernilai salah, kerjakan langkah d - f. (Kondisi berhenti jika perulangan mencapai maksimal perulangan atau perubahan rasio lebih kecil dari minimal eror)

d. Untuk masing-masing pelatihan vektor masukan, kerjakan:

$$Cj = \sqrt{\sum_{i=1}^{n} (xi - wji)^2}$$

   Perulang ke-1 dari 5
   Data ke-1 (x1)

Jarak pada :

1) Bobot ke-1 (w1)

$C_1 = \sqrt{((0-0)^2 + (1-1)^2 + (1-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-1)^2 + (1-0)^2 + (0-0)^2)} = \sqrt{2} =$ **1.4**

2) Bobot ke-2 (w2)

$C_1 = \sqrt{((0-1)^2 + (1-1)^2 + (1-1)^2 + (0-1)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (1-1)^2 + (1-1)^2 + (0-1)^2)} = \sqrt{5} =$ **2.2**

e. Dilihat jarak (C) terkecil yang diperoleh pada perhitungan terhadap x1. Kemudian diperiksa apakah target x1 sesuai dengan kelas bobot dengan jarak terkecil. Hasil dari perhitungan di atas menunjukan bahwa jarak terkecil ada pada bobot ke-1. Target x1 adalah kelas 1 dan kelas bobot ke-1 juga kelas 1, hal ini berarti target sesuai dengan hasil perhitungan.

f. Dilakukan perhitungan perubahan bobot terhadap bobot dengan jarak terecil yaitu bobot ke-1 dengan rumus:

   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)]

α= rasio pembelajaran

Dari contoh perhitungan berikut ini menggunakan rumus *Wj(baru) = Wj(lama) + α[x-Wj(lama)]*, karena kelas bobot sesuai dengan kelas target dengan nilai paling minimum, sehinggan perhitungannya sebagai berikut :

$W_{1,1} = 0 + 0{,}05 * (0-0) = 0$

$W_{1,2} = 1 + 0{,}05 * (1-1) = 1$

$W_{1,3} = 0 + 0{,}05 * (1-0) = 0.05$

$W_{1,4} = 0 + 0{,}05 * (0-0) = 0$

$W_{1,5} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,6} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,7} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,8} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,9} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,10} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,11} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,12} = 0 + 0{,}05 * (0 - 0) = 0$

$W_{1,13} = 1 + 0{,}05 * (1 - 1) = 1$

$W_{1,14} = 0 + 0{,}05 * (1 - 0) = 0.05$

$W_{1,15} = 0 + 0{,}05 * (0 - 0) = 0$

g. Mengurangi *learning rate*, dengan rumus:
   α = α * pengurangan rasio
   α = 0,05 * 0,1 = 0,005
   rasio = 0,05-0,005 = 0,045
h. Tes kondisi berhenti (kondisi berhenti jika perulangan mencapai maksimal perulangan atau perubahan rasio lebih kecil dari minimal eror). Disini kedua kondisi tersebut belum ada yang terpenuhi, sehingga perhitungan diulang hingga perulangan berhenti.
i. Pada Contoh ini perulangan berhenti pada perulangan ke-15, dimana perulangan sudah mencapai maksimal dan perubahan rasio sebesar 0,009 dengan hasil bobot ditunjukkan pada table 3.14.

Tabel 3.14 Bobot Hasil Pembelajaran

Wayang Kulit	Kelas	Vektor															
Yudistira	1	0	1	0.31	0	0	0	0	0	0	0	0	0	1	0.31	0	Bobot (w1)
Arjuna	2	0,68	1	1	1	0	0	1	0	0	1	0	0	0.68	1	1	Bobot (w2)

3.1.5. Proses Pengenalan

Proses pengenalan merupakan suatu proses dimana komputer mengenali data tes dengan menggunakan nilai bobot data pembelajaran yang sebelumnya sudah diperoleh. Disini data yang akan diuji merupakan citra wayang kulit Yudistira dengan *array* 1 dimensi seperti yang terlihat pada **Tabel 3.15**.

Tabel 3.15 *Array* Citra Masukan Yudsitira

0
1
1
0
0
0
0
0
0
0
0
0
1
1
0
[b]

3.1.5.1.Pengenalan *Learning Vector Quantization*

Proses pengenalan pada metode *Learning Vector Quantization* sama seperti proses perhitungan bobot pada proses pembelajaran *Learning Vector Quantization.* Berikut tahapan yang akan dilakukan :

a. Data tes dihitung terhadap setiap data bobot yang diperoleh dari hasil pembelajaran.

1. Bobot ke-1 (w1)

$$C_1 = \sqrt{((0-0)^2 + (1-1)^2 + (1-0{,}31)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (1-1)^2 + (1-0{,}31)^2 + (0-0)^2} = \sqrt{0{,}9522} = \mathbf{0,97}$$

2. Bobot Ke-2 (w2)

$$C_2 = \sqrt{((0-0{,}68)^2 + (1-1)^2 + (1-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2 + (1-0{,}68)^2 + (1-1)^2 + (0-1)^2} = \sqrt{4{,}5648} = \mathbf{2,13}$$

b. Dilihat dari jarak [C] terkecil yang diperoleh maka Bobot ke-1 merupakan hasil dari pengenalan, dimana (w1) merupakan citra golongan kelas 1 yaitu citra wayang kulit Yudistira.

3.2 Analisis Kebutuhan *Non* Fungsional

Analisis kebutuhan *non* fungsional menjelaskan pendukung sistem yang akan dijalankan. Adapun kebutuhan-kebutuhannya yaitu:

3.2.1 Kebutuhan Perangkat Keras

Pada aplikasi identifikasi pola citra wayang kulit ini, perangkat keras yang digunakan berupa sebuah komputer yang berbasis desktop. Adapun kebutuhan spesifikasi komputer yang digunakan dalam penelitian ini dapat dilihat pada Tabel 3.16

Tabel 3.16 Spesifikasi Perangkat Keras Komputer

Komputer		
No	Spesifikasi	
1	*Processor*	Core i5 2.26Ghz
2	*Memory RAM*	4GB
3	*VGA*	512 MB
4	*Hardisk*	500GB
5	*Keyboard*	-
6	*Mouse*	-

3.2.2 Kebutuhan Perangkat Lunak

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:

Tabel 3.17 Spesifikasi Perangkat Lunak

Perangkat Lunak		
No	Spesifikasi	
1	Sistem Operasi	*Windows 7 64bit*
2	*Tools*	*Matlab* 2015

3.2.3 Kebutuhan *User*

Untuk menjalankan aplikasi ini dibutuhkan *user* yang dapat menjalankan fungsi-fungsi dari aplikasi ini. Aplikasi ini hanya terdapat satu jenis *user* yaitu "penguji". Berikut adalah spesifikasi *user* yang diperlukan :

