

## BAB III

### METODOLOGI PENELITIAN

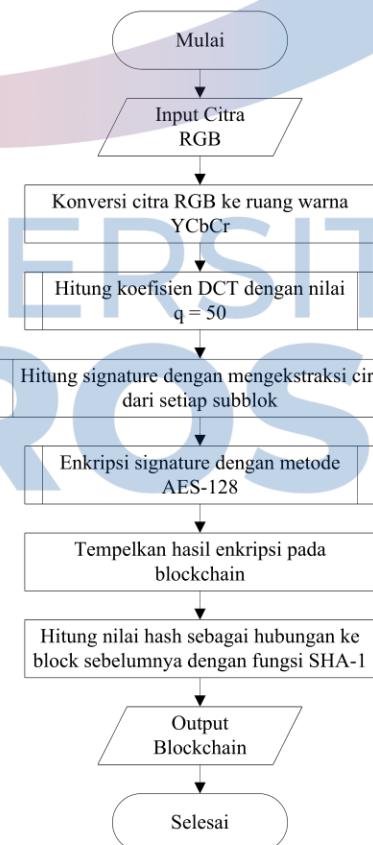
#### 3.1. Analisis

Analisis sistem adalah pembelajaran sebuah sistem dan komponen-komponennya sebagai prasyarat *system design* / desain sistem dan spesifikasi sebuah sistem yang baru. Berpindah dari definisi klasik analisis sistem ini ke suatu yang lebih kontemporer, analisis sistem adalah sebuah istilah yang secara kolektif mendeskripsikan fase-fase awal pengembangan sistem. Berdasarkan definisi dari analisis sistem tersebut, maka tahapan analisis dapat dibagi menjadi beberapa tahapan, yaitu analisis proses dan analisis kebutuhan.

##### 3.1.1 Analisis Proses

Proses kerja dari aplikasi implementasi *blockchain* untuk mendeteksi plagiarisme gambar pada aplikasi media sosial dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 3.1 berikut ini:

###### 1. Proses pembuatan *blockchain*



Gambar 3.1 *Flowchart* Proses Pembuatan *Blockchain*

Berdasarkan *flowchart* pada gambar 3.1, maka langkah kerja dari proses pembuatan *blockchain* adalah sebagai berikut:

a. Input citra RGB.

Misalkan dipilih sebuah citra berwarna RGB berukuran 16 x 16 piksel dengan perincian warna piksel seperti yang terlihat pada Tabel 3.1 sebagai berikut :

Tabel 3.1 Elemen Warna RGB dari Citra Input

105	94	96	128	119	248	76	255	181	114	54	95	71	148	85	75
177	167	168	201	191	112	63	250	64	86	166	245	157	58	224	214
65	55	56	89	79	247	240	215	106	153	166	162	71	63	244	148
249	72	97	140	190	129	172	123	136	212	84	193	112	196	146	120
180	74	184	232	219	181	127	117	141	67	205	156	209	174	66	193
136	107	89	230	127	89	50	221	207	60	101	161	68	229	172	227
240	249	169	134	179	65	130	94	131	232	253	120	151	107	240	226
135	201	86	67	72	156	93	68	74	76	80	116	146	66	168	51
115	198	86	119	77	161	199	156	154	197	199	238	92	85	229	233
130	228	126	85	148	150	130	156	138	55	229	97	123	118	179	64
193	220	247	103	77	97	101	134	241	87	199	109	114	187	153	146
109	179	206	68	87	251	68	196	70	212	237	55	127	145	164	118
159	88	240	221	210	176	194	189	169	132	83	124	242	213	155	255
209	156	252	217	154	171	224	231	179	61	86	212	71	73	167	210
172	174	118	254	206	111	74	238	164	86	213	68	84	182	156	104
189	58	76	163	158	186	194	123	89	255	164	106	141	69	253	180
138	160	136	238	64	215	223	209	219	121	110	226	57	77	237	233
76	203	214	107	63	143	230	191	123	247	60	145	253	155	73	90
202	158	122	135	172	84	201	119	221	150	111	97	147	187	251	224
97	108	214	131	96	167	59	169	125	171	85	227	232	153	130	244
99	63	218	135	94	98	154	247	120	89	78	208	168	58	86	247
72	66	93	156	171	176	170	215	229	105	218	176	236	249	155	252
255	78	199	131	94	226	135	131	191	73	160	114	194	136	57	177
54	79	72	134	244	210	111	126	232	93	152	249	140	242	108	114
220	227	208	134	160	51	146	187	137	202	135	141	135	239	154	56
111	140	218	119	57	177	234	123	56	65	152	148	118	69	191	176
110	254	232	86	173	73	188	254	192	140	186	214	175	118	100	141
181	234	98	99	99	219	76	230	198	116	247	74	221	148	237	246
141	81	134	242	113	81	166	91	74	107	184	77	132	78	105	53
229	195	66	253	61	170	168	118	168	242	210	147	55	82	165	198
62	71	242	106	176	128	189	186	70	144	241	50	65	200	161	249
135	185	80	89	124	185	161	239	182	224	246	228	92	201	126	129
143	183	95	63	211	139	136	57	244	54	197	192	116	208	166	168
153	167	111	150	251	237	91	124	240	159	170	69	151	228	90	211
189	186	51	165	165	71	190	120	219	166	160	158	214	174	141	180
172	227	99	159	57	132	236	181	185	199	240	141	160	71	229	120
215	192	237	109	251	182	253	96	201	64	189	113	246	184	132	209
99	128	88	214	227	233	241	154	198	111	244	132	152	112	181	170
76	248	231	234	75	181	70	51	226	65	173	209	246	95	185	200

150	111	253	202	131	179	134	250	53	237	66	98	74	163	251	192
209	197	76	93	249	130	121	96	202	168	244	57	176	161	138	126
226	141	188	198	210	80	136	253	71	237	157	73	62	254	50	240
68	219	214	102	222	76	214	199	154	96	225	75	203	59	55	157
118	237	134	78	69	254	79	155	62	145	52	146	98	60	120	152
54	143	87	127	161	224	173	87	177	251	231	108	241	222	212	251
210	53	203	195	157	216	248	59	239	240	60	201	207	127	130	53
180	247	144	97	72	53	69	144	123	223	121	240	71	118	164	225
195	159	156	134	178	172	129	213	75	180	100	74	169	131	226	211

b. Konversi citra RGB ke ruang warna YCbCr.

Proses konversi warna piksel citra dari ruang warna RGB pada tabel 3.1 ke ruang warna YCbCr menggunakan rumusan (1) pada bab 2. Karena nilai yang digunakan pada perhitungan selanjutnya hanya nilai elemen Y saja, maka hanya akan dihitung nilai elemen Y saja, seperti terlihat pada rincian perhitungan berikut:

Piksel (0, 0):

$$Y = 0.2990 * 105 + 0.5870 * 177 + 0.1140 * 65$$

$$Y = 48$$

Piksel (0, 1):

$$Y = 0.2990 * 94 + 0.5870 * 167 + 0.1140 * 55$$

$$Y = 39$$

Piksel (0, 2):

$$Y = 0.2990 * 96 + 0.5870 * 168 + 0.1140 * 56$$

$$Y = 40$$

Proses perhitungan akan dilakukan untuk setiap piksel. Hasil yang diperoleh dapat dilihat pada tabel 3.2.

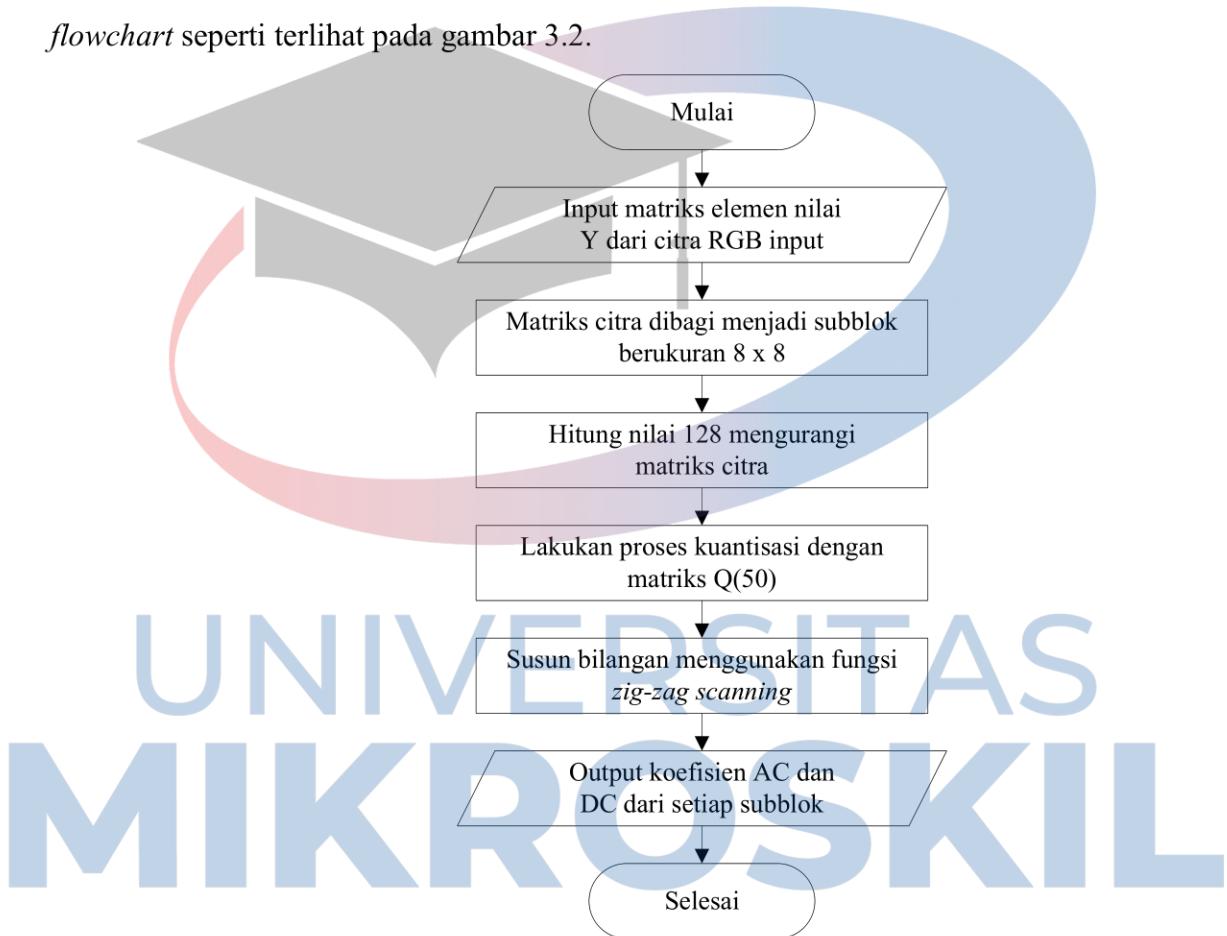
Tabel 3.2 Elemen Nilai Y dari Citra Input

