BAB 3
ANALISIS DAN PERANCANGAN

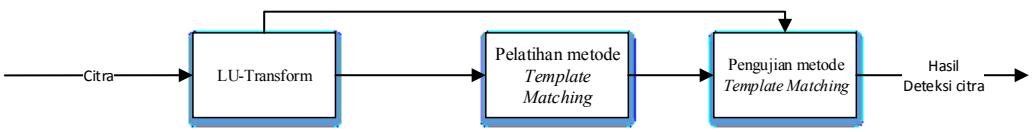
3.1 Analisis Masalah

Metode *LU-Transform* merupakan metode tektur deteksi yang sampai saat ini masih sedang dikembangkan. Metode *LU-Transform* ini berfungsi untuk memberikan ekstraksi ciri, dari nilai *eigen values* yang berguna untuk normalisasi objek dari pola karakteristik pada sebuah objek tersebut, sehingga dapat memisahkan antara tekstur asli sebuah objek dengan *background image* yang berada diluar karakteristik penyakit objek tersebut, yang nantinya tekstur penyakit tersebut dapat dijadikan pola untuk pengklasifikasian objek lebih lanjut dengan metode *Template matching correlation. Template matching correlation* merupakan metode sederhana yang digunakan untuk mengenali pola pada sebuah gambar. Selama ini sering digunakan dalam mendeteksi huruf, sidik jari, kulit, dan lain-lain. Metode *Template matching correlation* dapat melakukan klasifikasi objek dengan tingkat akurasi yang tinggi. Dengan membandingkan nilai korelasi dari tiap data uji dengan tiap-tiap data latih.

3.2 Analisis Sistem

Dalam penyelesaian masalah yang telah dijabarkan pada analisis masalah, proses yang dilakukan dapat dikelompokan pada tiga tahapan utama.

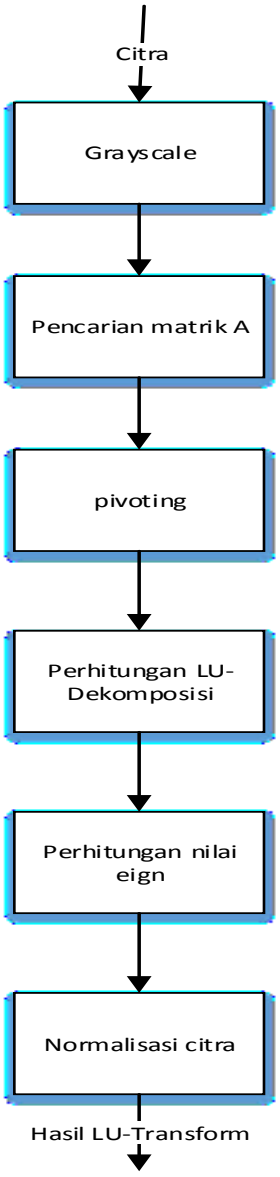
Tahapan proses tersebut dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Proses utama sistem

3.2.1 Proses Perhitungan LU-Transform

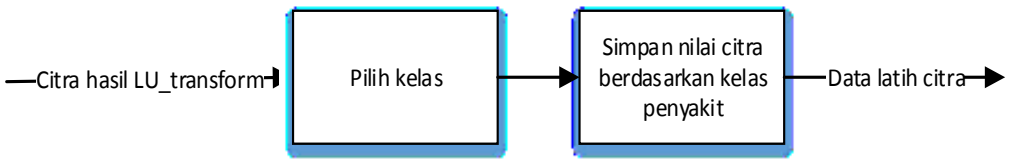
Gambar penuh yang dijadikan masukan data diproses dengan deteksi tekstur *LU-Transform* mulai dari perubahan warna *grayscale* pencarian matrik A, pivoting, perhitungan Eliminasi Gauss, pencarian nilai *eigen values*, dan proses normalisasi citra. Tahapan proses ini dapat dilihat pada gambar 3.2.



Gambar 3. 2 Alur Proses LU-Transform

3.2.2 Proses Pelatihan Teamplate Macthing

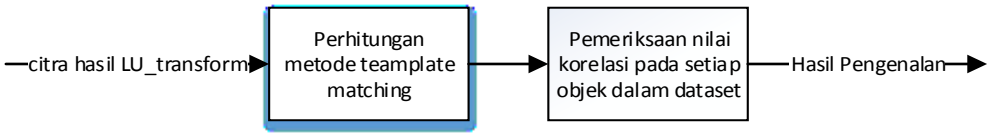
Dalam tahapan ini dilakukan pemilihan kelas penyakit, selanjutnya nilai objek setiap pikel akan langsung disimpan kedalam dataset sesuai dengan kelas penyakitnya tersebut. Proses pelatihan ini akan digunakan dalam proses selanjutnya yakni pengujian objek. Tahapan proses ini dapat dilihat pada gambar 3.3.



Gambar 3. 3 Alur Proses Pelatihan Template matching correlation

3.2.3 Proses Pengujian Teamplate Macthing

Dalam tahapan ini nilai inputan hasil *LU-Transform*, dicari perbandingan nilai korelasi objek uji tersebut dibandingkan dengan korelasi nilai objek lain yang telah disimpan dalam data latih. Alur proses pengenalan dapat dilihat pada Gambar 3.4.



Gambar 3. 4 Alur Proses Pengujian Template matching correlation

3.3 Analisis Data Masukan

Data masukan yang akan diproses merupakan data RGB dari objek penyakitnya kelinci, yang di rubah menjadi data bernilai *grayscale*, yang akan diproses ke *LU-Transform*, objek yang diambil khususnya bagian kulit yang terinfeksi penyakit tersebut. Bukan gambar dari objek kelinci secara keleluruhan. Contoh data gambar yang dapat digunakan dapat dilihat pada Gambar 3.5.



Gambar 3. 5 Contoh gambar penyakit Sore Hocks

Gambar 3.5 dimisalkan gambar memiliki ukuran 32x32 pixel dengan model warna RGB dan kedalaman warna 255bit, dan dirubah menjadi nilai *grayscale*. Nilai *grayscale* setiap pixel objek ke-1 dapat dilihat pada tabel 3.1.

Tabel 3. 1 Nilai pixel grayscale objek ke-1

f(i,j)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	188	194	197	198	196	191	193	195	194	195	197	196	190	188	182	179	176	168	150	146	136	131	136	153	164	166	161	175	173	170	157	143
1	197	201	199	200	202	196	196	198	196	197	197	195	189	186	181	177	172	160	145	136	99	100	97	110	140	158	156	174	173	177	154	149
2	201	203	202	201	204	201	202	201	199	198	195	192	186	181	176	175	164	153	136	102	88	71	80	105	123	145	141	175	170	176	160	154
3	205	203	203	204	205	203	204	203	200	197	195	190	183	172	168	167	157	144	122	81	76	76	73	75	94	135	105	170	171	176	65	153
4	206	205	203	206	206	204	206	207	201	197	194	189	180	165	160	158	155	135	101	77	78	58	69	60	75	101	87	161	172	171	165	148
5	204	206	205	206	205	206	207	206	201	198	193	189	176	155	150	152	144	104	81	58	53	32	44	36	51	78	36	153	176	177	163	143
6	205	206	207	205	206	206	205	204	200	197	192	187	172	139	139	139	126	83	61	34	30	23	18	14	28	38	35	133	173	176	163	136
7	206	206	206	207	204	207	201	203	199	196	191	183	167	128	126	123	119	68	32	26	21	20	23	16	17	22	32	79	165	174	157	128
8	206	205	206	206	206	204	198	205	200	194	187	182	163	119	122	118	111	43	27	27	19	24	29	21	18	18	29	44	158	174	156	125
9	206	204	204	206	205	202	197	204	200	195	185	180	159	125	105	116	83	32	22	24	22	28	33	26	18	21	27	38	147	173	162	134
10	206	201	206	205	204	201	201	203	199	192	185	180	157	132	106	114	59	29	21	23	27	33	35	27	16	25	25	46	150	172	160	136
11	204	202	205	206	202	197	204	201	198	191	185	177	157	135	106	109	45	25	19	19	32	41	32	27	18	25	29	53	134	172	159	133
12	205	202	204	204	199	192	205	200	196	191	184	174	150	133	112	104	37	28	22	22	30	44	35	29	24	41	43	50	121	168	160	138
13	203	201	205	204	201	192	203	200	194	191	181	174	144	120	109	82	29	30	19	23	26	38	48	41	31	41	41	60	121	159	160	145
14	202	202	207	204	202	194	200	200	192	188	181	170	138	106	109	45	20	28	18	23	28	40	53	48	36	38	30	53	113	157	163	145
15	200	197	206	205	203	199	199	200	190	188	178	167	134	103	105	22	16	23	23	30	33	48	56	48	41	45	33	50	103	156	157	144
16	200	199	204	204	202	203	198	199	188	184	173	168	119	119	77	18	19	23	29	36	38	35	45	32	49	51	42	48	112	144	150	144
17	200	200	202	203	204	201	199	195	188	181	175	160	107	87	36	17	16	20	31	36	34	37	44	40	49	53	41	56	107	140	148	147
18	200	198	202	202	204	201	198	192	185	180	173	149	111	81	35	16	15	22	35	34	34	32	46	42	46	65	40	61	114	144	155	157
19	196	196	199	199	205	201	198	190	182	178	171	143	113	84	39	21	17	24	37	38	26	30	41	32	55	50	48	52	121	144	154	158
20	195	191	198	198	204	201	198	188	178	175	168	142	106	92	40	21	17	24	36	38	24	30	41	37	51	51	45	73	122	147	150	162
21	196	188	195	198	201	201	197	186	178	172	166	145	99	94	34	19	23	24	31	35	34	50	31	47	61	38	43	74	135	143	149	166
22	192	188	193	196	197	199	194	186	180	171	160	133	106	93	49	20	23	22	26	35	25	42	36	53	46	40	63	87	138	145	155	165
23	185	182	191	195	189	195	193	188	182	170	161	124	124	96	87	19	19	24	19	28	28	50	38	64	42	67	71	118	138	145	160	165
24	182	184	191	193	186	190	187	186	181	161	157	123	33	98	93	19	18	29	23	29	31	44	29	39	61	76	88	143	140	149	161	174
25	183	189	193	188	184	183	181	182	174	150	154	130	126	110	85	23	27	24	27	31	34	30	30	28	81	104	124	149	144	152	167	179
26	185	186	192	188	185	180	178	179	161	139	152	138	119	130	79	42	29	28	31	35	35	27	27	39	105	101	151	151	150	155	169	183
27	184	180	190	189	188	180	179	177	158	142	137	145	135	157	118	97	40	26	39	39	41	41	36	73	139	152	159	155	156	158	176	189
28	186	180	188	189	190	178	177	172	165	155	149	147	150	166	147	152	130	67	33	48	47	34	47	137	160	163	164	157	163	162	182	192
29	186	181	183	188	190	176	178	169	153	163	155	143	153	167	161	154	160	157	95	81	91	107	155	166	171	168	166	159	168	168	185	194
30	186	181	179	186	187	174	171	172	155	163	155	145	163	168	168	157	166	167	182	164	140	164	178	173	172	169	169	166	170	176	192	198
31	182	179	177	184	184	173	162	168	163	155	157	150	180	170	169	169	171	184	187	166	135	171	190	181	172	176	172	172	177	182	196	196

3.4 Analisis Metode

Setelah objek gambar didapatkan, objek tersebut akan dihitung dengan metode *LU-Transform* untuk menghasilkan objek baru yang digunakan sebagai inputan ke dalam metode kalsifikasi metode *Template matching correlation*. Sebagai contoh kasus objek pertama dalam gambar dimisalkan dengan ukuran yang lebih kecil yaitu dengan skala 32x32.