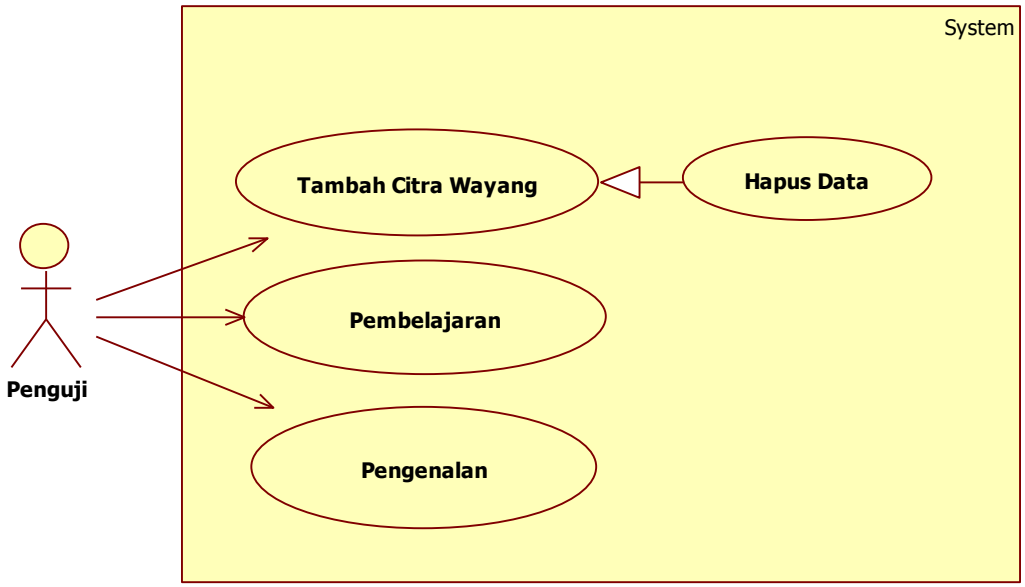
1. Dapat mengoperasikan komputer.
2. Mengerti maksud penggunaan algoritma *Learning Vector Quantization*.
3. Mengerti istilah-istilah dalam algoritma *Learning Vector Quantization*.

3.3 Analisis Data Fungsional

Analisis kebutuhan fungsional menjelaskan proses-proses yang terjadi di dalam aplikasi ini. Proses-proses tersebut dijelaskan dalam *usecase diagram, skenario diagram, activity diagram, sequence diagram* dan *class diagram*.

3.3.1 *Use Case Diagram*

Diagram *Use Case* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Diagram ini juga mendeskripsikan apa yang akan dilakukan oleh sistem. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, identifikasi *Use Case* dan skenario *Use Case.*



Gambar 3.14 *Use Case Diagram*

3.3.1.1 Identifikasi Aktor

Identifikasi Aktor berfungsi untuk menjelaskan Aktor yang terdapat pada *Use Case Diagram*. Definisi Aktor diterangkan pada Tabel 3.18

Tabel 3.18 Identifikasi Aktor

No	Aktor	Deskripsi
1	Penguji	Merupakan aktor yang akan menggunakan aplikasi pengenalan pola citra wayang kulit yang akan dibangun atau disebut sebagai penguji

3.3.1.2 Identifikasi *Use Case*

Definisi *Use Case* berfungsi untuk menjelaskan fungsi *Use Case* yang terdapat pada *Use Case Diagram*. Definisi *Use Case* diterangkan pada Tabel 3.19

Tabel 3.19 Identifikasi *Use Case*

No	Use Case	Deskripsi
1	Tambah Citra Wayang	Proses menambah data informasi dan gambar wayang kulit
2	Hapus Data	Proses untuk menghapus data wayang kulit
3	Pembelajaran	Proses untuk melakukan pembelajaran terhadap citra wayang kulit
4	Pengenalan	Proses untuk melakukan pengenalan terhadap citra wayang kulit

3.3.1.3 Skenario *Use Case*

Skenario *Use Case* menggambarkan alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang aktor, seseorang, atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara. Skenario *use case* pada aplikasi ini akan dijelaskan pada Tabel 3.20 sampai Tabel 3.23

Tabel 3.20 Skenario *Use Case* Tambah Data Wayang Kulit

Identifikasi	
Nomor	1
Nama	Tambah Citra Wayang
Tujuan	Menambahkan data wayang kulit ke sistem
Aktor	Penguji
Kondisi Awal	Menampilkan tampilan pengenalan citra

Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Memilih jenis wayang kulit	
	2. Menampilkan form citra
3. Melakukan aksi *Input* citra	
	4. Menampilkan popup pencarian citra
5. Memilih citra yang di inginkan	
	6. Menampilkan citra yang dipilih
7. Melakukan aksi simpan	
	8. Melakukan proses pre-processing.
	9. Menyimpan gambar dan hasil pre-processing.
Skenario Alternatif (Data Belum Dipilih)	
1. Memilih jenis wayang kulit	
	2. Menampilkan form citra
3. Melakukan aksi simpan	
	4. Menampilkan pesan "Tentukan Terlebih Dahulu Citra ke-x"
Kondisi Akhir	Aktor dapat melakukan proses penambahan citra wayang.

Tabel 3.21 Skenario *Use Case* Hapus Data

Identifikasi	
Nomor	2
Nama	Hapus Data
Tujuan	Menghapus data wayang kulit
Aktor	Penguji

Kondisi Awal	Aktor berada tampilan pengenalan pola citra wayang kulit
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Memilih jenis wayang kulit	
	2. Menampilkan citra wayang kulit yang dipilih
3. Melakukan aksi reset atau hapus data	
	4. Menampilkan pesan konfirmasi penghapusan data wayang kulit
5. Memilih aksi “Ya”	
	6. Menghapus citra wayang kulit
Skenario Alternatif (Batal Menghapus)	
1. Memilih jenis wayang kulit	
	2. Menampilkan citra wayang kulit yang dipilih
3. Melakukan aksi reset reset hapus data	
	4. Menampilkan pesan konfirmasi penghapusan data wayang kulit
5. Memilih aksi “Tidak”	
	6. Batal menghapus data citra wayang kulit
Kondisi Akhir	Data wayang kulit sudah terhapus dari sistim

Tabel 3.22 Skenario *Use Case* Pembelajaran

Identifikasi	
Nomor	3
Nama	Pembelajaran

Tujuan	Melakukan pembelajaran dengan menggunakan algoritma LVQ
Aktor	Penguji
Kondisi Awal	Data Citra belum dipelajari
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Mengisi parameter yang akan digunakan dalam pembelajaran	
2. Melakukan aksi proses pembelajaran	
	3. Mengambil semua data citra wayang kulit dari sistem
	4. Melakukan proses pembelajaran dengan metode LVQ
	5. Menyimpan bobot W hasil pembelajaran LVQ
	6. Menampilkan data hasil pembelajaran
Kondisi Akhir	Data pembelajaran telah dipelajari. Dan bobot hasil pembelajaran disimpan.

