

BAB III

ANALISIS DAN PERANCANGAN

3.1. Analisis

Sebelum memulai proses perancangan perangkat lunak, maka terlebih dahulu dilakukan analisis perangkat lunak yang akan dibangun. Perangkat lunak terdiri atas dua bagian yaitu aplikasi dan pengujian. Proses analisis yang dilakukan mencakup analisis proses dan analisis kebutuhan.

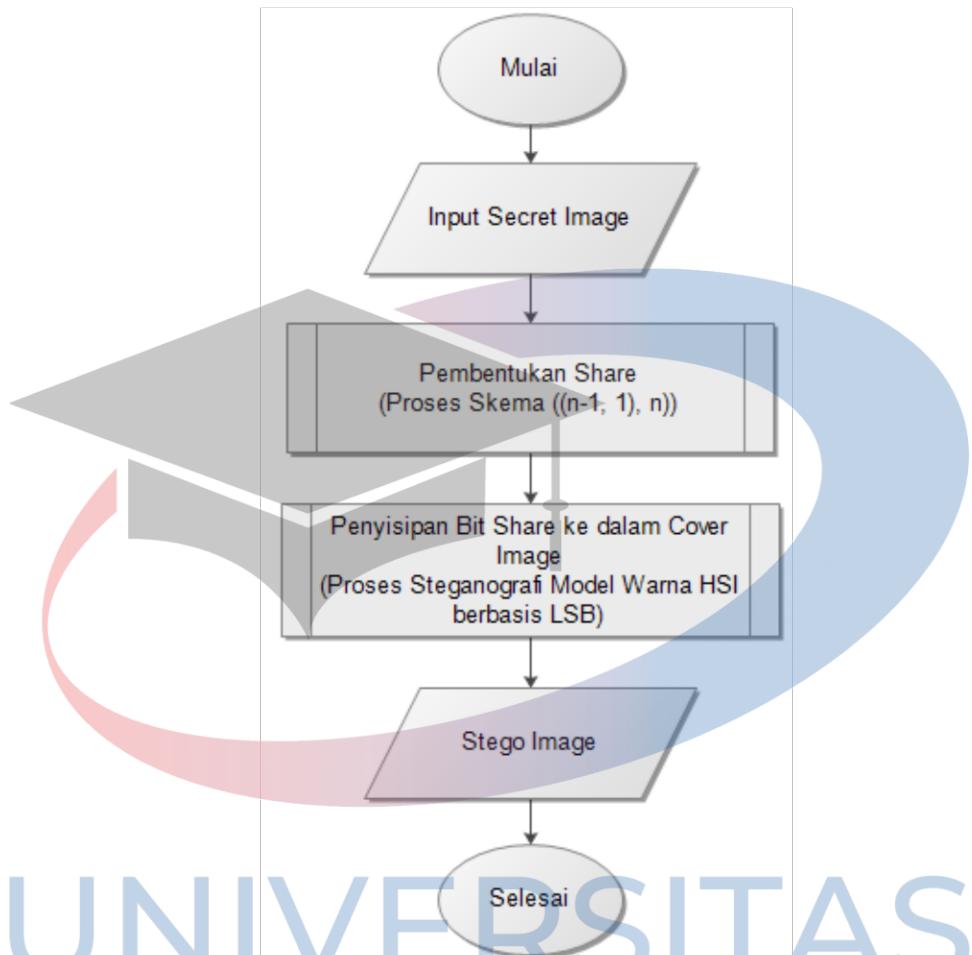
3.1.1. Analisis Proses

Analisis proses digunakan untuk menjelaskan proses kerja pada perangkat lunak untuk menyelesaikan permasalahan yang ada, meliputi proses *embedding* dan ekstrak terhadap *secret image*.

3.1.1.1. Analisis Proses *Embedding*

Proses *embedding* bertujuan untuk menghasilkan *stego image*, yaitu melakukan pembentukan *share* dari *secret image* kemudian membungkusnya ke dalam *cover image*. Proses *embedding* meliputi: pembentukan *share* dan penyisipan *share*. Proses pembentukan *share* menggunakan skema $((n-1, 1), n)$ dimana $(n-1)$ *natural image* digunakan sebagai *input* untuk menghasilkan *share*. Pada proses ini terdiri dari dua tahapan yaitu Fitur Ekstraksi dan Enkripsi. Pada Fitur Ekstraksi, $(n-1)$ *natural image* dilakukan ekstraksi fitur sehingga menghasilkan $(n-1)$ *natural share image*, kemudian $(n-1)$ *natural share image* digabungkan. Pada tahap Enkripsi, hasil penggabungan *natural share image* dilakukan operasi XOR dengan *secret image* sehingga menghasilkan *share image*. Proses penyisipan *share* menggunakan citra warna HSI dengan teknik LSB. Pada tahap ini, *share image* dan *natural share image* yang telah dihasilkan serta *password* akan dibungkus ke dalam *cover image*. Proses pembungkusan dilakukan dengan cara melakukan konversi *cover image* (RGB) ke citra warna HSI, kemudian bit *share image*, *natural share image* dan *password* disisipkan ke nilai intensitas citra HSI

sehingga menghasilkan n buah *stego image*. Secara garis besar, proses *embedding* dapat dilihat pada Gambar 3. 1 berikut ini:

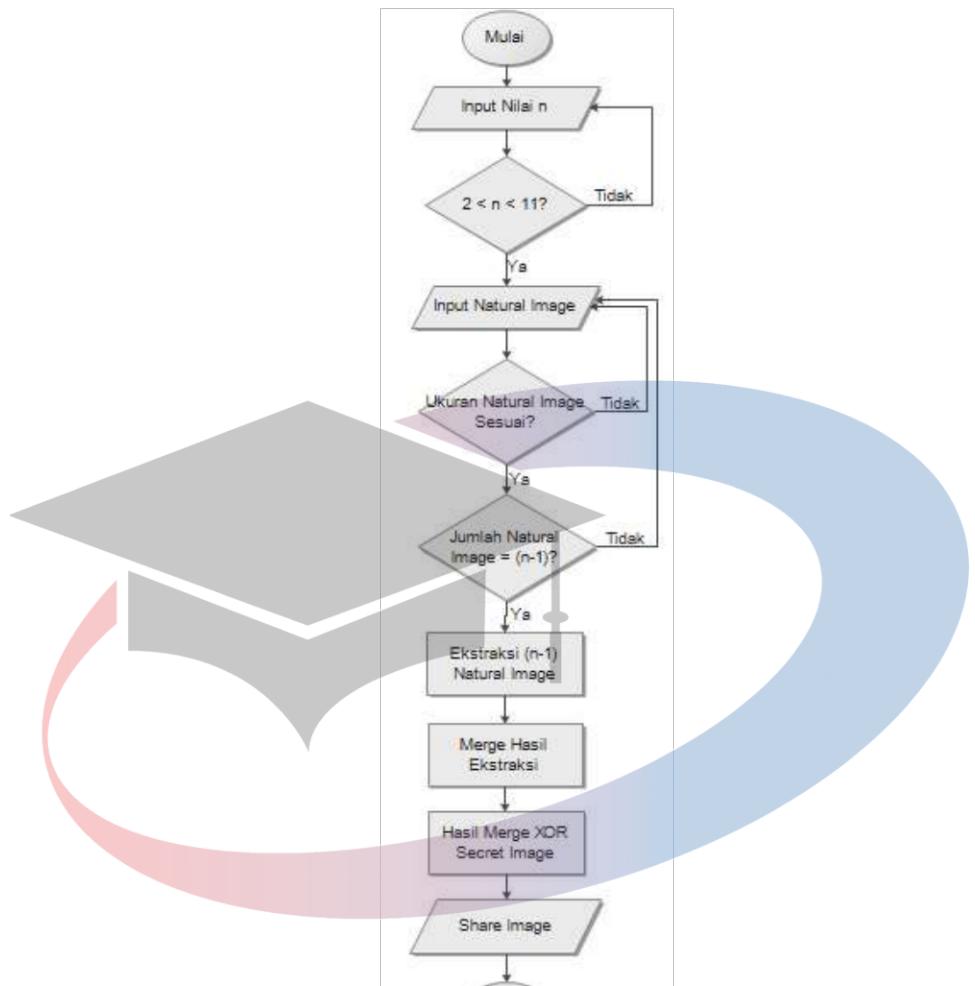


Gambar 3. 1 Flowchart Proses *Embedding*

Berikut merupakan langkah-langkah dalam proses *embedding*:

1. Pembentukan *Share*

Proses pembentukan *share* bertujuan untuk menghasilkan *share* dari *secret image*. Proses ini menggunakan skema $((n - 1, 1), n)$. Adapun *flowchart* proses pembentukan *share* dapat dilihat pada Gambar 3. 2 berikut ini:



Gambar 3. 2 Flowchart Pembentukan *Share*

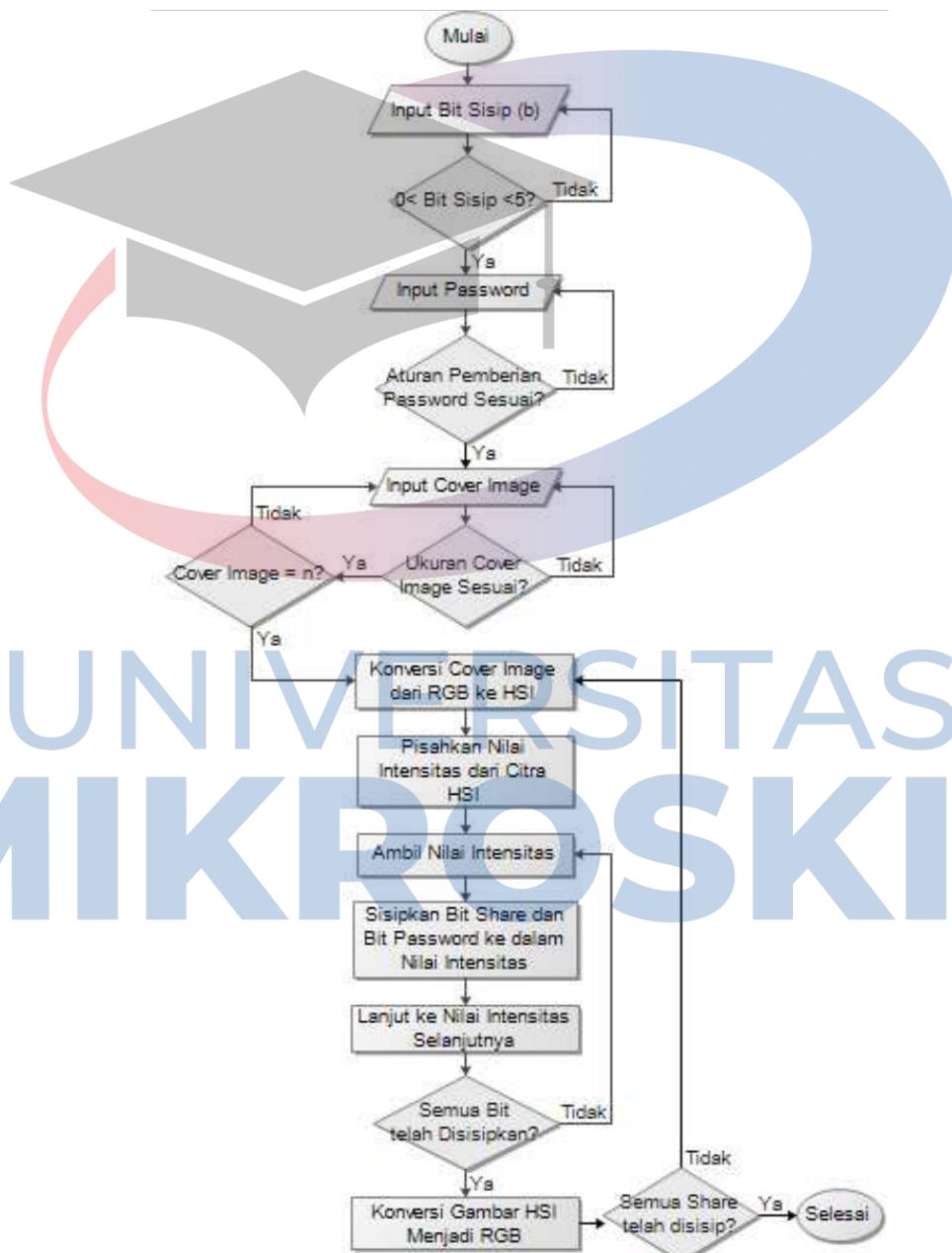
UNIVERSITAS MIKROSKILL

Proses pembentukan *share* terdiri dari dua tahap, yaitu Ekstraksi Fitur dan Enkripsi. Pada Ekstraksi Fitur, $(n - 1)$ *natural image* diekstrak menjadi $(n - 1)$ *natural share image* kemudian dilakukan penggabungan. Pada proses enkripsi hasil penggabungan *natural share image* dilakukan operasi XOR dengan *secret image* sehingga menghasilkan *share image*.

2. Penyisipan *Share*

Proses penyisipan *share* bertujuan untuk menghasilkan *stego image*, yaitu citra yang berisi sisipan *share image* dan *natural share image*. Proses ini dilakukan dengan mengkonversi *cover image* (RGB) menjadi citra HSI, kemudian

menyisipkan bit *share*, *natural share image* dan *password* pada nilai intensitas citra HSI menggunakan teknik LSB. Jumlah bit nilai intensitas yang akan digantikan oleh bit *share*, *natural share image* dan *password* tergantung nilai b yang di-*input* ($1 \leq b \leq 4$). Semakin besar nilai b maka *cover image* yang dibutuhkan akan semakin kecil, dan sebaliknya. *Flowchart* proses penyisipan *share* dapat dilihat pada Gambar 3. 3 berikut ini:



Gambar 3. 3 Flowchart Penyisipan Share

3.1.1.2. Contoh Perhitungan Proses Embedding

1. Pembentukan Share

Pada contoh ini *secret image* yang digunakan berukuran 8×8 piksel dan $n = 3$ jadi, *natural image* yang harus di-*input* adalah 2.

Secret Image:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	142	55	100	127	253	115	39	102	52	112	175	14
1	223	160	24	135	69	222	118	119	229	255	213	97
2	245	190	180	196	43	151	232	98	108	202	117	2
3	41	86	33	133	198	153	47	135	169	84	140	84
4	6	230	82	79	60	252	102	117	148	149	141	57
5	122	111	36	43	238	84	173	155	128	43	36	25
6	71	253	175	64	84	12	174	75	189	216	232	110
7	165	27	206	39	86	204	80	125	180	174	13	66

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	50	239	64	29	147	68	75	6	66	51	171	139
1	53	222	250	7	206	104	238	61	72	207	139	27
2	83	184	126	214	206	88	15	84	89	212	95	187
3	196	32	37	12	134	79	75	218	4	130	52	110
4	32	158	197	95	46	19	130	46	96	201	244	130
5	238	201	139	87	216	2	179	114	211	83	48	223
6	174	250	210	241	214	41	200	34	82	189	232	214
7	22	93	180	98	114	232	30	31	154	71	164	250

Natural image 1:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	61	46	27	74	57	31	88	63	23	87	62	21
1	118	111	95	146	134	112	162	142	105	188	166	125
2	111	113	108	126	124	111	130	119	89	170	154	118
3	5	15	24	14	18	19	113	106	88	113	99	72
4	8	14	46	15	17	38	103	93	92	146	128	114
5	2	15	50	13	20	48	55	52	59	78	65	57
6	0	20	46	0	18	38	38	47	54	87	88	83
7	0	19	40	0	20	36	6	21	26	40	46	44

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	67	51	17	62	40	3	92	55	2	154	108	48
1	154	139	106	121	99	62	145	105	53	154	106	44
2	138	125	93	99	77	38	151	110	56	145	95	34
3	76	61	32	129	104	64	145	100	43	154	99	34
4	107	85	62	150	116	78	121	69	9	161	99	26
5	83	64	47	90	60	26	202	151	96	138	79	11
6	96	90	78	94	74	50	86	47	6	112	61	14
7	88	88	80	75	62	45	94	61	28	76	32	0

Natural image 2:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	79	6	11	191	193	198	37	215	202	160	173	27
1	111	138	84	54	56	237	78	113	193	254	10	187
2	159	238	33	155	31	44	77	45	133	81	223	50
3	236	13	93	209	194	96	154	6	159	130	164	154
4	230	240	101	247	118	238	213	167	26	178	39	103
5	95	79	125	22	61	94	32	71	119	148	188	250
6	105	30	190	114	206	13	66	222	50	160	92	101
7	223	170	157	126	21	247	210	75	15	16	96	98

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	22	207	57	232	198	59	42	114	27	228	78	84
1	255	4	144	220	209	40	189	220	2	202	184	28
2	65	172	75	116	194	223	171	219	75	240	58	191
3	112	29	106	241	159	101	39	239	73	216	180	35
4	137	176	161	226	203	103	113	78	35	86	165	204
5	209	216	17	49	229	156	116	98	47	251	51	225
6	227	116	169	115	32	114	149	129	158	205	227	159
7	40	12	111	23	250	110	206	8	243	134	81	244

a. Fitur Ekstraksi

Tahap ini dilakukan untuk menghasilkan gambar fitur (*Feature Image*) yang nantinya merupakan menjadi *natural share image* dengan tahapan sebagai berikut:

- Inisialisasi *FI* (*Feature Image*) untuk semua bit pada tiap nilai R, G dan B dengan nilai 0. *FI* merupakan pemetaan dari *natural image*. Oleh karena itu,

jumlah dan ukuran FI sama dengan *natural image* yang di-input. Sehingga menjadi:

FI_1 :

Piksel	0			1		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	2			3		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	4			5		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	6			7		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

$FI_2:$

Piksel	0			1		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	2			3		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	4			5		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	6			7		
	R	G	B	R	G	B
0	00000000	00000000	00000000	00000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

- ii. Ganti tiap bit i pada masing-masing FI ($1 \leq i \leq 8$) untuk masing-masing nilai R, G dan B dengan citra biner ($F\alpha$). $F\alpha$ diperoleh dengan tahapan sebagai berikut:

→ $i=1$

Untuk nilai R:

- Untuk semua *natural image* ambil koordinat piksel secara acak (x_1, y_1) dan (x_2, y_2) , lalu tukarkan kedua nilai pada koordinat piksel tersebut untuk masing-masing nilai R, G dan B.

Koordinat piksel secara acak sebanyak 32 untuk R (x_1, y_1) adalah:

(x ₁ ,y ₁)							
(4,2)	(2,0)	(1,7)	(6,6)	(3,7)	(3,2)	(0,0)	(3,4)
(1,3)	(2,6)	(1,0)	(5,6)	(4,3)	(3,3)	(5,0)	(1,2)
(6,0)	(6,1)	(0,7)	(4,0)	(6,7)	(5,5)	(0,2)	(4,5)
(2,3)	(2,5)	(4,6)	(4,7)	(5,7)	(1,5)	(7,6)	(7,3)

Koordinat piksel secara acak sebanyak 32 untuk R (x_2, y_2) adalah:

(x ₂ ,y ₂)							
(7,7)	(6,4)	(0,3)	(2,2)	(0,6)	(0,5)	(2,5)	(6,3)
(1,4)	(7,0)	(3,5)	(2,1)	(6,2)	(4,4)	(5,1)	(6,5)
(3,0)	(5,4)	(1,6)	(7,2)	(0,1)	(2,4)	(4,1)	(0,4)
(2,7)	(3,1)	(7,1)	(5,2)	(7,4)	(5,3)	(1,1)	(7,5)

Hasil *natural image* setelah pengacakan nilai R untuk tiap piksel.

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	145	46	27	112	57	31	15	63	23	154	62	21
1	129	111	95	94	134	112	94	142	105	154	166	125
2	96	113	108	202	124	111	86	119	89	145	154	118
3	0	15	24	99	18	19	62	106	88	107	99	72
4	6	14	46	88	17	38	76	93	92	38	128	114
5	13	15	50	2	20	48	161	52	59	121	65	57
6	5	20	46	83	18	38	146	47	54	76	88	83
7	151	19	40	121	20	36	8	21	26	75	46	44

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	150	51	17	113	40	3	154	55	2	145	108	48
1	188	139	106	78	99	62	154	105	53	87	106	44
2	90	125	93	14	77	38	0	110	56	170	95	34
3	87	61	32	118	104	64	61	100	43	92	99	34
4	113	85	62	67	116	78	0	69	9	55	99	26
5	0	64	47	138	60	26	126	151	96	88	79	11
6	111	90	78	162	74	50	130	47	6	74	61	14
7	138	88	80	40	62	45	146	61	28	103	32	0

- Lakukan operasi XOR dimana untuk mendapatkan nilai R pada koordinat tertentu merupakan hasil XOR nilai R piksel sebelumnya dan nilai R pada piksel tersebut. Begitu seterusnya hingga semua piksel dibaca.