48	39	40	68	60	38	50	121	52	55	61	66	70	61	64	73
49	82	79	101	113	106	27	62	63	59	55	90	109	85	69	72
58	63	77	69	124	107	74	125	62	59	68	113	144	104	66	73
80	97	74	54	59	71	91	66	63	58	71	122	154	106	70	69
18	34	33	46	64	61	32	37	67	61	68	104	126	88	68	70
149	108	80	106	116	61	73	92	79	65	60	70	77	68	58	75
211	233	159	88	107	158	161	109	85	71	64	59	55	61	65	83
212	104	40	44	71	136	113	66	87	79	69	68	65	76	78	94
124	85	161	160	135	76	138	113	73	89	203	100	165	123	79	67
147	165	103	126	122	136	184	155	66	105	187	23	145	178	90	90

263	162	199	118	192	150	128	124	134	189	47	65	163	88	89	216
280	176	177	192	211	213	153	85	111	45	56	78	80	99	67	146
177	112	94	191	179	114	159	176	109	210	103	154	95	76	208	167
176	100	91	115	204	211	234	170	89	89	156	130	100	56	244	93
139	61	118	109	98	91	137	154	94	221	174	56	201	187	137	218
110	180	162	85	147	137	145	161	65	190	210	84	211	64	180	86

c. Hitung koefisien DCT dengan nilai  $q = 50$ .

Proses kerja dari perhitungan koefisien DCT dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 3.2.



Gambar 3.2 *Flowchart* dari Proses Perhitungan Koefisien DCT

Elemen nilai Y dari citra input berukuran  $16 \times 16$  seperti terlihat pada tabel 3.2 akan dibagi menjadi 4 buah subblok berukuran  $8 \times 8$ , seperti terlihat pada tabel berikut:

Blok 1

48	39	40	68	60	38	50	121
49	82	79	101	113	106	27	62
58	63	77	69	124	107	74	125
80	97	74	54	59	71	91	66
18	34	33	46	64	61	32	37
149	108	80	106	116	61	73	92

211	233	159	88	107	158	161	109
212	104	40	44	71	136	113	66

Blok 2

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Blok 3

124	85	161	160	135	76	138	113
147	165	103	126	122	136	184	155
263	162	199	118	192	150	128	124
280	176	177	192	211	213	153	85
177	112	94	191	179	114	159	176
176	100	91	115	204	211	234	170
139	61	118	109	98	91	137	154
110	180	162	85	147	137	145	161

Blok 4

73	89	203	100	165	123	79	67
66	105	187	23	145	178	90	90
134	189	47	65	163	88	89	216
111	45	56	78	80	99	67	146
109	210	103	154	95	76	208	167
89	89	156	130	100	56	244	93
94	221	174	56	201	187	137	218
65	190	210	84	211	64	180	86

Setelah itu, nilai 128 mengurangi data matriks citra, sehingga diperoleh:

Blok 1

80	89	88	60	68	90	78	7
79	46	49	27	15	22	101	66
70	65	51	59	4	21	54	3
48	31	54	74	69	57	37	62
110	94	95	82	64	67	96	91
-21	20	48	22	12	67	55	36
-83	-105	-31	40	21	-30	-33	19
-84	24	88	84	57	-8	15	62

Blok 2

76	73	67	62	58	67	64	55
65	69	73	38	19	43	59	56
66	69	60	15	-16	24	62	55
65	70	57	6	-26	22	58	59
61	67	60	24	2	40	60	58
49	63	68	58	51	60	70	53
43	57	64	69	73	67	63	45
41	49	59	60	63	52	50	34

Blok 3

4	43	-33	-32	-7	52	-10	15
-19	-37	25	2	6	-8	-56	-27
-135	-34	-71	10	-64	-22	0	4
-152	-48	-49	-64	-83	-85	-25	43
-49	16	34	-63	-51	14	-31	-48
-48	28	37	13	-76	-83	-106	-42
-11	67	10	19	30	37	-9	-26
18	-52	-34	43	-19	-9	-17	-33

Blok 4

55	39	-75	28	-37	5	49	61
62	23	-59	105	-17	-50	38	38
-6	-61	81	63	-35	40	39	-88
17	83	72	50	48	29	61	-18
19	-82	25	-26	33	52	-80	-39
39	39	-28	-2	28	72	-116	35
34	-93	-46	72	-73	-59	-9	-90
63	-62	-82	44	-83	64	-52	42

Kemudian, dilakukan perkalian matriks T dengan data matriks citra, hasil perkalian dengan matriks T kemudian dikalikan lagi dengan matriks  $T^t$ , sehingga menghasilkan matriks seperti tabel berikut:

Blok 1

337.3389	-25.8408	-38.9151	-57.4226	-11.503	35.32662	-4.44511	7.586736
144.4996	91.88461	89.45234	90.6304	-44.7384	35.5756	-14.6629	6.097781
-78.9428	-20.9337	-52.9283	-36.9063	-8.08287	-12.0341	-11.8445	15.05902
37.37209	-14.948	13.02361	51.74496	-33.9559	-0.42611	19.13175	0.307841
145.0382	32.06024	-29.6005	-12.9175	-3.75099	-32.2951	1.708737	5.784931
-109.84	-15.024	-68.3647	27.74453	41.17039	-4.6076	19.15903	-15.161
36.76266	40.31021	-17.8433	0.419904	-72.4224	-10.5781	-5.55937	-16.4202

10.71012	42.27061	7.747972	6.633022	16.75859	19.34757	-13.7857	-3.02715
----------	----------	----------	----------	----------	----------	----------	----------

Blok 2

-162.293	-4.49104	-33.5831	-71.883	-48.2627	-39.3897	2.546769	-62.7779
-23.809	-75.7225	23.92541	2.065701	-10.3372	-29.9468	6.290881	0.086207
133.1419	40.94502	-16.3563	108.3477	58.02272	23.95572	8.768604	-12.3422
61.87257	115.4788	16.12619	12.2714	-47.0118	-55.6787	34.08594	7.662562
-15.7542	-44.8391	55.86159	33.20916	-12.7534	3.387736	12.03768	12.63529
10.21847	-66.9056	32.01399	-13.4062	-14.011	-45.4074	-57.5585	-27.7063
-58.1332	-7.33714	15.26723	-37.1887	17.90765	26.80993	-24.1351	-47.3412
71.62126	-18.1393	-11.315	74.13768	-57.5988	-7.68442	-12.5504	-19.6325

Blok 3

415.4845	30.18842	61.19879	-27.244	-56.1398	20.09456	2.384855	-0.46477
-4.46484	21.85557	60.74996	-10.253	-13.1462	7.085636	8.531416	-4.87875
46.83577	-7.36869	-77.1134	24.55962	28.91249	-9.93002	-5.41227	5.649152
48.5397	-12.0675	-34.0956	14.75881	10.24156	-6.29422	-1.82903	-1.94452
-12.1282	6.554011	13.1966	3.952046	1.875494	-1.74533	2.786686	-3.13608
7.739455	-2.90679	-2.38375	5.940574	2.379162	-0.94148	-4.3034	-1.84838
1.028516	-0.18258	-0.41332	2.413859	0.876468	3.019578	-4.11947	0.661553
0.165732	-0.1433	1.066605	4.193465	1.171441	0.097529	-0.50182	-1.67495

Blok 4

34.8842	43.00978	-10.9733	51.57235	70.64362	101.1744	38.5075	-104.34
104.3346	-14.5258	24.70445	-45.1987	-43.1172	17.08441	-106.31	35.62857
-45.3678	-44.5578	125.8027	1.007746	87.56306	70.70679	-57.6214	-105.568
-23.1784	3.608131	45.24247	32.87712	54.29843	-112.691	6.965589	-8.16681
40.88578	-37.5023	39.55656	26.48598	-17.8797	-54.7489	24.61045	3.231303
12.21023	22.26027	-19.2486	-22.9488	-68.2319	55.16528	-64.1124	31.6722
24.07136	-20.2348	-0.88339	24.97865	-43.7164	-68.9916	82.15349	-31.5063
-114.396	-10.6249	-58.128	17.34362	-17.8599	99.70104	67.41211	48.47362

Kemudian dilanjutkan dengan proses kuantisasi dengan matriks Q50, sehingga menghasilkan tabel berikut:

Blok 1

21	-2	-4	-4	0	1	0	0
12	8	6	5	-2	1	0	0
-6	-2	-3	-2	0	0	0	0
3	-1	1	2	-1	0	0	0
8	1	-1	0	0	0	0	0
-5	0	-1	0	1	0	0	-2
1	1	0	0	-1	0	0	0
0	0	0	0	0	0	0	0

Blok 2

-10	0	-3	-4	-2	-1	0	-1
-2	-6	2	0	0	-1	0	0
10	3	-1	5	1	0	0	0
4	7	1	0	-1	-1	0	0
-1	-2	2	1	0	0	0	0
0	-2	1	0	0	0	-1	-3
-1	0	0	0	0	0	0	0
1	0	0	1	-1	0	0	0