Tabel 3.23 Skenario *Use Case* Pengenalan

Identifikasi	
Nomor	4
Nama	Pengenalan
Tujuan	Melakukan pengenalan citra wayang kulit dengan algoritma LVQ
Aktor	Penguji
Kondisi Awal	Citra wayang kulit yang belum dikenali

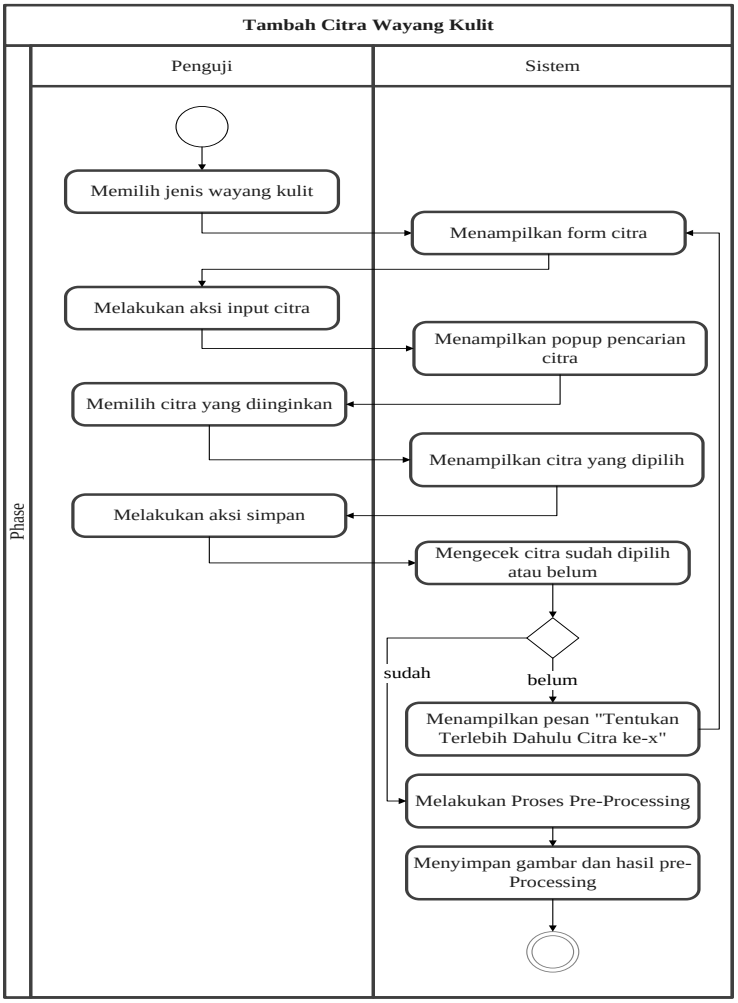
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Melakukan aksi *Input* citra	
	2. Menampilkan popup pencarian citra
3. Memilih citra wayang kulit yang ingin dikenali	
	4. Menampilkan gambar yang dipilih
5. Mekalukan aksi proses pengenalan	
	6. Mengambil data bobot LVQ dari system penyimpanan
	7. Melakukan proses pengenalan dengan metode LVQ
	8. Menampilkan hasil pengenalan

3.3.2 *Activity Diagram*

Diagram aktivitas atau *Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. 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 sistem. Activity diagram sistem ini ditunjukkan pada Gambar 3.15 sampai 3.18

1. *Activity Diagram* Tambah Citra Wayang Kulit

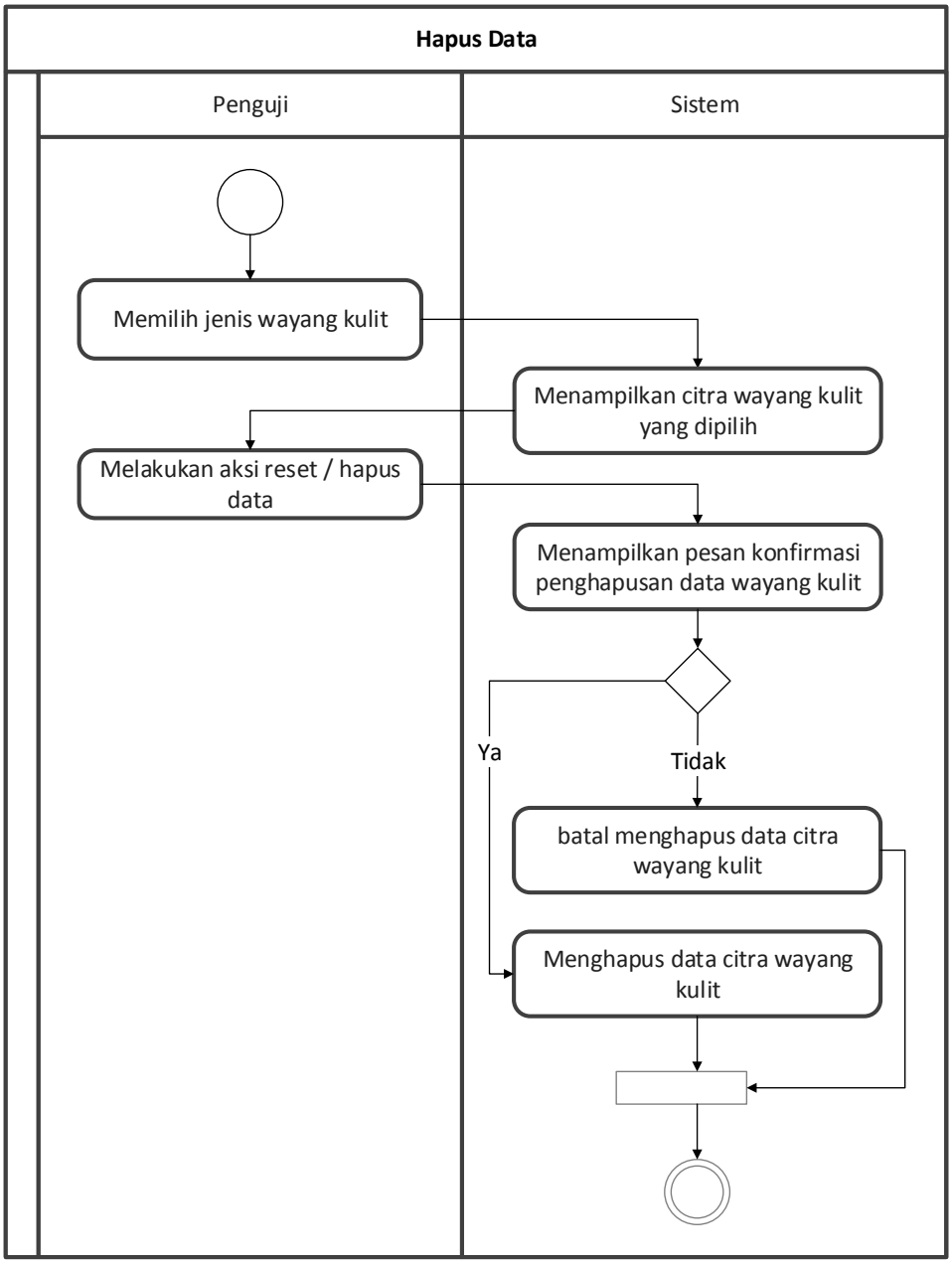
   Activity Diagram tambah data wayang kulit menjelaskan alir kerja *actor* untuk melakukan proses tambah data citra wayang kulit.