3.4.1 Analisis Perhitungan LU-Transform

1. Mencari nilai matrik A yang akan dihitung menggunakan perhitungan Eliminasi Gauss, dengan rumus (2.2):

$$A(j,i) = \begin{pmatrix} f(j,i) & f(j,i+\delta,) & \cdots & f(j,i+w-\delta,) \\ f(j+\delta,i) & f(j+\delta,i+\delta) & \cdots & f(j+\delta,i+w-\delta) \\ \vdots & \vdots & \ddots & \vdots \\ f(j+h-\delta,i) & f(j+h-\delta,i+\delta) & \cdots & f(i+h-\delta,i+w-\delta) \end{pmatrix}$$

Dimana : space (δ) =2

Banyak matrik A(*w)* =8

Awal nilai keofisien matrik A(*l)* $= \frac{w}{2} = 4$

Sehingga didapatkan posisi matrik A sebagai berikut:

Tabel 3. 2 Nilai hasil pencarian matrik A objek ke-1

f(i,j)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	188	194	197	198	196	191	193	195	194	195	197	196	190	188	182	179	176	168	150	146	136	131	136	153	164	166	161	175	173	170	157	143
1	197	201	199	200	202	196	196	198	196	197	197	195	189	186	181	177	172	160	145	136	99	100	97	110	140	158	156	174	173	177	154	149
2	201	203	202	201	204	201	202	201	199	198	195	192	186	181	176	175	164	153	136	102	88	71	80	105	123	145	141	175	170	176	160	154
3	205	203	203	204	205	203	204	203	200	197	195	190	183	172	168	167	157	144	122	81	76	76	73	75	94	135	105	170	171	176	65	153
4	206	205	203	206	206	204	206	207	201	197	194	189	180	165	160	158	155	135	101	77	78	58	69	60	75	101	87	161	172	171	165	148
5	204	206	205	206	205	206	207	206	201	198	193	189	176	155	150	152	144	104	81	58	53	32	44	36	51	78	36	153	176	177	163	143
6	205	206	207	205	206	206	205	204	200	197	192	187	172	139	139	139	126	83	61	34	30	23	18	14	28	38	35	133	173	176	163	136
7	206	206	206	207	204	207	201	203	199	196	191	183	167	128	126	123	119	68	32	26	21	20	23	16	17	22	32	79	165	174	157	128
8	206	205	206	206	206	204	198	205	**200**	194	187	**182**	163	119	**122**	118	111	**43**	27	27	**19**	24	29	**21**	18	18	**29**	44	158	**174**	156	125
9	206	204	204	206	205	202	197	204	200	195	185	180	159	125	105	116	83	32	22	24	22	28	33	26	18	21	27	38	147	173	162	134
10	206	201	206	205	204	201	201	203	199	192	185	180	157	132	106	114	59	29	21	23	27	33	35	27	16	25	25	46	150	172	160	136
11	204	202	205	206	202	197	204	201	**198**	191	185	**177**	157	135	**106**	109	45	**25**	19	19	**32**	41	32	**27**	18	25	**29**	53	134	**172**	159	133
12	205	202	204	204	199	192	205	200	196	191	184	174	150	133	112	104	37	28	22	22	30	44	35	29	24	41	43	50	121	168	160	138
13	203	201	205	204	201	192	203	200	194	191	181	174	144	120	109	82	29	30	19	23	26	38	48	41	31	41	41	60	121	159	160	145
14	202	202	207	204	202	194	200	200	**192**	188	181	**170**	138	106	**109**	45	20	**28**	18	23	**28**	40	53	**48**	36	38	**30**	53	113	**157**	163	145
15	200	197	206	205	203	199	199	200	190	188	178	167	134	103	105	22	16	23	23	30	33	48	56	48	41	45	33	50	103	156	157	144
16	200	199	204	204	202	203	198	199	188	184	173	168	119	119	77	18	19	23	29	36	38	35	45	32	49	51	42	48	112	144	150	144
17	200	200	202	203	204	201	199	195	**188**	181	175	**160**	107	87	**36**	17	16	**20**	31	36	**34**	37	44	**40**	49	53	**41**	56	107	**140**	148	147
18	200	198	202	202	204	201	198	192	185	180	173	149	111	81	35	16	15	22	35	34	34	32	46	42	46	65	40	61	114	144	155	157
19	196	196	199	199	205	201	198	190	182	178	171	143	113	84	39	21	17	24	37	38	26	30	41	32	55	50	48	52	121	144	154	158
20	195	191	198	198	204	201	198	188	**178**	175	168	**142**	106	92	**40**	21	17	**24**	36	38	**24**	30	41	**37**	51	51	**45**	73	122	**147**	150	162
21	196	188	195	198	201	201	197	186	178	172	166	145	99	94	34	19	23	24	31	35	34	50	31	47	61	38	43	74	135	143	149	166
22	192	188	193	196	197	199	194	186	180	171	160	133	106	93	49	20	23	22	26	35	25	42	36	53	46	40	63	87	138	145	155	165
23	185	182	191	195	189	195	193	188	**182**	170	161	**124**	124	96	**87**	19	19	**24**	19	28	**28**	50	38	**64**	42	67	**71**	118	138	**145**	160	165
24	182	184	191	193	186	190	187	186	181	161	157	123	33	98	93	19	18	29	23	29	31	44	29	39	61	76	88	143	140	149	161	174
25	183	189	193	188	184	183	181	182	174	150	154	130	126	110	85	23	27	24	27	31	34	30	30	28	81	104	124	149	144	152	167	179
26	185	186	192	188	185	180	178	179	**161**	139	152	**138**	119	130	**79**	42	29	**28**	31	35	**35**	27	27	**39**	105	101	**151**	151	150	**155**	169	183
27	184	180	190	189	188	180	179	177	158	142	137	145	135	157	118	97	40	26	39	39	41	41	36	73	139	152	159	155	156	158	176	189
28	186	180	188	189	190	178	177	172	165	155	149	147	150	166	147	152	130	67	33	48	47	34	47	137	160	163	164	157	163	162	182	192
29	186	181	183	188	190	176	178	169	**153**	163	155	**143**	153	167	**161**	154	160	**157**	95	81	**91**	107	155	**166**	171	168	**166**	159	168	**168**	185	194
30	186	181	179	186	187	174	171	172	155	163	155	145	163	168	168	157	166	167	182	164	140	164	178	173	172	169	169	166	170	176	192	198
31	182	179	177	184	184	173	162	168	163	155	157	150	180	170	169	169	171	184	187	166	135	171	190	181	172	176	172	172	177	182	196	196

2. Pada matrik A tersebut lakukanlah pivoting

Matrik A

200	182	122	43	19	21	29	174
198	177	106	25	32	27	29	172
192	170	**109**	28	28	48	30	157
188	160	36	20	34	40	41	140
178	142	40	24	24	37	45	147
182	124	87	24	28	64	71	145
161	138	79	28	35	39	151	155
153	143	161	157	91	166	166	168

200	182	122	43	19	21	29	174
198	177	106	25	32	27	29	172
153	143	**161**	157	91	166	166	168
188	160	36	**20**	34	40	41	140
178	142	40	24	24	37	45	147
182	124	87	24	28	64	71	145
161	138	79	28	35	39	151	155
192	170	**109**	**28**	28	48	30	157

200	182	122	43	19	21	29	174
198	177	106	25	32	27	29	172
153	143	161	157	91	166	166	168
192	170	109	**28**	28	48	30	157
178	142	40	24	**24**	37	45	147
182	124	87	24	28	64	71	145
161	138	79	28	**35**	39	151	155
188	160	36	**20**	34	40	41	140

200	182	122	43	19	21	29	174
198	177	106	25	32	27	29	172
153	143	161	157	91	166	166	168
192	170	109	28	28	48	30	157
161	138	79	28	**35**	39	151	155
182	124	87	24	28	**64**	71	145
178	142	40	24	**24**	37	**45**	147
188	160	36	20	34	40	41	140

Gambar 3. 6 Alur proses pivoting pada matrik A

Sehingga didapatkan matrik A hasil pivoting:

200	182	122	43	19	21	29	174
198	177	106	25	32	27	29	172
153	143	161	157	91	166	166	168
192	170	109	28	28	48	30	157
161	138	79	28	35	39	151	155
182	124	87	24	28	64	71	145
178	142	40	24	24	37	45	147
188	160	36	20	34	40	41	140

Gambar 3. 7 matrik A hasil Pivoting

3. Setelah mendapatkan nilai matrik A lakukanlah perhitungan LU-Gauss untuk mendapatkan matrik segitiga bawah U dengan rumus LU-gauus (2.6).

R_{21}- $(\frac{R21}{R11})$ R_{11}

200	182	122	43	19	21	29	174
198	177	106	25	32	27	29	172
153	**143**	161	157	91	166	166	168
192	**170**	**109**	28	28	48	30	157
161	**138**	**79**	**28**	35	39	151	155
182	**124**	**87**	**24**	**28**	64	71	145
178	**142**	**40**	**24**	**24**	**37**	45	147
188	**160**	**36**	**20**	**34**	**40**	**41**	140

200	182	122	43	19	21	29	174
0	-41,62	-24,02	-15,13	10,71	44,89	44,61	-13,34
0	0	-72,29	-16,39	13,29	8,31	1,86	-20,01
0	0	0	107,88	89,48	161,53	149,54	15,55
0	0	0	0	21,17	21,47	15,23	6,29
0	0	0	0	0	21,25	-0,31	-10,33
0	0	0	0	0	0	107,62	15,4
0	0	0	0	0	0	0	15,36

Gambar 3. 8 Hasil perhitungan Eliminasi Gauss untuk mendapatkan matrik U

4. Setelah mendapatkan matrik U kita cari nilai *eigen values* matrik U, yang akan digunakan untuk normalisasi. Dengan rumus (2.3):

$$\Omega(l,w) = \sum_{k=1}^{w} ||u_{kk||}1 \leq l \leq w$$

nilai absolute dari nilai *eigen values* matrik U dengan rumus (2.4).

$$||Ukk|| = \sqrt{\sum UKK^2}$$

Dimana semua nilai *eigen values* matrik U semua berubah menjadi nilai positif.