Blok 3

Blok 4

2	4	-1	3	3	3	1	-2
9	-1	2	-2	-2	0	-2	1
-3	-3	8	0	2	1	-1	-2
-2	0	2	1	1	-1	0	0
2	-2	1	0	0	-1	0	0
1	1	0	0	-1	1	-1	4
0	0	0	0	0	-1	1	0
-2	0	-1	0	0	1	1	0

Terakhir, dilakukan pembacaan bilangan secara *zig-zag scanning* pada setiap blok, sehingga menghasilkan barisan blok berikut :

Blok 2 : -10, 0, -2, 10, -6, -3, -4, 2, 3, 4, -1, 7, -1, 0, -2, -1, 0, 5, 1, -2, 0, -1, -2, 2, 0, 1, -1, 0, -1, 0, 0, 0, -1, 1, 1, 0, 1, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, -1, 0, -1, 0, 0, -3, 0, 0, 0

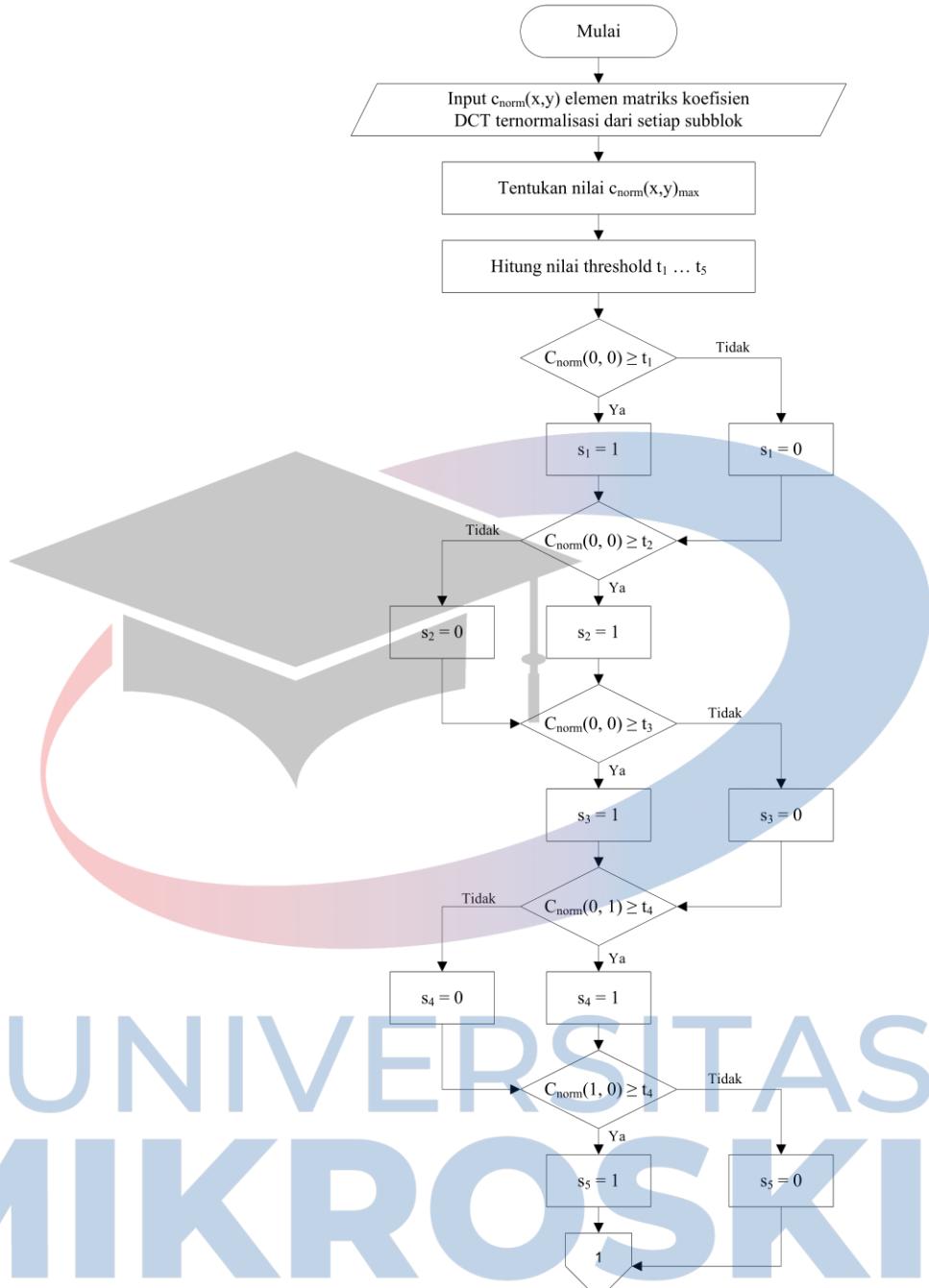
Blok 4 : 2, 4, 9, -3, -1, -1, 3, 2, -3, -2, 2, 0, 8, -2, 3, 3, -2, 0, 2, -2, 1, 0, 1, 1, 1, 2, 0, 1, -2, -2, 1, 1, 0, 0, 0, 0, -2, 0, 0, 0, 0, -1, -1, 1, -2, 0, -1, -1, 0, -1, 0, 0, 1, 0, 0, 0, -1, -1, 0, 1, 1, 4, 0, 1, 0

- d. Hitung *signature* dengan mengekstraksi ciri dari setiap subblok.

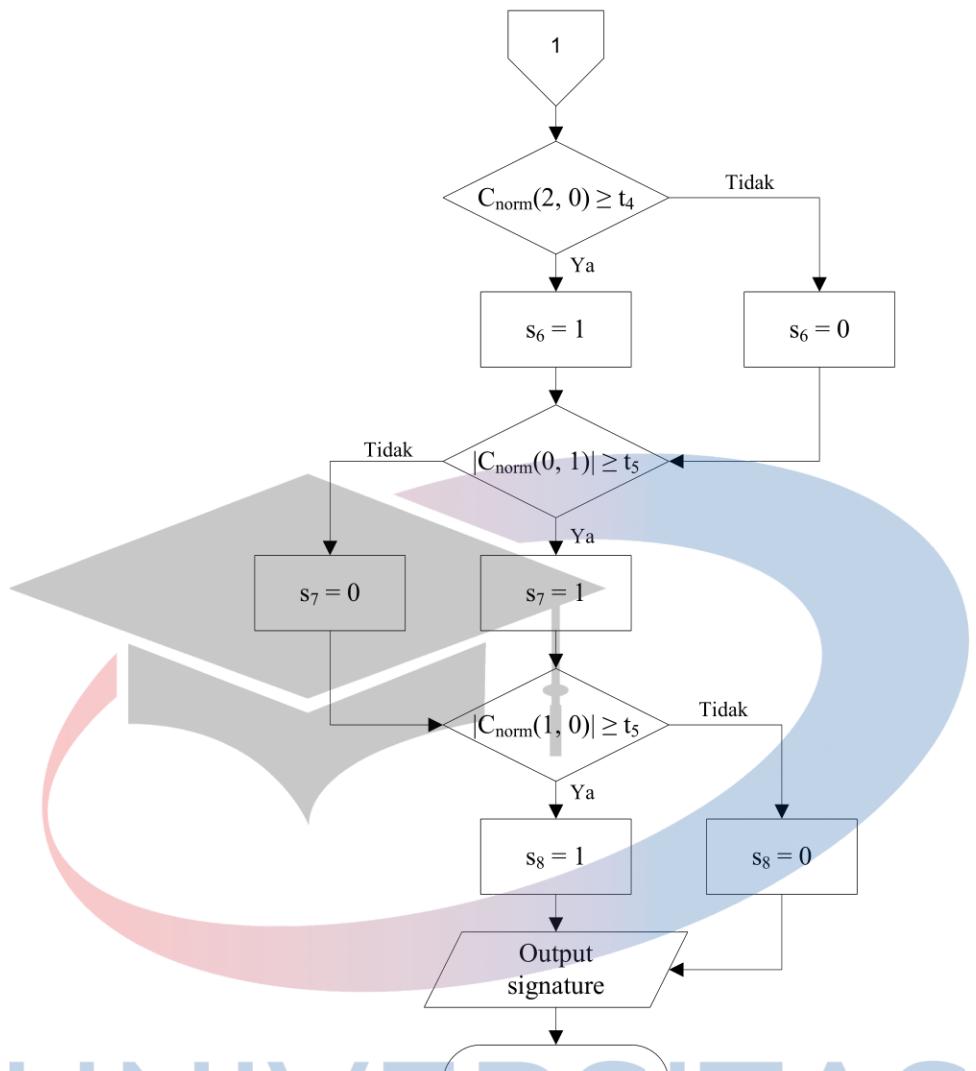
Proses perhitungan *signature* dengan ekstraksi ciri dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 3.3 & 3.4.



# UNIVERSITAS MIKROSKIL



Gambar 3.3 Flowchart Signature dengan Ekstraksi Ciri



Gambar 3.4 Lanjutan Flowchart Signature dengan Ekstraksi Ciri

Proses kerja dari perhitungan *signature* dengan mengekstraksi ciri dari setiap subblok dapat dilihat pada rincian perhitungan berikut:

- 1) Tentukan nilai  $c_{norm}(0,0)_{max}$ .