Gambar 3.15 *Activity* Diagram Tambah Citra Wayang

2. *Activity Diagram* Hapus Data

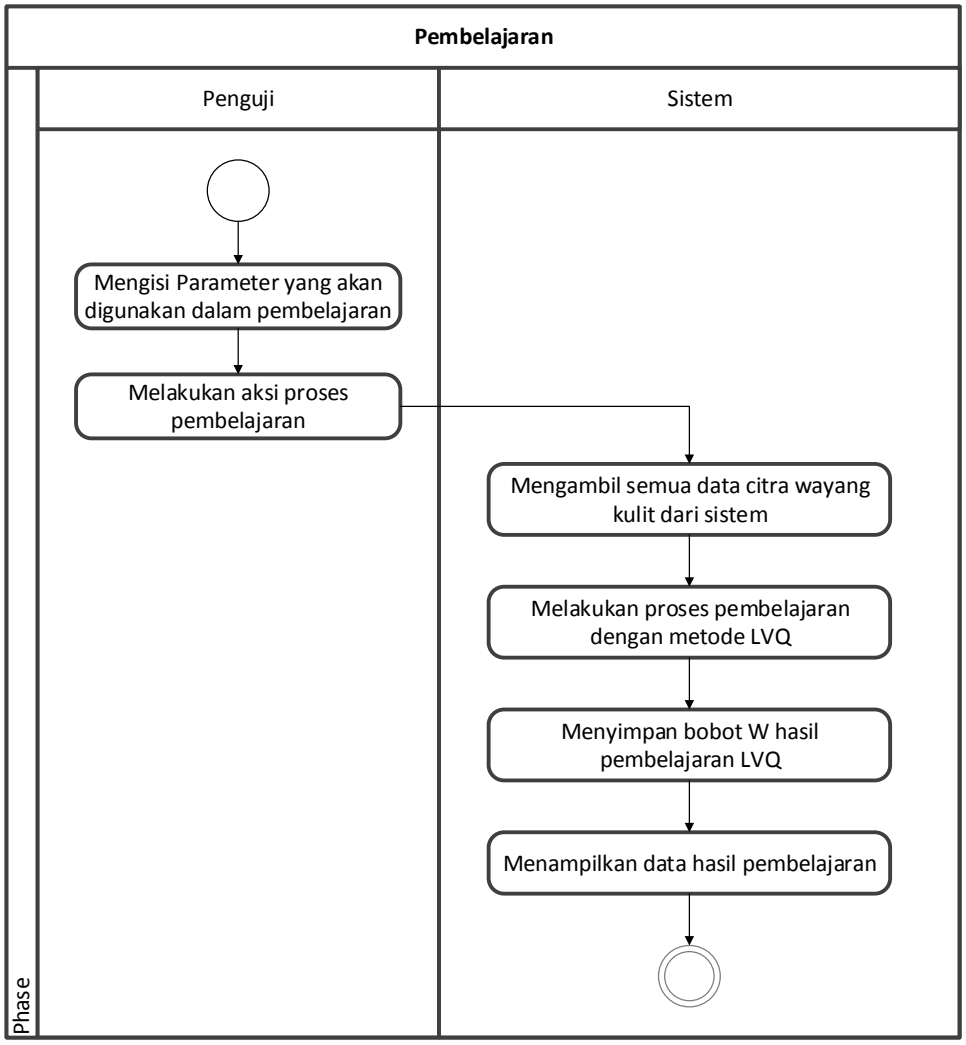
   Activity Diagram Hapus Data menjelaskan alir kerja *actor* (penguji) untuk melakukan proses menghapus data wayang kulit dari sistem.



Gambar 3.16 *Activity Diagram* Hapus Data

3. *Activity Diagram* Pembelajaran

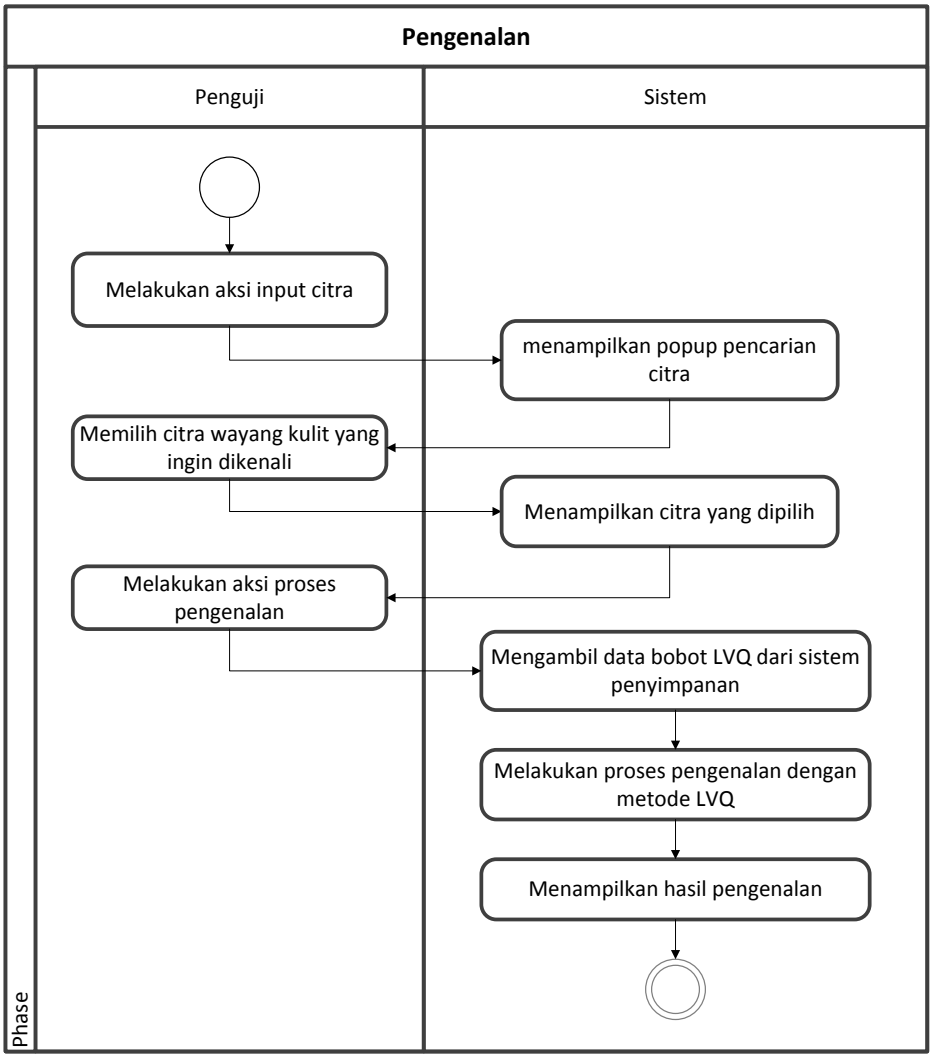
   Activity Diagram Pembelajaran menjelaskan alir kerja *actor* (penguji) untuk melakukan proses pembelajaran dengan menggunakan algoritma LVQ.