Piksel 1:

0 : 00000000

145 : 10010001 XOR

10010001 : 145

Piksel 2 = nilai R pada piksel 1 XOR nilai R pada piksel 2:

145 : 10010001

112 : 01110000 XOR

11100001 : 225

Piksel 64 = nilai R pada piksel 63 XOR nilai R pada piksel 64:

3 : 00000011

103 : 01100111 XOR

01100100 : 100

Hasil operasi XOR:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	145	46	27	255	57	31	238	63	23	116	62	21
1	25	111	95	71	134	112	25	142	105	131	166	125
2	220	113	108	22	124	111	64	119	89	209	154	118
3	47	15	24	76	18	19	114	106	88	25	99	72
4	95	14	46	7	17	38	75	93	92	109	128	114
5	101	15	50	103	20	48	198	52	59	191	65	57
6	22	20	46	69	18	38	215	47	54	155	88	83
7	9	19	40	112	20	36	120	21	26	51	46	44

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	226	51	17	147	40	3	9	55	2	152	108	48
1	63	139	106	113	99	62	235	105	53	188	106	44
2	139	125	93	133	77	38	133	110	56	47	95	34
3	78	61	32	56	104	64	5	100	43	89	99	34
4	28	85	62	95	116	78	95	69	9	104	99	26
5	191	64	47	53	60	26	75	151	96	19	79	11
6	244	90	78	86	74	50	212	47	6	158	61	14
7	185	88	80	145	62	45	3	61	28	100	32	0

Koordinat piksel secara acak sebanyak 32 untuk G (x_1, y_1) adalah:

(x ₁ ,y ₁)							
(0,0)	(6,0)	(1,1)	(1,0)	(0,6)	(1,3)	(3,5)	(4,2)
(7,6)	(3,1)	(7,1)	(7,2)	(2,6)	(0,3)	(0,4)	(0,2)
(7,4)	(1,5)	(6,6)	(2,3)	(7,7)	(6,2)	(2,1)	(6,7)
(4,1)	(3,6)	(7,3)	(3,7)	(5,3)	(5,4)	(7,0)	(2,4)

Koordinat piksel secara acak sebanyak 32 untuk G (x_2, y_2) adalah:

(x ₂ ,y ₂)							
(2,7)	(3,2)	(3,4)	(6,4)	(4,5)	(5,2)	(3,3)	(0,5)
(4,3)	(6,3)	(1,7)	(4,7)	(6,5)	(5,5)	(5,7)	(4,4)
(6,1)	(2,5)	(2,2)	(5,0)	(1,2)	(3,0)	(5,6)	(2,0)
(1,6)	(1,4)	(5,1)	(4,6)	(4,0)	(7,5)	(0,1)	(0,7)

Hasil *natural image* setelah pengacakan nilai G untuk tiap piksel.

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	61	95	27	74	19	31	88	85	23	87	60	21
1	118	90	95	146	61	112	162	32	105	188	52	125
2	111	61	108	126	151	111	130	47	89	170	15	118
3	5	47	24	14	88	19	113	20	88	113	104	72
4	8	65	46	15	105	38	103	40	92	146	61	114
5	2	154	50	13	46	48	55	166	59	78	14	57
6	0	106	46	0	88	38	38	15	54	87	18	83
7	0	57	40	0	106	36	6	99	26	40	20	44

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	67	79	17	62	93	3	92	116	2	154	125	48
1	154	100	106	121	77	62	145	17	53	154	20	44
2	138	108	93	99	99	38	151	74	56	145	46	34
3	76	134	32	129	99	64	145	139	43	154	69	34
4	107	63	62	150	55	78	121	99	9	161	21	26
5	83	62	47	90	62	26	202	124	96	138	51	11
6	96	111	78	94	110	50	86	119	6	112	113	14
7	88	18	80	75	64	45	94	128	28	76	142	0

- Lakukan operasi XOR antara nilai piksel untuk nilai G sebelumnya dan nilai G pada piksel tersebut, begitu seterusnya hingga semua piksel dibaca.

Piksel 1:

0 : 00000000

95 : 01011111 XOR

01011111 : 95

Piksel 2 = nilai G pada piksel 1 XOR nilai G pada piksel 2:

95 : 01011111

19 : 00010011 XOR

01001100 : 76

Piksel 64 = nilai G pada piksel 63 XOR nilai G pada piksel 64:

110 : 01101110

142 : 10001110 XOR

11100000 : 224

Hasil operasi XOR:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	61	95	27	74	76	31	88	25	23	87	37	21
1	118	100	95	146	89	112	162	121	105	188	77	125
2	111	92	108	126	203	111	130	228	89	170	235	118
3	5	175	24	14	247	19	113	227	88	113	139	72
4	8	225	46	15	136	38	103	160	92	146	157	114
5	2	121	50	13	87	48	55	241	59	78	255	57
6	0	218	46	0	130	38	38	141	54	87	159	83
7	0	161	40	0	203	36	6	168	26	40	188	44

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	67	106	17	62	55	3	92	67	2	154	62	48
1	154	41	106	121	100	62	145	117	53	154	97	44
2	138	135	93	99	228	38	151	174	56	145	128	34
3	76	13	32	129	110	64	145	229	43	154	160	34
4	107	162	62	150	149	78	121	246	9	161	227	26
5	83	193	47	90	255	26	202	131	96	138	176	11
6	96	240	78	94	158	50	86	233	6	112	152	14
7	88	174	80	75	238	45	94	110	28	76	224	0

Koordinat piksel secara acak sebanyak 32 untuk B (x_1, y_1) adalah:

(x ₁ ,y ₁)							
(7,6)	(6,4)	(4,5)	(0,2)	(5,7)	(3,6)	(2,0)	(6,6)
(4,3)	(7,0)	(4,7)	(0,6)	(6,3)	(1,7)	(5,4)	(7,5)
(4,6)	(3,0)	(0,7)	(5,0)	(6,7)	(3,7)	(6,2)	(5,6)
(0,5)	(1,1)	(3,2)	(5,3)	(3,4)	(1,6)	(2,5)	(1,3)

Koordinat piksel secara acak sebanyak 32 untuk B (x_2, y_2) adalah:

(x ₂ ,y ₂)							
(1,2)	(0,3)	(6,5)	(2,3)	(5,1)	(0,0)	(5,2)	(1,5)
(2,4)	(7,1)	(7,7)	(2,2)	(2,1)	(1,0)	(4,0)	(5,5)
(0,1)	(3,1)	(6,1)	(4,4)	(3,5)	(1,4)	(2,7)	(6,0)
(4,2)	(7,2)	(0,4)	(2,6)	(7,3)	(3,3)	(7,4)	(4,1)

Hasil *natural image* setelah pengacakan nilai B untuk tiap piksel.

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	61	46	43	74	57	9	88	63	118	87	62	78
1	118	111	44	146	134	26	162	142	28	188	166	38
2	111	113	59	126	124	83	130	119	2	170	154	23
3	5	15	19	14	18	24	113	106	17	113	99	53
4	8	14	47	15	17	125	103	93	3	146	128	93
5	2	15	62	13	20	11	55	52	108	78	65	56
6	0	20	96	0	18	48	38	47	34	87	88	111
7	0	19	36	0	20	40	6	21	112	40	46	32

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	67	51	88	62	40	92	92	55	89	154	108	38
1	154	139	34	121	99	6	145	105	72	154	106	95
2	138	125	114	99	77	80	151	110	57	145	95	54
3	76	61	44	129	104	14	145	100	27	154	99	106
4	107	85	50	150	116	50	121	69	31	161	99	0
5	83	64	46	90	60	45	202	151	46	138	79	48
6	96	90	21	94	74	78	86	47	62	112	61	64
7	88	88	38	75	62	26	94	61	105	76	32	26

- Lakukan operasi XOR antara nilai piksel untuk nilai B sebelumnya dan nilai B pada piksel tersebut, begitu seterusnya hingga semua piksel dibaca.

Piksel 1:

0 : 00000000

43 : 00101011 XOR

00101011 : 43

Piksel 2 = nilai B pada piksel 1 XOR nilai B pada piksel 2:

43 : 00101011

9 : 00001001 XOR

00100010 : 34

Piksel 64 = nilai B pada piksel 63 XOR nilai B pada piksel 64:

44 : 00101100

26 : 00011010 XOR

00110110 : 54

Hasil operasi XOR:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	61	46	43	74	57	34	88	63	84	87	62	26
1	118	111	77	146	134	87	162	142	75	188	166	109
2	111	113	101	126	124	54	130	119	52	170	154	35
3	5	15	29	14	18	5	113	106	20	113	99	33
4	8	14	93	15	17	32	103	93	35	146	128	126
5	2	15	95	13	20	84	55	52	56	78	65	0
6	0	20	125	0	18	77	38	47	111	87	88	0
7	0	19	1	0	20	41	6	21	89	40	46	121

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	67	51	66	62	40	30	92	55	71	154	108	97
1	154	139	79	121	99	73	145	105	1	154	106	94
2	138	125	81	99	77	1	151	110	56	145	95	14
3	76	61	13	129	104	3	145	100	24	154	99	114
4	107	85	76	150	116	126	121	69	97	161	99	97
5	83	64	46	90	60	3	202	151	45	138	79	29
6	96	90	21	94	74	91	86	47	101	112	61	37
7	88	88	95	75	62	69	94	61	44	76	32	54

Hasil R, G dan B setelah pengacakan dan operasi XOR:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	145	95	43	255	76	34	238	25	84	116	37	26
1	25	100	77	71	89	87	25	121	75	131	77	109
2	220	92	101	22	203	54	64	228	52	209	235	35
3	47	175	29	76	247	5	114	227	20	25	139	33
4	95	225	93	7	136	32	75	160	35	109	157	126
5	101	121	95	103	87	84	198	241	56	191	255	0
6	22	218	125	69	130	77	215	141	111	155	159	0
7	9	161	1	112	203	41	120	168	89	51	188	121

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	226	106	66	147	55	30	9	67	71	152	62	97
1	63	41	79	113	100	73	235	117	1	188	97	94
2	139	135	81	133	228	1	133	174	56	47	128	14
3	78	13	13	56	110	3	5	229	24	89	160	114
4	28	162	76	95	149	126	95	246	97	104	227	97
5	191	193	46	53	255	3	75	131	45	19	176	29
6	244	240	21	86	158	91	212	233	101	158	152	37
7	185	174	95	145	238	69	3	110	44	100	224	54

▪ Hitung $H_{\alpha}^{x,y}$, merupakan jumlah nilai R, G dan B untuk tiap piksel.

Piksel 1:

R	G	B
145	95	43
$\rightarrow 145 + 95 + 43 = 283$		

Piksel 2:

R	G	B
255	76	34
$\rightarrow 255 + 76 + 34 = 365$		

Piksel 64:

R	G	B
100	224	54
$\rightarrow 100 + 224 + 54 = 378$		

H_α :

Piksel	0	1	2	3	4	5	6	7
0	283	365	347	179	398	232	147	311
1	202	247	221	317	398	286	353	379
2	413	279	344	479	355	362	363	189
3	251	328	361	197	104	169	258	363
4	413	175	270	392	266	370	438	428
5	317	274	495	446	430	311	251	224
6	365	276	467	314	505	335	546	347
7	171	356	377	360	454	452	157	378

- Hitung M_α^β , merupakan rata-rata per blok, dimana tiap blok terdiri dari 8×8 piksel. Dalam contoh ini, kebetulan citra yang digunakan berukuran 8×8 piksel.

$$\rightarrow \frac{283+365+\dots+378}{64} = 326$$

$$M_\alpha^\beta = 326$$

- Cari F_α dengan cara:

Cek apakah $H_{\alpha,y}^{x,y} < M_\alpha^\beta$

Jika ya, jadikan nilai piksel untuk F_α menjadi 1

Jika tidak, jadikan 0

Piksel 1: $283 < 326$, maka di piksel 1 F_α dijadikan 1.

Piksel 2: $365 > 326$, maka di piksel 2 F_α dijadikan 0.

.

.

.

Piksel 64: $378 > 326$, maka di piksel 64 F_α dijadikan 0.

F_α :

Piksel	0	1	2	3	4	5	6	7
0	1	0	0	1	0	1	1	1
1	1	1	1	1	0	1	0	0
2	0	1	0	0	0	0	0	1
3	1	0	0	1	1	1	1	0
4	0	1	1	0	1	0	0	0
5	1	1	0	0	0	1	1	1
6	0	1	0	1	0	0	0	0
7	1	0	0	0	0	0	1	0

- Set nilai FI_1 untuk bit ke 1 nilai R menggunakan F_α , sehingga menjadi:

Piksel	0			1		
	R	G	B	R	G	B
0	10000000	00000000	00000000	00000000	00000000	00000000
1	10000000	00000000	00000000	10000000	00000000	00000000
2	00000000	00000000	00000000	10000000	00000000	00000000
3	10000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	10000000	00000000	00000000
5	10000000	00000000	00000000	10000000	00000000	00000000
6	00000000	00000000	00000000	10000000	00000000	00000000
7	10000000	00000000	00000000	00000000	00000000	00000000

Piksel	2			3		
	R	G	B	R	G	B
0	00000000	00000000	00000000	10000000	00000000	00000000
1	10000000	00000000	00000000	10000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	00000000	00000000	00000000	10000000	00000000	00000000
4	10000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	00000000	00000000	00000000
6	00000000	00000000	00000000	10000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	4			5		
	R	G	B	R	G	B
0	00000000	00000000	00000000	10000000	00000000	00000000
1	00000000	00000000	00000000	10000000	00000000	00000000
2	00000000	00000000	00000000	00000000	00000000	00000000
3	10000000	00000000	00000000	10000000	00000000	00000000
4	10000000	00000000	00000000	00000000	00000000	00000000
5	00000000	00000000	00000000	10000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	00000000	00000000	00000000	00000000	00000000	00000000

Piksel	6			7		
	R	G	B	R	G	B
0	10000000	00000000	00000000	10000000	00000000	00000000
1	00000000	00000000	00000000	00000000	00000000	00000000
2	00000000	00000000	00000000	10000000	00000000	00000000
3	10000000	00000000	00000000	00000000	00000000	00000000
4	00000000	00000000	00000000	00000000	00000000	00000000
5	10000000	00000000	00000000	10000000	00000000	00000000
6	00000000	00000000	00000000	00000000	00000000	00000000
7	10000000	00000000	00000000	00000000	00000000	00000000

Untuk nilai G:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	121	99	3	8	76	33	99	98	19	101	30	53
1	204	42	3	75	63	59	129	46	20	194	107	31
2	49	87	87	81	105	123	2	49	117	148	13	21
3	180	79	15	131	33	101	201	21	54	88	59	44
4	11	81	55	138	223	43	208	203	9	210	252	33
5	14	71	36	66	122	9	76	16	0	76	30	54
6	213	203	41	155	191	41	226	134	16	169	241	72
7	186	239	94	48	178	96	103	52	32	111	11	127

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	217	125	5	169	114	30	241	242	62	175	101	47
1	83	85	12	110	0	40	1	90	6	160	23	105
2	148	85	59	226	63	46	180	45	94	46	103	65
3	88	47	46	87	75	52	38	35	54	35	16	106
4	138	159	77	32	141	87	6	43	108	156	36	34
5	206	111	95	169	245	23	11	153	97	153	246	28
6	215	152	80	77	199	71	218	212	117	228	192	44
7	51	118	84	100	253	100	100	221	66	105	224	100

H_α :

Piksel	0	1	2	3	4	5	6	7
0	223	117	216	184	347	313	545	323
1	249	197	195	332	180	150	97	288
2	223	309	168	182	292	335	319	214
3	274	265	276	191	181	214	127	157
4	147	404	420	495	374	260	157	226
5	121	197	92	160	412	437	261	427
6	457	387	376	482	447	347	547	464
7	519	322	187	249	253	453	387	429

$$M_\alpha^\beta = 290$$

F_α :

Piksel	0	1	2	3	4	5	6	7
0	1	1	1	1	0	0	0	0
1	1	1	1	0	1	1	1	1
2	1	0	1	1	0	0	0	1
3	1	1	1	1	1	1	1	1
4	1	0	0	0	0	1	1	1
5	1	1	1	1	0	0	1	0
6	0	0	0	0	0	0	0	0
7	0	0	1	1	1	0	0	0

Untuk nilai B:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	86	166	93	204	155	29	178	245	78	196	137	112
1	38	74	90	135	61	73	13	112	85	155	94	27
2	135	9	102	223	55	88	77	177	66	61	244	96
3	133	8	13	203	29	125	178	61	51	225	18	3
4	240	11	118	188	60	78	118	95	76	244	5	126
5	72	18	52	210	6	46	142	41	37	128	7	3
6	246	61	63	186	0	6	43	99	51	186	44	36
7	123	137	99	123	135	81	250	149	97	149	212	78

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	78	253	93	45	165	71	26	180	27	113	221	83
1	193	81	119	195	12	24	229	96	46	175	53	12
2	126	231	123	64	105	125	69	86	91	18	53	45
3	238	43	27	232	107	4	163	85	10	174	31	38
4	110	111	107	210	92	22	163	198	73	49	153	16
5	241	111	47	107	30	44	12	113	4	174	41	63
6	228	172	124	236	207	14	209	251	40	123	244	3
7	149	252	96	4	152	113	100	138	24	100	224	54

H_α :

Piksel	0	1	2	3	4	5	6	7
0	345	388	501	445	424	281	233	417
1	202	269	210	276	393	231	371	240
2	246	366	320	401	480	294	246	116
3	154	357	290	246	308	343	258	243
4	369	326	289	375	328	324	434	218
5	142	262	220	138	399	181	129	278
6	370	192	193	266	524	457	500	370
7	359	339	496	439	497	269	262	378

$$M_\alpha^\beta = 316$$

F_α :

Piksel	0	1	2	3	4	5	6	7
0	0	0	0	0	0	1	1	0
1	1	1	1	1	0	1	0	1
2	1	0	0	0	0	1	1	1
3	1	0	1	1	1	0	1	1
4	0	0	1	0	0	0	0	1
5	1	1	1	1	0	1	1	1
6	0	1	1	1	0	0	0	0
7	0	0	0	0	0	1	1	0

→ i=2

Untuk nilai R:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	129	46	59	240	118	29	129	72	19	22	93	78
1	41	44	86	87	108	84	25	27	108	63	38	34
2	234	212	117	120	188	118	125	50	72	11	103	8
3	146	64	84	206	84	95	132	41	95	220	108	112
4	242	108	60	96	52	10	250	126	12	245	29	22
5	150	247	40	12	196	120	79	174	99	237	244	72
6	163	80	71	232	112	93	84	19	123	206	2	95
7	107	132	115	161	235	32	249	213	0	251	251	46

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	33	73	71	128	126	40	142	86	88	131	56	116
1	88	40	12	88	168	85	210	148	119	129	160	71
2	154	13	29	11	2	2	92	110	107	16	125	52
3	165	251	106	152	135	28	251	200	37	172	231	16
4	154	135	59	204	246	39	128	149	54	190	186	68
5	119	114	0	41	27	38	41	38	62	41	66	45
6	190	67	34	228	30	10	114	125	86	18	219	65
7	253	194	30	253	253	32	245	242	110	100	224	54

H_α :

Piksel	0	1	2	3	4	5	6	7
0	234	387	220	193	177	294	316	303
1	171	279	160	135	140	341	477	360
2	563	426	247	122	196	15	309	193
3	294	385	268	440	522	315	488	419
4	410	158	388	296	348	489	331	444
5	437	328	352	553	233	106	141	152
6	314	437	226	303	291	268	325	302
7	354	428	462	548	477	538	597	378

$$M_\alpha^\beta = 325$$

F_α

Piksel	0	1	2	3	4	5	6	7
0	1	0	1	1	1	1	1	1
1	1	1	1	1	1	0	0	0
2	0	0	1	1	1	1	1	1
3	1	0	1	0	0	1	0	0
4	0	1	0	1	0	0	0	0
5	0	0	0	0	1	1	1	1
6	1	0	1	1	1	1	1	1
7	0	0	0	0	0	0	0	0

Untuk nilai G:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	202	19	125	104	75	75	52	30	81	57	10	31
1	144	34	11	1	120	47	15	61	64	96	50	110
2	205	41	21	87	113	70	96	72	8	202	90	31
3	105	29	38	251	157	37	71	236	30	16	211	65
4	139	173	95	10	38	76	144	91	108	144	49	64
5	100	142	119	254	156	73	116	188	75	39	179	84
6	208	198	62	160	172	98	247	185	68	134	150	118
7	213	176	35	131	190	19	131	221	97	18	83	13

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	82	71	127	195	13	65	67	113	121	206	77	32
1	58	165	103	18	203	87	62	252	10	174	179	51
2	107	78	5	105	39	20	121	26	68	16	41	102
3	54	183	65	125	159	109	150	194	119	235	131	7
4	7	151	41	121	244	67	15	201	101	60	230	121
5	107	167	34	225	248	13	94	155	56	183	248	22
6	128	168	62	133	185	53	154	249	23	141	196	37
7	100	63	24	40	185	53	113	206	45	100	224	54

H_α :

Piksel	0	1	2	3	4	5	6	7
0	346	254	163	98	280	273	301	315
1	189	168	140	256	326	308	324	404
2	267	270	176	323	190	164	215	159
3	172	445	337	292	302	393	463	373
4	407	124	343	257	199	432	317	411
5	361	483	379	302	308	486	305	453
6	468	430	500	402	358	371	426	374
7	424	340	449	114	187	278	364	378

$$M_\alpha^\beta = 314$$

F_α :

Piksel	0	1	2	3	4	5	6	7
0	0	1	1	1	1	1	1	0
1	1	1	1	1	0	1	0	0
2	1	1	1	0	1	1	1	1
3	1	0	0	1	1	0	0	0
4	0	1	0	1	1	0	0	0
5	0	0	0	1	1	0	1	0
6	0	0	0	0	0	0	0	0
7	0	0	0	1	1	1	0	0

Untuk nilai B:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	8	69	38	95	86	121	197	97	85	79	2	85
1	95	20	102	121	126	66	121	36	114	33	179	76
2	248	96	10	166	3	33	39	48	54	39	109	16
3	154	204	112	234	76	101	221	12	72	150	30	34
4	107	140	32	26	230	16	113	242	77	50	116	123
5	220	56	94	70	96	7	209	53	50	143	68	16
6	189	156	81	179	161	121	182	175	22	52	224	40
7	179	170	90	202	215	26	134	163	17	132	192	40

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	79	97	74	222	29	4	79	113	114	213	76	84
1	88	138	110	56	165	109	169	210	119	245	72	27
2	181	9	112	186	39	40	203	8	16	164	134	25
3	60	95	114	118	212	15	37	234	1	107	229	115
4	76	96	64	26	45	66	128	63	10	218	80	68
5	195	123	48	181	221	64	214	227	83	119	141	125
6	136	212	46	223	232	114	73	200	110	235	151	64
7	22	212	26	113	193	53	89	206	45	100	224	54

H_α :

Piksel	0	1	2	3	4	5	6	7
0	115	302	379	166	250	255	306	373
1	217	313	271	288	336	330	498	344
2	354	202	141	164	302	265	227	323
3	470	411	305	214	269	345	272	451
4	279	272	432	289	236	137	201	366
5	370	173	312	227	366	466	524	385
6	426	461	379	316	394	569	383	450
7	439	443	314	364	260	359	340	378

$$M_\alpha^\beta = 314$$

F_α :