200	182	122	43	19	21	29	174
0	**41,62**	-24,02	-15,13	10,71	44,89	44,61	-13,34
0	0	**72,29**	-16,39	13,29	8,31	1,86	-20,01
0	0	0	**107,88**	89,48	161,53	149,54	15,55
0	0	0	0	**21,17**	21,47	15,23	6,29
0	0	0	0	0	**21,25**	-0,31	-10,33
0	0	0	0	0	0	**107,62**	15,4
0	0	0	0	0	0	0	**15,36**

Gambar 3. 9 koefisien nilai tengah matrik A

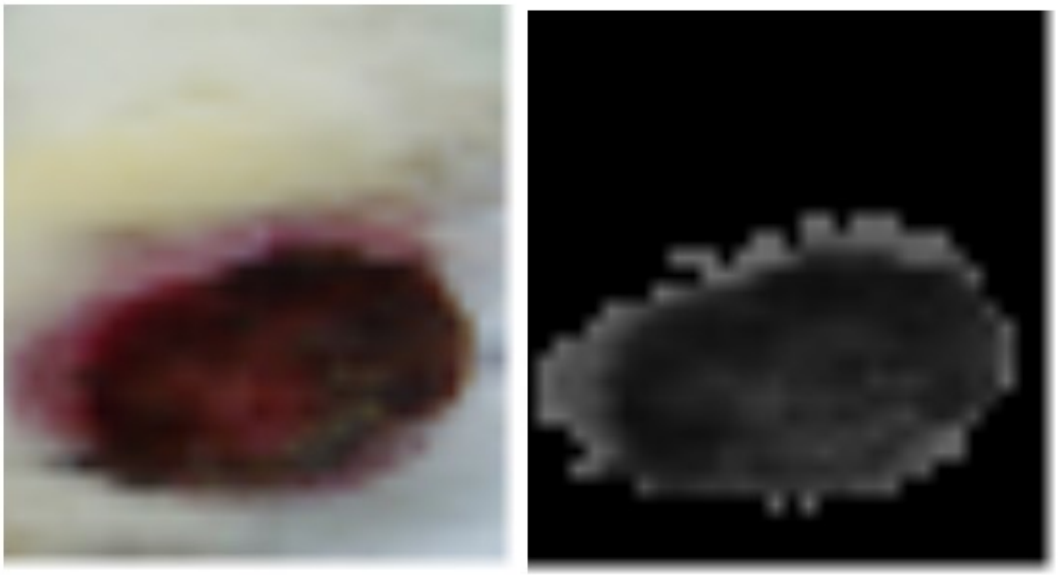
Nilai *eigen values* jumlahkan semua nilai tengah matrik U dimulai dari kordinat tengah ke 4 ($\ell = \frac{w}{2} = \frac{8}{2} = 4$)

$\sqrt{21{,}172^2 + 21{,}252^2 + 107{,}622^2 + 15{,}362^2}$ = 112,775

5. Setelah mendapatkan nilai *eigen values* maka langsung lakukan normalisasi gambar dengan rumus (2.5).

$$f(i,j) \begin{cases} \text{Gambar asli, } f(i,j) < ||Ukk|| \\ 0, \; f(i,j) > ||Ukk|| \end{cases}$$

Sehingga dapat dilihat gambar hasil *LU-Transform* pada gambar 3.10 dan nilai matrik gabar hasil LU-Transform pada tabel 3.3.



Gambar 3. 10 Hasil Proses LU-Transform penyakit sore hocks (data latih)

Tabel 3. 3 Nilai matrik hasil LU-Transform objek ke-1

f(i,j)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**107**	**111**	**0**	106	99	106	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**106**	**103**	**0**	**87**	**81**	**84**	**92**	**94**	**93**	**96**	**98**	**110**	**0**	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	**105**	**106**	**106**	**0**	**109**	**109**	**105**	**77**	**36**	**35**	**39**	**40**	**34**	**49**	**87**	**93**	**85**	**79**	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	**109**	**104**	**182**	**45**	**22**	**18**	**17**	**16**	**21**	**21**	**19**	**20**	**19**	**19**	**23**	**42**	**97**	0	0	0	0
16	0	0	0	0	0	0	0	0	**111**	**83**	**59**	**45**	**37**	**29**	**20**	**16**	**19**	**16**	**15**	**17**	**17**	**23**	**23**	**19**	**18**	**27**	**29**	**40**	0	0	0	0
17	0	0	0	0	0	**104**	**83**	**68**	**43**	**32**	**29**	**25**	**28**	**30**	**28**	**23**	**23**	**20**	**22**	**24**	**24**	**24**	**22**	**24**	**29**	**24**	**28**	**26**	**67**	0	0	0
18	0	0	0	0	**101**	**81**	**61**	**32**	**27**	**22**	**21**	**19**	**22**	**19**	**18**	**23**	**29**	**31**	**35**	**37**	**36**	**31**	**26**	**19**	**23**	**27**	**31**	**39**	**33**	**95**	0	0
19	0	0	**102**	**81**	**77**	**58**	**34**	**26**	**27**	**24**	**23**	**19**	**22**	**23**	**23**	**30**	**36**	**36**	**34**	**38**	**38**	**35**	**35**	**28**	**29**	**31**	**35**	**39**	**48**	**81**	0	0
20	0	**99**	**88**	**76**	**78**	**53**	**30**	**21**	**19**	**22**	**27**	**32**	**30**	**26**	**28**	**33**	**38**	**34**	**34**	**26**	**24**	**34**	**25**	**28**	**31**	**34**	**35**	**41**	**47**	**91**	0	0
21	0	**100**	**71**	**76**	**58**	**32**	**23**	**20**	**24**	**28**	**33**	**41**	**44**	**38**	**40**	**48**	**35**	**37**	**32**	**30**	**30**	**50**	**42**	**50**	**44**	**30**	**27**	**41**	**34**	**107**	0	0
22	0	**97**	**80**	**73**	**69**	**44**	**18**	**23**	**29**	**33**	**35**	**32**	**35**	**48**	**53**	**56**	**45**	**44**	**46**	**41**	**41**	**31**	**36**	**38**	**29**	**30**	**27**	**36**	**47**	0	0	0
23	0	**110**	**105**	**75**	**60**	**36**	**14**	**16**	**21**	**26**	**27**	**27**	**29**	**41**	**48**	**48**	**32**	**40**	**42**	**32**	**37**	**47**	**53**	**64**	**39**	**28**	**39**	**73**	0	0	0	0
24	0	0	0	**94**	**75**	**51**	**28**	**17**	**18**	**18**	**16**	**18**	**24**	**31**	**36**	**41**	**49**	**49**	**46**	**55**	**51**	**61**	**46**	**42**	**61**	**81**	**105**	0	0	0	0	0
25	0	0	0	0	**101**	**78**	**38**	**22**	**18**	**21**	**25**	**25**	**41**	**41**	**38**	**45**	**51**	**53**	**65**	**50**	**51**	**38**	**40**	**67**	**76**	**104**	**101**	0	0	0	0	0
26	0	0	0	**105**	**87**	**36**	**35**	**32**	**29**	**27**	**25**	**29**	**43**	**41**	**30**	**33**	**42**	**41**	**40**	**48**	**45**	**43**	**63**	**71**	**88**	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	**79**	**44**	**38**	**46**	**53**	**50**	**60**	**53**	**50**	**48**	**56**	**61**	**52**	**73**	**74**	**87**	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**103**	0	**107**	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3.4.2 Analisis Perhitungan Template matching correlation

3.4.2.1 Penentuan kelas target

Target kelas yang akan disimpan kedalam data latih berupa kelas *scabies* dan kelas *sorehock* yang dapat dipilih menggunakan *Combobox* yang telah disediakan. Setiap objek dalam kelas target memiliki nilai korelasi yang berbeda-beda. Nilai korelasi pada tiap objek tersebut yang akan menjadi target dalam pengujian pada *Template matching correlation* dilakukan.

3.4.2.2 Pelatihan Template matching correlation

Setelah kelas penyakit dipilih, data obejek akan disimpan. Data yang disimpan kedalam data latih *Template matching correlation* berupa nilai matrik hasil LU-Transform tiap pikelnya.

Pelatihan *Template matching correlation* objek ke-1 (kelas sorehocks). Data input matrik hasil LU-Transform penyakit sorehocks dapat dilihat pada tabel 3.3.

Data ini disimpan kedalam data latih *Template matching correlation* kelas sorehocks, dan dihitung nilai rata-rata citra tersebut yang digunakan untuk perbandingan nilai korelasi dihitung dengan rumus (2.8).

$$x_i = \frac{1}{n} \sum_{k=1}^{n} x_{ik}$$

$x_i = \frac{16567}{1024}$ = 16,1787

dengan nilai rata-rata objek 16,1787

1. Pelatihan objek ke-2 (kelas Scabies)

Pada gambar 3.11 dilihat penyakit *scabies*, Data matrik hasil *LU-Transform* penyakit scabies dengan matrik 32x32 dapat dilihat pada tabel 3.4.



Gambar 3. 11 Hasil Proses LU-Transform penyakit Scabies (data latih)

Tabel 3. 4 Nilai matrik hasil LU-Transform objek ke-2

f(i,j)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	84	80	75	58	79	84	94	68	82	72	70	94	66	73	74	61	62	83	58	68	48	63	61	59	66	63	71	95	75	72	76	105
1	59	84	64	91	0	55	105	78	96	86	83	81	71	80	70	70	71	61	69	70	71	49	69	68	65	71	101	55	74	89	100	86
2	57	64	76	94	83	69	83	85	90	103	46	78	84	76	78	65	60	68	70	73	77	105	86	70	0	0	81	89	88	84	0	105
3	88	71	74	87	81	92	78	97	81	97	0	0	0	88	70	73	105	76	66	79	68	81	61	0	82	0	74	106	98	0	103	0
4	102	83	77	79	92	96	88	89	104	0	102	104	0	0	0	95	93	53	79	101	95	76	62	92	68	69	73	96	0	106	103	0
5	93	73	70	71	79	87	96	90	88	104	91	0	0	0	0	0	0	98	90	83	71	78	61	73	88	74	98	100	0	103	0	0
6	72	71	63	74	67	68	96	81	98	94	86	103	93	0	0	0	0	0	0	96	85	83	80	96	0	0	0	99	72	90	103	83
7	71	63	77	85	96	96	0	100	88	0	100	0	93	0	0	0	0	0	0	0	0	96	85	68	98	86	0	98	105	0	0	0
8	48	55	78	107	0	0	0	0	0	0	0	0	94	91	101	0	0	0	0	0	0	0	0	103	0	0	0	0	104	0	0	0
9	47	59	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	101	0	0	0	0	0	0	0
10	68	74	84	106	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95	0	101	107
11	61	78	0	103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	106	99	107	106
12	64	90	0	82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0
13	63	96	85	0	0	0	0	106	0	103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	79	0	0
14	73	0	107	84	90	0	0	78	98	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	91
15	65	100	105	68	73	0	0	0	104	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	100	0	86
16	76	72	0	68	104	0	106	0	104	101	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	106	100	93	99
17	95	83	92	68	82	106	75	65	105	81	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	102	94
18	92	90	97	70	93	61	0	0	99	74	92	0	0	0	0	0	69	0	0	0	0	0	0	0	0	0	0	0	0	102	94	106
19	75	103	81	97	96	99	0	0	93	72	73	96	94	0	0	0	0	0	82	71	101	0	0	0	0	0	0	0	0	0	95	98
20	60	63	78	67	89	95	0	0	104	72	88	91	0	100	89	0	0	76	0	0	0	0	0	0	0	0	0	0	0	0	107	0
21	54	60	84	64	84	79	0	0	48	68	77	98	79	106	0	101	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	61	68	80	70	77	92	0	70	93	62	68	65	82	77	86	81	91	0	0	0	0	0	0	0	0	0	0	0	0	0	0	103
23	76	70	86	67	61	74	72	75	79	76	96	88	71	68	96	0	0	0	0	0	85	0	0	0	0	0	0	0	0	0	67	0
24	51	56	57	60	74	81	77	82	67	57	55	84	93	88	86	0	0	0	0	0	68	101	105	102	103	0	0	0	89	0	0	0
25	91	87	74	56	86	81	87	70	58	45	43	78	0	82	0	97	0	0	0	100	91	98	79	95	0	0	0	0	76	81	87	91
26	0	48	80	63	47	79	67	81	59	39	52	62	89	102	0	0	0	0	0	87	68	82	85	0	96	0	87	78	72	84	88	64
27	44	43	72	79	71	92	39	38	58	46	37	56	64	85	71	92	0	0	100	80	74	64	57	73	79	85	102	79	81	92	60	48
28	57	56	64	56	71	83	34	76	61	50	58	70	82	100	78	65	97	83	90	93	70	62	63	75	61	71	82	83	72	77	65	56
29	37	80	75	71	63	60	65	56	74	54	68	55	76	67	55	83	82	78	77	69	66	50	50	78	69	75	76	68	77	77	75	65
30	66	55	84	73	59	0	57	0	102	53	46	68	56	68	76	85	60	63	51	60	61	58	44	67	76	71	62	67	62	81	76	75
31	58	86	85	86	81	0	52	67	67	75	54	57	86	61	60	78	72	81	83	60	61	65	43	50	63	89	72	70	68	80	76	69

Data input matrik hasil *LU-Transform* penyakit *scabies* yang akan disimpan kedalam data latih *Template matching correlation* dihitung dengan rumus (2.8).