Gambar 3.17 *Activity Diagram* Pembelajaran

4. *Activity Diagram* Pengenalan

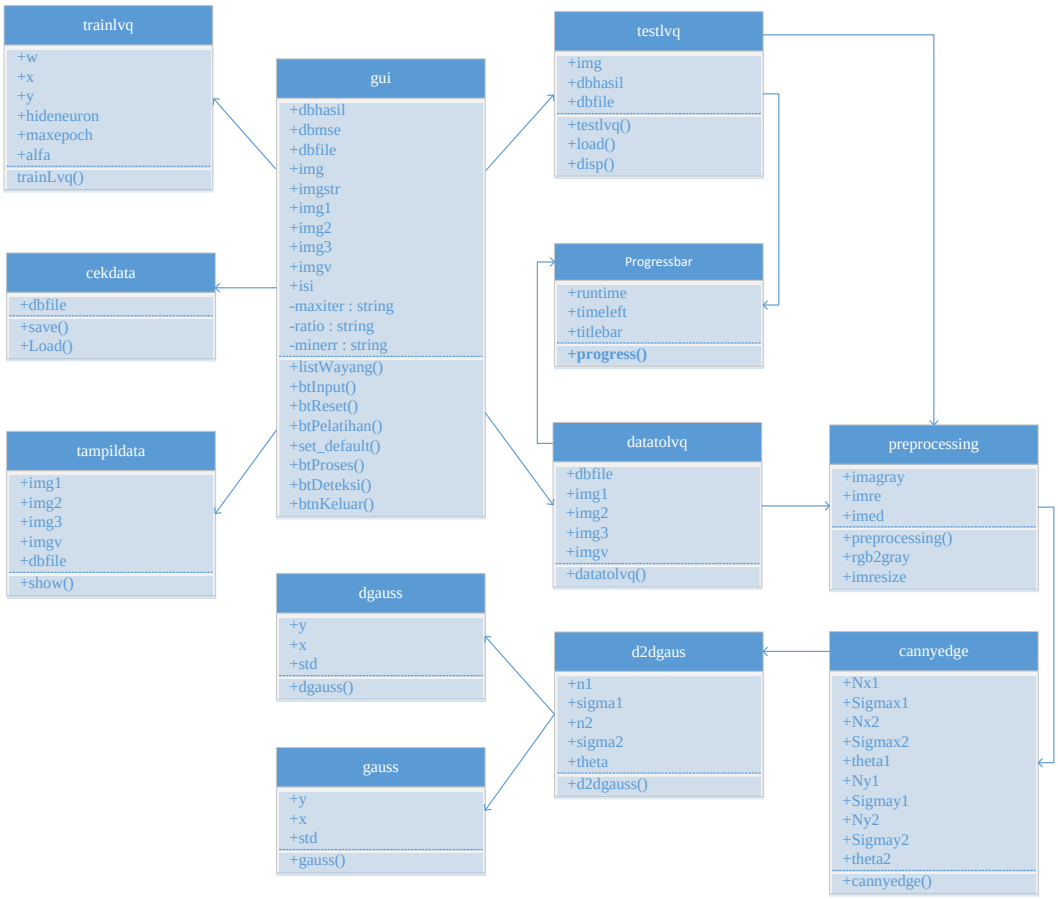
   Activity Diagram Pengenalan menjelaskan alir kerja actor untuk melakukan proses pengenalan atau identifikasi citra wayang kulit.



Gambar 3.18 *Activity Diagram* Pengenalan

3.3.2.1 *Class Diagram*

Class diagram digunakan untuk memberi gambaran suatu sistem. Dapat dilihat dari *class-class* yang ada dan bagaimana relasinya. *Class diagram* pada aplikasi ini dapat dilihat pada gambar 3.19



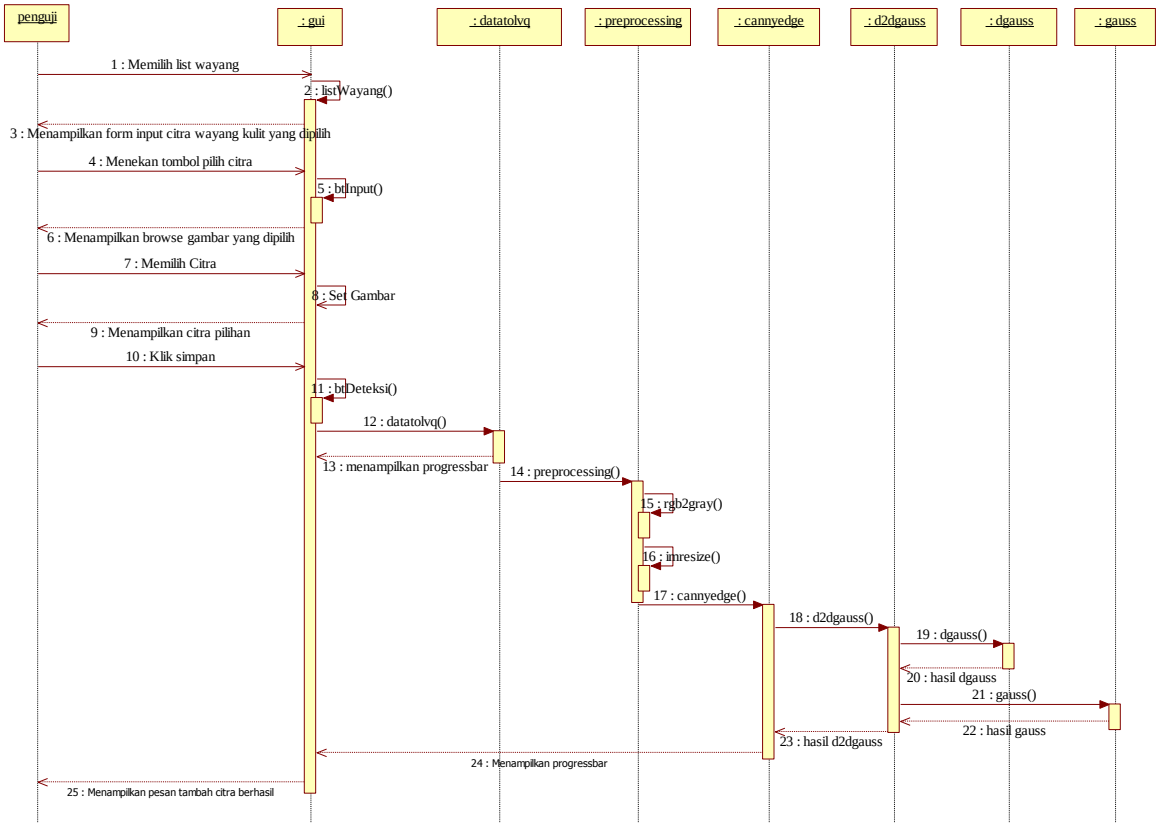
Gambar 3.19 *Class Diagram*

3.3.2.2 *Sequence Diagram*

Sequence diagram merupakan gambaran antara interaksi antar objekobjeknya. Merupakan langkah-langkah dari beberapa event untuk mendapatkan suatu *Output* tertentu.