Subblok 1:  $c_{norm}(0,0) = 21$

Subblok 2:  $c_{norm}(0,0) = -10$

Subblok 3:  $c_{norm}(0,0) = 26$

Subblok 4:  $c_{norm}(0,0) = 2$

$$c_{norm}(0,0)_{max} = 26$$

- 2) Hitung nilai *threshold*  $t_1 \dots t_5$ .

$$t_1 = C_{norm}(0,0)_{max}/4 = 26/4 = 6.5$$

$$t_2 = C_{norm}(0,0)_{max}/2 = 26/2 = 13$$

$$t_3 = 3 * C_{norm}(0,0)_{max}/4 = 3 * 26/4 = 19.5$$

$$t_4 = 0$$

$$t_5 = 3$$

3) Hitung nilai *signature* untuk setiap subblok:

Subblok 1:

$$c_{\text{norm}}(0,0) = 21$$

$$c_{\text{norm}}(0,0) \geq t_1 \rightarrow 21 \geq 6.5 \rightarrow s_1 = 1$$

$$c_{\text{norm}}(0,0) \geq t_2 \rightarrow 21 \geq 13 \rightarrow s_2 = 1$$

$$c_{\text{norm}}(0,0) \geq t_3 \rightarrow 21 \geq 19.5 \rightarrow s_3 = 1$$

$$c_{\text{norm}}(0,1) \geq t_4 \rightarrow -2 \geq 0 \rightarrow s_4 = 0$$

$$c_{\text{norm}}(1,0) \geq t_4 \rightarrow 12 \geq 0 \rightarrow s_5 = 1$$

$$c_{\text{norm}}(2,0) \geq t_4 \rightarrow -6 \geq 0 \rightarrow s_6 = 0$$

$$|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 2 \geq 3 \rightarrow s_7 = 0$$

$$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 12 \geq 3 \rightarrow s_8 = 1$$

$$\text{Signature} = 11101001$$

Subblok 2:

$$c_{\text{norm}}(0,0) = 21$$

$$c_{\text{norm}}(0,0) \geq t_1 \rightarrow -10 \geq 6.5 \rightarrow s_1 = 0$$

$$c_{\text{norm}}(0,0) \geq t_2 \rightarrow -10 \geq 13 \rightarrow s_2 = 0$$

$$c_{\text{norm}}(0,0) \geq t_3 \rightarrow -10 \geq 19.5 \rightarrow s_3 = 0$$

$$c_{\text{norm}}(0,1) \geq t_4 \rightarrow 0 \geq 0 \rightarrow s_4 = 1$$

$$c_{\text{norm}}(1,0) \geq t_4 \rightarrow -2 \geq 0 \rightarrow s_5 = 0$$

$$c_{\text{norm}}(2,0) \geq t_4 \rightarrow 10 \geq 0 \rightarrow s_6 = 1$$

$$|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 2 \geq 3 \rightarrow s_7 = 0$$

$$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 10 \geq 3 \rightarrow s_8 = 1$$

$$\text{Signature} = 00010101$$

Subblok 3:

$$c_{\text{norm}}(0,0) = 26$$

$$c_{\text{norm}}(0,0) \geq t_1 \rightarrow 26 \geq 6.5 \rightarrow s_1 = 1$$

$$c_{\text{norm}}(0,0) \geq t_2 \rightarrow 26 \geq 13 \rightarrow s_2 = 1$$

$$c_{\text{norm}}(0,0) \geq t_3 \rightarrow 26 \geq 19.5 \rightarrow s_3 = 1$$

$$c_{\text{norm}}(0,1) \geq t_4 \rightarrow 3 \geq 0 \rightarrow s_4 = 1$$

$$c_{\text{norm}}(1,0) \geq t_4 \rightarrow 0 \geq 0 \rightarrow s_5 = 1$$

$$c_{\text{norm}}(2,0) \geq t_4 \rightarrow 3 \geq 0 \rightarrow s_6 = 1$$

$$|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 0 \geq 3 \rightarrow s_7 = 0$$

$$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 3 \geq 3 \rightarrow s_8 = 1$$

Signature = 11111101

Subblok 4:

$$c_{\text{norm}}(0,0) = 2$$

$$c_{\text{norm}}(0,0) \geq t_1 \rightarrow 2 \geq 6.5 \rightarrow s_1 = 0$$

$$c_{\text{norm}}(0,0) \geq t_2 \rightarrow 2 \geq 13 \rightarrow s_2 = 0$$

$$c_{\text{norm}}(0,0) \geq t_3 \rightarrow 2 \geq 19.5 \rightarrow s_3 = 0$$

$$c_{\text{norm}}(0,1) \geq t_4 \rightarrow 4 \geq 0 \rightarrow s_4 = 1$$

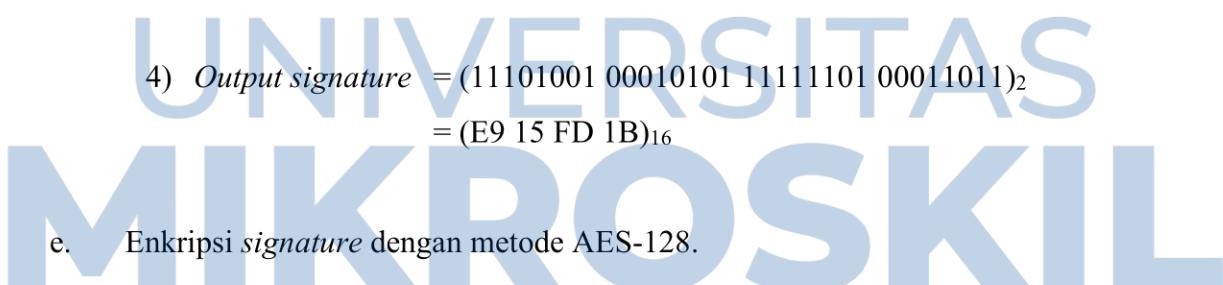
$$c_{\text{norm}}(1,0) \geq t_4 \rightarrow 9 \geq 0 \rightarrow s_5 = 1$$

$$c_{\text{norm}}(2,0) \geq t_4 \rightarrow -3 \geq 0 \rightarrow s_6 = 0$$

$$|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 4 \geq 3 \rightarrow s_7 = 1$$

$$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 9 \geq 3 \rightarrow s_8 = 1$$

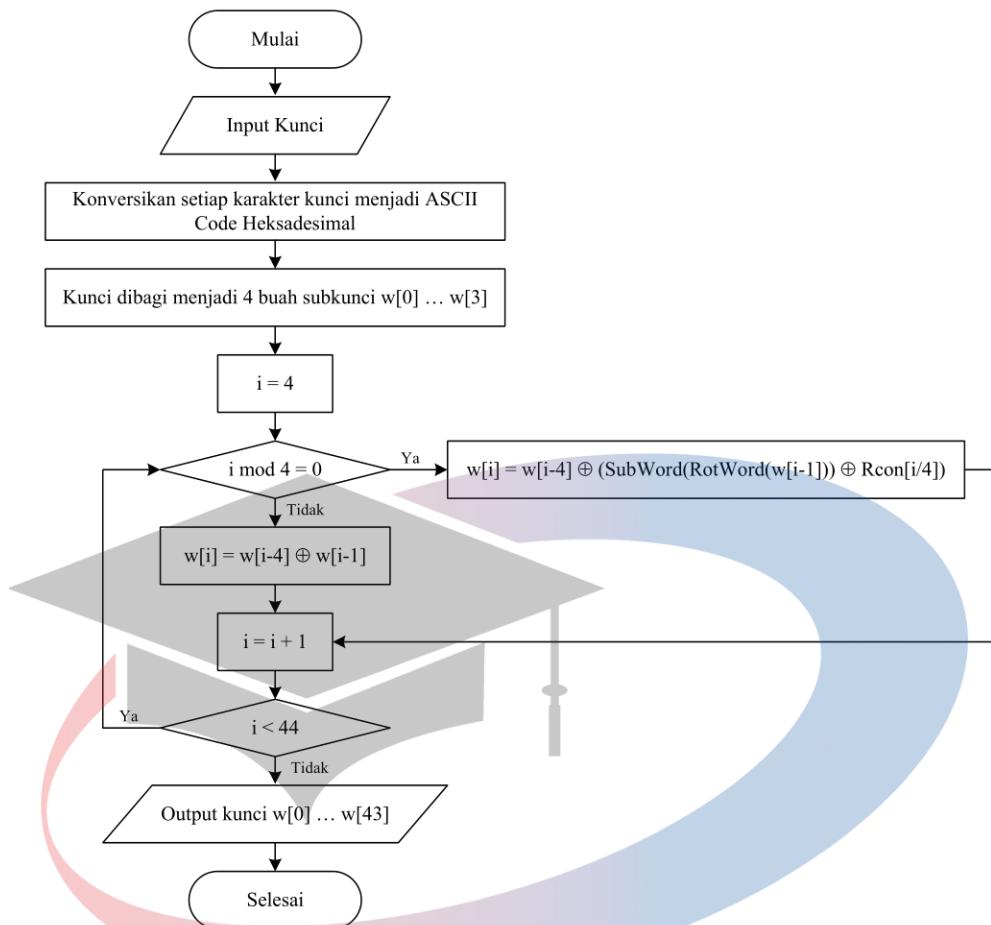
Signature = 00011011



4) Output signature =  $(11101001\ 00010101\ 11111101\ 00011011)_2$   
= (E9 15 FD 1B)<sub>16</sub>

e. Enkripsi signature dengan metode AES-128.

Untuk melakukan proses enkripsi dengan metode AES-128 diperlukan proses pembentukan kunci terlebih dahulu. Proses pembentukan kunci pada metode AES-128 dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 3.5.



Gambar 3.5 Proses Pembentukan Kunci dengan Metode AES 128 Bit

Langkah kerja dari proses pembentukan kunci dengan menggunakan metode AES (semua angka pada perhitungan dalam format heksadesimal) adalah sebagai berikut:

- 1) Input kunci

Kunci = TUGASAKHIRAES128

- 2) Konversikan setiap karakter kunci menjadi ASCII Code Heksadesimal

T = 54

U = 55

G = 47

A = 41

S = 53

A = 41

K = 4B

H = 48

I = 49

R = 52

A = 41

E = 45

S = 53

1 = 31

2 = 32

8 = 38

Kunci setelah diubah ke notasi heksadesimal (disimbolkan dengan 'X')

X = 5455474153414B484952414553313238

- 3) Kunci dibagi menjadi 4 buah subkunci w[0] ... w[3]

w[0] = 54554741

w[1] = 53414B48

w[2] = 49524145

w[3] = 53313238

- 4)  $i = 4$ .