Piksel	0	1	2	3	4	5	6	7
0	1	1	0	1	1	1	1	0
1	1	1	1	1	0	0	0	0
2	0	1	1	1	1	1	1	1
3	0	0	1	1	1	0	1	0
4	1	1	0	1	1	1	1	0
5	0	1	1	1	0	0	0	0
6	0	0	0	1	0	0	0	0
7	0	0	1	0	1	0	0	0

→ i=3

⋮

⋮

Untuk nilai R:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	40	79	2	102	23	44	247	3	117	134	13	83
1	52	14	60	120	114	32	1	79	115	66	213	92
2	106	130	31	252	232	55	64	165	45	49	177	80
3	84	129	19	31	237	26	17	222	0	103	85	118
4	88	58	97	194	156	79	192	255	61	154	193	17
5	61	198	122	27	72	92	186	103	51	209	4	17
6	171	23	73	60	96	123	111	42	110	49	189	91
7	166	103	8	60	104	16	76	238	1	214	193	10

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	142	99	97	176	119	97	206	10	1	84	55	58
1	232	218	113	122	200	90	248	161	54	248	194	8
2	187	233	16	25	180	46	78	219	49	55	229	121
3	103	106	117	80	87	40	86	69	14	14	101	81
4	27	212	41	27	142	103	87	228	116	56	165	88
5	137	42	53	3	111	3	146	58	25	204	43	57
6	60	61	50	173	76	110	173	36	94	241	19	6
7	217	245	68	19	221	20	89	206	45	100	224	54

H_α :

Piksel	0	1	2	3	4	5	6	7
0	121	169	367	230	392	313	140	58
1	126	266	195	371	435	499	496	8
2	267	539	274	306	221	343	333	121
3	232	294	239	306	284	195	129	81
4	243	429	508	364	210	418	337	88
5	381	191	340	230	167	207	272	57
6	267	279	263	329	299	319	354	6
7	277	180	315	417	308	315	369	54

$$M_\alpha^\beta = 268$$

F_α :

Piksel	0	1	2	3	4	5	6	7
0	1	1	0	1	0	0	1	1
1	1	1	1	0	0	0	0	1
2	1	0	0	0	1	0	0	1
3	1	0	1	0	0	1	1	1
4	1	0	0	0	1	0	0	1
5	0	1	0	1	1	1	0	1
6	1	0	1	0	0	0	0	1
7	0	1	0	0	0	0	0	1

Untuk nilai G:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	6	134	24	92	8	2	44	53	109	182	36	113
1	45	61	2	45	94	2	175	77	48	225	121	92
2	86	97	90	244	5	124	162	106	107	149	253	73
3	160	61	26	49	47	66	187	175	100	26	160	75
4	205	145	46	149	230	40	214	206	117	92	247	7
5	129	25	4	205	36	74	173	97	25	55	28	39
6	214	20	57	87	76	68	38	36	105	24	113	69
7	193	215	94	91	136	112	89	200	72	14	133	89

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	185	77	27	183	95	83	233	108	12	115	81	50
1	130	35	106	19	31	99	116	66	65	39	117	106
2	236	91	109	126	100	69	86	75	112	234	40	115
3	26	209	73	70	239	90	75	207	104	129	219	49
4	92	227	62	34	128	18	91	252	100	202	146	36
5	111	83	9	73	92	19	65	114	115	214	76	35
6	110	235	11	252	170	30	249	201	110	174	163	85
7	164	170	5	207	192	11	89	206	45	100	224	54

H_α :

Piksel	0	1	2	3	4	5	6	7
0	164	102	206	331	305	424	208	50
1	108	141	300	438	156	281	221	106
2	273	373	375	475	335	230	386	115
3	247	162	462	261	382	372	452	49
4	396	419	537	346	224	361	448	36
5	158	315	295	122	174	198	405	35
6	291	231	179	206	433	480	447	85
7	502	339	361	236	404	306	369	54

$$M_\alpha^\beta = 279$$

F_α :

Piksel	0	1	2	3	4	5	6	7
0	1	1	1	0	0	0	1	1
1	1	1	0	0	1	0	1	1
2	1	0	0	0	0	1	0	1
3	1	1	0	1	0	0	0	1
4	0	0	0	0	1	0	0	1
5	1	0	0	1	1	1	0	1
6	0	1	1	1	0	0	0	1
7	0	0	0	1	0	0	0	1

Untuk nilai B:

Hasil pengacakan dan operasi XOR untuk masing-masing nilai RGB dari *natural image*:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	88	77	36	2	92	18	190	250	120	187	146	70
1	207	0	86	69	15	30	34	87	38	92	69	14
2	224	105	71	188	65	91	234	120	68	133	118	66
3	106	123	11	32	7	0	182	21	62	198	1	41
4	167	74	28	144	69	69	10	47	69	44	59	126
5	2	189	41	13	232	122	135	104	19	241	50	75
6	72	35	114	226	79	47	130	91	82	21	49	107
7	194	18	1	83	156	8	241	248	46	99	235	44

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	187	189	92	247	39	124	156	73	19	207	119	9
1	92	49	40	84	126	14	206	65	32	183	84	82
2	4	55	88	92	11	122	98	122	105	240	84	39
3	145	98	27	218	61	54	212	64	3	214	41	51
4	182	12	46	23	49	114	89	5	50	200	54	2
5	146	81	103	204	27	41	202	59	42	202	126	4
6	100	82	7	105	212	53	16	187	23	129	44	25
7	47	179	76	113	243	60	89	206	45	100	224	54

H_α :

Piksel	0	1	2	3	4	5	6	7
0	201	112	560	403	378	353	345	9
1	293	114	159	175	250	285	299	82
2	400	344	422	317	191	342	429	39
3	240	39	265	240	306	330	258	51
4	269	282	126	229	118	208	304	2
5	232	367	258	366	334	302	370	4
6	221	352	303	177	324	256	196	25
7	213	247	535	378	432	355	369	54

$$M_\alpha^\beta = 257$$

F_α :

Piksel	0	1	2	3	4	5	6	7
0	1	1	0	0	0	0	0	1
1	0	1	1	1	1	0	0	1
2	0	0	0	0	1	0	0	1
3	1	1	0	1	0	0	0	1
4	0	0	1	1	1	1	0	1
5	1	0	0	0	0	0	0	1
6	1	0	0	1	0	1	1	1
7	1	1	0	0	0	0	0	1

Setelah 8 kali iterasi maka diperoleh FI_1 :

Piksel	0			1		
	R	G	B	R	G	B
0	11111111	10011111	01111111	00000011	11011111	01111011
1	11101111	11000001	11101000	11001111	11000011	11010001
2	00001011	11010011	10111000	10100110	01010000	01011000
3	11101001	11110101	10001011	00101000	10100101	00001101
4	00000111	10101100	01001010	11001100	01101110	01000010
5	10011000	10111101	10110001	10011001	10111110	11111000
6	01000001	00001000	00000011	10000000	00001101	10000000
7	10010000	00000010	00000001	00100001	00100010	00100001

Piksel	2			3		
	R	G	B	R	G	B
0	01011010	11000111	00000110	11000011	11100110	01011000
1	11011111	11010100	11110001	11100110	01010000	11010001
2	01000110	11011010	01111000	01010100	10010010	01101100
3	01101001	10100000	11001010	10101100	11100011	11100011
4	10110100	00000110	10000101	01110000	01100000	01000101
5	00010000	10100110	11110010	00001011	11100101	11110010
6	01100001	00100001	10110000	11010010	00101101	11100001
7	00001000	10010000	01000000	00011100	11000001	00100000

Piksel	4			5		
	R	G	B	R	G	B
0	01011100	01100000	01001000	11011010	01100100	11011000
1	01111110	10010001	00010001	10100110	11000010	10011110
2	01010101	01011010	01100001	01000100	01110011	11111000
3	10100000	11101110	11000100	11101001	10110110	00001010
4	10010001	01111011	01000111	00011000	10111010	01100011
5	01000001	01011101	00001110	11000001	00010001	10001110
6	01011110	00000000	00010000	01010010	00000000	00100001
7	00010000	11011000	01000110	00000000	01000000	10000000

Piksel	6			7		
	R	G	B	R	G	B
0	11101011	01000101	11001100	11111111	00111111	00111111
1	00011100	10000001	00010000	00111111	10111111	10111111
2	01100000	01000010	11111000	11111111	11111111	11111111
3	10000001	10000110	11000110	00111111	10111111	10111111
4	00000010	10100000	01101100	00111111	10111111	10111111
5	11000000	11100100	10010110	11111111	00111111	10111111
6	01010110	00000100	00000101	01111111	00111111	00111111
7	10000000	00000000	10000000	00111111	00111111	00111111

FI_1 dalam desimal:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	255	159	127	3	223	123	90	199	6	195	230	88
1	239	193	232	207	195	209	223	212	241	230	80	209
2	11	211	184	166	80	88	70	218	120	84	146	108
3	233	245	139	40	165	13	105	160	202	172	227	227
4	7	172	74	204	110	66	180	6	133	112	96	69
5	152	189	177	153	190	248	16	166	242	11	229	242
6	65	8	3	128	13	128	97	33	176	210	45	225
7	144	2	1	33	34	33	8	144	64	28	193	32

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	92	96	72	218	100	216	235	69	204	255	63	63
1	126	145	17	166	194	158	28	129	16	63	191	191
2	85	90	97	68	115	248	96	66	248	255	255	255
3	160	238	196	233	182	10	129	134	198	63	191	191
4	145	123	71	24	186	99	2	160	108	63	191	191
5	65	93	14	193	17	142	192	228	150	255	63	191
6	94	0	16	82	0	33	86	4	5	127	63	63
7	16	216	70	0	64	128	128	0	128	63	63	63

Lakukan juga iterasi = 1 sampai 8 untuk *natural image* 2, sehingga diperoleh FI_2 :

Piksel	0			1		
	R	G	B	R	G	B
0	11101111	00010110	11111101	00010011	01100101	00001010
1	01000110	10011100	11011000	00101101	11110001	00111011
2	11001011	01101101	10110001	11100101	11010000	11110011
3	01100101	00100101	01001101	01101010	01111011	00101100
4	00110111	01010011	01101101	00110010	10100010	10111100
5	00100000	01100010	01011101	00111001	10000011	11001100
6	11110010	00010100	11001011	11100101	01000010	00110001
7	00001010	01101101	01000110	11011111	00101100	00111010

Piksel	2			3		
	R	G	B	R	G	B
0	11110010	10000001	00111011	11011000	11000010	00010100
1	01011110	11101010	11010101	00111000	10110000	00100111
2	10001111	11111000	10100001	01000000	00011010	00011000
3	10010101	01110110	11001111	10100100	10111010	00111001
4	10100000	11000001	11010010	11001001	00101001	10101110
5	11100101	11011010	01011101	10000110	00000111	10001010
6	11110100	00011101	10011001	11101010	11010111	10111011
7	01000011	11110000	10101011	00000000	00001111	01000101

Piksel	4			5		
	R	G	B	R	G	B
0	10111010	11110011	10010111	11010010	00000011	01011001
1	11001101	10001000	10000001	10100000	10011010	10000110
2	00100001	01001001	00011011	10111000	01011110	00100110
3	00100000	01010110	00011010	00111001	10000010	01010011
4	00111101	00001110	10001001	01010111	01100101	01100100
5	01011011	10101101	11111100	00010001	01101011	11011000
6	10101010	11101111	11101111	01100101	11111110	11010100
7	11001100	10101000	01110100	11011111	00101000	00000100

Piksel	6			7		
	R	G	B	R	G	B
0	10010101	11100100	10100000	10101110	11111011	10000011
1	01000011	10001011	01100110	01110011	11011100	00100100
2	01011000	10101111	01110110	11111111	00100010	10100110
3	10000110	00000101	10110010	11100111	00111101	00100100
4	00001111	01111100	10010101	01011110	00100011	01010010
5	11101100	11100101	01111011	01111110	00000011	10110110
6	11010011	11110101	01101010	10101000	10011111	11111011
7	00000000	00000000	00000000	00000000	00000000	00000000

FI_2 dalam desimal:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	239	22	253	19	101	10	242	129	59	216	194	20
1	70	156	216	45	241	59	94	234	213	56	176	39
2	203	109	177	229	208	243	143	248	161	64	26	24
3	101	37	77	106	123	44	149	118	207	164	186	57
4	55	83	109	50	162	188	160	193	210	201	41	174
5	32	98	93	57	131	204	229	218	93	134	7	138
6	242	20	203	229	66	49	244	29	153	234	215	187
7	10	109	70	223	44	58	67	240	171	0	15	69

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	186	243	151	210	3	89	149	228	160	174	251	131
1	205	136	129	160	154	134	67	139	102	115	220	36
2	33	73	27	184	94	38	88	175	118	255	34	166
3	32	86	26	57	130	83	134	5	178	231	61	36
4	61	14	137	87	101	100	15	124	149	94	35	82
5	91	173	252	17	107	216	236	229	123	126	3	182
6	170	239	239	101	254	212	211	245	106	168	159	251
7	204	168	116	223	40	4	0	0	0	0	0	0

- iii. Lakukan penggabungan semua FI , dalam hal ini FI_1 dan FI_2 dengan cara melakukan operasi XOR untuk masing-masing piksel.

Piksel 1: Untuk nilai R:

$$FI_1 : 255 : 11111111$$

$$FI_2 : 239 : \underline{11101111} \quad \text{XOR}$$

$$00010000 : 16$$

Untuk nilai G:

$$FI_1 : 159 : 10011111$$

$$FI_2 : 22 : \underline{00010110} \quad \text{XOR}$$

$$10001001 : 137$$

Untuk nilai B:

$$FI_1 : 127 : 01111111$$

$$FI_2 : 253 : \underline{11111101} \quad \text{XOR}$$

$$10000010 : 130$$

Piksel 64: Untuk nilai R:

$$FI_1 : 63 : 00111111$$

$$FI_2 : 0 : \underline{00000000} \quad \text{XOR}$$

$$00111111 : 63$$

Untuk nilai G:

$$FI_1 : 63 : 00111111$$

$$FI_2 : 0 : \underline{00000000} \text{ XOR}$$

$$00111111 : 63$$

Untuk nilai B:

$$FI_1 : 63 : 00111111$$

$$FI_2 : 0 : \underline{00000000} \text{ XOR}$$

$$00111111 : 63$$

Gabungan FI:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	16	137	130	16	186	113	168	70	61	27	36	76
1	169	93	48	226	50	234	129	62	36	222	224	246
2	192	190	9	67	128	171	201	34	217	20	136	116
3	140	208	198	66	222	33	252	214	5	8	89	218
4	48	255	39	254	204	254	20	199	87	185	73	235
5	184	223	236	160	61	52	245	124	175	141	226	120
6	179	28	200	101	79	177	149	60	41	56	250	90
7	154	111	71	254	14	27	75	96	235	28	206	101

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	230	147	223	8	103	129	126	161	108	81	196	188
1	179	25	144	6	88	24	95	10	118	76	99	155
2	116	19	122	252	45	222	56	237	142	0	221	89
3	128	184	222	208	52	89	7	131	116	216	130	155
4	172	117	206	79	223	7	13	220	249	97	156	237
5	26	240	242	208	122	86	44	1	237	129	60	9
6	244	239	255	55	254	245	133	241	111	215	160	196
7	220	112	50	223	104	132	128	0	128	63	63	63

b. Enkripsi

Pada tahap ini lakukan operasi XOR untuk tiap nilai R, G dan B *secret image* dan gabungan FI sehingga menghasilkan *share image*.

Piksel 1: Untuk nilai R:

$$\begin{array}{r} 142 : 10001110 \\ 16 : \underline{00010000} \quad \text{XOR} \\ \hline 10011110 : 158 \end{array}$$

Untuk nilai G:

$$\begin{array}{r} 55 : 00110111 \\ 137 : \underline{10001001} \quad \text{XOR} \\ \hline 10111110 : 190 \end{array}$$

Untuk nilai B:

$$\begin{array}{r} 100 : 01100100 \\ 130 : \underline{10000010} \quad \text{XOR} \\ \hline 11100110 : 230 \end{array}$$

Piksel 64: Untuk nilai R:

$$\begin{array}{r} 71 : 01000111 \\ 63 : \underline{00111111} \quad \text{XOR} \\ \hline 01111000 : 120 \end{array}$$

Untuk nilai G:

$$\begin{array}{r} 164 : 10100100 \\ 63 : \underline{00111111} \quad \text{XOR} \\ \hline 10011011 : 155 \end{array}$$

Untuk nilai B:

$$\begin{array}{r} 250 : 11111010 \\ 63 : \underline{00111111} \quad \text{XOR} \\ \hline 11000101 : 197 \end{array}$$

Share image:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	158	190	230	111	71	2	143	32	9	107	139	66
1	118	253	40	101	119	52	247	73	193	33	53	151
2	53	0	189	135	171	60	33	64	181	222	253	118
3	165	134	231	199	24	184	211	81	172	92	213	142
4	54	25	117	177	240	2	114	178	195	44	196	210
5	194	176	200	139	211	96	88	231	47	166	198	97
6	244	225	103	37	27	189	59	119	148	224	18	52
7	63	116	137	217	88	215	27	29	95	178	195	39

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	212	124	159	21	244	197	53	167	46	98	111	55
1	134	199	106	1	150	112	177	55	62	131	232	128
2	39	171	4	42	227	134	55	185	215	212	130	226
3	68	152	251	220	178	22	76	89	112	90	182	245
4	140	235	11	16	241	20	143	242	153	168	104	111
5	244	57	121	135	162	84	159	115	62	210	12	214
6	90	21	45	198	40	220	77	211	61	106	72	18
7	202	45	134	189	26	108	158	31	26	120	155	197

2. Penyisipan Share

Pada contoh ini, *share* dengan ukuran 3 x 3 dan *password* dengan panjang 3 karakter akan disisipkan ke dalam *cover* dengan ukuran 8 x 8, dimana jumlah bit nilai intensitas yang akan digantikan dengan bit *share* dan *password* adalah 4. *Password* yang digunakan 3 karakter karena hanya untuk menjelaskan proses penyisipan.

Share image:

Piksel	0			1			2		
	R	G	B	R	G	B	R	G	B
0	25	113	138	76	216	52	226	53	212
1	186	197	136	195	175	31	248	77	9
2	51	197	66	219	238	90	8	180	83

Share image dalam biner:

Piksel	0			1			2		
	R	G	B	R	G	B	R	G	B
0	00011001	01110001	10001010	01001100	11011000	00110100	11100010	00110101	11010100
1	10111010	11000101	10001000	11000011	10101111	00011111	11111000	01001101	00001001
2	00110011	11000101	01000010	11011011	11101110	01011010	00001000	10110100	01010011

Password: As2

Karakter	A	s	2
Desimal	65	115	50
Biner	01000001	01110011	00110010

Cover:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	165	121	238	161	234	241	227	15	98	144	1	15
1	63	233	246	170	148	158	6	178	234	14	83	181
2	215	189	163	10	120	35	125	80	235	67	115	235
3	92	13	205	240	147	90	244	7	244	187	144	210
4	196	11	68	195	71	239	232	179	179	164	107	67
5	147	36	128	201	146	229	73	28	18	227	70	143
6	222	191	228	207	198	8	210	109	218	53	231	167
7	16	10	241	101	148	155	202	2	56	62	54	38

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	186	200	255	61	76	155	250	244	192	182	74	111
1	253	136	147	190	128	100	252	251	183	13	15	142
2	221	56	89	197	133	50	80	154	24	54	181	223
3	253	184	155	166	253	193	229	133	196	130	83	195
4	158	78	194	85	89	213	238	13	16	167	164	163
5	211	105	156	169	56	252	123	118	85	65	124	4
6	42	91	220	138	96	175	73	172	8	82	136	119
7	180	21	184	93	195	255	46	239	56	223	131	131

Ubah cover menjadi citra warna HSI dengan contoh perhitungan sebagai berikut:

Piksel 1:

R	G	B
165	121	238

Lakukan normalisasi menggunakan rumus:

$$r = \frac{R}{R + G + B}$$

$$g = \frac{G}{R + G + B}$$

$$b = \frac{B}{R + G + B}$$

$$r = \frac{165}{165 + 121 + 238} = 0.3148854961832061$$

$$g = \frac{121}{165 + 121 + 238} = 0.23091603053435114$$

$$b = \frac{238}{165 + 121 + 238} = 0.45419847328244273$$

Karena $b > g$, maka untuk nilai H dihitung menggunakan rumus:

$$h = 2\pi - \cos^{-1} \left[\frac{0.5 \times \{(r-g) + (r-b)\}}{\sqrt{[(r-g)^2 + (r-b)(g-b)]}} \right]$$

$$h = 2\pi - \cos^{-1} \left[\frac{0.5 \times \{(0.3149 - 0.2309) + (0.3149 - 0.4542)\}}{\sqrt{[(0.3149 - 0.2309)^2 + (0.3149 - 0.4542)(0.2309 - 0.4542)]}} \right]$$

$$h = 261.856042021813$$

Nilai s dihitung menggunakan rumus:

$$s = 1 - 3 \times \min(r, g, b)$$

$$s = 1 - 3 \times \min(0.3149, 0.2309, 0.4542)$$

$$s = 1 - 3 \times 0.2309$$

$$s = 0.3072519083969466$$

Nilai i dihitung menggunakan rumus:

$$i = (R + G + B)/(3 \times 255)$$

$$i = (165 + 121 + 238)/(3 \times 255)$$

$$i = 0.68496732026143792$$

Untuk representasi nilai i diubah ke dalam range nilai 0, 255. Sehingga i dikalikan dengan 255, menjadi:

$$i = 0.68496732026143792 \times 255$$

$$i = 175$$

Lakukan hal yang sama untuk semua piksel, sehingga diperoleh citra HSI:

Piksel	0			1			2		
	H	S	I	H	S	I	H	S	I
0	261.856	0.307	175	183.650	0.651	181	30.000	0.138	189
1	184.531	0.241	212	333.004	0.067	159	132.520	0.818	55
2	337.140	0.868	113	193.630	0.957	139	256.390	0.455	147
3	354.906	0.981	53	215.725	0.849	93	223.898	0.518	139
4	228.935	0.129	214	355.117	0.239	179	349.107	0.541	122
5	231.459	0.373	97	17.696	0.282	139	34.268	0.605	127
6	54.603	0.160	229	59.276	0.200	229	94.571	0.721	86
7	340.302	0.395	122	239.225	0.771	57	193.808	0.646	153

Piksel	3			4			5		
	H	S	I	H	S	I	H	S	I
0	264.162	0.874	103	342.494	0.880	92	309.208	0.653	104
1	22.111	0.434	159	285.372	0.578	168	280.637	0.240	192
2	300.000	0.958	165	0.000	0.090	197	9.826	0.546	40
3	279.924	0.201	180	24.222	0.405	113	332.316	0.523	147
4	16.740	0.215	197	282.352	0.456	143	331.248	0.333	157
5	137.648	0.186	204	238.425	0.341	129	275.050	0.648	159
6	319.773	0.285	186	359.334	0.854	89	53.046	0.218	109
7	264.699	0.390	136	13.898	0.010	165	89.449	0.938	64

Piksel	6			7		
	H	S	I	H	S	I
0	291.311	0.106	214	241.306	0.888	89
1	57.705	0.942	138	186.845	0.250	135
2	296.225	0.391	179	344.873	0.977	87
3	159.212	0.647	150	40.893	0.260	51
4	224.546	0.643	118	298.767	0.836	128
5	272.093	0.296	136	201.487	0.486	181
6	96.826	0.905	84	122.637	0.595	114
7	162.070	0.270	112	0.000	0.190	162

Nilai Intensitas citra HSI:

Piksel	0	1	2	3	4	5	6	7
0	175	181	189	103	92	104	214	89
1	212	159	55	159	168	192	138	135
2	113	139	147	165	197	40	179	87
3	53	93	139	180	113	147	150	51
4	214	179	122	197	143	157	118	128
5	97	139	127	204	129	159	136	181
6	229	229	86	186	89	109	84	114
7	122	57	153	136	165	64	112	162

Nilai Intensitas citra HSI dalam biner:

Piksel	0	1	2	3	4	5	6	7
0	10101111	10110101	10111101	01100111	01011100	01101000	11010110	01011001
1	11010100	10011111	00110111	10011111	10101000	11000000	10001010	10000111
2	01110001	10001011	10010011	10100101	11000101	00101000	10110011	01010111
3	00110101	01011101	10001011	10110100	01110001	10010011	10010110	00110011
4	11010110	10110011	01111010	11000101	10001111	10011101	01110110	10000000
5	01100001	10001011	01111111	11001100	10000001	10011111	10001000	10110101
6	11100101	11100101	01010110	10111010	01011001	01101101	01010100	01110010
7	01111010	00111001	10011001	10001000	10100101	01000000	01110000	10100010

Ganti bit intensitas dengan bit *share image*, *natural share image* dan *password*. Pada contoh ini jumlah bit yang akan digantikan adalah 4 bit terakhir. Adapun urutan penyisipan bit adalah panjang *password*, panjang *width*, *password* kemudian *shares*.