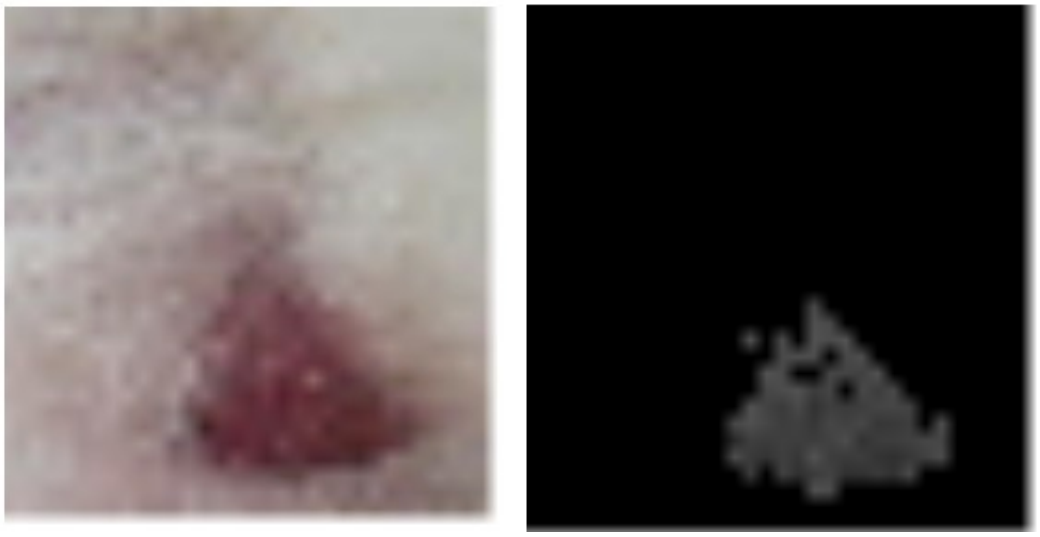
$$x_j = \frac{1}{n}\sum_{k=1}^{n} x_{jk}$$

$$x_j = \frac{47334}{1024} = 46{,}2246$$

dengan nilai rata-rata objek 46,2246

3.4.2.3 Pengujian Template matching correlation

Pada pengujian *Template matching correlation* ini menggunakan objek ke-3 penyakit *sore hocks* pada kelinci. Gambar contoh penyakit dapat dilihat pada gambar 3.12 dan dan matrik hasil *LU-Transform* dapat dilihat tabel 3.5.



Gambar 3. 12 Hasil Proses LU-Transform penyakit Scabies (data uji)

Tabel 3. 5 Nilai matrik hasil LU-Transform objek ke-3

f(i,j)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	74	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	79	0	71	80	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84	75	0	0	71	68	65	83	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	70	77	83	61	70	0	73	61	74	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	67	58	74	50	63	59	62	69	70	67	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	73	80	71	61	64	55	71	65	72	55	57	62	0	81	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	40	63	60	63	60	63	62	61	66	78	50	61	62	65	71	70	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	79	58	52	63	55	84	61	59	62	52	48	56	65	77	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	62	0	66	49	59	67	49	50	59	50	76	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82	77	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Data input matrik hasil *LU-Transform* penyakit *sore hocks* yang akan disimpan kedalam data latih *Template matching correlation* dihitung dengan rumus (2.8).

$$x_j = \frac{1}{n}\sum_{k=1}^{n} x_{jk}$$

$x_j = \frac{5755}{1024} = 5{,}6201$

dengan nilai rata-rata objek 5,6201

Lakukan perhitungan *Template Matcing Correlation* untuk mendapatkan nilai korelasi, yang berfungsi membandingkan dengan nilai korelasi uji dengan data latih. Menggunakan rumus (2.7). Nilai korelasi berkisar antara -1 hingga 1, semakin mendekati nilai 1 maka tingkat kemiripan semakin besar. Pertama lakukan perhitungan perbandingan data uji objek ke-3 dengan penyakit *sore hocks* data latih objek ke-1.

$$r = \frac{\sum_{k=1}^{n}(x_{ik} - x_i) * (x_{jk} - x_j)}{\sqrt{[\sum_{k=1}^{n}(x_{ik} - x_i)^{\,2} * \sum_{k=1}^{n}(x_{jk} - x_j)^{\,2}]}}$$

$$r = \frac{(0\text{-}5{,}6201)\text{x}(0\text{-}16{,}1787)+(0\text{-}5{,}6201)\text{x}(0\text{-}16{,}1787)+ \ldots+(0\text{-}5{,}6201)\text{x}(0\text{-}16{,}1787)}{\sqrt{(0-5{,}6201)^2+(0-5{,}6201)^2+ \ldots +(0-5{,}6201)^2 \text{ x } (0-16{,}1787)^2+(0-16{,}1787)^2+ \ldots.+(0-16{,}1787)^2}}$$

$$r = \frac{455731{,}2}{\sqrt{356995{,}2 \text{ x } 791144{,}3}}$$

$$r = \frac{455731{,}2}{\sqrt{282434717607{,}36}}$$

$$r = \frac{455731{,}2}{531445{,}87}$$

$r = 0{,}0857$

Kedua lakukan perhitungan perbandingan dengan penyakit scabies objek ke-2 (data latih).

$$r = \frac{\sum_{k=1}^{n}(x_{ik} - x_i) * (x_{jk} - x_j)}{\sqrt{[\sum_{k=1}^{n}(x_{ik} - x_i)^{2} * \sum_{k=1}^{n}(x_{jk} - x_j)^{2}]}}$$

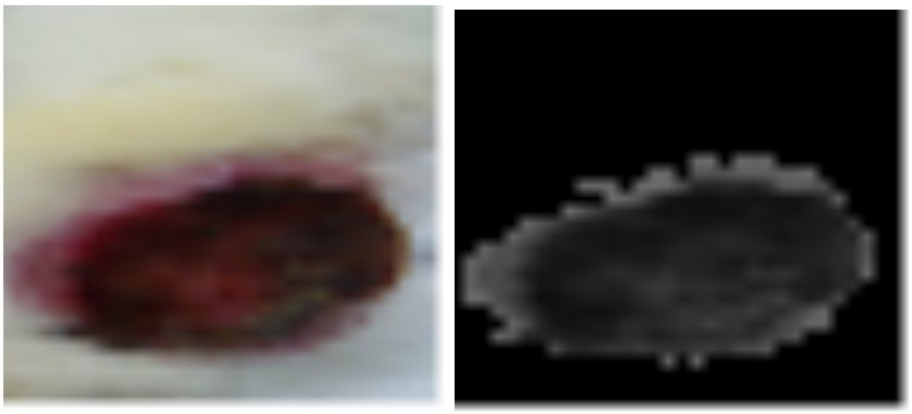
$$r = \frac{(0\text{-}5{,}6201)x(84\text{-}46{,}2246)+(0\text{-}5{,}6201)x(80\text{-}46{,}2246)+...+(0\text{-}5{,}6201)x(69\text{-}46{,}2246)}{\sqrt{(0-5{,}6201)^2 + (0-5{,}6201)^2 + \ldots + (0-5{,}6201)^2 x (84-46{,}2246)^2 + (80-46{,}2246)^2 + \ldots + (69-46{,}2246)^2}}$$

$$r = \frac{896104{,}7}{\sqrt{356995{,}2 \text{ x } 1090548{,}66}}$$

$$r = \frac{896104{,}7}{\sqrt{389322636986{,}43}}$$

$$r = \frac{896104{,}7}{623955{,}64}$$

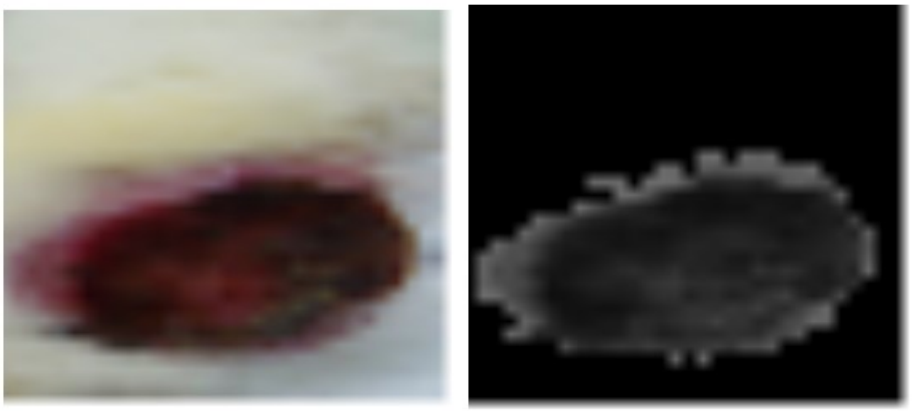
r = 0,01436



Gambar 3. 13 Data latih objek ke-1 penyakit sore hocks



Gambar 3. 14 Data latih objek ke-2 penyakit scabies



Gambar 3. 15 Data uji objek ke-3 penyakit sore hocks

Dari hasil pengujian *Template Matcing Corelation* tersebut, perbandingan antara data uji objek ke-3 dengan data latih objek ke-1 (penyakit sore hocks), menunjukan bila perbandingan data uji objek ke-3 dengan data latih objek ke-1, menunjukan nilai korelasi 0,0857.

Sedangkan hasil nilai korelasi perbandingan data uji objek ke-3 dengan data latih objek ke-2 (penyakit scabies) menunjukan nilai korelasi 0,01436. Dari hasil pengujian *Template Matcing Correlation* dengan data uji objek ke-3 disimpulkan bila data uji objek ke-3 termasuk kedalam klasifikasi penyakit *sore hocks*. Dengan ketentuan dimana nilai korelasi berkisar antara -1 hingga 1, semakin mendekati nilai 1 maka tingkat kemiripan semakin besar.