- 5)  $i \bmod 4 = 0$ , maka:

$w[i] = w[i-4] \text{ XOR } (\text{SubWord}(\text{RotWord}(w[i-1])) \text{ XOR } Rcon[i/4])$

$w[4] = w[0] \text{ XOR } (\text{SubWord}(\text{RotWord}(w[3]))) \text{ XOR } Rcon[1]$

$w[4] = 54554741 \text{ XOR } (\text{SubWord}(\text{RotWord}(53313238))) \text{ XOR } 01000000$

RotWord = Lakukan sekali rotasi kiri sebanyak 1 byte.

$\text{RotWord}(53313238) = 31323853$

SubWord = substitusi dengan menggunakan S-Box

$\text{SubWord}(31323853) = C72307ED$

$w[4] = 54554741 \text{ XOR } (C72307ED \text{ XOR } 01000000)$

$w[4] = 54554741 \text{ XOR } C62307ED$

$w[4] = 927640AC$

- 6)  $i = i + 1 = 4 + 1 = 5$

- $i \bmod 4 \neq 0$ , maka:

$w[i] = w[i-4] \text{ XOR } w[i-1]$

$w[5] = w[1] \text{ XOR } w[4]$

$w[5] = 53414B48 \text{ XOR } 927640AC$

$w[5] = C1370BE4$

- 7)  $i = i + 1 = 5 + 1 = 6$

- $i \bmod 4 \neq 0$ , maka:

$w[i] = w[i-4] \text{ XOR } w[i-1]$

$w[6] = w[2] \oplus w[5]$

$w[6] = 49524145 \text{ XOR } C1370BE4$

$w[6] = 88654AA1$

8)  $i = i + 1 = 6 + 1 = 7$

$i \bmod 4 \neq 0$ , maka:

$w[i] = w[i-4] \oplus w[i-1]$

$w[7] = w[3] \oplus w[6]$

$w[7] = 53313238 \text{ XOR } 88654AA1$

$w[7] = DB547899$

9)  $i = i + 1 = 7 + 1 = 8$

$i \bmod 4 = 0$ , maka:

$w[i] = w[i-4] \oplus (\text{SubWord}(\text{RotWord}(w[i-1])) \oplus Rcon[i/4])$

$w[8] = w[4] \oplus (\text{SubWord}(\text{RotWord}(w[7])) \oplus Rcon[2])$

$w[8] = 927640AC \text{ XOR } (\text{SubWord}(\text{RotWord}(DB547899)) \oplus 02000000)$

$\text{RotWord} = \text{Lakukan sekali rotasi kiri sebanyak 1 byte.}$

$\text{RotWord}(DB547899) = 547899DB$

$\text{SubWord} = \text{substitusi dengan menggunakan S-Box}$

$\text{SubWord}(547899DB) = 20BCEEB9$

$w[8] = 927640AC \text{ XOR } (20BCEEB9 \oplus 02000000)$

$w[8] = 927640AC \text{ XOR } 22BCEEB9$

$w[8] = B0CAAE15$

10)  $i = i + 1 = 8 + 1 = 9$

$i \bmod 4 \neq 0$ , maka:

$w[i] = w[i-4] \oplus w[i-1]$

$w[9] = w[5] \oplus w[8]$

$w[9] = C1370BE4 \text{ XOR } B0CAAE15$

$w[9] = 71FDA5F1$

11)  $i = i + 1 = 9 + 1 = 10$

$i \bmod 4 \neq 0$ , maka:

$w[i] = w[i-4] \oplus w[i-1]$

$w[10] = w[6] \oplus w[9]$

$w[10] = 88654AA1 \text{ XOR } 71FDA5F1$

$w[10] = F998EF50$

12) Proses diatas akan dilakukan hingga  $W[43]$ .

13) Output kunci

Subkunci[0] = w[0] w[1] w[2] w[3]  
= 5455474153414B484952414553313238

Subkunci[1] = w[4] w[5] w[6] w[7]  
= 927640ACC1370BE488654AA1DB547899

Subkunci[2] = w[8] w[9] w[10] w[11]  
= B0CAAE1571FDA5F1F998EF5022CC97C9

Subkunci[3] = w[12] w[13] w[14] w[15]  
= FF4273868EBFD6777727392755EBAEEE

Subkunci[4] = w[16] w[17] w[18] w[19]  
= 1EA65B7A90198D0DE73EB42AB2D51AC4

Subkunci[5] = w[20] w[21] w[22] w[23]  
= 0D04474D9D1DCA407A237E6AC8F664AE

Subkunci[6] = w[24] w[25] w[26] w[27]  
= 6F47A3A5F25A69E58879178F408F7321

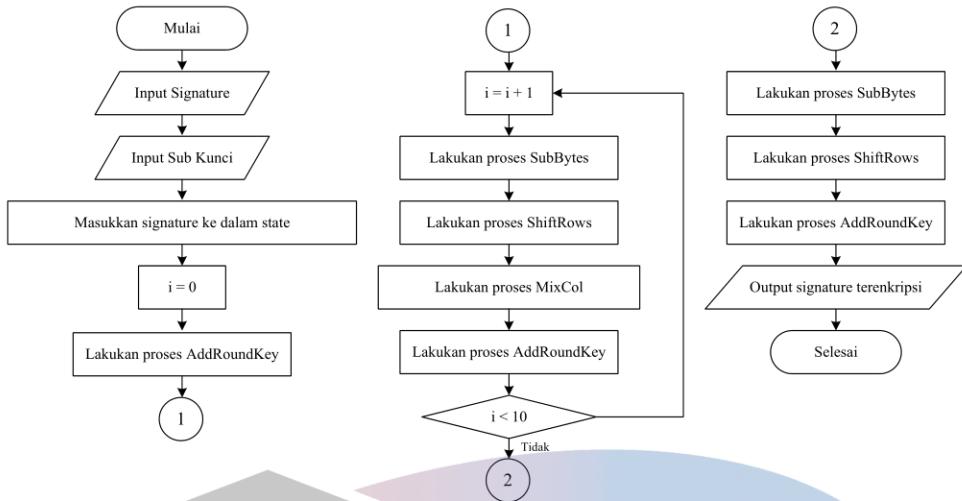
Subkunci[7] = w[28] w[29] w[30] w[31]  
= 5CC85EACAE92374926EB20C6666453E7

Subkunci[8] = w[32] w[33] w[34] w[35]  
= 9F25CA9F31B7FDD6175CDD1071388EF7

Subkunci[9] = w[36] w[37] w[38] w[39]  
= 833CA23CB28B5FEAA5D782FAD4EF0C0D

Subkunci[10] = w[40] w[41] w[42] w[43]  
= 6AC27574D8492A9E7D9EA864A971A469

Proses enkripsi *signature* dengan metode AES-128 dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 3.6.



Gambar 3.6 Proses Enkripsi dengan Metode AES 128 Bit

Langkah kerja dari proses enkripsi *signature* dengan metode AES-128 adalah sebagai berikut:

- 1) Input *signature* = (E9 15 FD 1B)<sub>16</sub>.
- 2) Input sub kunci, seperti terlihat pada langkah (13) diatas.
- 3) Masukkan *signature* ke dalam *state*.

E9 15 FD 1B (heksadesimal)

Masukkan ke dalam *State* :

*State*

E9 00 00 00  
15 00 00 00  
FD 00 00 00  
1B 00 00 00

- 4) *i* = 0.

- 5) Lakukan proses AddRoundkey pada *State* dengan Subkunci[0] yaitu:

State	XOR	Subkunci[0]	=	AddRoundkey
E9 00 00 00	54 53 49 53			BD 53 49 53
15 00 00 00	55 41 52 31			40 41 52 31
FD 00 00 00	47 4B 41 32			BA 4B 41 32
1B 00 00 00	41 48 45 38			5A 48 45 38

- 6) *i* = *i* + 1 = 0 + 1 = 1.

- 7) Lakukan proses SubBytes

$$\begin{array}{ccc} \text{AddRoundkey} & \rightarrow & \text{SubByte} \\ \text{BD 53 49 53} & & \text{7A ED 3B ED} \end{array}$$

40 41 52 31	09 83 00 C7
BA 4B 41 32	F4 B3 83 23
5A 48 45 38	BE 52 6E 07

8) Lakukan proses ShiftRows

SubByte	→	ShiftRow
7A ED 3B ED		7A ED 3B ED
09 83 00 C7		83 00 C7 09
F4 B3 83 23		83 23 F4 B3
BE 52 6E 07		07 BE 52 6E

9) Lakukan proses MixCol

ShiftRow	→	MixCol
7A ED 3B ED		EE 5C 82 07
83 00 C7 09		FE 36 FB 5F
83 23 F4 B3		ED 72 F9 2B
07 BE 52 6E		80 68 DA 4A

10) Lakukan proses AddRoundKey

MixCol	XOR	Subkunci[1]	=	AddRoundkey
EE 5C 82 07		92 C1 88 DB		7C 9D 0A DC
FE 36 FB 5F		76 37 65 54		88 01 9E 0B
ED 72 F9 2B		40 0B 4A 78		AD 79 B3 53
80 68 DA 4A		AC E4 A1 99		2C 8C 7B D3

11)  $i = i + 1 = 1 + 1 = 2$ .