Sehingga menjadi:

Piksel	0	1	2	3	4	5	6	7
0	10100011	10111000	10110100	01100001	01010111	01100011	11010011	01010010
1	11010001	10011001	00110111	10010001	10101000	11001010	10000100	10001100
2	01111101	10001000	10010011	10100100	11001110	00100010	10110011	01010101
3	00111101	01010100	10001011	10111010	01111100	10010101	10011000	00111000
4	11011100	10110011	01111010	11001111	10000001	10011111	01111111	10001000
5	011000100	10001101	01110000	11001001	10000011	10010011	10001100	10110101
6	11100100	11100010	01011101	10111011	01011110	01101110	01010101	01111010
7	01110000	00111000	10011011	10000100	10100101	01000011	01110000	10100010

Nilai Intensitas setelah dilakukan penyisipan:

Piksel	0	1	2	3	4	5	6	7
0	163	184	180	97	87	99	211	82
1	209	153	55	145	168	202	132	140
2	125	136	147	164	206	34	179	85
3	61	84	139	186	124	149	152	56
4	220	179	122	207	129	159	127	136
5	100	141	112	201	131	147	140	181
6	228	226	93	187	94	110	85	122
7	112	56	155	132	165	67	112	162

Setelah melakukan penyisipan konversi kembali dari citra warna HSI ke RGB dengan contoh perhitungan sebagai berikut:

Piksel 1:

$$h = 261.856042021813$$

$$s = 0.3072519083969466$$

$$i = 163/255$$

$$i = 0.63921568627450975$$

Karena $h > 240$, maka $h - 240$.

$$h = 21.856042021812982$$

$$x = i \cdot x \cdot (1 - s)$$

$$x = 0.63921568627450975 \cdot x \cdot (1 - 0.3072519083969466)$$

$$x = 0.44281544678940271$$

$$y = i \cdot x \left[1 + \frac{s \cdot x \cos(h)}{\cos(\frac{\pi}{3} h)} \right]$$

$$y = 0.63921568627450975 \cdot x \left[1 + \frac{0.3072519083969466 \cdot x \cos(21.856042021812982)}{\cos(\frac{\pi}{3} 21.856042021812982)} \right]$$

$$y = 0.87099236641221378$$

$$z = 3 \cdot x \cdot i - (x + y)$$

$$z = 3 \cdot x \cdot 0.63921568627450975 - (0.44281544678940271 + 0.87099236641221378)$$

$$z = 0.60383924562191282$$

$$R = z \cdot 255$$

$$G = x \cdot 255$$

$$B = y \cdot 255$$

$$R = 0.60383924562191282 \cdot 255$$

$$R = 154$$

$$G = 0.44281544678940271 \cdot 255$$

$$G = 113$$

$$B = 0.87099236641221378 \cdot 255$$

$$B = 222$$



Lakukan untuk semua piksel, sehingga diperoleh citra RGB (*stego image*):

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	154	113	222	140	203	209	361	24	156	262	2	27
1	73	270	285	164	143	152	2	70	92	22	130	283
2	142	125	108	25	297	87	125	80	236	79	136	277
3	54	8	121	127	78	48	206	6	206	193	149	217
4	470	26	163	207	75	254	144	111	111	301	197	123
5	142	35	123	148	107	168	206	79	51	311	96	196
6	237	204	243	340	325	13	109	57	113	66	287	208
7	20	13	303	42	62	64	361	4	100	159	139	98

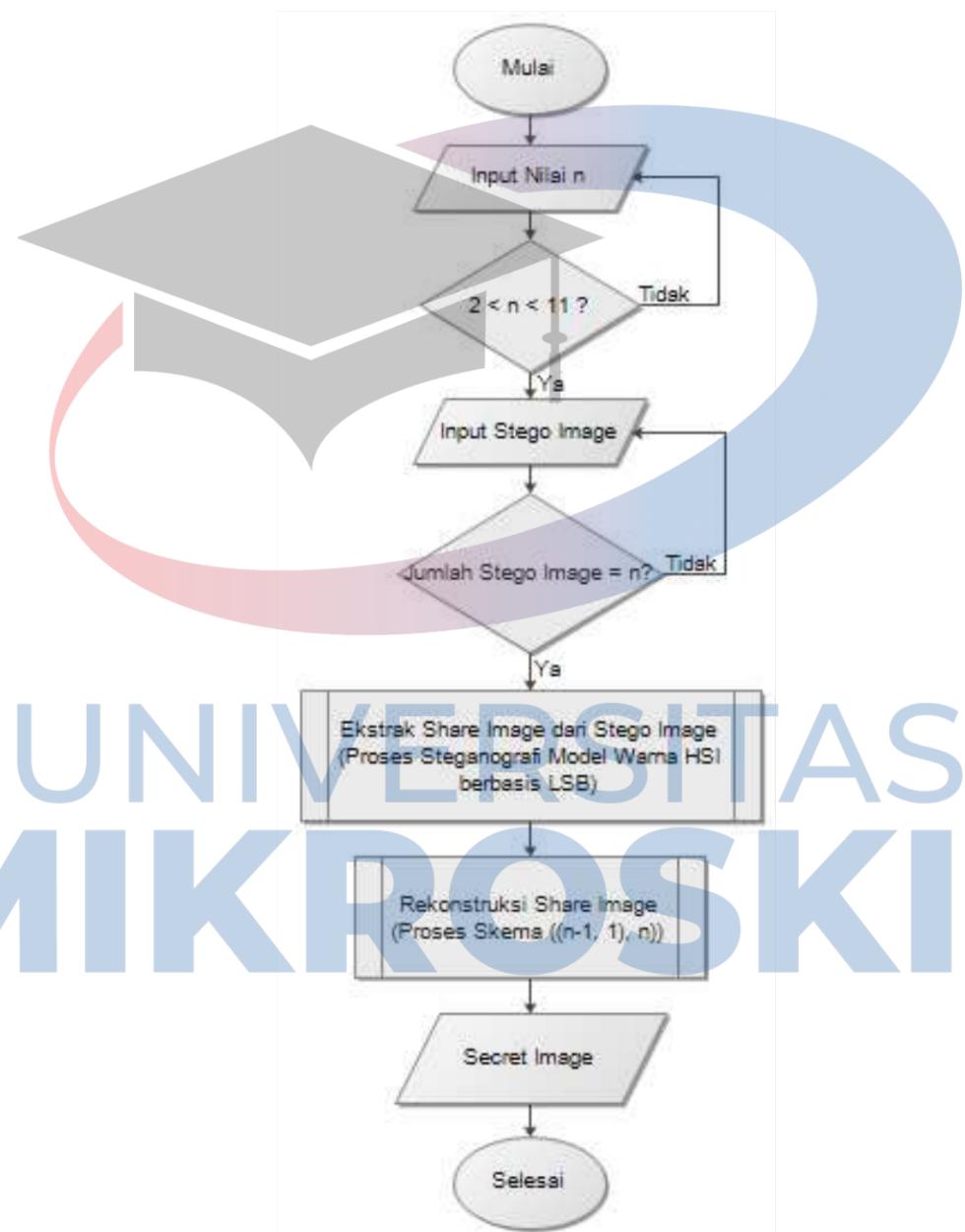
Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	76	81	104	62	77	158	231	225	177	122	50	74
1	238	128	138	275	186	145	145	145	106	32	37	351
2	373	95	150	53	36	13	167	321	50	30	101	124
3	159	116	97	121	185	141	187	109	160	54	34	80
4	142	70	175	105	110	263	340	19	23	138	135	135
5	176	87	130	156	52	233	158	152	110	183	349	11
6	34	73	176	111	77	141	74	173	8	89	148	129
7	231	27	237	34	72	94	45	235	55	223	131	131

Lakukan proses tersebut untuk penyisipan semua *share* ke dalam masing-masing *cover image*.

Dalam proses penyisipan bit ke dalam citra HSI yang dalam hal ini adalah nilai intensitas, memungkinkan nilai intensitas lebih dari angka 1. Sehingga jika direpresentasikan ke dalam warna RGB yaitu dengan mengalikan dengan 255 membuat warna tidak bisa direpresentasikan, dimana nilai maksimum warna adalah 255. Hal ini terjadi jika nilai RGB sebelum dikonversi ke HSI mendekati angka 255. Adapun solusi yang kami tawarkan terkait masalah tersebut adalah dengan mengurangi nilai maksimum dari R, G atau B sebelum konversi ke HSI, setelah itu ubah ke dalam model HSI kemudian sisip kembali sampai hasil konversi ulang ke RGB tidak melebihi 255.

3.1.1.3. Analisis Proses Ekstrak

Proses ekstrak bertujuan untuk mendapatkan kembali *secret image* setelah proses *embedding*. Tahapan proses ekstrak terdiri dari ekstrak *share* dan rekonstruksi *share*. Adapun *flowchart* proses ekstrak dapat dilihat pada Gambar 3. 4 berikut ini:

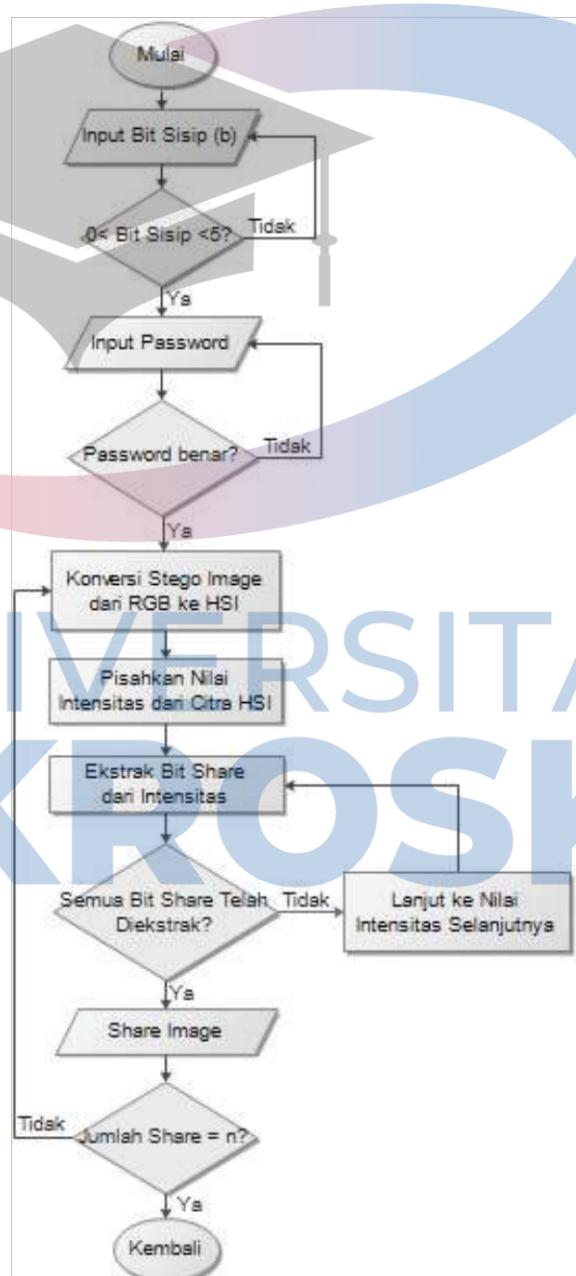


Gambar 3. 4 *Flowchart* Proses Ektrak

Berikut merupakan langkah-langkah dalam proses ekstrak:

1. Ekstrak *Share*

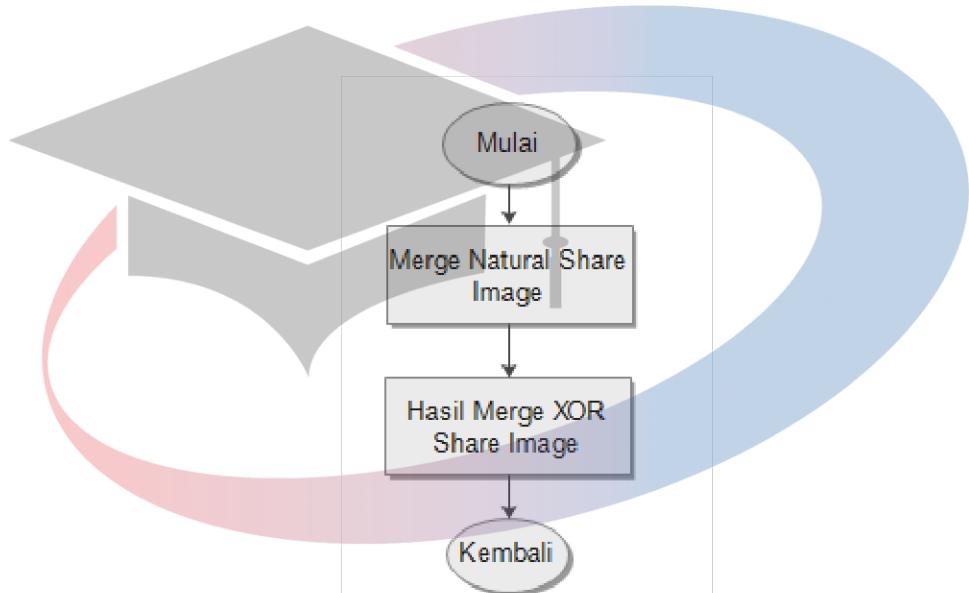
Proses ini bertujuan untuk mendapatkan *share* yang telah disisipkan pada *stego image*. Untuk mendapatkan *share*, *stego image* (RGB) dikonversi ke warna HSI kemudian mengambil bit *share* yaitu b bit trakhir nilai intensitas, dimana b merupakan jumlah bit nilai intensitas yang digantikan dengan bit *share*. Adapun *flowchart* proses ekstrak dapat dilihat pada Gambar 3. 5 berikut ini:



Gambar 3. 5 Flowchart Ekstrak *Share*

2. Rekonstruksi Share

Proses ini bertujuan untuk mendapatkan kembali *secret image* setelah pembentukan *share* pada proses *embedding*. Untuk mendapatkan *secret image* dilakukan rekonstruksi *share* yaitu dengan melakukan penggabungan $(n - 1)$ *natural share image*, hasil penggabungan kemudian dilakukan operasi XOR dengan *share image*. Adapun *flowchart* proses Rekonstruksi *Share* dapat dilihat pada Gambar 3. 6 berikut ini:



Gambar 3. 6 *Flowchart* Rekonstruksi Share

3.1.1.4. Contoh Perhitungan Proses Ekstrak

Pada tahap ini dilakukan pengembalian *secret image* yang terdiri dari dua tahap yaitu ekstrak *share* dan rekonstruksi *share*.

1. Ekstrak *Share*

Tahap ini bertujuan untuk mengambil *share image* yang telah disisipkan kedalam *cover image*. Proses pengembalian dilakukan dengan mengkonversi *stego image* (citra RGB) menjadi cinta HSI, kemudian mengambil 4 bit trakhir nilai Intensitas sebagai bit *share*.

Stego image:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	154	113	222	140	203	209	361	24	156	262	2	27
1	73	270	285	164	143	152	2	70	92	22	130	283
2	142	125	108	25	297	87	125	80	236	79	136	277
3	54	8	121	127	78	48	206	6	206	193	149	217
4	470	26	163	207	75	254	144	111	111	301	197	123
5	142	35	123	148	107	168	206	79	51	311	96	196
6	237	204	243	340	325	13	109	57	113	66	287	208
7	20	13	303	42	62	64	361	4	100	159	139	98

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	76	81	104	62	77	158	231	225	177	122	50	74
1	238	128	138	275	186	145	145	145	106	32	37	351
2	373	95	150	53	36	13	167	321	50	30	101	124
3	159	116	97	121	185	141	187	109	160	54	34	80
4	142	70	175	105	110	263	340	19	23	138	135	135
5	176	87	130	156	52	233	158	152	110	183	349	11
6	34	73	176	111	77	141	74	173	8	89	148	129
7	231	27	237	34	72	94	45	235	55	223	131	131

Konversi ke citra HSI dengan contoh perhitungan sebagai berikut:

Piksel 1:

R	G	B
154	113	222

Lakukan normalisasi sehingga diperoleh nilai r, g dan b adalah:

$$r = \frac{154}{154 + 113 + 222} = 0.31492842535787319$$

$$g = \frac{113}{154 + 113 + 222} = 0.2310838445807771$$

$$b = \frac{222}{154 + 113 + 222} = 0.45398773006134968$$

Karena $b > g$, maka untuk nilai H dihitung menggunakan rumus:

$$h = 2\pi - \cos^{-1} \left[\frac{0.5 \times \{(r-g) + (r-b)\}}{\sqrt{[(r-g)^2 + (r-b)(g-b)]}} \right]$$

$$h = 2\pi - \cos^{-1} \left[\frac{0.5 \times \{(0.3149 - 0.2311) + (0.3149 - 0.4540)\}}{\sqrt{[(0.3149 - 0.2311)^2 + (0.3149 - 0.4540)(0.2311 - 0.4540)]}} \right]$$

$$h = 261.86112573683351$$

Nilai s dihitung menggunakan rumus:

$$s = 1 - 3 \times \min(r, g, b)$$

$$s = 1 - 3 \times \min(0.3149, 0.2311, 0.4540)$$

$$s = 1 - 3 \times 0.2311$$

$$s = 0.30674846625766872$$

Nilai i dihitung menggunakan rumus:

$$i = (R + G + B) / (3 \times 255)$$

$$i = (154 + 113 + 222) / (3 \times 255)$$

$$i = 0.63921568627450975$$

Untuk representasi nilai i diubah ke dalam range nilai 0, 255. Sehingga i dikalikan dengan 255, menjadi:

$$i = 0.63921568627450975 \times 255$$

$$i = 163$$

Lakukan untuk semua piksel sehingga diperoleh citra HSI sebagai berikut:

Nilai intensitas:

Piksel	0	1	2	3	4	5	6	7
0	163	184	180	97	87	99	211	82
1	209	153	55	145	168	202	132	140
2	125	136	147	164	206	34	179	85
3	61	84	139	186	124	149	152	56
4	220	179	122	207	129	159	127	136
5	100	141	112	201	131	147	140	181
6	228	226	93	187	94	110	85	122
7	112	56	155	132	165	67	112	162

Ubah nilai intensitas menjadi biner kemudian ambil 4 bit terakhir sebagai data yang telah disisipkan.

i = 163: 10100011

Lakukan untuk semua piksel, sehingga diperoleh:

Piksel	0	1	2	3	4	5	6	7
0	10100011	10111000	10110100	01100001	01010111	01100011	11010011	01010010
1	11010001	10011001	00110111	10010001	10101000	11001010	10000100	10001100
2	01111101	10001000	10010011	10100100	11001110	00100010	10110011	01010101
3	00111101	01010100	100001011	10111010	01111100	100010101	10011000	00111000
4	11011100	10110011	01111010	11001111	10000001	10011111	01111111	10001000
5	01100100	10000101	01110000	11001001	10000011	10010011	100001100	10110101
6	11100100	11100010	01011101	10111011	01011110	01101110	01010101	01111010
7	01110000	00111000	10011011	10000100	10100101	01000011	01110000	10100010

Ambil bit share sehingga diperoleh bit share:

Piksel	0			1			2		
	R	G	B	R	G	B	R	G	B
0	00011001	01110001	10001010	01001100	11011000	00110100	11100010	00110101	11010100
1	10111010	11000101	10001000	11000011	10101111	00011111	11111000	01001101	00001001
2	00110011	11000101	01000010	11011011	11101110	01011010	00001000	10110100	01010011

Ubah ke dalam nilai desimal, sehingga diperoleh:

Share image:

Piksel	0			1			2		
	R	G	B	R	G	B	R	G	B
0	25	113	138	76	216	52	226	53	212
1	186	197	136	195	175	31	248	77	9
2	51	197	66	219	238	90	8	180	83

Lakukan hal tersebut untuk semua *stego image* (n buah *stego image*) untuk mendapatkan n buah *share*.