3.5 Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam implementasi metode *LU-Transform* dan *Template Matcing Correlation* untuk mendeteksi penyakit kulit pada kelinci ini meliputi beberapa kebutuhan, diantaranya:

3.5.1 Analisis Kebutuhan Non Fungsional

Kebutuhan *non fungsional* yang dibutuhkan untuk implementasi kedalam sistem mencakup dua hal yakni kebutuhan perangkat keras dan perangkat lunak.

Adapun spesifikasi perangkat keras yang dibutuhkan untuk implementasi kedalam sistem antara lain :

1. Processor minimal memiliki kecepatan 2 GHz.
2. RAM berkapasitas minimal 2 GB.
3. Hard disk (HDD) berkapasitas minimal 20 GB atau disesuaikan dengan sistem operasi yang digunakan.

3.5.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk implementasi kedalam sistem antara lain :

1. Sistem Operasi Windows 7 atau Windows 8, Windows 8.1.
2. Visual Studio 2012

3.5.3 Analisis Kebutuhan Fungsional

Fungsional merupakan tindakan mengidentifikasi proses yang akan berjalan didalam sistem, yang akan dikembangkan dan menjelaskan kebutuhan yang diperlukan. Sehingga sistem dapat berjalan seperti yang diharapkan.

Analisis fungsional dimodelkan dengan menggunakan UML (Unified Modeling Language). Tahapan pemodelan dalam analisis tersebut antara lain mengidentifikasi aktor, pembuatan diagram use case, scenario use case, diagram sequence, diagram class dan diagram package.

3.6 Perancangan Sistem

3.6.1 Identifikasi Aktor

Aktor yang teridentifikasi dan terlibat dalam sistem hanya ada satu yakni *user*. Karakteristik aktor dapat di lihat dari tabel berikut.

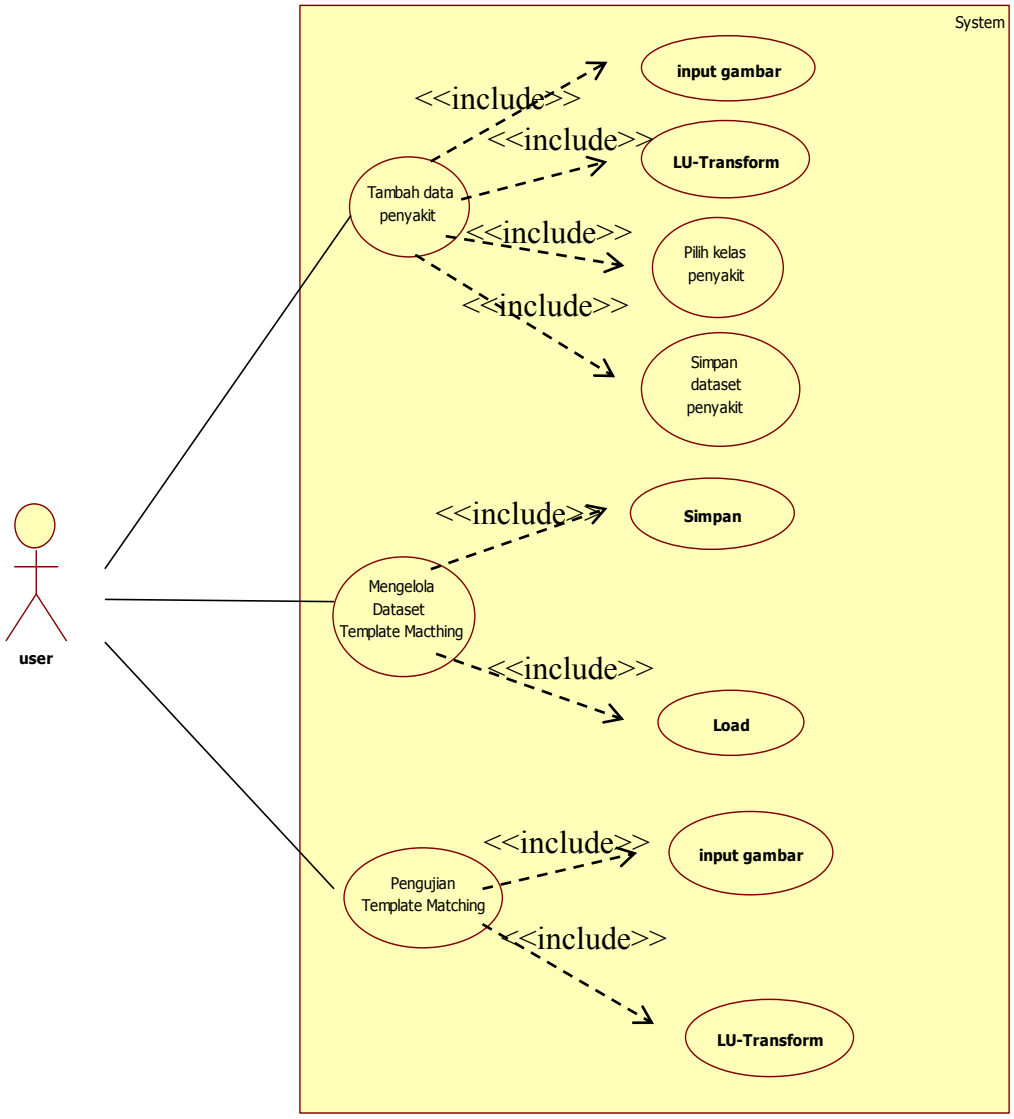
Tabel 3. 6 Identifikasi Aktor

Kategori Pengguna	Tugas	Hak Akses	Kemampuan yang harus dimiliki
User	Menjalankan Aplikasi Pendeteksi Penyakit Kulit Kelinci	Memasukan data jenis-jenis penyakit kulit pada kelinci Mengelola dat-data penyakit yang sudah ada Menjalankan aplikasi untuk mengecek jenis-jenis penyakit kulit pada kelinci	Pengetahuan dasar tentang jenis-jenis penyakit kulit pada kelinci Kecakapan dalam menjalankan aplikasi

3.6.2 Diagram Use Case

Use Case adalah interaksi yang terjadi antara sistem dan aktor yang terlibat, didalamnya termasuk pertukaran pesan dan tindakan yang dilakukan oleh sistem. *Use Case diagram* menggambarkan setiap aktivitas yang dilakukan oleh sistem dari sudut pengamatan tertentu.

Use case diagram dalam sistem ini dapat dilihat pada gambar 3.16 menunjukan diagram usecase untuk sistem ini.



Gambar 3. 16 Diagram Use case

3.6.3 Definisi Use Case

Use case berfungsi untuk mewakili apa yang sistem bisa lakukan. Definisi *use case* bisa dilihat pada tabel 3.7

Tabel 3. 7 Definisi use case

No	*Use Case*	Deskripsi
1	Tambah Data Penyakit	Untuk melakukan proses Tambah Data Penyakit : 1. Input gambar 2. Melakukan proses LU-Transform 3. Melakukan proses pemilihan kelas penyakit 4. Simpan dataset penyakit
2	Mengelola Dataset Template Macthing Correlation	Untuk melakukan proses Mengelola Dataset Template Macthing Correlation: 1. Pengguna menekan tombol *load dataset* 2. Sistem menampilkan dataset 3. Pengguna menekan tombol *Simpan* 4. Sistem akan menyimpan data training
3	Pengujian Template matching correlation	Untuk melakukan proses Pengujian Template matching correlation: 1. Pengguna menekan tombol *input gambar* 2. Sistem menampilkan gambar yang diinputkan 3. Pengguna menekan tombol *testing* 4. Sistem akan memproses dan menampilkan hasil testing

3.6.4 Skenario Use Case

Skenario use case merupakan skenario (flow of event) interaksi aktor yang terlibat dengan beberapa usecase utama. Skenario usecase menggambarkan urutan interaksi aktor dengan use case tersebut, dari awal sampai akhir. Skenario usecase utama untuk implementasi sistem dapat dilihat dibawah ini:

Skenario Use Case

Use case : Tambah Data Penyakit

Aktor : User

Pre-condition : aktor ingin menginputkan jenis-jenis penyakit kulit kelinci pada aplikasi

Post-condition : aktor telah melakukan jenis-jenis penyakit kulit kelinci kedalam aplikasi

Deskripsi : aktor memasukan data jenis-jenis penyakit pada aplikasi untuk menjadi sumber dataset pada aplikasi pendeteksi penyakit kulit tersebut.

Tabel 3. 8 Skenario Use case Menginput data

Identifikasi		
No	1	
Nama	Tambah Data Penyakit	
Tujuan	Menambah dataset Penyakit	
Deskripsi	Proses untuk menambahkan dataset	
Aktor	User	
Skenario Utama		
Kondisi Awal	User berada di tampilan Tekstur Deteksi LU-Transform	
Aksi Aktor		Reaksi Sistem
1. User memasukan gambar inputan		

	2. Menampilkan gambar inputan
3. User menekan tombol LU-Transform	
	4. Memproses LU-Transform
	5. Menampilkan hasil LU-Transform dengan Matriknya
6. Memilih kelas penyakit	
7. User menekan tombol simpan	
Kondisi akhir	8. melakukan proses simpan dataset

Use case : mengelola data penyakit

Aktor : user

Pre-condition : aktor ingin mengelola jenis-jenis penyakit kulit kelinci pada dataset penyakit kelinci

Post-condition : aktor telah memastikan kecocokan jenis-jenis penyakit kulit kelinci pada aplikasi

Deskripsi : aktor mengelola data jenis-jenis penyakit pada aplikasi, aktor dapat load dataset yang telah disimpan dalam memory komputer. Jugadapat menyimpan dataset baru pada komputer.

Tabel 3. 9 Skenario Use case Mengelola data penyakit

Identifikasi	
No	2
Nama	Mengelola data penyakit
Tujuan	Mendapatkan nilai training
Deskripsi	Proses untuk mendapatkan nilai training

Aktor	User
Skenario Utama	
Kondisi Awal	User berada di tampilan Dataset Template Macthing Correlation
Aksi Aktor	Reaksi Sistem
1. menekan tombol load dataset	
	2. menampilkan dataset
3. menekan tombol Simpan	
	4. Melakukan proses Simpan
Kondisi akhir	5. Menampilkan dataset

Use case : Pengujian Template matching correlation

Aktor : user

Pre-condition : aktor ingin mencoba mendeteksi jenis-jenis penyakit kulit kelinci pada aplikasi

Post-condition : aktor mendapatkan klasifikasi jenis-jenis penyakit kulit kelinci apa yang telah dideteksi pada aplikasi

Deskripsi : aktor mencoba mengecek jenis-jenis penyakit kulit pada aplikasi, dan aktor mendapatkan deskripsi jenis penyakit apa pada data yang baru diambil sebelumnya.