12) Lakukan proses SubBytes

AddRoundkey	→	SubByte
7C 9D 0A DC		10 5E 67 86
88 01 9E 0B		C4 7C 0B 2B
AD 79 B3 53		95 B6 6D ED
2C 8C 7B D3		71 64 21 66

13) Lakukan proses ShiftRows

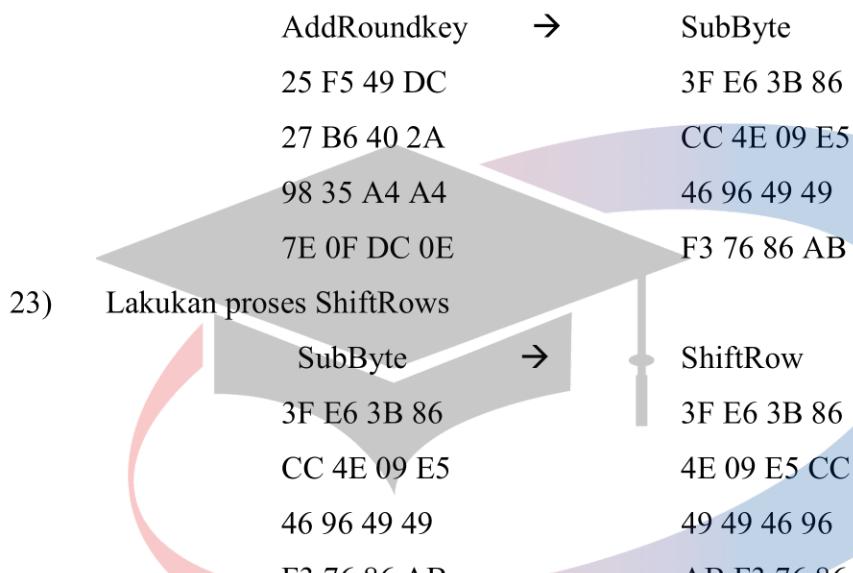
SubByte	→	ShiftRow
10 5E 67 86		10 5E 67 86
C4 7C 0B 2B		7C 0B 2B C4
95 B6 6D ED		6D ED 95 B6

- 71 64 21 66                      66 71 64 21
- 14) Lakukan proses MixCol
- |             |   |             |
|-------------|---|-------------|
| ShiftRow    | → | MixCol      |
| 10 5E 67 86 |   | AF 3D 42 D7 |
| 7C 0B 2B C4 |   | 39 15 F1 F5 |
| 6D ED 95 B6 |   | 1C 07 D1 56 |
| 66 71 64 21 |   | ED E6 DF A1 |
- 15) Lakukan proses AddRoundKey
- |             |     |             |   |             |
|-------------|-----|-------------|---|-------------|
| MixCol      | XOR | Subkunci[2] | = | AddRoundkey |
| AF 3D 42 D7 |     | B0 71 F9 22 |   | 1F 4C BB F5 |
| 39 15 F1 F5 |     | CA FD 98 CC |   | F3 E8 69 39 |
| 1C 07 D1 56 |     | AE A5 EF 97 |   | B2 A2 3E C1 |
| ED E6 DF A1 |     | 15 F1 50 C9 |   | F8 17 8F 68 |
- 16)  $i = i + 1 = 2 + 1 = 3.$
- 17) Lakukan proses SubBytes
- |             |   |             |
|-------------|---|-------------|
| AddRoundkey | → | SubByte     |
| 1F 4C BB F5 |   | C0 29 EA E6 |
| F3 E8 69 39 |   | 0D 9B F9 12 |
| B2 A2 3E C1 |   | 37 3A B2 78 |
| F8 17 8F 68 |   | 41 F0 73 45 |
- 18) Lakukan proses ShiftRows
- |             |   |             |
|-------------|---|-------------|
| SubByte     | → | ShiftRow    |
| C0 29 EA E6 |   | C0 29 EA E6 |
| 0D 9B F9 12 |   | 9B F9 12 0D |
| 37 3A B2 78 |   | B2 78 37 3A |
| 41 F0 73 45 |   | 45 41 F0 73 |
- 19) Lakukan proses MixCol
- |             |   |             |
|-------------|---|-------------|
| ShiftRow    | → | MixCol      |
| C0 29 EA E6 |   | DA 7B 3E 89 |
| 9B F9 12 0D |   | 65 09 67 C1 |
| B2 78 37 3A |   | EB E3 9D 0A |
| 45 41 F0 73 |   | F8 78 FB E0 |
- 20) Lakukan proses AddRoundKey
- |        |     |             |   |             |
|--------|-----|-------------|---|-------------|
| MixCol | XOR | Subkunci[3] | = | AddRoundkey |
|--------|-----|-------------|---|-------------|

DA 7B 3E 89	FF 8E 77 55	25 F5 49 DC
65 09 67 C1	42 BF 27 EB	27 B6 40 2A
EB E3 9D 0A	73 D6 39 AE	98 35 A4 A4
F8 78 FB E0	86 77 27 EE	7E 0F DC 0E

21)  $i = i + 1 = 3 + 1 = 4.$

22) Lakukan proses SubBytes



24) Lakukan proses MixCol



25) Lakukan proses AddRoundKey

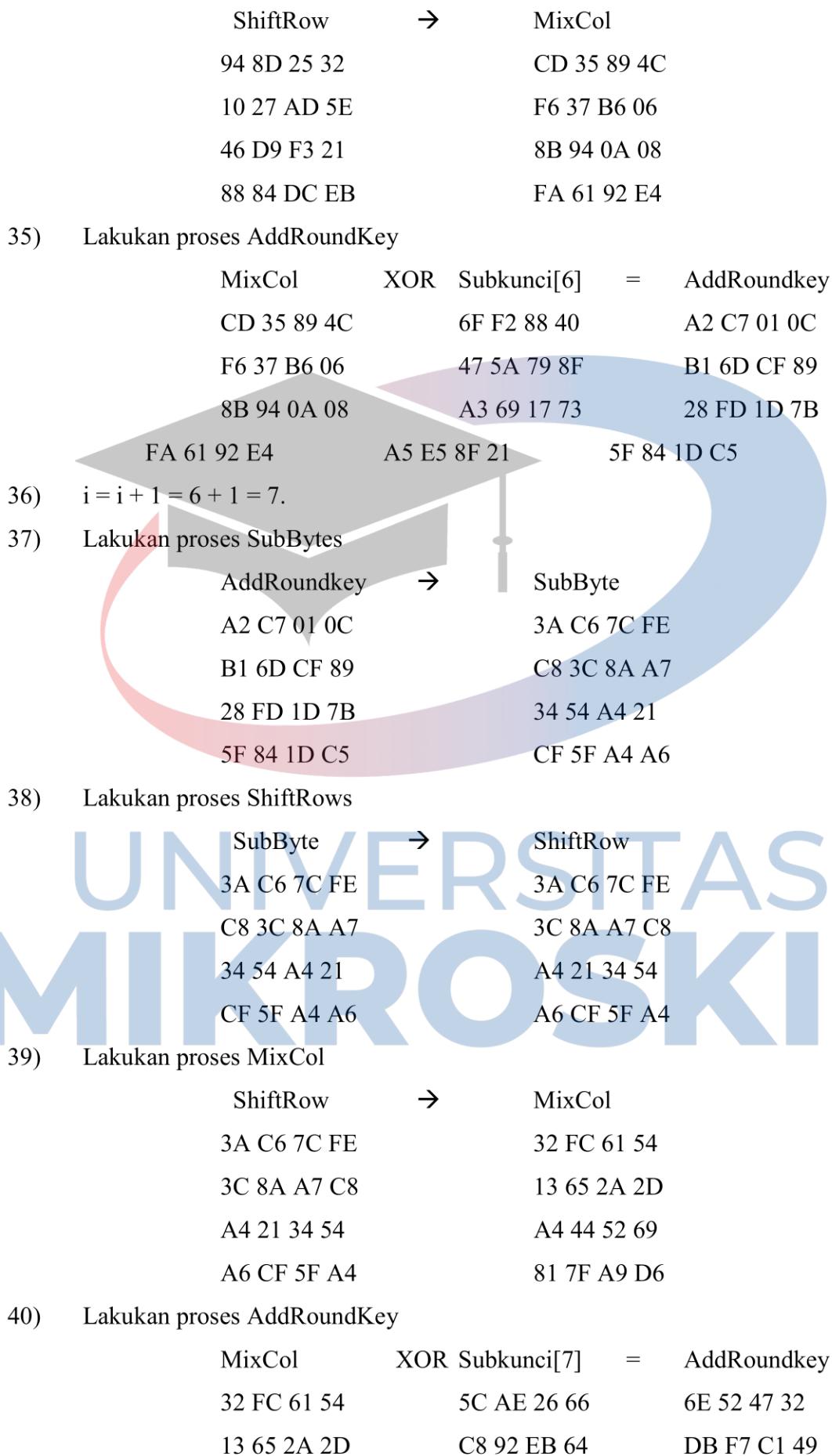
MixCol	XOR	Subkunci[4]	=	AddRoundkey
4E 76 72 48		1E 90 E7 B2		50 E6 95 FA
D3 DC 56 22		A6 19 3E D5		75 C5 68 F7
05 73 C8 EC		5B 8D B4 1A		5E FE 7C F6
0B 8C 02 DC		7A 0D 2A C4		71 81 28 18

26)  $i = i + 1 = 4 + 1 = 5.$

27) Lakukan proses SubBytes



- 5E FE 7C F6  
71 81 28 18
- 58 BB 10 42  
A3 0C 34 AD
- 28) Lakukan proses ShiftRows
- |             |               |             |
|-------------|---------------|-------------|
| SubByte     | $\rightarrow$ | ShiftRow    |
| 53 8E 2A 2D |               | 53 8E 2A 2D |
| A6 45 68 9D |               | A6 45 68 9D |
| 58 BB 10 42 |               | 10 42 58 BB |
| A3 0C 34 AD |               | AD A3 0C 34 |
- 29) Lakukan proses MixCol
- |             |               |             |
|-------------|---------------|-------------|
| ShiftRow    | $\rightarrow$ | MixCol      |
| 53 8E 2A 2D |               | EA 29 B8 69 |
| A6 45 68 9D |               | 99 61 1E EE |
| 10 42 58 BB |               | 39 B1 E6 81 |
| AD A3 0C 34 |               | 02 D3 56 39 |
- 30) Lakukan proses AddRoundKey
- |             |     |             |   |             |
|-------------|-----|-------------|---|-------------|
| MixCol      | XOR | Subkunci[5] | = | AddRoundkey |
| EA 29 B8 69 |     | 0D 9D 7A C8 |   | E7 B4 C2 A1 |
| 99 61 1E EE |     | 04 1D 23 F6 |   | 9D 7C 3D 18 |
| 39 B1 E6 81 |     | 47 CA 7E 64 |   | 7E 7B 98 E5 |
- 31)  $i = i + 1 = 5 + 1 = 6.$
- 32) Lakukan proses SubBytes
- |             |               |             |
|-------------|---------------|-------------|
| AddRoundkey | $\rightarrow$ | SubByte     |
| E7 B4 C2 A1 |               | 94 8D 25 32 |
| 9D 7C 3D 18 |               | 5E 10 27 AD |
| 7E 7B 98 E5 |               | F3 21 46 D9 |
| 4F 93 3C 97 |               | 84 DC EB 88 |
- 33) Lakukan proses ShiftRows
- |             |               |             |
|-------------|---------------|-------------|
| SubByte     | $\rightarrow$ | ShiftRow    |
| 94 8D 25 32 |               | 94 8D 25 32 |
| 5E 10 27 AD |               | 10 27 AD 5E |
| F3 21 46 D9 |               | 46 D9 F3 21 |
| 84 DC EB 88 |               | 88 84 DC EB |
- 34) Lakukan proses MixCol