2. Rekonstruksi *Share*

Tahap ini bertujuan untuk mendapatkan *secret image* setelah proses pembentukan *share* pada proses *embedding*. Proses rekonstruksi *share* dilakukan dengan melakukan penggabungan ($n - 1$) *natural share image* menggunakan operasi XOR sehingga menghasilkan satu citra *share* gabungan. Kemudian citra *share* hasil gabungan dilakukan operasi XOR dengan *share image* sehingga menghasilkan *secret image*. Pada contoh ini, jumlah *natural share image* yang digunakan adalah 2, sesuai dengan jumlah *natural image* pada saat pembentukan *share* pada proses *embedding*.

Natural share image 1:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	255	159	127	3	223	123	90	199	6	195	230	88
1	239	193	232	207	195	209	223	212	241	230	80	209
2	11	211	184	166	80	88	70	218	120	84	146	108
3	233	245	139	40	165	13	105	160	202	172	227	227
4	7	172	74	204	110	66	180	6	133	112	96	69
5	152	189	177	153	190	248	16	166	242	11	229	242
6	65	8	3	128	13	128	97	33	176	210	45	225
7	144	2	1	33	34	33	8	144	64	28	193	32

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	92	96	72	218	100	216	235	69	204	255	63	63
1	126	145	17	166	194	158	28	129	16	63	191	191
2	85	90	97	68	115	248	96	66	248	255	255	255
3	160	238	196	233	182	10	129	134	198	63	191	191
4	145	123	71	24	186	99	2	160	108	63	191	191
5	65	93	14	193	17	142	192	228	150	255	63	191
6	94	0	16	82	0	33	86	4	5	127	63	63
7	16	216	70	0	64	128	128	0	128	63	63	63

Natural share image 2:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	239	22	253	19	101	10	242	129	59	216	194	20
1	70	156	216	45	241	59	94	234	213	56	176	39
2	203	109	177	229	208	243	143	248	161	64	26	24
3	101	37	77	106	123	44	149	118	207	164	186	57
4	55	83	109	50	162	188	160	193	210	201	41	174
5	32	98	93	57	131	204	229	218	93	134	7	138
6	242	20	203	229	66	49	244	29	153	234	215	187
7	10	109	70	223	44	58	67	240	171	0	15	69

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	186	243	151	210	3	89	149	228	160	174	251	131
1	205	136	129	160	154	134	67	139	102	115	220	36
2	33	73	27	184	94	38	88	175	118	255	34	166
3	32	86	26	57	130	83	134	5	178	231	61	36
4	61	14	137	87	101	100	15	124	149	94	35	82
5	91	173	252	17	107	216	236	229	123	126	3	182
6	170	239	239	101	254	212	211	245	106	168	159	251
7	204	168	116	223	40	4	0	0	0	0	0	0

Lakukan penggabungan semua *natural share image*, dalam hal ini *natural share image*₁ dan *natural share image*₂ dengan contoh perhitungan sebagai berikut:

Piksel 1: Untuk nilai R:

*Natural share image*₁ : 255 : 11111111

*Natural share image*₂ : 239 : 11101111 XOR
00010000 : 16

Untuk nilai G:

*Natural share image*₁ : 159 : 10011111

*Natural share image*₂ : 22 : 00010110 XOR
10001001 : 137

Untuk nilai B:

*Natural share image*₁ : 127 : 01111111

*Natual share image*₂ : 253 : 11111101 XOR
10000010 : 130

Piksel 64: Untuk nilai R:

*Natural share image*₁ : 63 : 00111111

*Natual share image*₂ : 0 : 00000000 XOR
 00111111 : 63

Untuk nilai G:

Natural share image₁ : 63 : 00111111

Untuk nilai B:

*Natural share image*₁ : 63 : 00111111

$$Natural\ share\ image_2 : 0 : \frac{00000000}{00111111} : 63$$

Sehingga menjadi:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	16	137	130	16	186	113	168	70	61	27	36	76
1	169	93	48	226	50	234	129	62	36	222	224	246
2	192	190	9	67	128	171	201	34	217	20	136	116
3	140	208	198	66	222	33	252	214	5	8	89	218
4	48	255	39	254	204	254	20	199	87	185	73	235
5	184	223	236	160	61	52	245	124	175	141	226	120
6	179	28	200	101	79	177	149	60	41	56	250	90
7	154	111	71	254	14	27	75	96	235	28	206	101

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	230	147	223	8	103	129	126	161	108	81	196	188
1	179	25	144	6	88	24	95	10	118	76	99	155
2	116	19	122	252	45	222	56	237	142	0	221	89
3	128	184	222	208	52	89	7	131	116	216	130	155
4	172	117	206	79	223	7	13	220	249	97	156	237
5	26	240	242	208	122	86	44	1	237	129	60	9
6	244	239	255	55	254	245	133	241	111	215	160	196
7	220	112	50	223	104	132	128	0	128	63	63	63

Share mage:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	158	190	230	111	71	2	143	32	9	107	139	66
1	118	253	40	101	119	52	247	73	193	33	53	151
2	53	0	189	135	171	60	33	64	181	222	253	118
3	165	134	231	199	24	184	211	81	172	92	213	142
4	54	25	117	177	240	2	114	178	195	44	196	210
5	194	176	200	139	211	96	88	231	47	166	198	97
6	244	225	103	37	27	189	59	119	148	224	18	52
7	63	116	137	217	88	215	27	29	95	178	195	39

Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	212	124	159	21	244	197	53	167	46	98	111	55
1	134	199	106	1	150	112	177	55	62	131	232	128
2	39	171	4	42	227	134	55	185	215	212	130	226
3	68	152	251	220	178	22	76	89	112	90	182	245
4	140	235	11	16	241	20	143	242	153	168	104	111
5	244	57	121	135	162	84	159	115	62	210	12	214
6	90	21	45	198	40	220	77	211	61	106	72	18
7	202	45	134	189	26	108	158	31	26	120	155	197

Lakukan operasi XOR hasil gabungan dengan share image.

Piksel 1: Untuk nilai R:

Hasil Merge : 16 : 00010000

Share image : 158 : 10011110 XOR

10001110 : 142

Untuk nilai G:

Hasil Merge : 137 : 10001001

Share image : 190 : 10111110 XOR
00110111 : 55

Untuk nilai B:

Hasil Merge : 130 : 10000010

Share image : 230 : 11100110 XOR
01100100 : 100

Piksel 64: Untuk nilai R:

Hasil Merge : 63 : 00111111

Share image : 120 : 01111000 XOR
01000111 : 71

Untuk nilai G:

Hasil Merge : 63 : 00111111

Share image : 155 : 10011011 XOR
10100100 : 164

Untuk nilai B:

Hasil Merge : 63 : 00111111

Share image : 197 : 11000101 XOR
11111010 : 250

Sehingga menjadi:

Piksel	0			1			2			3		
	R	G	B	R	G	B	R	G	B	R	G	B
0	142	55	100	127	253	115	39	102	52	112	175	14
1	223	160	24	135	69	222	118	119	229	255	213	97
2	245	190	180	196	43	151	232	98	108	202	117	2
3	41	86	33	133	198	153	47	135	169	84	140	84
4	6	230	82	79	60	252	102	117	148	149	141	57
5	122	111	36	43	238	84	173	155	128	43	36	25
6	71	253	175	64	84	12	174	75	189	216	232	110
7	165	27	206	39	86	204	80	125	180	174	13	66

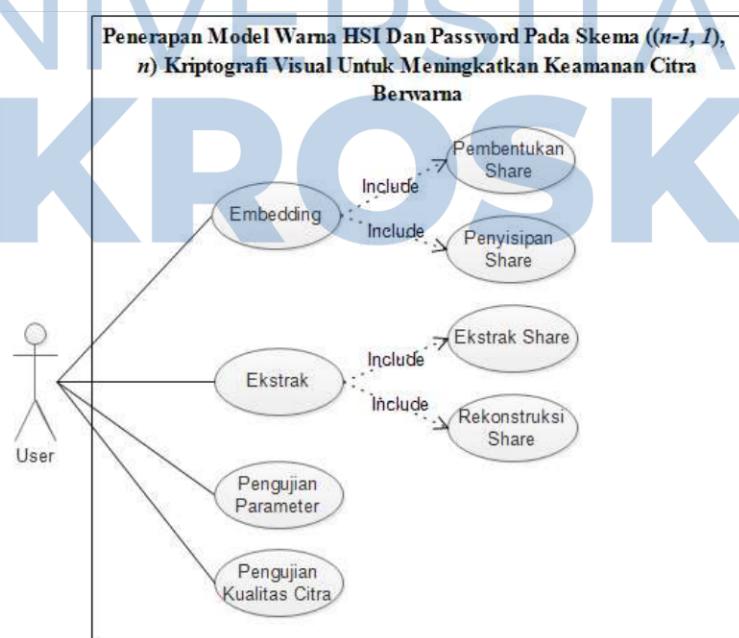
Piksel	4			5			6			7		
	R	G	B	R	G	B	R	G	B	R	G	B
0	50	239	64	29	147	68	75	6	66	51	171	139
1	53	222	250	7	206	104	238	61	72	207	139	27
2	83	184	126	214	206	88	15	84	89	212	95	187
3	196	32	37	12	134	79	75	218	4	130	52	110
4	32	158	197	95	46	19	130	46	96	201	244	130
5	238	201	139	87	216	2	179	114	211	83	48	223
6	174	250	210	241	214	41	200	34	82	189	232	214
7	22	93	180	98	114	232	30	31	154	71	164	250

3.1.2. Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan penting dalam pengembangan sistem untuk mengetahui secara detail sistem yang akan dikembangkan. Analisis kebutuhan dibagi menjadi dua, yaitu analisis kebutuhan fungsional yang merupakan spesifikasi inti mengenai hal-hal yang bisa dilakukan oleh sistem dan analisis kebutuhan non-fungsional yang merupakan komponen pendukung pada sistem.

3.1.2.1. Analisis Kebutuhan Fungsional

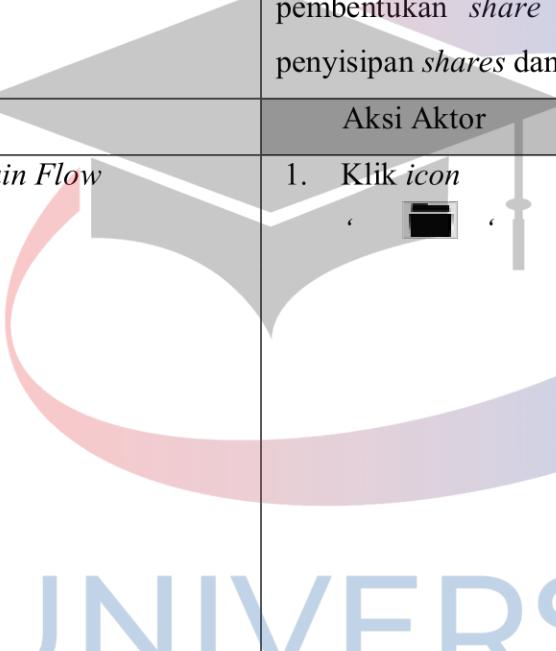
Adapun kebutuhan fungsional digambarkan dengan menggunakan *Use Case Diagram* pada Gambar 3. 7 berikut ini:



Gambar 3. 7 Use Case Diagram Perangkat Lunak

Berikut narasi untuk *use case* diagram di atas:

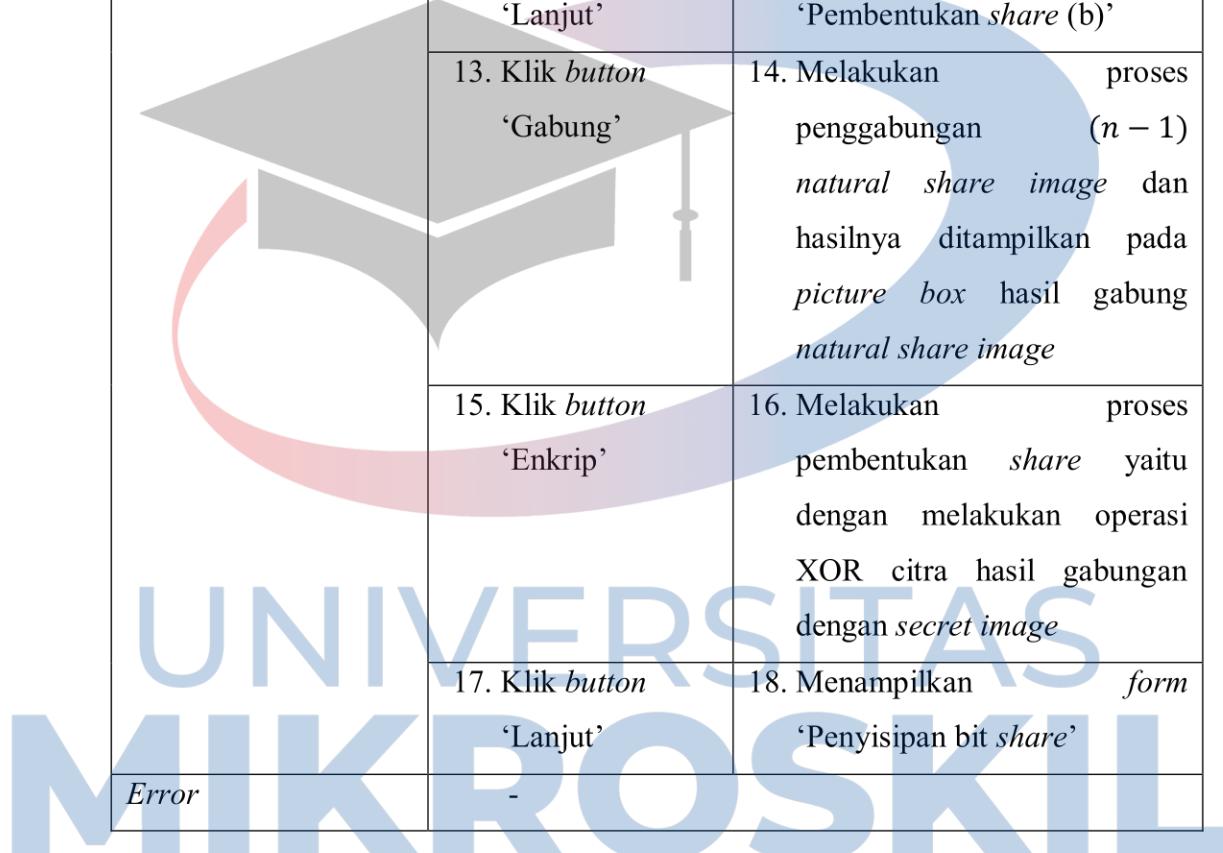
Tabel 3. 1 Tabel Narasi *Use Case* Proses *Embedding*

Nama <i>use case</i>	<i>Embedding</i>	
Case terkait	Pembentukan <i>share</i> dan Penyisipan <i>share</i>	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses <i>embedding</i> , yang meliputi pembentukan <i>share</i> (a), pembentukan <i>share</i> (b) dan penyisipan <i>shares</i> dan <i>password</i>	
	Aksi Aktor	Respon Sistem
<i>Main Flow</i>	1. Klik icon  	2. Menampilkan kotak dialog untuk membuka <i>secret image</i> 3. Melakukan pengecekan ukuran <i>secret image</i> , jika bukan persegi atau bukan merupakan kelipatan 8 maka akan dilakukan <i>scalling</i> dan hasilnya merupakan citra persegi dan merupakan kelipatan 8 yang paling mendekati ukuran citra <i>input</i> 4. Menampilkan <i>secret image</i> pada <i>picture box secret image</i>
<i>Error</i>	-	

Tabel 3. 2 Tabel Narasi *Use Case* Proses Pembentukan *Share*

Nama <i>use case</i>	Pembentukan <i>Share</i>
Aktor	<i>User</i>

Deskripsi	<i>Use case</i> ini menjelaskan proses pembentukan <i>share</i> , yang meliputi pembentukan <i>share</i> (a) dan pembentukan <i>share</i> (b)		
	Aksi Aktor	Respon Sistem	
<i>Main flow</i>	<p>1. <i>Input</i> nilai n</p>  <p>3. Klik icon ‘’</p>	<p>2. Melakukan pengecekan nilai minimum dan maksimum nilai n. Jika nilai yang di-<i>input</i> lebih kecil dari nilai minimum maka nilai minimum akan dijadikan sebagai nilai n secara otomatis dan jika nilai yang di-<i>input</i> lebih besar dari nilai maksimum maka nilai maksimum akan dijadikan sebagai nilai n</p> <p>4. Menampilkan kotak dialog untuk membuka <i>natural image</i></p> <p>5. Melakukan pengecekan ukuran <i>natural image</i>, jika tidak sama dengan <i>secret image</i> yang telah di-<i>input</i> maka akan dilakukan <i>scalling</i> otomatis sehingga ukurannya sama</p> <p>6. Menampilkan <i>natural image</i> pada <i>picture box</i> <i>natural image</i></p> <p>7. Klik button ‘Hapus’</p> <p>8. Menghapus <i>natural image</i> dari <i>picture box</i></p>	



	9. Klik button 'Ekstraksi Fitur'	10. Melakukan ekstraksi fitur untuk tiap <i>natural image</i> dan hasilnya berupa <i>natural share image</i> yang kemudian ditampilkan pada <i>picture box</i> hasil ekstrak <i>natural image</i>
	11. Klik button 'Lanjut'	12. Menampilkan <i>form</i> 'Pembentukan share (b)'
	13. Klik button 'Gabung'	14. Melakukan proses penggabungan $(n - 1)$ <i>natural share image</i> dan hasilnya ditampilkan pada <i>picture box</i> hasil gabung <i>natural share image</i>
	15. Klik button 'Enkrip'	16. Melakukan proses pembentukan <i>share</i> yaitu dengan melakukan operasi XOR citra hasil gabungan dengan <i>secret image</i>
	17. Klik button 'Lanjut'	18. Menampilkan <i>form</i> 'Penyisipan bit share'
Error	-	

Tabel 3. 3 Tabel Narasi Use Case Proses Penyisipan Share

Nama use case	Penyisipan Share	
Aktor	User	
Deskripsi	<i>Use case</i> ini menjelaskan proses penyisipan <i>shares</i> dan <i>password</i>	
	Aksi Aktor	Respon Sistem

<i>Main flow</i>	<p>1. <i>Input</i> jumlah bit sisip (<i>b</i>)</p> <p>2. Melakukan pengecekan nilai minimum dan maksimum nilai <i>b</i>. Jika nilai yang <i>di-input</i> lebih kecil dari nilai minimum maka nilai minimum akan dijadikan sebagai nilai <i>b</i> secara otomatis dan jika nilai yang <i>di-input</i> lebih besar dari nilai maksimum maka nilai maksimum akan dijadikan sebagai nilai <i>b</i></p> <p>3. <i>Input password</i></p> <p>4. Melakukan pengecekan panjang minimal dan maksimal <i>password</i> serta kombinasi <i>password</i></p> <p>5. Klik icon </p> <p>6. Menampilkan kotak dialog untuk membuka <i>cover image</i></p> <p>7. Melakukan pengecekan ukuran <i>cover image</i>. Jika bukan merupakan citra persegi maka akan dilakukan <i>scalling</i> otomatis dan hasilnya akan menjadi persegi yang ukurannya mendekati ukuran citra <i>input</i> atau menjadi ukuran minimum <i>cover</i> jika ukuran (<i>height</i> atau <i>width</i>) citra <i>input</i> lebih kecil dari ukuran <i>cover</i> minimum</p>
------------------	--

		8. Menampilkan <i>cover image</i> pada <i>picture box cover image</i>
	9. Klik <i>button</i> ‘Sisip’	10. Melakukan proses penyisipan bit <i>share</i> dan <i>password</i> ke dalam <i>cover image</i>
	11. Klik <i>button</i> ‘Simpan’	12. Menyimpan <i>stego image</i> ke lokasi yang telah ditentukan
	13. Klik <i>button</i> ‘Selesai’	14. Mengakhiri proses <i>embedding</i> dan kembali ke <i>form</i> utama
<i>Error</i>	1. Panjang <i>password</i> lebih kecil dari 6 atau lebih besar dari 14 karakter atau tidak terdiri dari gabungan huruf kapital, non-kapital, karakter khusus dan angka	

Tabel 3. 4 Tabel Narasi *Use Case* Proses Ekstrak

Nama <i>use case</i>	Ekstrak	
Case terkait	Ekstrak <i>share</i> dan Rekonstruksi <i>share</i>	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses ekstrak, yang meliputi ekstrak <i>share</i> dan rekonstruksi <i>share</i>	
Main Flow	Aksi Aktor	Respon Sistem
	1. Klik <i>icon</i> ‘  ’,	2. Menampilkan kotak <i>dialog</i> untuk membuka <i>stego image</i> 3. Menampilkan <i>stego image</i> pada <i>picture box stego image</i>
<i>Error</i>	-	