1. *Sequence Diagram* Tambah Citra

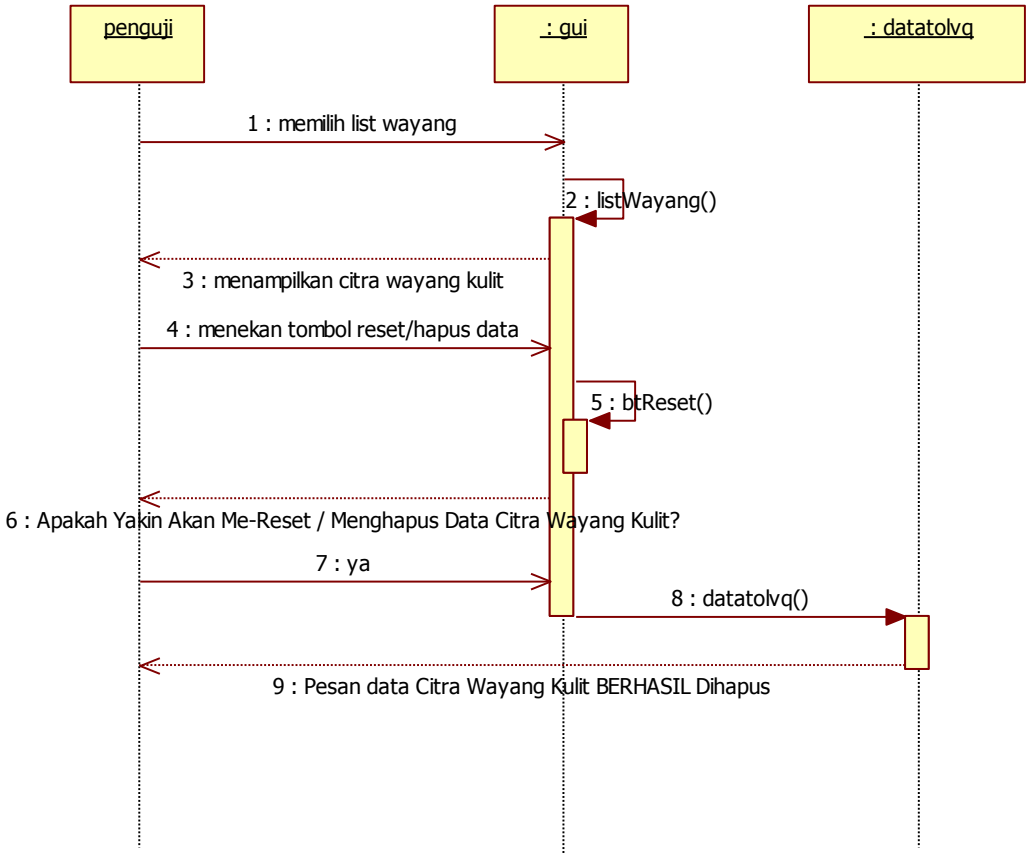
   Berikut adalah *sequence diagram* dalam melakukan pilih citra dapat dilihat pada gambar 3.20



Gambar 3.20 *Sequence Diagram* Tambah Citra Wayang

2. *Sequence Diagram* Hapus Data

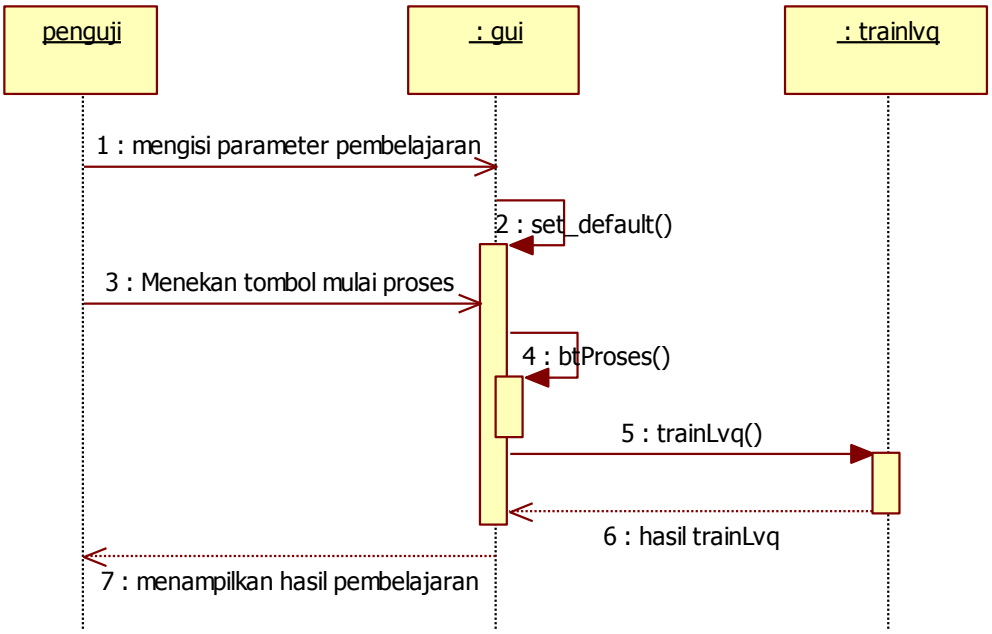
   Berikut adalah *sequence diagram* dalam melakukan hapus data dapat dilihat pada gambar 3.21



Gambar 3.21 *Sequence Diagram* Hapus Data

3. *Sequence Diagram* Pembelajaran

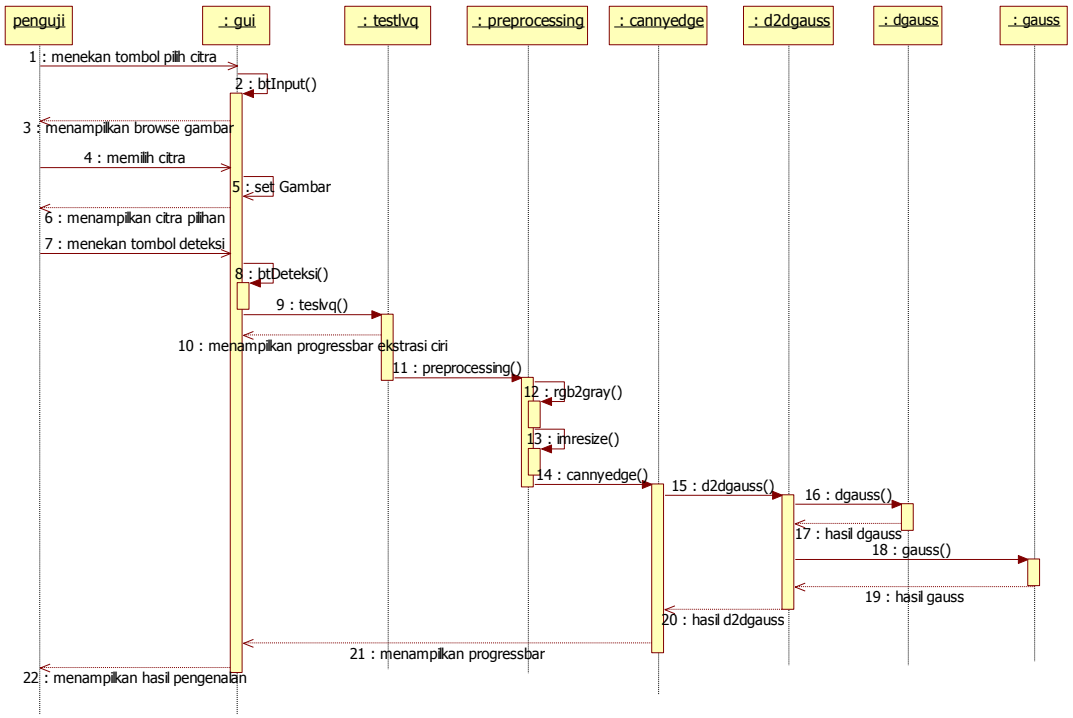
   Berikut adalah *sequence diagram* dalam melakukan pembelajaran dapat dilihat pada gambar 3.22