Tabel 3. 10 Skenario Use case Pengujian Template matching correlation

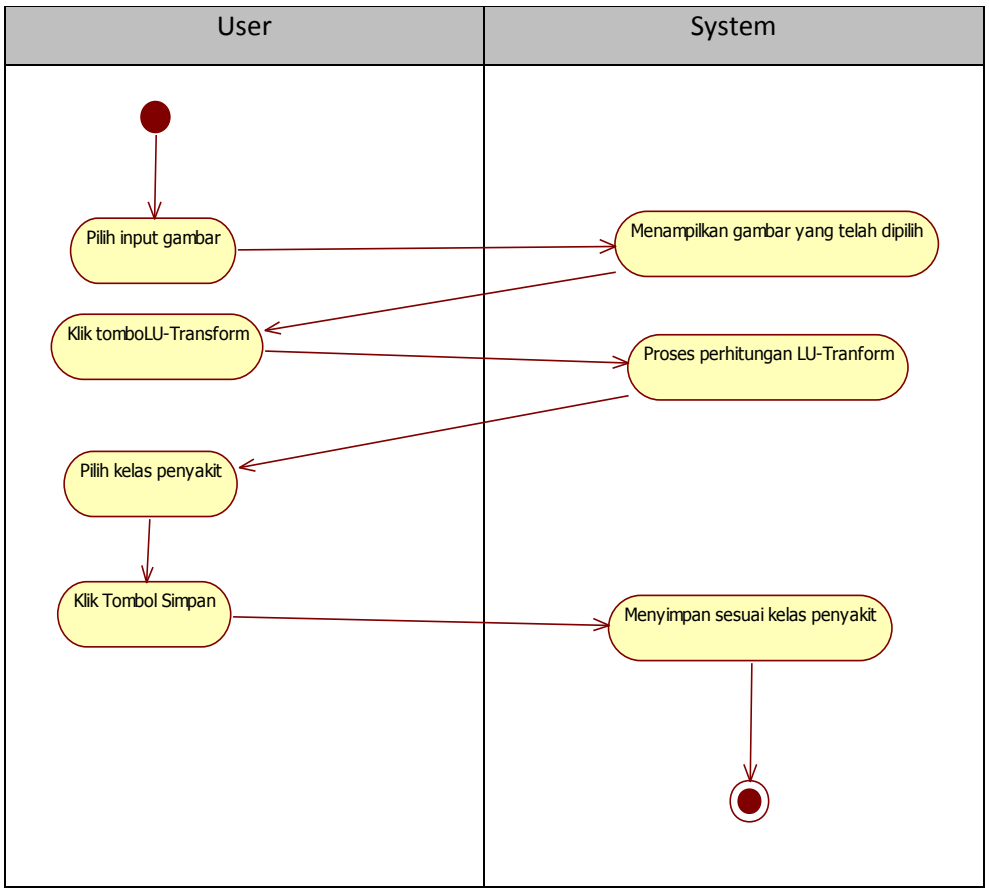
Identifikasi	
No	3
Nama	Pengujian Template matching correlation
Tujuan	Memperlihatkan hasil pengujian
Deskripsi	Proses untuk mendapatkan hasil pengujian
Aktor	User
Skenario Utama	
Kondisi Awal	User berada di tampilan Pengujian Template matching correlation

Aksi Aktor	Reaksi Sistem
1. Memasukan citra testing	
	2. Menampilkan citra testing
3. Menekan tombol Pengujian	
	4. Melakukan proses LU-Transform
	5. Melakukan proses Pengujian Template matching correlation
Kondisi akhir	6. Menampilkan hasil Pengujian Template matching correlation

3.6.5 Activity Diagram

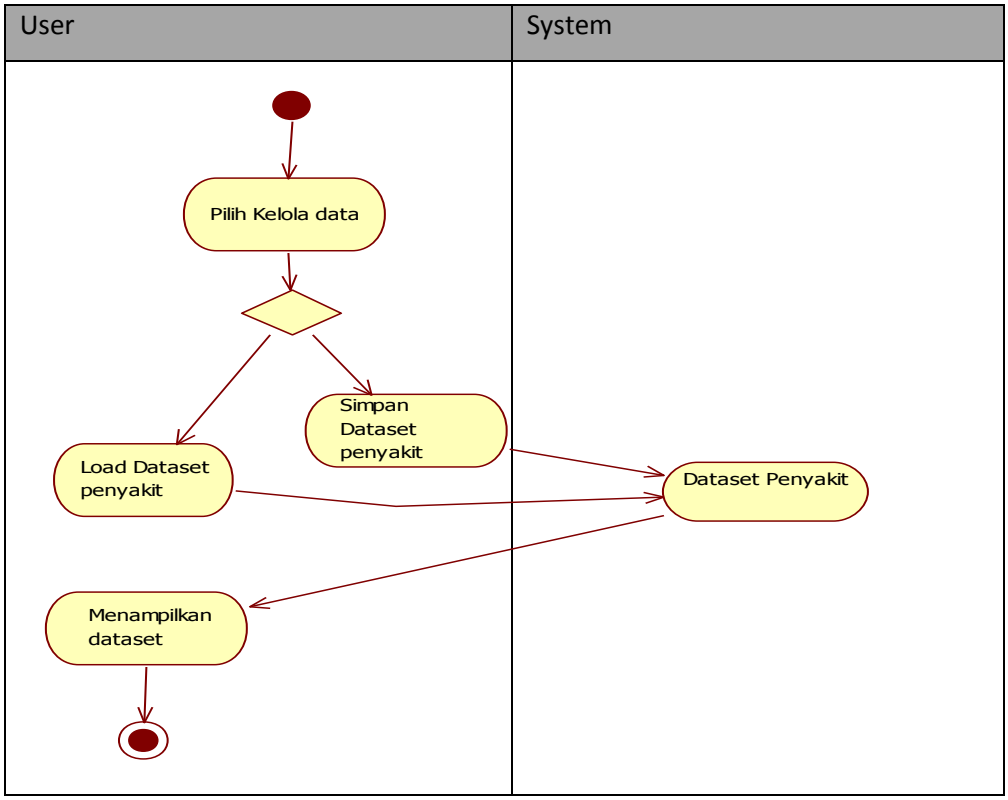
Activity diagram merupakan teknik untuk menggunakan logika prosedural, proses bisnis, dan jalur kerja. Dari skenario yang dibuat, berikut akan digambarkan activity diagramnya. Activity diagram untuk implementasi sistem terbagi menjadi 3 bagian diantaranya:

1. Input data



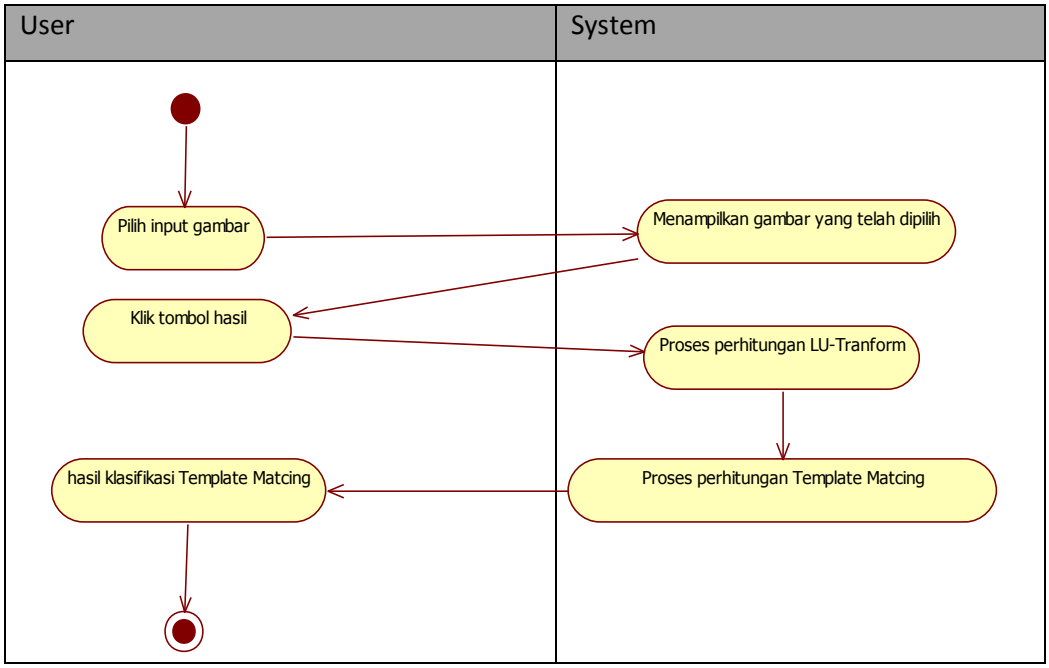
Gambar 3. 17 Activity diagram input data