Tabel 3. 5 Tabel Narasi *Use Case* Proses Ekstrak *Share*

Nama <i>use case</i>	Ekstrak Share
Aktor	<i>User</i>
Deskripsi	<i>Use case</i> ini menjelaskan proses ekstrak untuk mendapatkan <i>n</i> buah <i>shares</i>

	Aksi Aktor	Respon Sistem
<i>Main flow</i>	1. <i>Input</i> bit sisip 2. <i>Input password</i>	- 3. Melakukan pengecekan apakah <i>password</i> sama dengan <i>password</i> yang disisipkan ke dalam <i>stego image</i>
	4. Klik button ‘Ekstrak’ 6. Klik button ‘Lanjut’	5. Mengambil bit <i>share</i> dari <i>stego image</i> kemudian menampilkan <i>share</i> pada <i>picture box share image</i> 7. Menampilkan form Rekonstruksi <i>share</i>
<i>Error</i>		1. <i>Password</i> tidak sama sama dengan <i>password</i> yang disisipkan ke dalam <i>stego image</i>

Tabel 3. 6 Tabel Narasi Use Case Proses Rekonstruksi Share

Nama <i>use case</i>	Rekonstruksi Share	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini menjelaskan proses rekonstruksi <i>share</i> sehingga menghasilkan <i>secret image</i>	
	Aksi Aktor	Respon Sistem
<i>Main flow</i>	1. Klik button ‘Gabung’ 3. Klik button ‘Dekrip’ 5. Klik button ‘Simpan’	2. Melakukan proses penggabungan $(n - 1)$ <i>natural share image</i> dan hasilnya ditampilkan pada <i>picture box hasil gabung natural share image</i> 4. Melakukan proses rekonstruksi <i>share</i> yaitu dengan melakukan operasi XOR citra hasil gabungan dengan <i>share image</i> , hasilnya ditampilkan pada <i>picture box secret image</i> 6. Menyimpan <i>secret image</i> ke lokasi yang telah ditentukan

	7. Klik <i>button</i> ‘Selesai’	8. Mengakhiri proses ekstrak dan kembali ke <i>form</i> utama
<i>Error</i>	-	

Tabel 3. 7 Tabel Narasi Use Case Pengujian Parameter

Nama <i>use case</i>	Pengujian Parameter	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk melihat pengaruh parameter (bit sisip (<i>b</i>) dan kontras <i>cover</i>) terhadap hasil <i>stego</i>	
	Aksi Aktor	Respon Sistem
<i>Main Flow</i>	1. Klik icon ‘  ’,	2. Menampilkan kotak <i>dialog</i> untuk membuka <i>stego image</i> 3. Menampilkan <i>stego image</i> pada <i>picture box stego image</i>
	4. Klik icon ‘  ’,	5. Menampilkan kotak <i>dialog</i> untuk membuka <i>cover image</i> sebelum disisip 6. Menampilkan <i>cover image</i> pada <i>picture box cover image</i>
	7. Klik <i>button</i> ‘Histogram’	8. Menampilkan histogram, nilai MSE dan PSNR dari masing-masing gambar
<i>Error</i>	-	

Tabel 3. 8 Tabel Narasi Use Case Pengujian Kualitas Citra

Nama <i>use case</i>	Pengujian Kualitas Citra	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk melihat kualitas <i>secret image</i> hasil rekonstruksi setelah <i>stego image</i> diberi <i>noise</i>	
	Aksi Aktor	Respon Sistem
<i>Main Flow</i>	1. Klik icon ‘  ’,	2. Menampilkan kotak <i>dialog</i> untuk membuka <i>stego share image</i>

		3. Menampilkan <i>stego image</i> pada <i>picture box stego share image</i>
	4. Klik icon 	5. Menampilkan kotak <i>dialog</i> untuk membuka <i>stego natural share image</i> 6. Menampilkan <i>stego natural share image</i> pada <i>picture box stego natural share image</i>
	7. Klik button 'Hapus'	8. Menghapus <i>stego natural share image</i> dari <i>picture box</i>
	9. Pilih jenis <i>noise</i>	-
	10. <i>Input</i> nilai probabilitas <i>noise</i>	11. Melakukan pengecekan apakah probabilitas lebih besar atau sama dengan 0 dan lebih kecil atau sama dengan 0,1
	12. Klik button 'Beri Noise Stego Image'	13. Melakukan proses pemberian <i>noise</i> pada <i>stego share image</i> secara <i>random</i> sesuai dengan jenis dan probabilitas <i>noise</i> yang dipilih
	14. <i>Input</i> bit sisip (<i>b</i>)	-
	15. <i>Input</i> password	16. Melakukan pengecekan apakah <i>password</i> sama dengan <i>password</i> yang disisipkan ke dalam <i>stego image</i>
	17. Klik button 'Ekstrak'	18. Mengambil bit <i>share</i> dari <i>stego image</i> kemudian menampilkan <i>share</i> pada <i>picture box share image</i>
	19. Klik button 'Rekonstruksi'	20. Melakukan proses rekonstruksi sehingga menghasilkan <i>secret image</i>

		dan menampilkannya pada <i>picture box</i> ‘Citra Rahasia Noise’
	21. Klik icon 	22. Menampilkan kotak <i>dialog</i> untuk membuka <i>secret image</i> (citra asli) 23. Menampilkan <i>secret image</i> pada <i>picture box secret image</i>
	24. Klik button ‘Hasil Pengujian’	25. Menampilkan hasil pengujian <i>noise</i> berupa tabel yang berisi data ukuran citra, bit sisip, jenis <i>noise</i> , probabilitas, MSE dan PSNR
	26. Klik button ‘Simpan’	27. Menyimpan <i>secret image</i> hasil rekonstruksi setelah <i>stego image</i> diberi <i>noise</i>
Error		1. Nilai probabilitas <i>noise</i> lebih kecil dari 0 dan lebih besar dari 0,1 2. <i>Password</i> tidak sama sama dengan <i>password</i> yang disisipkan ke dalam <i>stego image</i>

3.1.2.2. Analisis Kebutuhan Non-Fungsional

Untuk merumuskan persyaratan non-fungsional dari perangkat lunak, maka dilakukan analisis terhadap kinerja, informasi, ekonomi, pengendalian, efisiensi, dan pelayanan. Panduan ini dikenal dengan analisis PIECES (*Performance, Information, Economic, Control, Efficiency, dan Services*).

1. *Performance*

Perangkat lunak dapat menyelesaikan proses *embedding* yaitu pembentukan *share* dan penyisipan bit *share* dalam waktu yang relatif singkat yaitu kurang dari 1 menit untuk $n=3$.

2. *Information*

- Perangkat lunak dapat memberikan informasi berupa ukuran citra dan ketentuan *password* yang valid.

- b. Perangkat lunak dapat memberikan informasi berupa *progressbar* ketika melakukan *embedding* dan ekstrak.
 - c. Perangkat lunak dapat memberikan informasi berupa status proses jika telah selesai.
3. *Economic*
- Perangkat lunak tidak memerlukan perangkat pendukung lainnya dalam proses eksekusi dengan adanya *.NET Framework* sehingga perangkat dapat dengan mudah digunakan.
4. *Control*
- Perangkat lunak melakukan validasi terhadap parameter *input-an*, seperti:
- a. Perangkat lunak melakukan validasi terhadap ukuran *secret image* dan *natural image* yang di-*input* berdasarkan ketentuan.
 - b. Perangkat lunak membatasi nilai n yaitu dengan menjadikan nilai minimum sebagai default dan maksimum nilai yang boleh adalah 10.
 - c. Perangkat lunak membatasi nilai jumlah bit sisip yaitu dengan menjadikan nilai minimum sebagai default dan maksimum nilai yang boleh adalah 4.
 - d. Perangkat lunak melakukan validasi terhadap *password* yang di-*input*.
 - e. Perangkat lunak melakukan validasi terhadap ukuran *cover* yang di-*input* berdasarkan ketentuan.
 - f. Perangkat lunak melakukan validasi terhadap jenis noise yang di-*input*.
 - g. Perangkat lunak melakukan validasi terhadap nilai probabilitas *noise* yang di-*input*.
5. *Efficiency*
- a. Sistem akan melakukan *scalling* otomatis untuk *input-an* citra jika ukuran citra tidak sesuai ketentuan.
 - b. *User* dapat menghapus semua *natural image* secara bersamaan.
 - c. *User* dapat menyimpan citra *output* dengan pemberian nama file secara otomatis.
6. *Services*

Perangkat lunak menyediakan fasilitas bantuan berupa panduan dan teori pendukung sebagai referensi pemahaman untuk menggunakan perangkat lunak.

3.2. Perancangan

Pada tahap ini dilakukan analisis perancangan perangkat lunak yaitu berupa perancangan tampilan. Adapun rancangan tampilan perangkat lunak adalah sebagai berikut:

3.2.1. Form Utama

Rancangan *form* ‘Utama’ dapat dilihat pada Gambar 3. 8 berikut ini. Rancangan ini merupakan *form* yang pertama kali muncul saat perangkat lunak dijalankan dan berfungsi untuk menyediakan *link* yang memanggil *form* yang ada pada perangkat lunak.



Gambar 3. 8 Rancangan *form* ‘Utama’

Keterangan:

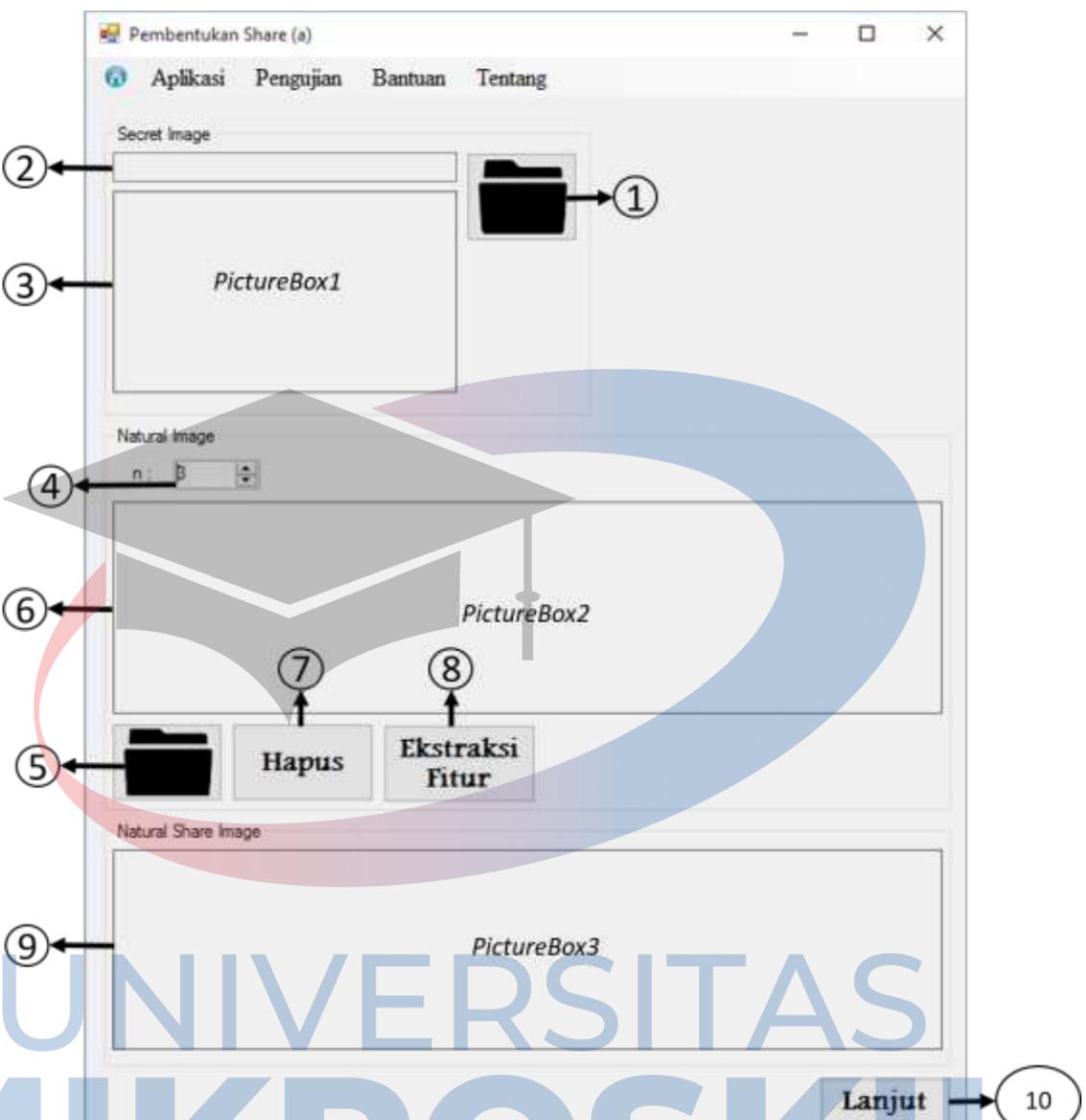
1. *Menu ‘Aplikasi’* untuk menampilkan *Submenu ‘Embedding’, ‘Ekstrak’ dan ‘Keluar’*. ‘Embedding’ untuk melakukan proses *embedding* citra, ‘Ekstrak’ untuk melakukan proses ekstrak citra dan ‘Keluar’ untuk keluar dari aplikasi.
2. *Menu ‘Pengujian’* untuk menampilkan *Submenu ‘Perbandingan Citra Stego’ dan ‘Pemberian Noise’*. ‘Perbandingan Citra Stego’ untuk melakukan pengujian terhadap *stego image* berdasarkan pengaruh nilai *b* dan kontras *cover*, sementara ‘Pemberian Noise’ untuk melakukan pengujian kualitas citra hasil rekonstruksi setelah *stego image* diberi *noise*.
3. *Menu ‘Bantuan’* untuk menampilkan *Submenu ‘Panduan Embedding’, ‘Panduan Ekstrak’ dan ‘Teori’*. ‘Panduan Embedding’ untuk memberikan panduan berupa penjelasan tentang proses *embedding*, ‘Panduan Ekstrak’ untuk memberikan panduan berupa penjelasan tentang proses ekstrak dan ‘Teori’ untuk menampilkan *Submenu ‘Kriptografi Visual’* dan ‘HSI’ dimana ‘Kriptografi Visual’ untuk menampilkan teori tentang kriptografi visual, sementara ‘HSI’ untuk menampilkan teori tentang model warna HIS.
4. *Menu ‘Tentang’* untuk menampilkan *profil* pembuat perangkat lunak.

3.2.2. *Submenu Embedding*

Pada *submenu ‘Embedding’* terdapat tiga *form*, yaitu *form* Pembentukan Share (a), *form* Pembentukan Share (b) dan *form* Penyisipan Bit Share.

3.2.2.1. *Form Pebentukan Share (a)*

Rancangan *form* ‘Pebentukan Share (a)’ dapat dilihat pada Gambar 3. 9 berikut ini. Rancangan ini merupakan *form* untuk melakukan *sub-proses embedding*, *user* harus meng-*input secret image*, nilai *n* dan *natural image* yaitu sebanyak $(n - 1)$. Setelah *user* menekan *button* Ekstraksi Fitur, maka akan dilakukan ekstraksi fitur $(n - 1)$ *natural image* sehingga menghasilkan $(n - 1)$ *natural share image*.



Gambar 3. 9 Rancangan form ‘Pembentukan Share (a)’

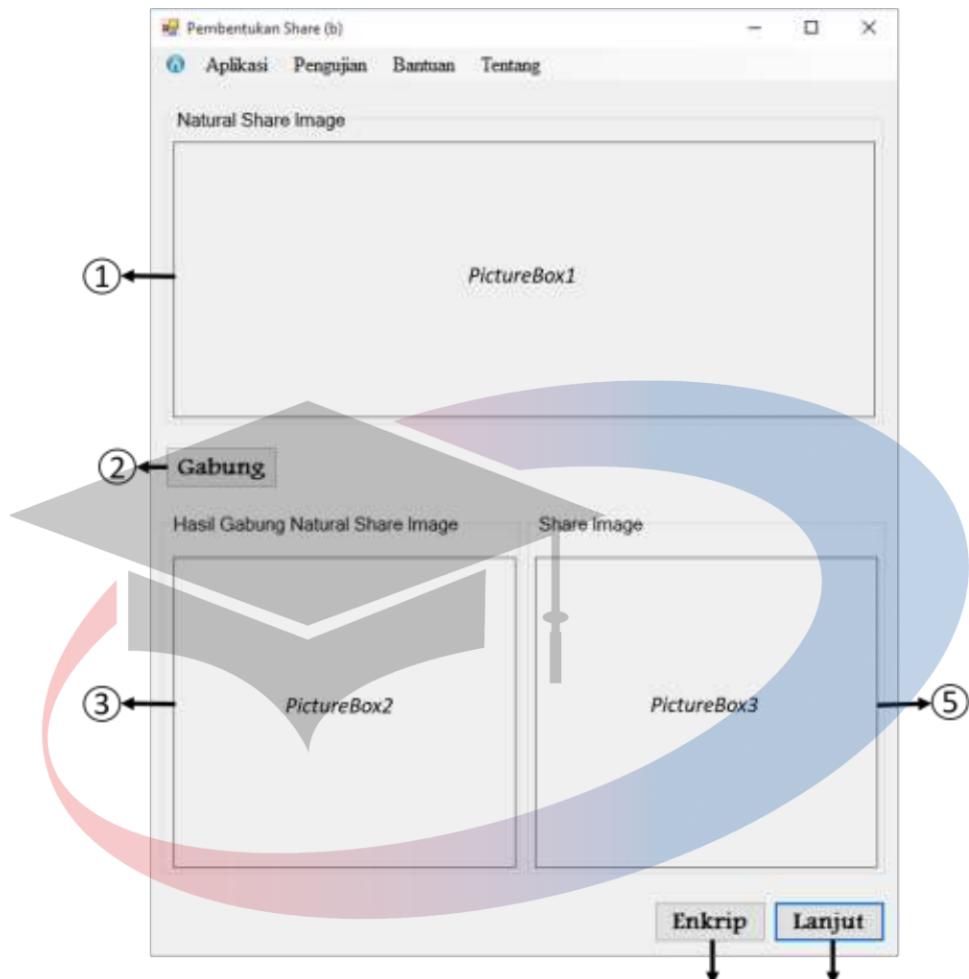
Keterangan:

1. *Icon* untuk memilih *secret image* dan menampilkannya di *pictureBox1*.
Jika ukuran tidak sesuai aturan maka akan dilakukan *scalling* untuk ukuran citra *input*.
2. *TextBox* untuk menampilkan *path* dari *secret image*.
3. *PictureBox1* untuk menampilkan *secret image*.

4. *NumericUpDown* untuk memasukkan nilai n , dimana $(n - 1)$ merupakan jumlah *natural image*. Nilai n yang dapat di-input minimal 3 dan maksimal 10.
5. *Icon*  untuk memilih *natural image* dan menampilkannya di *pictureBox2*. Jika ukuran tidak sesuai aturan maka akan dilakukan *scalling* untuk ukuran citra *input*.
6. *PictureBox2* untuk menampilkan *natural image*.
7. *Button* ‘Hapus’ untuk menghapus *natural image* pada *pictureBox2*.
8. *Button* ‘Ekstraksi Fitur’ untuk ekstraksi fitur $(n - 1)$ *natural image* sehingga menghasilkan $(n - 1)$ *natural share image* dan akan ditampilkan pada *pictureBox3*.
9. *PictureBox3* untuk menampilkan $(n - 1)$ *natural share image*.
10. *Button* ‘Lanjut’ untuk masuk ke *form* ‘Pembentukan Share (b)’.

3.2.2.2. Form Pembentukan Share (b)

Rancangan *form* ‘Pebentukan Share (b)’ dapat dilihat pada Gambar 3. 10 berikut ini. Rancangan ini merupakan *form* untuk melakukan *sub-proses embedding*. Pada *form* ini, ditampilkan $(n - 1)$ *natural share image* yang merupakan hasil pada proses sebelumnya (proses Pembentukan Share (a)). Setelah *user* menekan *button* ‘Gabung’, maka $(n - 1)$ *natural share image* akan digabungkan. Kemudian setelah *user* menekan *button* ‘Enkrip’, hasil gabungan akan dilakukan proses XOR dengan *secret image* sehingga menghasilkan *share image*.