Gambar 3.22 *Sequence Diagram* Pembelajaran

4. *Sequence Diagram* Pengenalan

   Berikut adalah *sequence diagram* dalam melakukan pengenalan dapat dilihat pada gambar 3.23



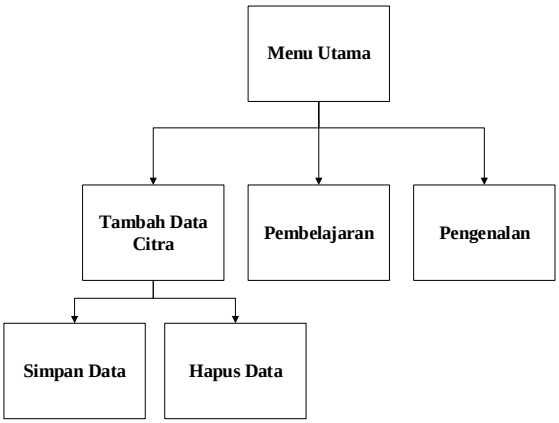
Gambar 3.23 *Sequence Diagram* Pengenalan

3.3.3 Perancangan Antarmuka

Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap tampilan yang akan digunakan. Perancanang antarmuka disini memiliki 3 sub judul yang akan dibuat, yaitu perancangan struktur menu, perancangan form dan perancangan pesan.

3.3.3.1 Perancangan Struktur Menu

Arsitektur struktur menu merupakan gambaran perancangan menu yang akan dimiliki oleh sistem. Adapun perancangan struktur menu yang terdapat dalam Aplikasi Pengenalan Pola Citra Wayang Kulit dapat dilihat pada Gambar 3.24.



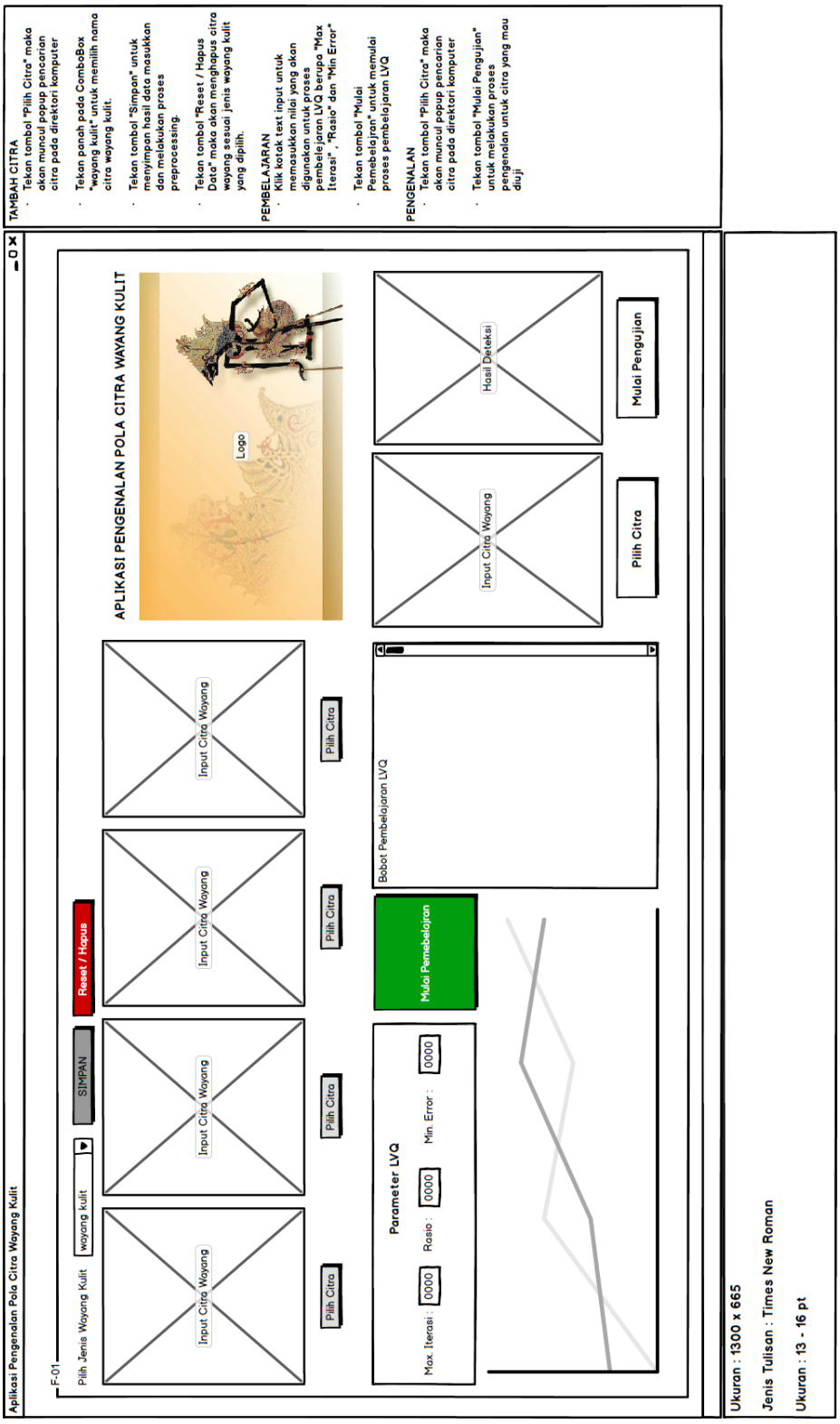
Gambar 3.24 Struktur Menu Pengenalan Pola Wayang Kulit

3.3.3.2 Perancangan *Form*

Perancangan *form* disini dibagi menjadi 6 *form* inti, yaitu menu utama, tambah data, lihat data, pembelajaran, pengenalan dan hasil (pembelajaran dan pengenalalan).

1. **Perancangan Tampilan Pengenalan**

   Perancangan tampilan pengenalan disini merupakan tampilan dari aplikasi yang akan dibangun, dimana terdapat beberapa proses yang dapat dilakukan oleh penguji, seperti proses tambah citra, hapus citra, pembelajaran, dan pengenalan. Perancangan tampilan pengenalan dapat dilihat pada Gambar 3.25.