A4 44 52 69  
81 7F A9 D6

5E 37 20 53  
AC 49 C6 E7

FA 73 72 3A  
2D 36 6F 31

41)  $i = i + 1 = 7 + 1 = 8.$

42) Lakukan proses SubBytes

AddRoundkey	→	SubByte
6E 52 47 32		9F 00 A0 23
DB F7 C1 49		B9 68 78 3B
FA 73 72 3A		2D 8F 40 80
2D 36 6F 31		D8 05 A8 C7

43) Lakukan proses ShiftRows

SubByte	→	ShiftRow
9F 00 A0 23		9F 00 A0 23
B9 68 78 3B		78 3B B9 68
2D 8F 40 80		2D 8F 40 80
D8 05 A8 C7		A8 C7 D8 05

44) Lakukan proses MixCol

ShiftRow	→	MixCol
9F 00 A0 23		28 05 13 7B
78 3B B9 68		B0 3B D1 6D
2D 8F 40 80		5E 6C EA 5F
A8 C7 D8 05		A4 21 A9 87

45) Lakukan proses AddRoundKey

MixCol	XOR	Subkunci[8]	=	AddRoundkey
28 05 13 7B		9F 31 17 71		B7 34 04 0A
B0 3B D1 6D		25 B7 5C 38		95 8C 8D 55
5E 6C EA 5F		CA FD DD 8E		94 91 37 D1
A4 21 A9 87		9F D6 10 F7		3B F7 B9 70

46)  $i = i + 1 = 8 + 1 = 9.$

47) Lakukan proses SubBytes

AddRoundkey	→	SubByte
B7 34 04 0A		A9 18 F2 67
95 8C 8D 55		2A 64 5D FC
94 91 37 D1		22 81 9A 3E
3B F7 B9 70		E2 68 56 51

48) Lakukan proses ShiftRows

SubByte	→	ShiftRow
A9 18 F2 67		A9 18 F2 67
2A 64 5D FC		64 5D FC 2A
22 81 9A 3E		9A 3E 22 81
E2 68 56 51		51 E2 68 56

49) Lakukan proses MixCol

ShiftRow	→	MixCol
A9 18 F2 67		2E 0B AA 67
64 5D FC 2A		85 02 1F FD
9A 3E 22 81		11 04 F2 AE
51 E2 68 56		BC 94 03 AE

50) Lakukan proses AddRoundKey

MixCol	XOR	Subkunci[9]	=	AddRoundkey
2E 0B AA 67		83 B2 A5 D4		AD B9 0F B3
85 02 1F FD		3C 8B D7 EF		B9 89 C8 12
11 04 F2 AE		A2 5F 82 0C		B3 5B 70 A2
BC 94 03 AE		3C EA FA 0D		80 7E F9 A3

51)  $i = i + 1 = 9 + 1 = 10$ .

52) Lakukan proses SubBytes

AddRoundkey	→	SubByte
AD B9 0F B3		95 56 76 6D
B9 89 C8 12		56 A7 E8 C9
B3 5B 70 A2		6D 39 51 3A
80 7E F9 A3		CD F3 99 0A

53) Lakukan proses ShiftRows

SubByte	→	ShiftRow
95 56 76 6D		95 56 76 6D
56 A7 E8 C9		A7 E8 C9 56
6D 39 51 3A		51 3A 6D 39
CD F3 99 0A		0A CD F3 99

54) Lakukan proses AddRoundKey

ShiftRow	XOR	Subkunci[10]	=	AddRoundkey
95 56 76 6D		6A D8 7D A9		FF 8E 0B C4

A7 E8 C9 56	C2 49 9E 71	65 A1 57 27
51 3A 6D 39	75 2A A8 A4	24 10 C5 9D
0A CD F3 99	74 9E 64 69	7E 53 97 F0

*Ciphertext = FF65247E8EA110530B57C597C4279DF0*

- f. Tempelkan hasil enkripsi pada *blockchain*.

*Blockchain:*

Transaksi ID = 1

*Ciphertext = FF65247E8EA110530B57C597C4279DF0*

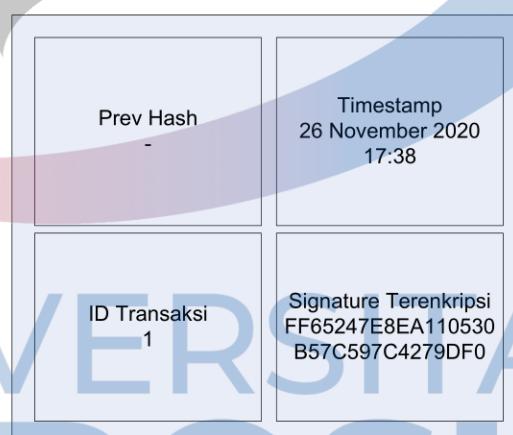
*Timestamp = 26 November 2020 17:38*

- g. Hitung nilai *hash* sebagai hubungan blok dengan fungsi SHA-1.

Input string = 1, FF65247E8EA110530B57C597C4279DF0, 26 November 2020 17:38

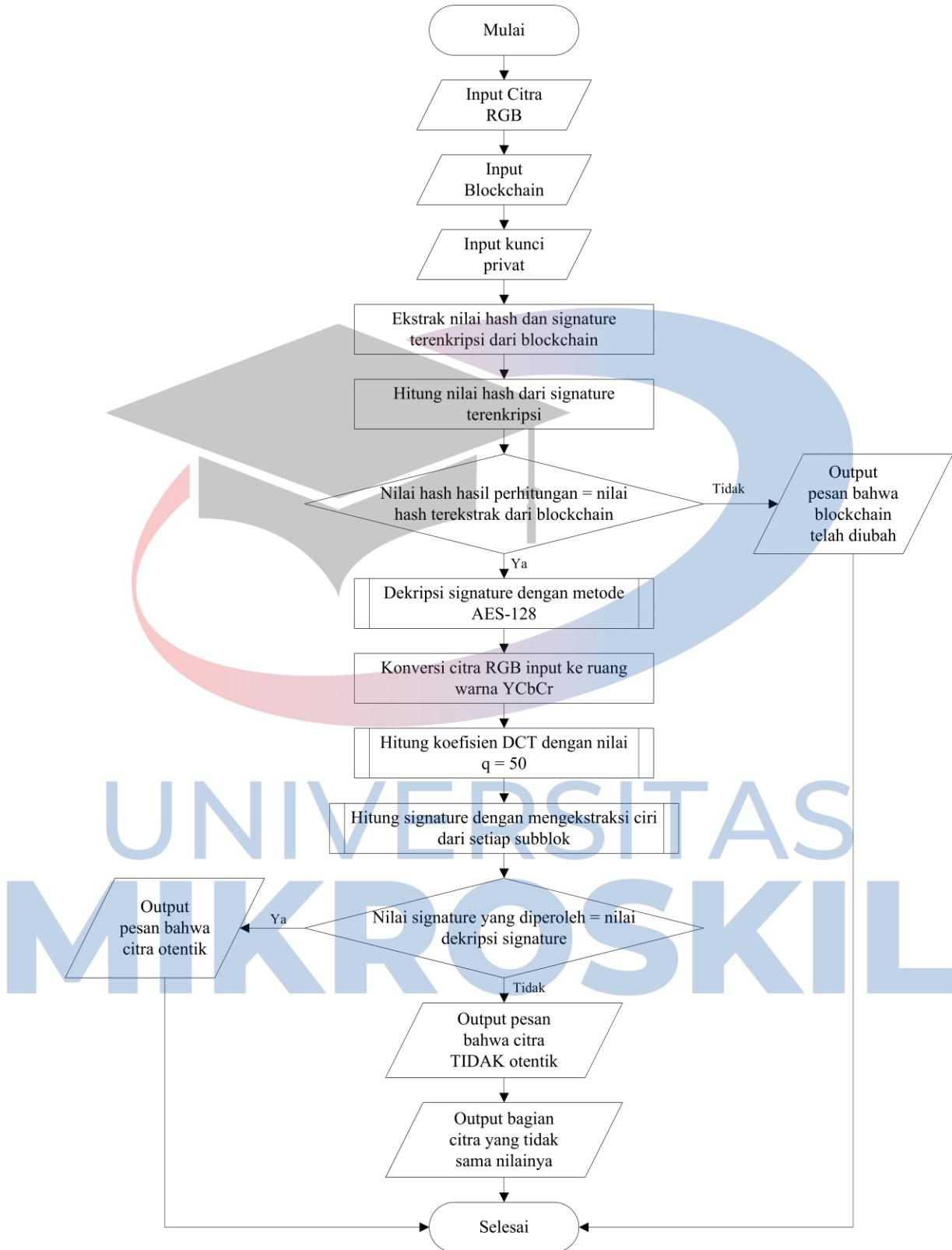
Nilai *Hash SHA-1* = 6E04E269B6AC7FA525A7BA0BD6D4D04E9EAA513A

- h. *Output blockchain.*



Gambar 3.7 Hasil *Blockchain*

## 2. Proses Pengecekan (Autentikasi) Citra untuk Mendeteksi Plagiarisme



Gambar 3.8 *Flowchart* Proses Autentikasi Citra

Berdasarkan *flowchart* pada gambar 3.8, maka langkah kerja dari proses autentikasi citra untuk mendeteksi plagiarisme adalah sebagai berikut:

a. *Input* citra RGB.