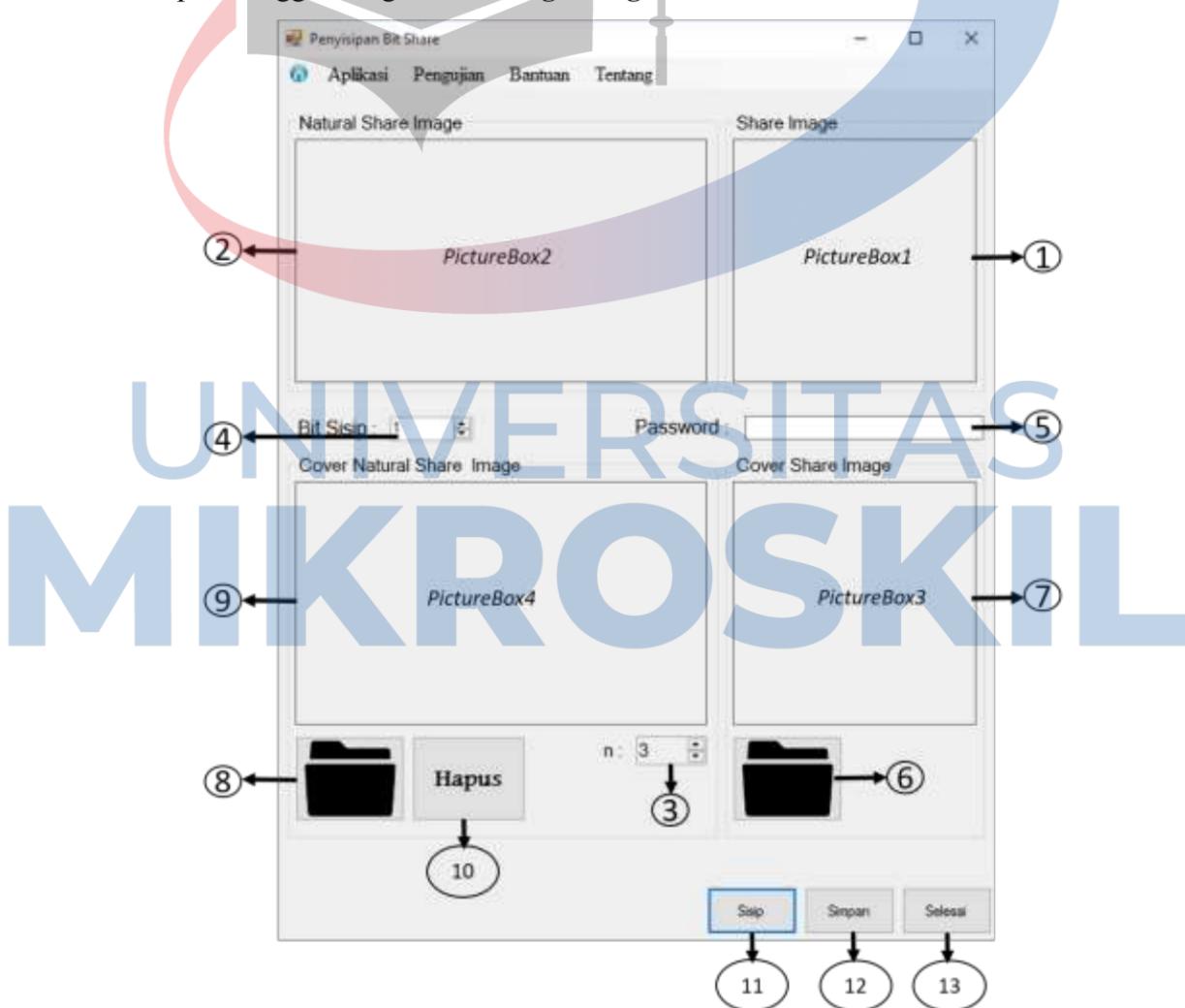
Gambar 3. 10 Rancangan form ‘Pembentukan Share (b)’

Keterangan:

1. *PictureBox1* untuk menampilkan $(n - 1)$ *natural share image*.
2. *Button ‘Gabung’* untuk melakukan proses penggabungan $(n - 1)$ *natural share image*.
3. *PictureBox2* untuk menampilkan gabungan $(n - 1)$ *natural share image*.
4. *Button ‘Enkrip’* untuk melakukan pembentukan *share*.
5. *PictureBox3* untuk menampilkan *share image*.
6. *Button ‘Lanjut’* untuk masuk ke *form ‘Penyisipan Bit Share’*.

3.2.2.3. Form Penyisipan Bit Share

Rancangan *form* ‘Penyisipan Bit Share’ dapat dilihat pada Gambar 3. 11 berikut ini. Rancangan ini merupakan *form* untuk melakukan *sub-proses embedding*. Pada *form* ini, ditampilkan *share image* dan *natural share image* yang merupakan hasil pada proses sebelumnya (proses Pembentukan Share (b)). Untuk melakukan penyisipan bit *share*, *user* harus meng-*input* jumlah bit sisip (b), *cover image* dan *password*. Jumlah bit merupakan jumlah bit nilai Intensitas citra HSI yang akan digantikan dengan bit *share* dan *password*. *Password* bertujuan untuk otentikasi, sehingga *stego image* hanya bisa direkonstruksi jika memiliki *password* yang valid. Proses penyisipan bit *share* akan dilakukan setelah *user* menekan *button* ‘Sisip’ sehingga menghasilkan *stego image*.



Gambar 3. 11 Rancangan *form* ‘Penyisipan Bit Share’

Keterangan:

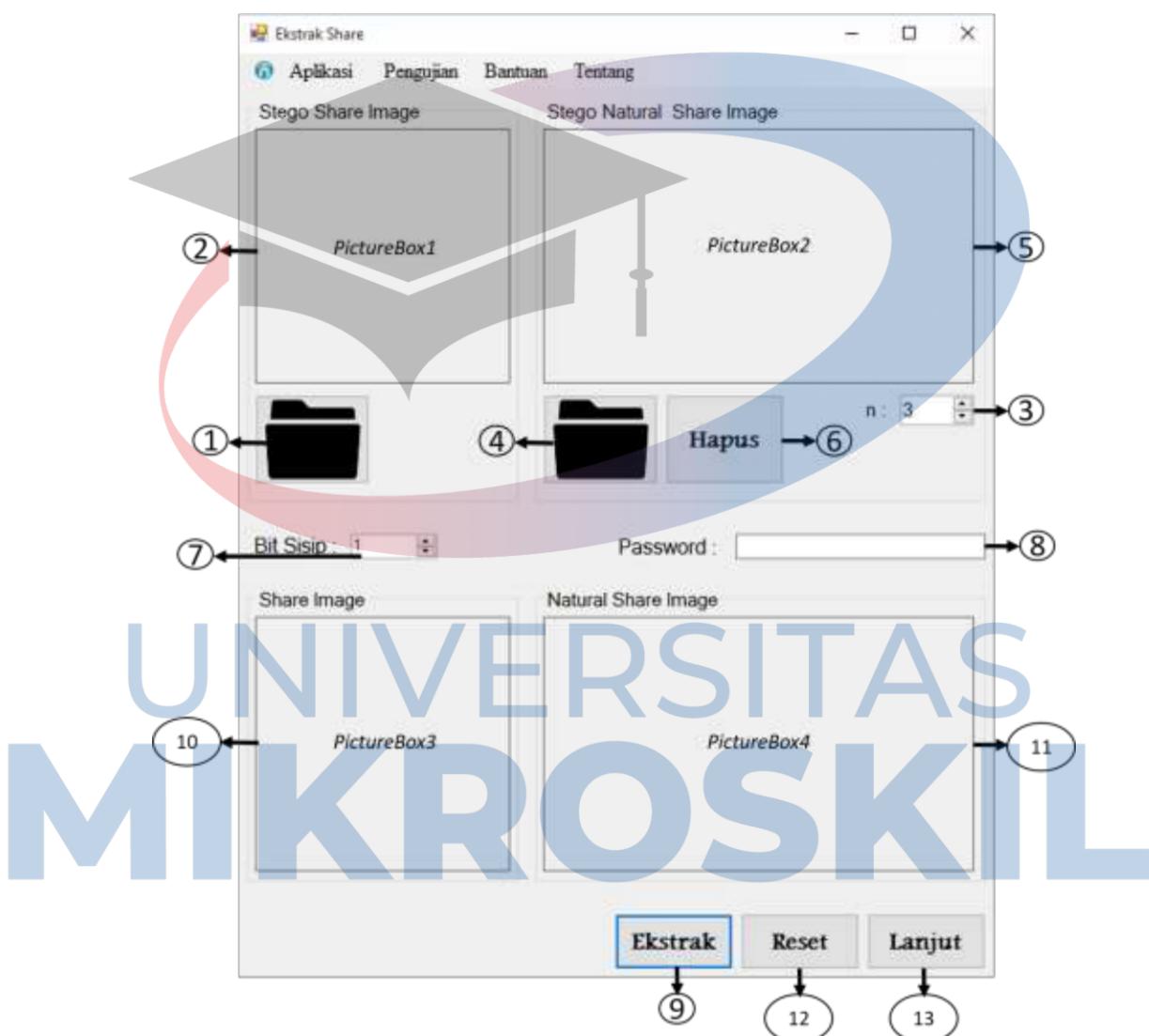
1. *PictureBox1* untuk menampilkan *share image*.
2. *PictureBox2* untuk menampilkan *natural share image*.
3. *NumericUpDown1* untuk menampilkan nilai n , dimana $n - 1$ merupakan jumlah *natural share image*.
4. *TextBox1* untuk meng-input *password*, dimana panjang *password* yaitu minimal 6 karakter dan maksimal 14 karakter yang terdiri dari kombinasi angka, huruf kapital dan huruf *non-kapital*.
5. *NumericUpDown* untuk meng-input jumlah bit sisip (b), dimana b merupakan jumlah bit intensitas yang akan digantikan dengan bit *share* atau *password*, dengan panjang minimal 1 dan maksimal 4.
6. *Icon*  untuk memilih *cover image* untuk *share image* dan menampilkannya di *pictureBox3*. Jika ukuran tidak sesuai aturan maka akan dilakukan *scaling* untuk ukuran citra *input*.
7. *PictureBox3* untuk menampilkan *cover image share image*.
8. *Icon*  untuk memilih *cover image* untuk *natural share image* dan menampilkannya di *pictureBox4*. Jika ukuran tidak sesuai aturan maka akan dilakukan *scaling* untuk ukuran citra *input*.
9. *PictureBox4* untuk menampilkan *cover image natural share image*.
10. *Button* ‘Hapus’ untuk menghapus gambar pada *pictureBox4*.
11. *Button* ‘Sisip’ untuk melakukan penyisipan *share* dan *password*.
12. *Button* ‘Simpan’ untuk menyimpan *stego image*.
13. *Button* ‘Selesai’ untuk mengakhiri proses *embedding* dan kembali ke *form* ‘Utama’.

3.2.3. Submenu Ekstrak

Pada submenu ‘Ekstrak’ terdapat dua *form*, yaitu *form* ‘Ekstrak Share’ dan *form* ‘Rekonstruksi Share’.

3.2.3.1. Form Ekstrak Share

Rancangan *form* ‘Ekstrak Share’ dapat dilihat pada Gambar 3. 12 berikut ini. Rancangan ini merupakan *form* untuk melakukan *sub-proses* ekstrak. Untuk melakukan ekstrak, *user* harus meng-*input stego image*, bit sisip dan *password* yang valid. Setelah *user* menekan *button* ‘Ekstrak’, maka akan dilakukan proses ekstrak sehingga menghasilkan *share image* dan *natural share image*.



Gambar 3. 12 Rancangan *form* ‘Ekstrak Share’

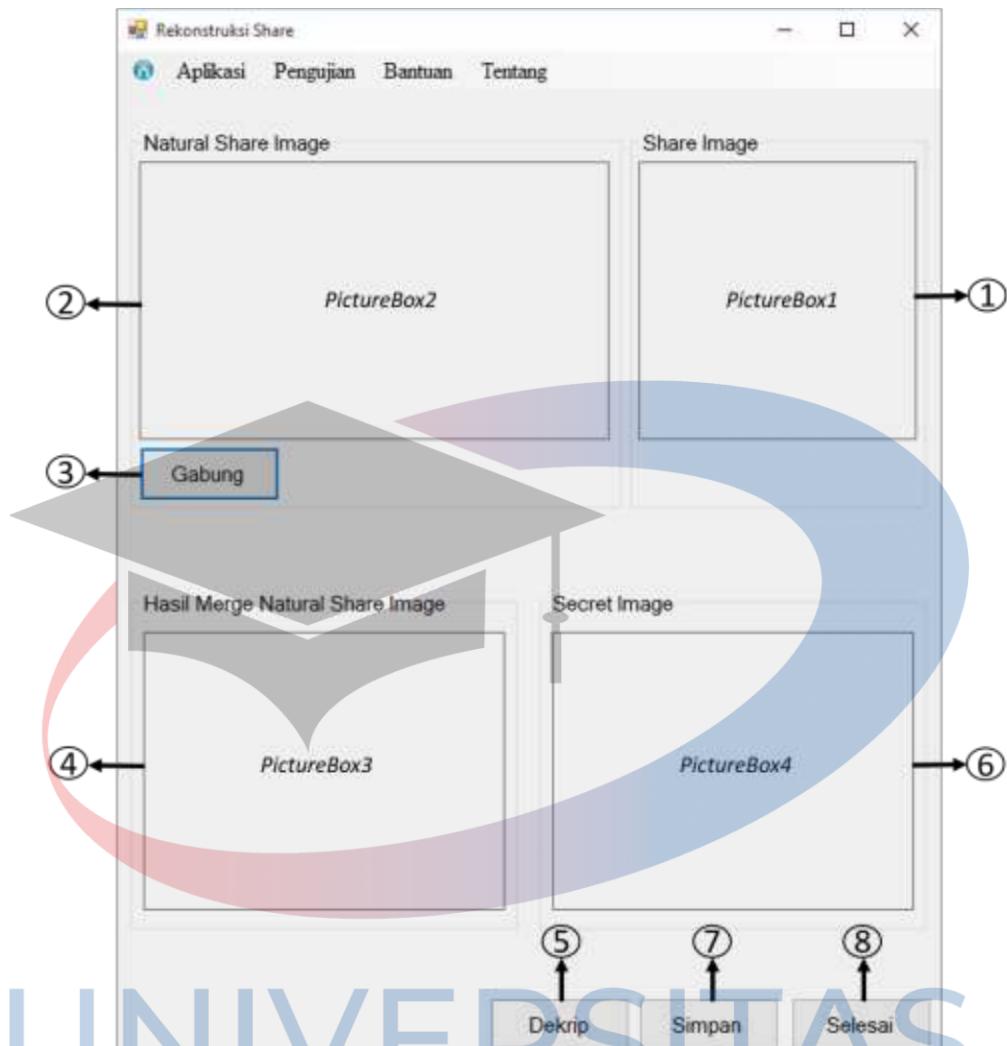
Keterangan:

1. *Icon* untuk memilih *stego share image* dan menampilkannya di *pictureBox1*.

2. *PictureBox1* untuk menampilkan *stego share image*.
3. *NumericUpDown* untuk meng-*input* nilai n , dimana n merupakan jumlah *share* atau *stego image* yang dihasilkan pada saat proses *embedding*.
4. *Icon*  untuk memilih *stego natural share image* dan menampilkannya di *pictureBox2*.
5. *PictureBox2* untuk menampilkan *stego natural share image*.
6. *Button* ‘Hapus’ untuk menghapus gambar pada *pictureBox2*.
7. *NumericUpDown* untuk meng-*input* jumlah bit sisip (b), bit sisip harus sama dengan bit sisip pada saat proses *embedding*.
8. *TextBox1* untuk meng-*input* *password*, dimana *password* harus sama dengan *password* yang disisipkan pada *stego image*.
9. *Button* ‘Ekstrak’ untuk melakukan ekstrak *share image* dan *natural share image* dari *stego image*.
10. *PictureBox3* untuk menampilkan *share image*.
11. *PictureBox4* untuk menampilkan *natural share image*.
12. *Button* ‘Reset’ untuk menghapus semua parameter dan gambar pada semua *pictureBox*.
13. *Button* ‘Lanjut’ untuk masuk ke *form* ‘Rekonstruksi Share’.

3.2.3.2. Form Rekonstruksi Share

Rancangan *form* ‘Rekonstruksi Share’ dapat dilihat pada Gambar 3. 13 berikut ini. Rancangan ini merupakan *form* untuk melakukan *sub-proses* ekstrak. Pada *form* ini, ditampilkan $(n - 1)$ *natural share image* yang merupakan hasil proses sebelumnya (proses Ekstrak Share). Setelah *user* menekan *button* ‘Gabung’, maka $(n - 1)$ *natural share image* akan digabungkan. Kemudian setelah *user* menekan *button* ‘Dekrip’, citra hasil gabungan akan dilakukan operasi XOR dengan *share image* sehingga menghasilkan *secret image*.



Gambar 3. 13 Rancangan form ‘Rekonstruksi Share’

Keterangan:

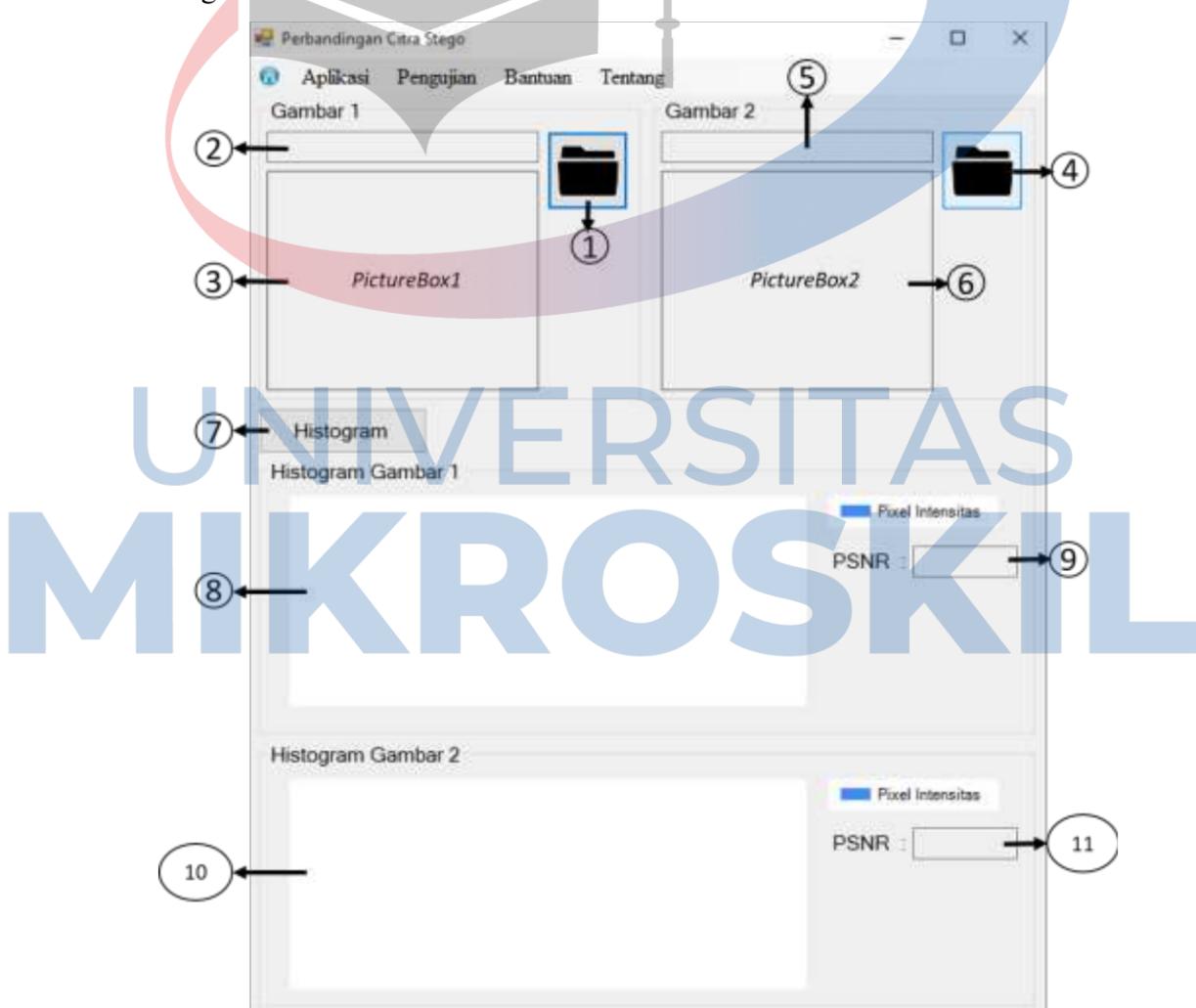
1. *PictureBox1* untuk menampilkan *share image*.
2. *PictureBox2* untuk menampilkan *natural share image*.
3. *Button ‘Gabung’* untuk menggabungkan $(n - 1)$ *natural share image*.
4. *PictureBox3* untuk menampilkan gabungan $(n - 1)$ *natural share image*.
5. *Button ‘Dekrip’* untuk melakukan rekonstruksi *share*.
6. *PictureBox4* untuk menampilkan *secret image*.
7. *Button ‘Simpan’* untuk menyimpan *secret image*.
8. *Button ‘Selesai’* untuk mengakhiri proses ekstrak dan kembali ke *form Utama*.

3.2.4. Pengujian

Pada menu ‘Pengujian’ terdapat dua *form*, yaitu *form* ‘Perbandingan Citra Stego’ dan *form* ‘Pemberian Noise’.

3.2.4.1. Form Perbandingan Citra Stego

Rancangan *form* ‘Perbandingan Citra Stego’ dapat dilihat pada Gambar 3. 14 berikut ini. Pada *form* ini, akan dilakukan perbandingan *stego image* untuk melihat pengaruh jumlah bit sisip (b) terhadap nilai PSNR. Untuk melihat perbandingan, *user* harus meng-input *stego image*. Histogram dan nilai PSNR dari masing-masing *stego image* akan ditampilkan setelah *user* menekan *button* ‘Histogram’.