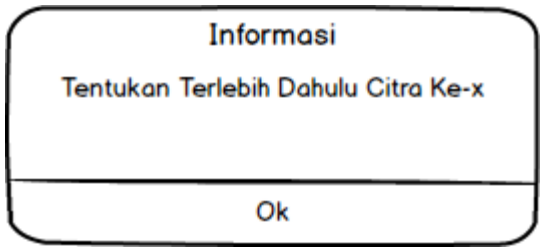
Gambar 3.25 Perancangan Tampilan Pengenalan

3.3.3.3 Perancangan Pesan

Perancangan pesan digunakan untuk memberi pemberithauan sebuah proses sudah selesai dilakukan, terdapat kesalahan dalam masukan data atau peringatan untuk menghapus data.

1. **Perancangan Pesan Tambah Data Pembelajaran Belum Dipilih [M-01]**

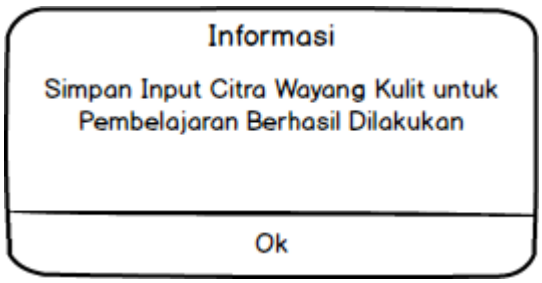
   Pesan tambah data pembelajaran belum dipilih digunakan untuk memberikan informasi bahwa proses tambah data belum dapat dilakukan. Perancangan pesan ini digunakan pada *Form* Pengenalan (F-01). Perancangan pesan tambah data pembelajaran belum dipilih ditunjukkan pada Gambar 3.26



Gambar 3.26 Perancangan Pesan Data Uji Belum Dipilih

2. **Perancangan Pesan Simpan Citra Wayang Kulit Berhasil [M-02]**

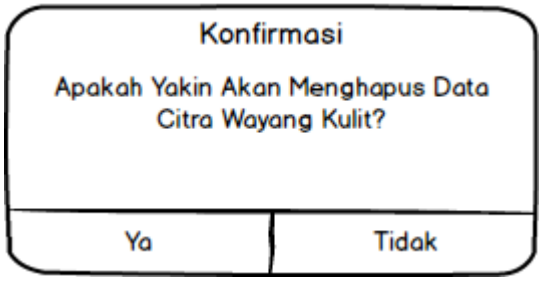
   Pesan simpan citra wayang kulit berhasil digunakan untuk memberikan informasi bahwa proses tambah data berhasil dilakukan. Perancangan pesan ini digunakan pada tampilan pengenalan (F-01). Perancangan pesan simpan citra wayang kulit berhasil ditunjukkan pada Gambar 3.27



Gambar 3.27 Perancangan Pesan Simpan Citra Berhasil

3. Perancangan Pesan Hapus Data [M-03]

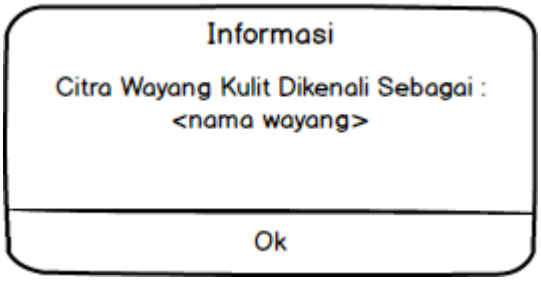
Pesan hapus data digunakan untuk memberikan penegasan apakah data yang dipilih yakin untuk dihapus. Perancangan pesan ini digunakan pada Form Pengenalan (F-01). Perancangan pesan hapus data ditunjukkan pada Gambar 3.28



Gambar 3.28 Perancangan Pesan Hapus Data

4. Perancangan Pesan Hasil Pengenalan [M-04]

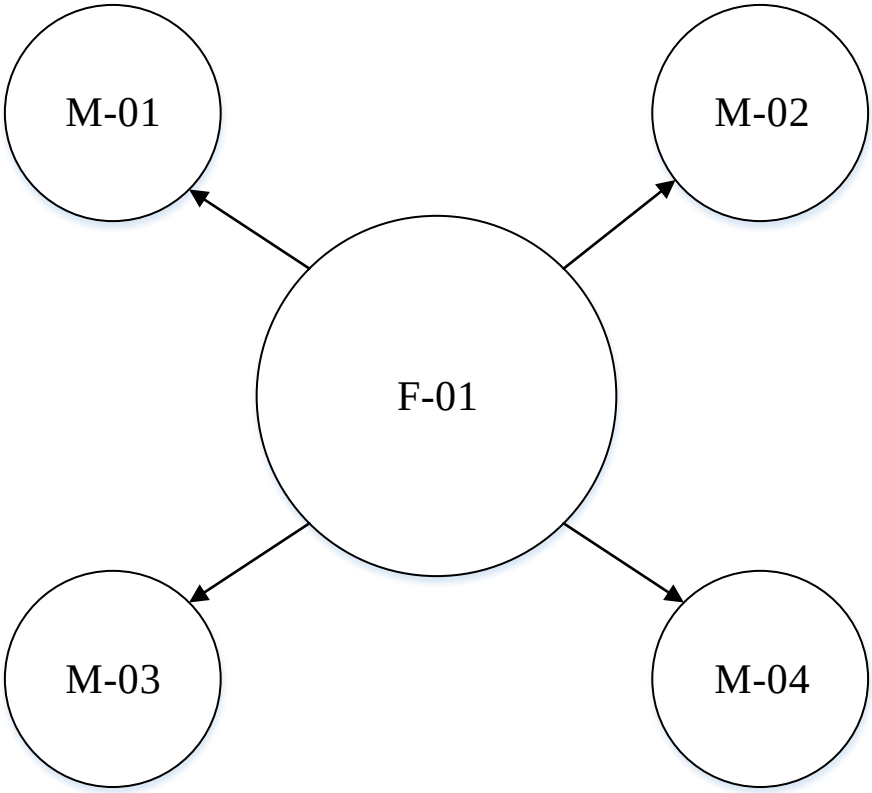
Pesan hasil pengenalan digunakan untuk memberikan informasi citra wayang kulit yang diuji dikenali sebagai salah satu citra wayang kulit tertentu. Perancangan pesan ini digunakan pada Form Pengenalan (F-01). Perancangan pesan hasil pengenalan ditunjukkan pada Gambar 3.29



Gambar 3.29 Perancangan Pesan Hasil Pengenalan

3.3.3.4 Jaringan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukan hubungan antar berbagai objek yang terdiri dari lingkaran-lingkaran sebagai objek dan informasi dari objek-objek tersebut. Adapun jaringan semantik untuk aplikasi Implementasi Algoritma *Learning Vector Quantization* Untuk Pengenalan Pola Citra Wayang Kulit adalah sebagai berikut.



Gambar 3.30 Jaringan Semantik aplikasi Pengenalan Pola Citra Wayang Kulit