2. Mengelola data



Gambar 3. 18 Activity diagram mengelola data

3. Menjalankan Aplikasi Pendeteksi Penyakit Kelinci

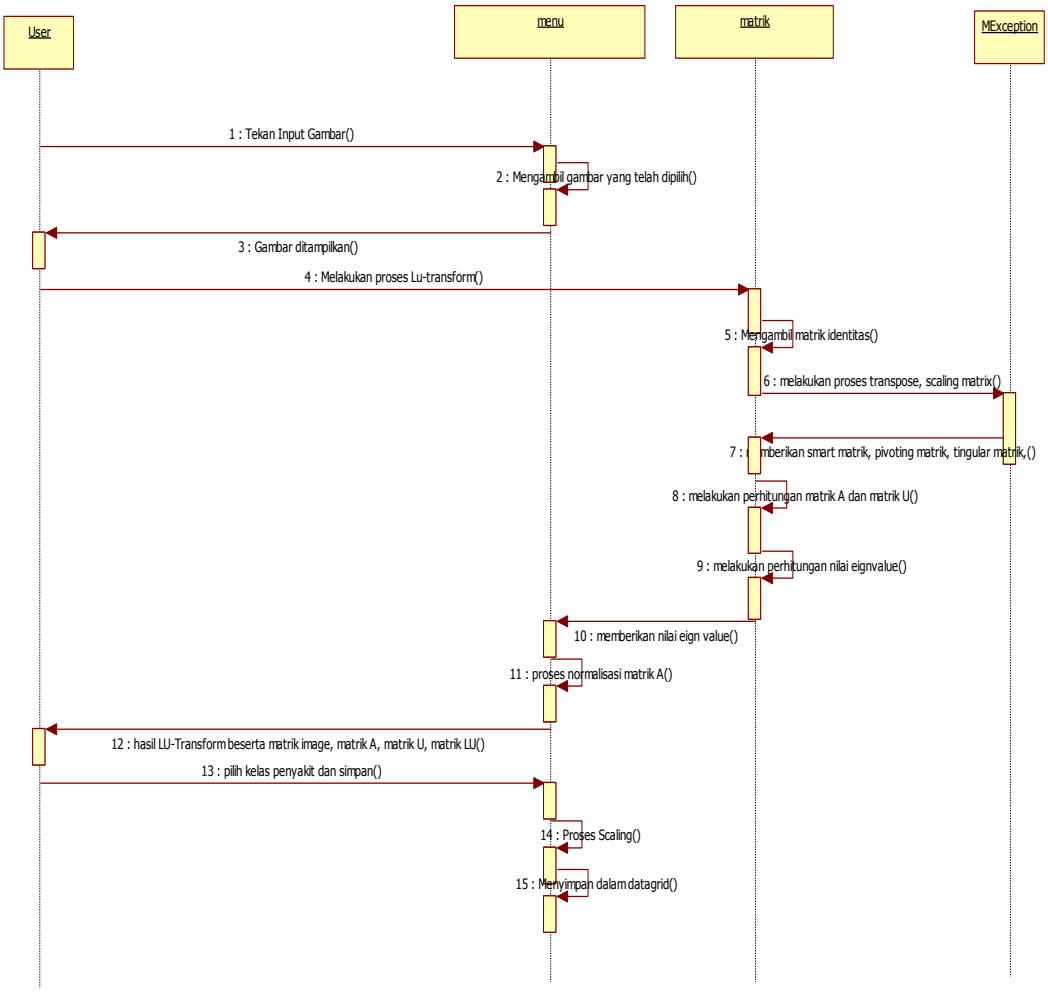


Gambar 3. 19 Activity diagram Pengujian Template Matching

3.6.6 Sequence Diagram

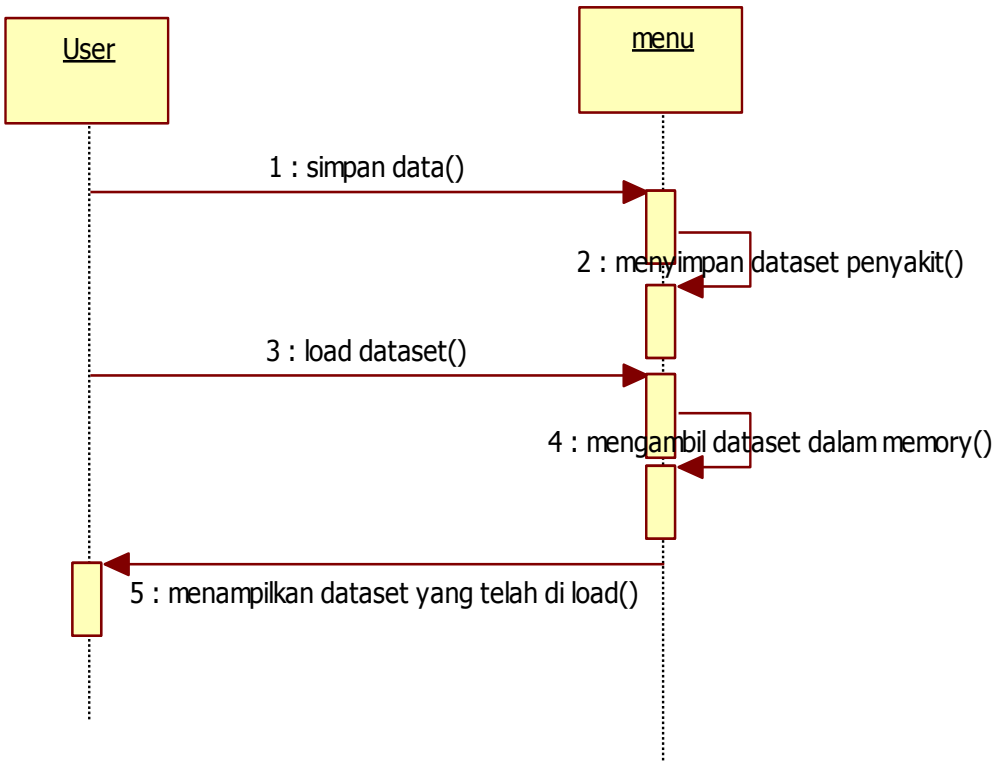
Sequence diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Diagram sequence untuk implemenasi sistem dibagi atas tiga bagian diantaranya:

1. Input data



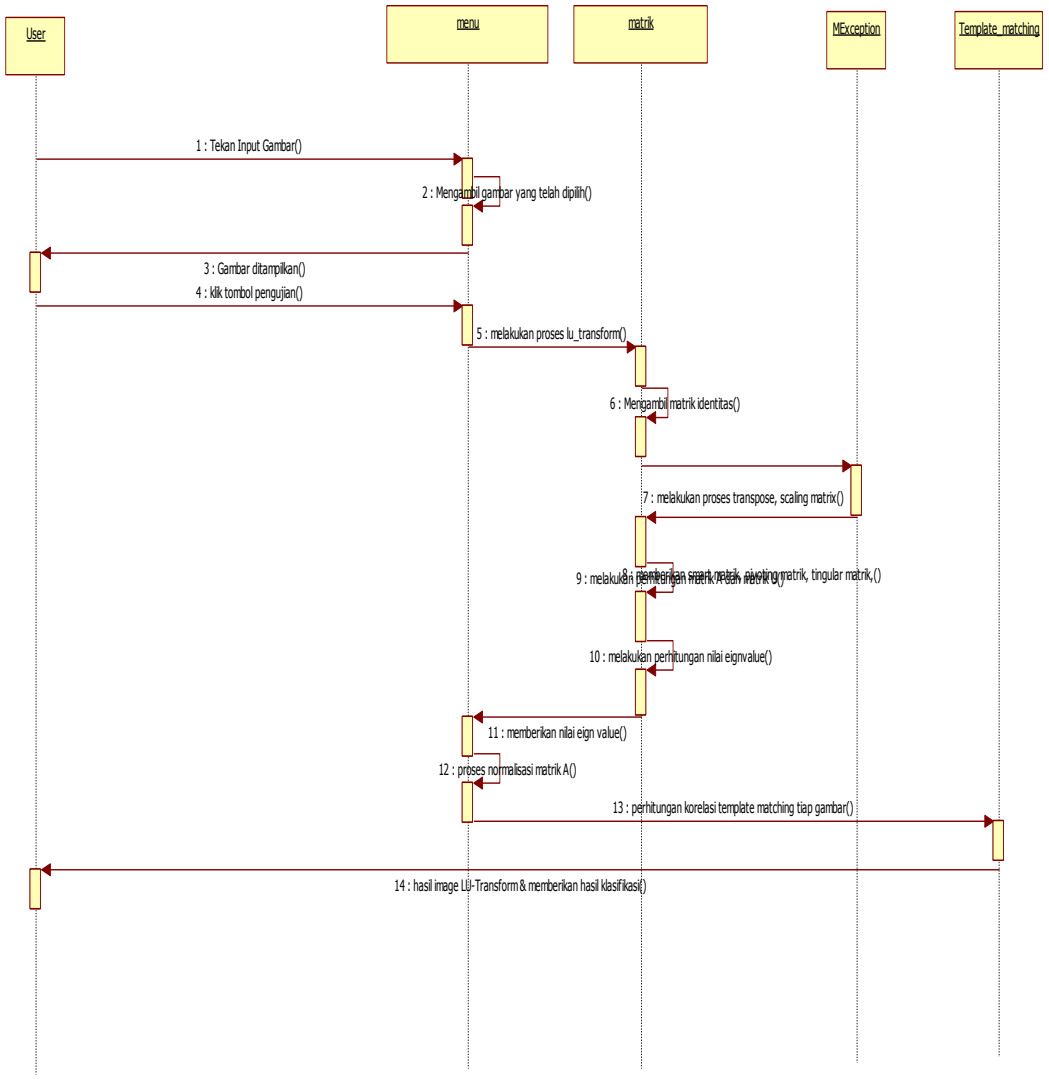
Gambar 3. 20 Sequence diagram Input data

2. Mengelola data



Gambar 3. 21 Sequence diagram Mengelola data

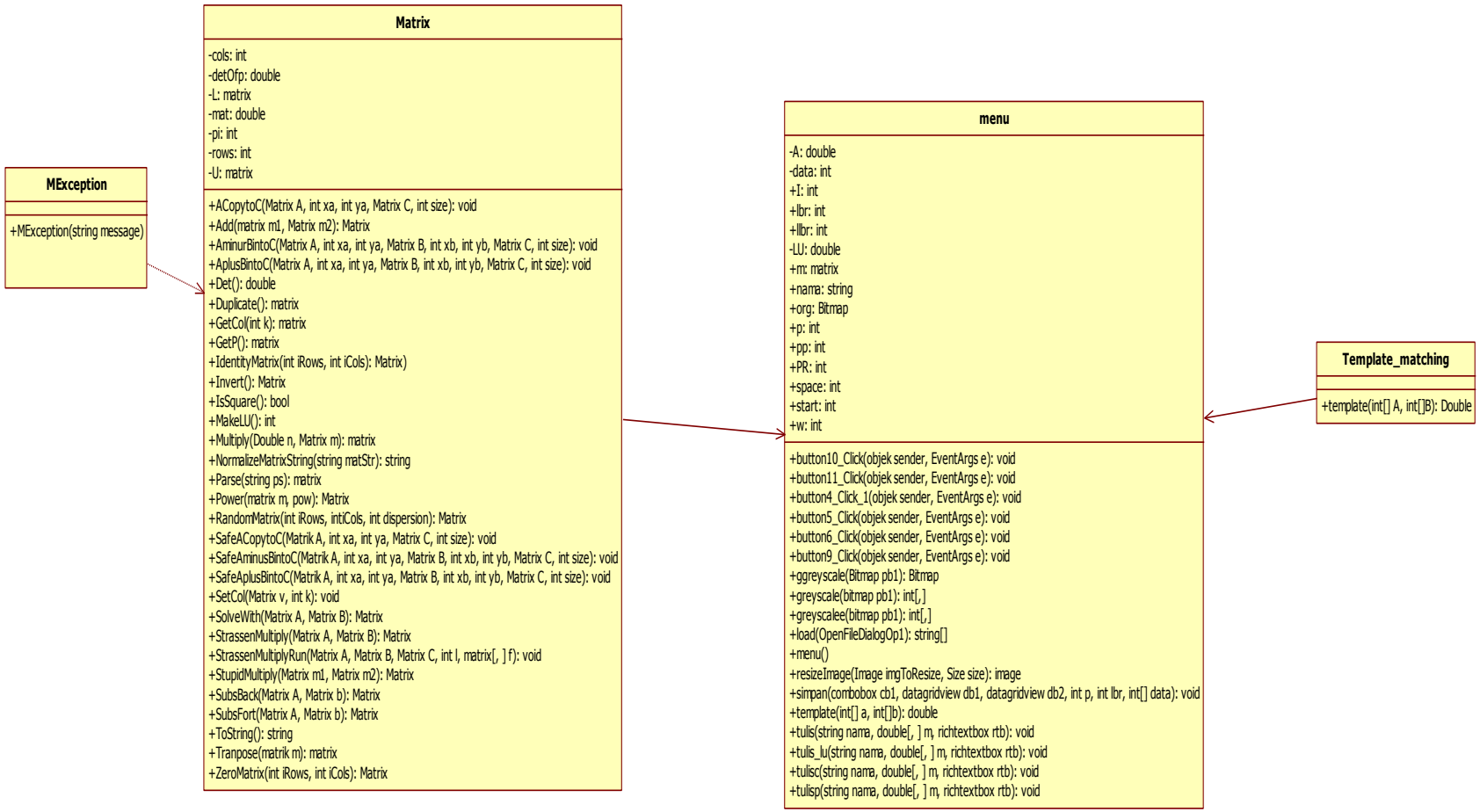
3. Menjalankan Pendeteksi Penyakit Kulit



Gambar 3. 22 Sequence diagram Pengujian Template matching correlation

3.6.7 Class Diagram

Class diagram mendeskripsikan jenis–jenis objek dalam sistem dan berbagai macam hubungan statis yang ada. Class diagram beserta deskripsinya untuk implementasi sistem ini dapat dilihat pada gambar 3.23.