Gambar 3. 14 Rancangan *form* ‘Perbandingan Citra Stego’

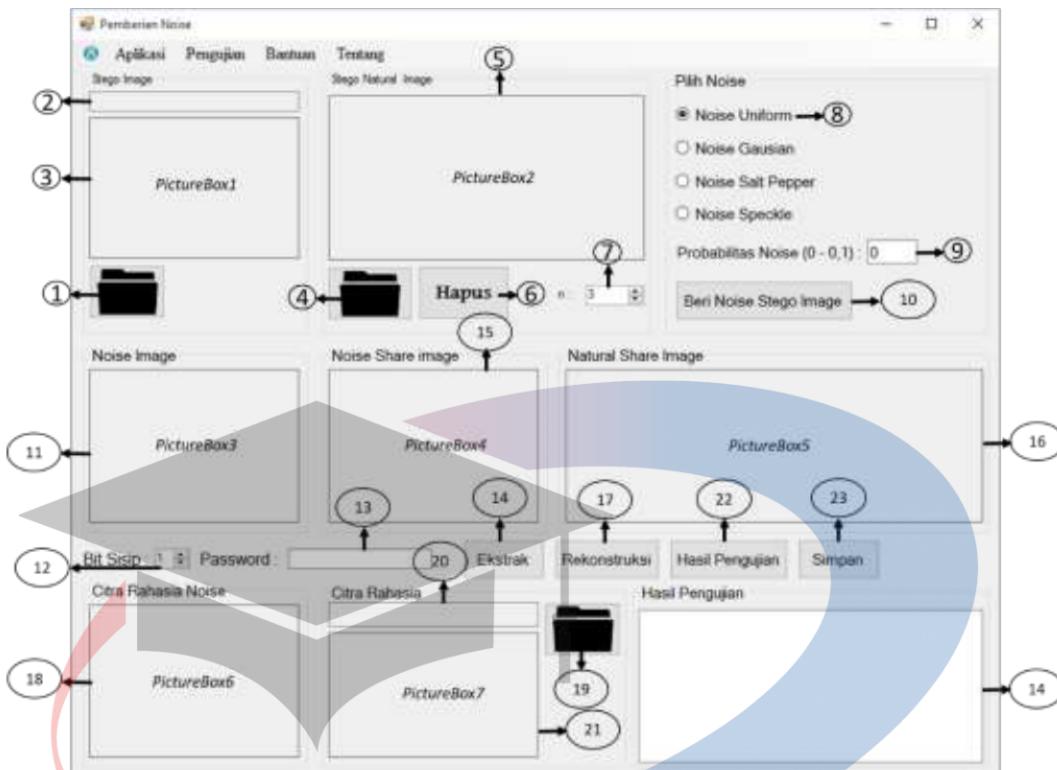
Keterangan:

1. *Icon* ‘’ untuk memilih *stego image* dan menampilkannya di *pictureBox1*.
2. *TextBox1* untuk menampilkan *path* dari *stego image*.
3. *PictureBox1* untuk menampilkan *stego image*.
4. *Icon* ‘’ untuk memilih *cover image* dan menampilkannya di *pictureBox2*.
5. *TextBox2* untuk menampilkan *path* dari *cover image*.
6. *PictureBox2* untuk menampilkan *cover image*.
7. *Button* ‘Histogram’ untuk menampilkan histogram dan nilai PSNR dari *stego image* dan *cover image*.
8. Histogram dari *stego image*.
9. *Label1* untuk menampilkan nilai PSNR dari *stego image*.
10. Histogram dari *cover image*.
11. *Label2* untuk menampilkan nilai PSNR dari *cover image*.

3.2.4.2. Form Pemberian Noise

Rancangan *form* ‘Pemberian Noise’ dapat dilihat pada Gambar 3. 15 berikut ini. Untuk melakukan pemberian *noise*, *user* harus meng *stego image* yang akan diberi *noise*, *stego natural share image*, jenis *noise* yang akan ditebar dan probabilitas dari *noise*. Setelah *user* menekan *button* ‘Beri Noise Stego Share’, proses pemberian *noise* akan dilakukan sehingga menghasilkan *stego image* yang diberi *noise*. Kemudian input bit sisip dan password untuk melakukan ekstrak share.

Tekan *button* ‘Ekstrak’ sehingga menghasilkan *n* buah *share*. Setelah *user* menekan *button* ‘Rekonstruksi’ maka akan dihasilkan *secret image* setelah pemberian *noise*. *User* dapat melihat tabel pengujian dengan menekan *button* ‘Hasil Pengujian’ dan dapat menyimpannya dengan menekan *button* ‘Simpan’. Disamping itu, *user* juga dapat melihat kembali *secret image* sebelum proses *embedding* dan pemberian *noise*, hal ini bertujuan untuk membandingkan secara visual data asli dengan *secret image* setelah pemberian *noise*.



Gambar 3. 15 Rancangan form ‘Pemberian Noise’

Keterangan:

1. Icon untuk memilih *stego image* yang berisi bit *share image* dan menampilkannya di *pictureBox1*.
2. *TextBox1* untuk menampilkan *path* dari *stego image*.
3. *PictureBox1* untuk menampilkan *stego image*.
4. Icon untuk memilih *stego natural share image* dan menampilkannya di *pictureBox2*.
5. *PictureBox1* untuk menampilkan *stego image*.
6. *Button* ‘Hapus’ untuk menghapus gambar pada *pictureBox2*.
7. *NumericUpDown* untuk meng-*input* nilai *n*, dimana *n* merupakan jumlah *share* atau *stego image* yang dihasilkan pada saat proses *embedding*.
8. *RadioButton* untuk memilih jenis *noise*.
9. *TextBox2* untuk *input* nilai probabilitas *noise*, dimana probabilitas *noise* yang dapat di-*input* minimal 0, 01 dan maksimal 0, 1 (10%).
10. *Button* ‘Beri Noise Stego Image’ untuk melakukan proses pemberian *noise*.

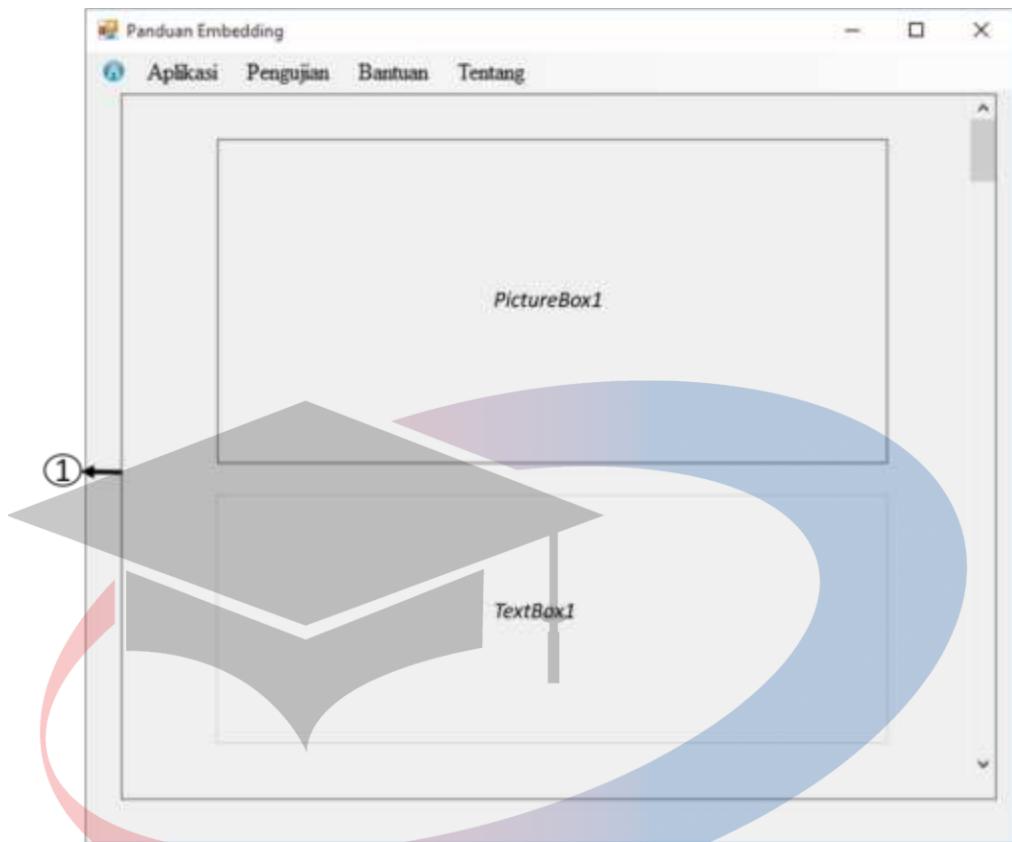
11. *PictureBox3* untuk menampilkan *stego image* setelah diberi *noise*.
12. *NumericUpDown* untuk meng-*input* jumlah bit sisip (*b*), bit sisip harus sama dengan bit sisip pada saat proses *embedding*.
13. *TextBox3* untuk meng-*input password*, dimana *password* harus sama dengan *password* yang disisipkan pada *stego image*.
14. *Button* ‘Ekstrak’ untuk melakukan proses ekstrak.
15. *PictureBox4* untuk menampilkan *share image* hasil ekstrak.
16. *PictureBox5* untuk menampilkan *natural share image* hasil ekstrak.
17. *Button* ‘Rekonstruksi’ untuk melakukan proses rekonstruksi sehingga menghasilkan *secret image*.
18. *PictureBox6* untuk menampilkan *secret image* hasil rekonstruksi.
19. *Icon*  untuk memilih *secret image* dan menampilkannya di *pictureBox7*.
20. *TextBox4* untuk menampilkan *path* dari *secret image*.
21. *PictureBox7* untuk menampilkan *secret image*.
22. *Button* ‘Hasil Pengujian’ untuk menampilkan hasil pengujian.
23. *Button* ‘Simpan’ untuk menyimpan *secret image* setelah pemberian *noise*.

3.2.5. Menu Bantuan

Pada menu ‘Bantuan’ terdapat *form* ‘Panduan Embedding’, *form* ‘Panduan Ekstrak’ dan submenu ‘Teori’.

3.2.5.1. Form Panduan Embedding

Rancangan *form* ‘Panduan Embedding’ dapat dilihat pada Gambar 3. 16 berikut ini. Rancangan ini merupakan *form* yang berisi penjelasan proses *embedding*.



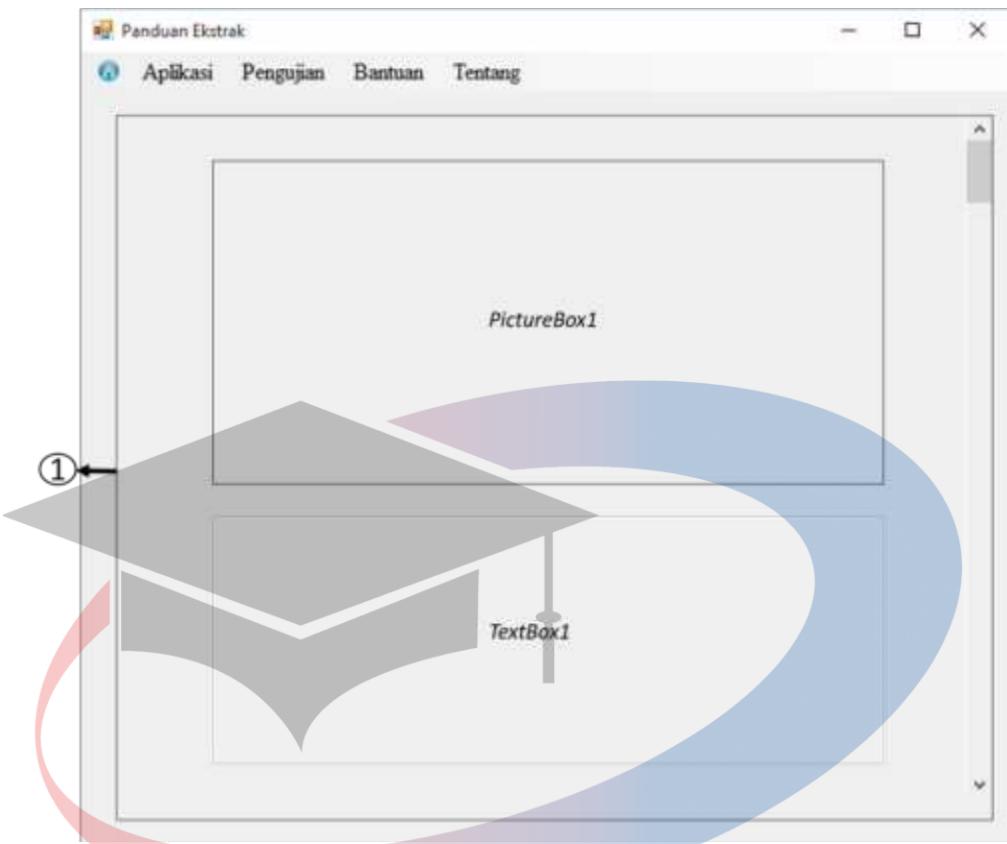
Gambar 3. 16 Rancangan *form* ‘Panduan Embedding’

Keterangan:

1. *Panel1* untuk menampilkan *screenshot* tampilan dan keterangan proses *embedding*.

3.2.5.2. Form Panduan Ekstrak

Rancangan *form* ‘Panduan Ekstrak’ dapat dilihat pada Gambar 3. 17 berikut ini. Rancangan ini merupakan *form* yang berisi penjelasan proses ekstrak.



Gambar 3. 17 Rancangan *form* ‘Panduan Ekstrak’

Keterangan:

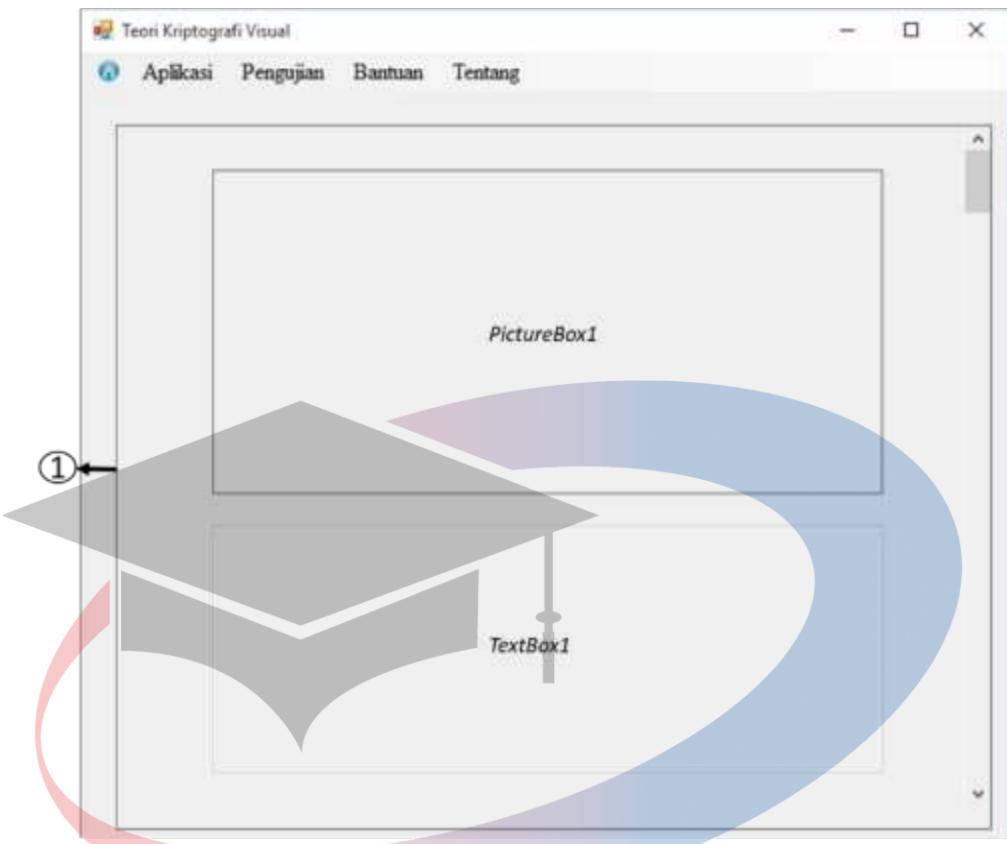
1. *Panel1* untuk menampilkan *screenshot* tampilan dan keterangan proses ekstrak.

3.2.5.3. Submenu Teori

Pada submenu ‘Teori’ terdapat dua *form* yaitu, *form* ‘Teori Kriptografi Visual’ dan *form* ‘Teori HSI’.

1. Form Teori Kriptografi Visual

Rancangan *form* ‘Teori Kriptografi Visual’ dapat dilihat pada Gambar 3. 18 berikut ini. Rancangan ini merupakan *form* yang berisi penjelasan teori kriptografi visual.



Gambar 3. 18 Rancangan *form* ‘Teori Kriptografi Visual’

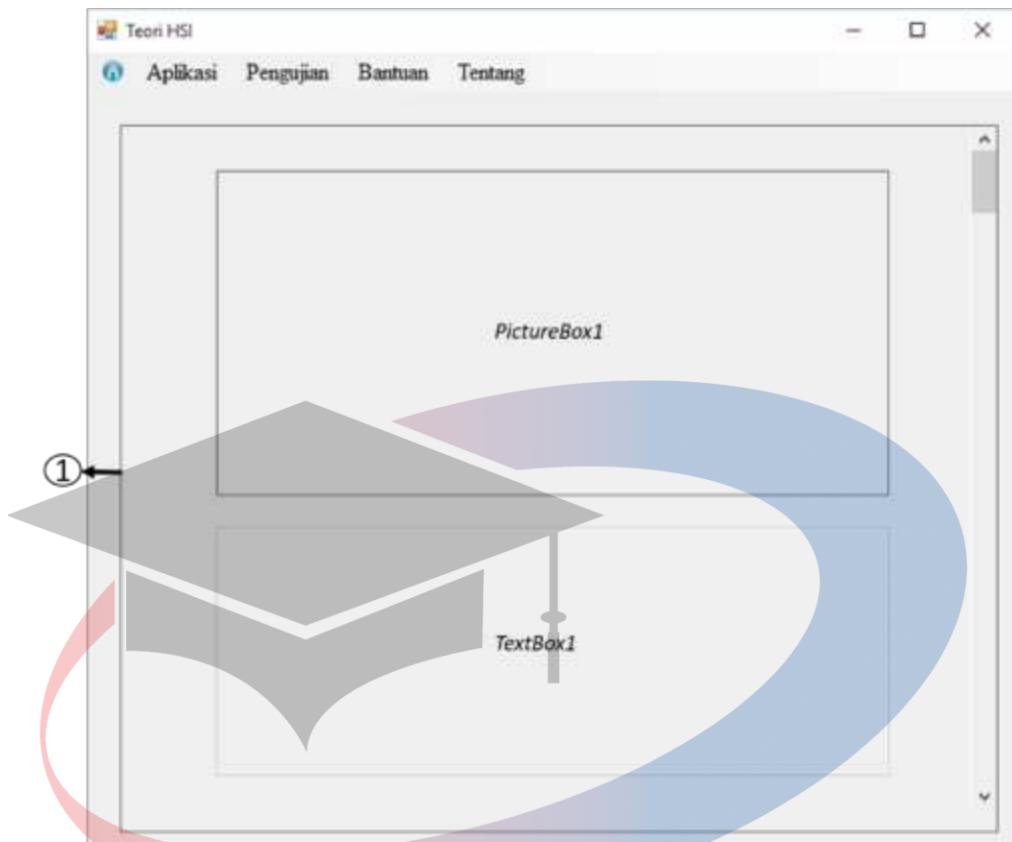
Keterangan:

1. *Panell1* untuk menampilkan gambar dan keterangan terkait teori kriptografi visual.

2. Form Teori HSI

Rancangan *form* ‘Teori HSI’ dapat dilihat pada Gambar 3. 19 berikut ini.

Rancangan ini merupakan *form* yang berisi penjelasan teori model warna HSI.



Gambar 3. 19 Rancangan *form* ‘Teori HSI’

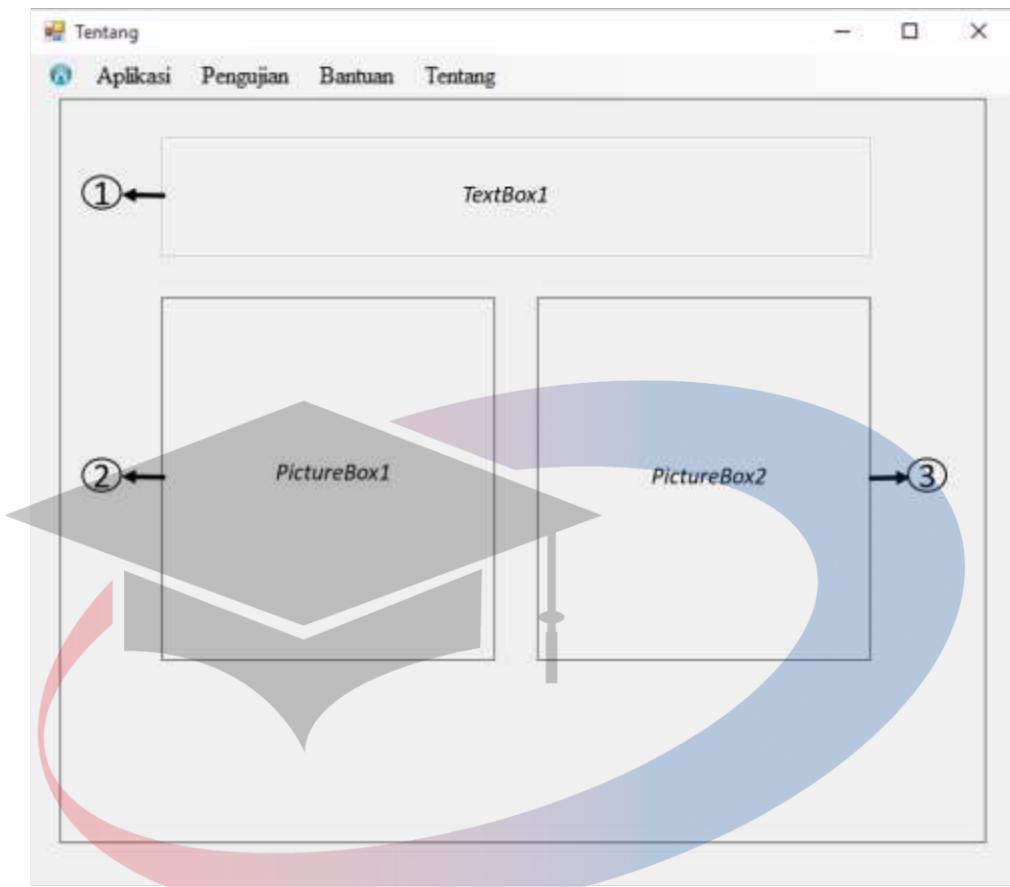
Keterangan:

1. *Panel1* untuk menampilkan gambar dan keterangan terkait teori model warna HSI.

3.2.6. Form Tentang

Rancangan *form* ‘Tentang’ dapat dilihat pada Gambar 3. 20 berikut ini.

Rancangan ini merupakan *form* yang berisi *profil* dari pembuat perangkat lunak.



Gambar 3. 20 Rancangan form ‘Tentang’

Keterangan:

1. *TextBox1* untuk menampilkan judul aplikasi.
2. *PictureBox1* untuk menampilkan foto dari pembuat aplikasi 1.
3. *PictureBox2* untuk menampilkan foto dari pembuat aplikasi 2.

UNIVERSITAS
MIKROSKIL