Gambar 3. 23 Diagram kelas

Penjelasan dari masing-masing class diagram dapat dilihat pada tabel 3.16

Tabel 3. 11 Dekripsi class diagram

User Application		
Class	**Jenis Class**	**Deskripsi**
Menu	View	Class yang menampilkan halaman awal aplikasi
Matrix	Control	Class yang berisikan ukuran matirx identitas, matrik LU, matrik P, matrik U, Matrik L, inisialisasi array
Mexceptions	Control	Class yang berisikan tranpose matrix, tringular matrix, dan inisialisasi matrix.
Template_matching	Control	Class yang berisikan perhitungan nilai korelasi template matching correlation

3.7 Perancangan sistem

Perancangan sistem merupakan penggambaran dan perencanaan dari beberapa elemen sistem yang terpisah ke dalam suatu kesatuan model yang utuh. Tahapan ini meliputi perancangan antarmuka dan jaringan semantik.

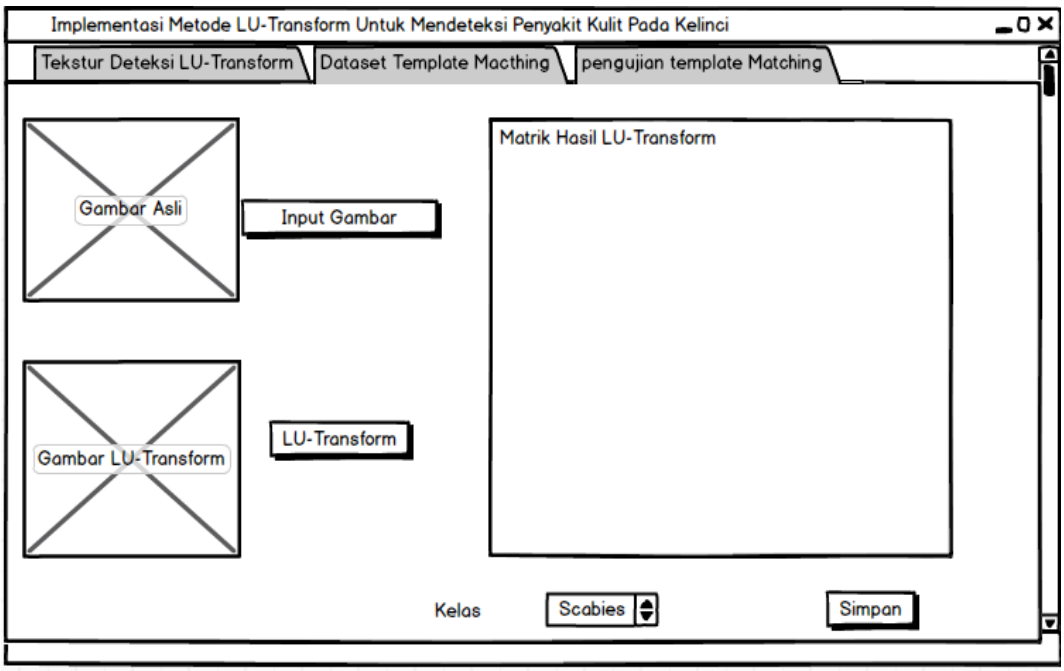
3.7.1 Perancangan Antar Muka

Perancangan antarmuka mendeskripsikan rencana tampilan dari setiap tampilan yang akan digunakan. Perancangan terdiri dari perancangan form dan pesan.

3.7.1.1 Perancangan form

1. Form menu utama

NO. T01



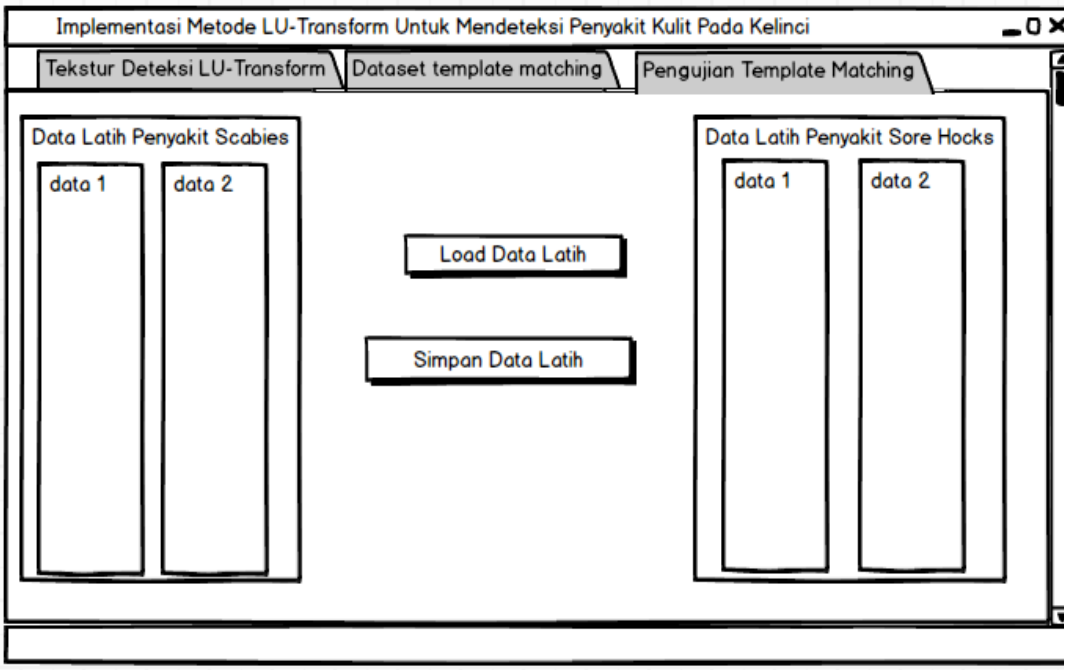
Keterangan:

- Input gambar : Tombol untuk mengambil foto penyakit kulit pada kelinci dari komputer
- LU-Transform : Tombol untuk melakukan proses perhitungan LU-Transform
- Combobox kelas: Tombol pilih kelas penyakit.
- Simpan : Tombol untuk menyimpan data latih sesuai dengan kelasnya
- Tabs bar Dataset Template Matching : Tombol untuk menuju form dataset
- Tabs bar Pengujian Template Matching : Tombol untuk menuju form pengujian
- Gambar asli : gambar asli yang ditampilkan
- Gambar LU-Transform: gambar hasil proses LU-transform yang ditampilkan

Gambar 3. 24 Form Menu Utama

2. Form data pelatihan template matching correlation

NO. T02

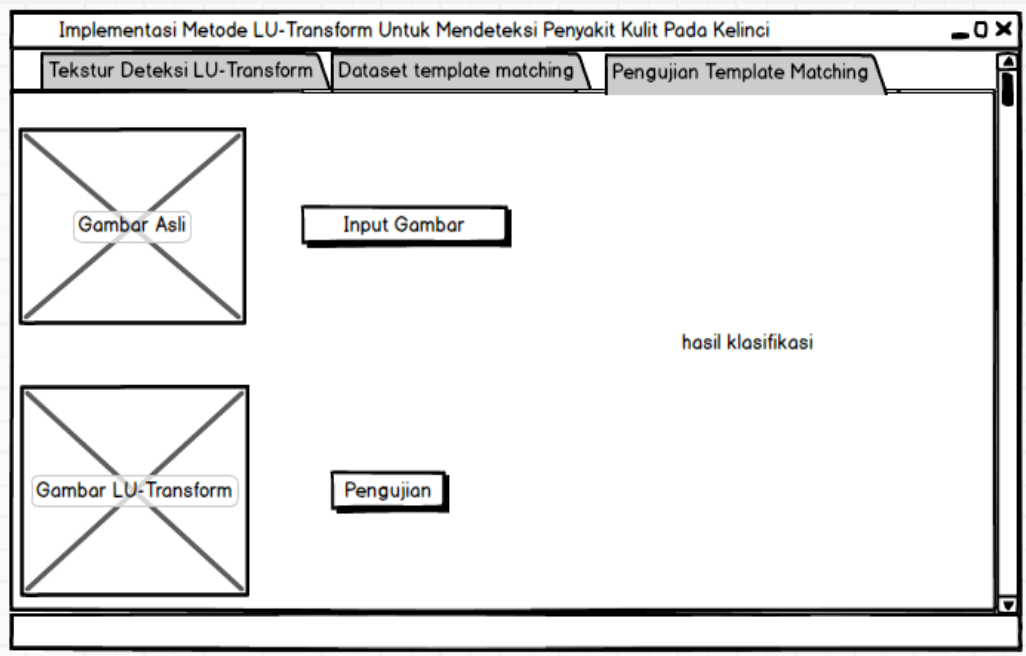


- Load data Latih : tombol untuk mengambil data-data bobot image template matching.
- Simpan data Latih : tombol untuk menyimpan file data latih template matching.
- Tabs bar Dataset Tektur deteksi LU-Transform : Tombol untuk menuju form tambah data
- Tabs bar Pengujian Template Matching : Tombol untuk menuju form pengujian
- Text area data latih penyakit scabies: menampilkan data-data penyakit scabies

Gambar 3. 25 Form menu data latih

3. Form pengujian template matching correlation

NO. T03



- Input gambar : tombol untuk mengambil foto penyakit kulit pada kelinci dari komputer
- Pengujiann : tombol untuk mendapatkan hasil klasifikasi penyakit kelinci hasil klasifikasi template matching
- Tabs bar Dataset Tektur deteksi LU-Transform : Tombol untuk menuju form tambah data
- Tabs bar Dataset Template Matching : Tombol untuk menuju form dataset
- Text hasil klasifikasi: menampilkan hasil pengujian
- Ukuran 1182: 769

Gambar 3. 26 Form Menu Data Uji

3.7.1.2 Perancangan pesan

1. **Perancangan pesan saat menyimpan data**

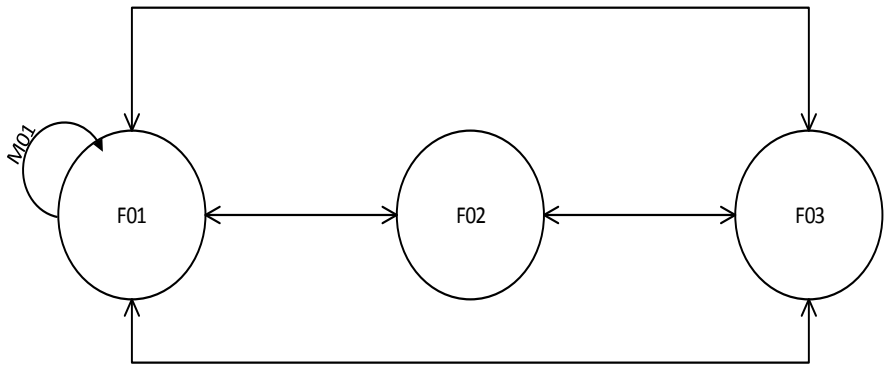
No M01
Data Telah tersimpan
Font 8,25, Ms Serif, warna hitam

Gambar 3. 27 Pesan simpan data

Gambar 3.27 menjelaskan bahwa objek penyakit telah disimpan kedalam dataset.

3.7.2 Jaringan Semantik

Jaringan Semantik merupakan gambaran pengetahuan grafis yang menunjukkan hubungan antar berbagai objek. Jaringan semantik terdiri dari lingkaran-lingkaran yang menunjukkan objek dan informasi tentang objek-objek tertentu. Jaringan semantik pada sistem ini dapat dilihat pada gambar 3.28.



Gambar 3. 28 Jaringan Semantik