Misalkan citra RGB diganti piksel (0,0) dan piksel (0,1) seperti terlihat pada tabel 3.3.

Tabel 3.3 Elemen Warna RGB dari Citra Input Setelah Dimodifikasi

88	60	96	128	119	248	76	255	181	114	54	95	71	148	85	75
100	65	168	201	191	112	63	250	64	86	166	245	157	58	224	214
60	20	56	89	79	247	240	215	106	153	166	162	71	63	244	148
249	72	97	140	190	129	172	123	136	212	84	193	112	196	146	120
180	74	184	232	219	181	127	117	141	67	205	156	209	174	66	193
136	107	89	230	127	89	50	221	207	60	101	161	68	229	172	227
240	249	169	134	179	65	130	94	131	232	253	120	151	107	240	226
135	201	86	67	72	156	93	68	74	76	80	116	146	66	168	51
115	198	86	119	77	161	199	156	154	197	199	238	92	85	229	233
130	228	126	85	148	150	130	156	138	55	229	97	123	118	179	64
193	220	247	103	77	97	101	134	241	87	199	109	114	187	153	146
109	179	206	68	87	251	68	196	70	212	237	55	127	145	164	118
159	88	240	221	210	176	194	189	169	132	83	124	242	213	155	255
209	156	252	217	154	171	224	231	179	61	86	212	71	73	167	210
172	174	118	254	206	111	74	238	164	86	213	68	84	182	156	104
189	58	76	163	158	186	194	123	89	255	164	106	141	69	253	180
138	160	136	238	64	215	223	209	219	121	110	226	57	77	237	233
76	203	214	107	63	143	230	191	123	247	60	145	253	155	73	90
202	158	122	135	172	84	201	119	221	150	111	97	147	187	251	224
97	108	214	131	96	167	59	169	125	171	85	227	232	153	130	244
99	63	218	135	94	98	154	247	120	89	78	208	168	58	86	247
72	66	93	156	171	176	170	215	229	105	218	176	236	249	155	252
255	78	199	131	94	226	135	131	191	73	160	114	194	136	57	177
54	79	72	134	244	210	111	126	232	93	152	249	140	242	108	114
220	227	208	134	160	51	146	187	137	202	135	141	135	239	154	56
111	140	218	119	57	177	234	123	56	65	152	148	118	69	191	176
110	254	232	86	173	73	188	254	192	140	186	214	175	118	100	141
181	234	98	99	99	219	76	230	198	116	247	74	221	148	237	246
141	81	134	242	113	81	166	91	74	107	184	77	132	78	105	53
229	195	66	253	61	170	168	118	168	242	210	147	55	82	165	198
62	71	242	106	176	128	189	186	70	144	241	50	65	200	161	249
135	185	80	89	124	185	161	239	182	224	246	228	92	201	126	129
143	183	95	63	211	139	136	57	244	54	197	192	116	208	166	168
153	167	111	150	251	237	91	124	240	159	170	69	151	228	90	211
189	186	51	165	165	71	190	120	219	166	160	158	214	174	141	180
172	227	99	159	57	132	236	181	185	199	240	141	160	71	229	120
215	192	237	109	251	182	253	96	201	64	189	113	246	184	132	209
99	128	88	214	227	233	241	154	198	111	244	132	152	112	181	170
76	248	231	234	75	181	70	51	226	65	173	209	246	95	185	200
150	111	253	202	131	179	134	250	53	237	66	98	74	163	251	192
209	197	76	93	249	130	121	96	202	168	244	57	176	161	138	126
226	141	188	198	210	80	136	253	71	237	157	73	62	254	50	240
68	219	214	102	222	76	214	199	154	96	225	75	203	59	55	157
118	237	134	78	69	254	79	155	62	145	52	146	98	60	120	152
54	143	87	127	161	224	173	87	177	251	231	108	241	222	212	251

210	53	203	195	157	216	248	59	239	240	60	201	207	127	130	53
180	247	144	97	72	53	69	144	123	223	121	240	71	118	164	225
195	159	156	134	178	172	129	213	75	180	100	74	169	131	226	211

- b. *Input blockchain*, seperti terlihat pada gambar 3.6.
- c. *Input kunci privat*, yaitu TUGASAKHIRAES128.
- d. Ekstrak nilai *hash* dan *signature* terenkripsi dari *blockchain*.

Nilai *hash* = 6E04E269B6AC7FA525A7BA0BD6D4D04E9EAA513A

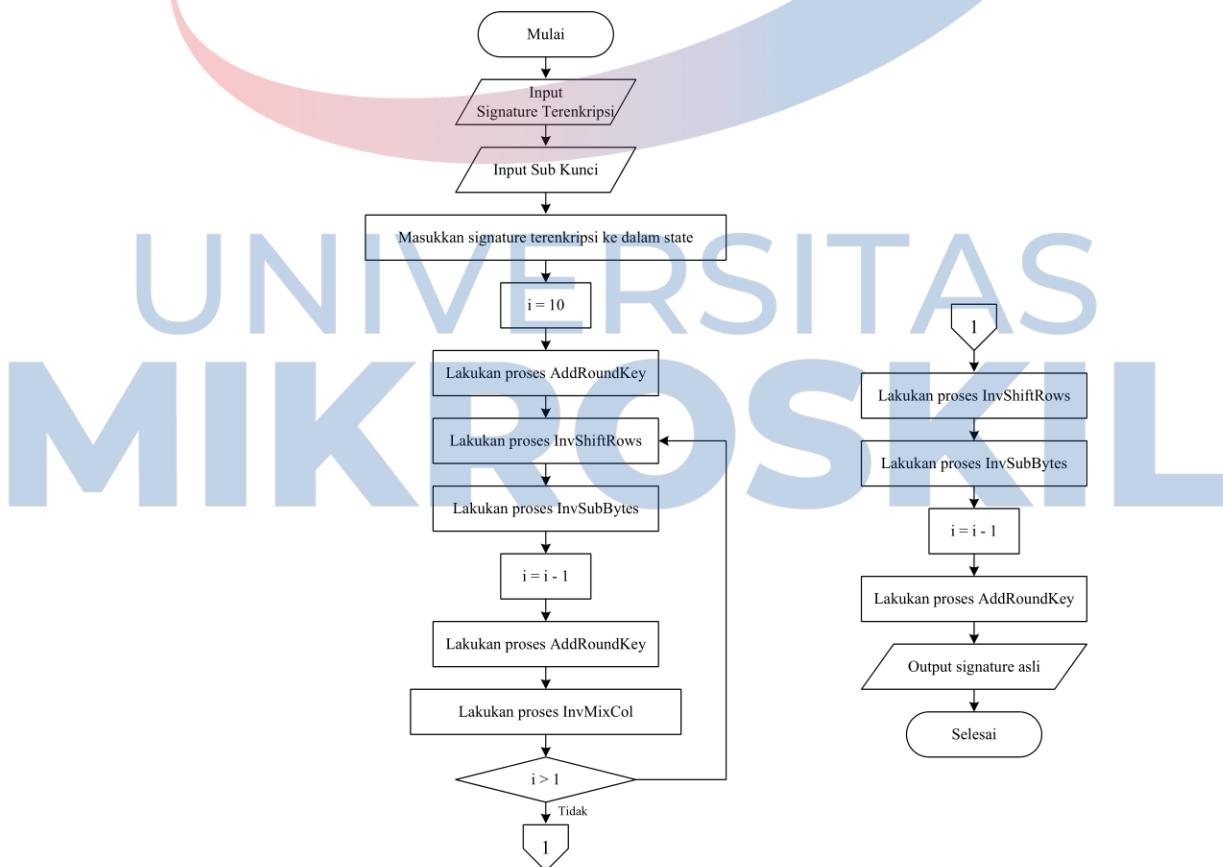
Nilai *signature* terenkripsi = FF65247E8EA110530B57C597C4279DF0

- e. Hitung nilai *hash* dari *signature* terenkripsi.

Nilai *hash SHA-1* = 6E04E269B6AC7FA525A7BA0BD6D4D04E9EAA513A

- f. Karena nilai *hash SHA-1* pada langkah (5) sama dengan langkah (4) berarti tidak terjadi penggantian data pada *blockchain* sehingga proses dilanjutkan.
- g. Dekripsi *signature* dengan metode AES 128.

Proses dekripsi *signature* dengan metode AES 128 dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar 3.9.



Gambar 3.9 Proses Dekripsi *Signature* dengan Metode AES 128

Langkah kerja dari proses dekripsi *signature* terenkripsi dengan metode AES-128 adalah sebagai berikut:

- 1) Input *signature* terenkripsi = (FF65247E8EA110530B57C597C4279DF0)<sub>16</sub>.
- 2) Input sub kunci, seperti terlihat pada langkah (13) proses pembentukan kunci diatas.
- 3) Masukkan *signature* terenkripsi ke dalam *state*.

FF65247E8EA110530B57C597C4279DF0 (heksadesimal)

Masukkan ke dalam *State* :

*State*

FF 8E 0B C4

65 A1 57 27

24 10 C5 9D

7E 53 97 F0

- 4)  $i = 10$ .

- 5) Lakukan proses AddRoundKey.

Ciphertext	XOR	Subkunci[10]	=	AddRoundkey
FF 8E 0B C4		6A D8 7D A9		95 56 76 6D
65 A1 57 27		C2 49 9E 71		A7 E8 C9 56
24 10 C5 9D		75 2A A8 A4		51 3A 6D 39
7E 53 97 F0		74 9E 64 69		0A CD F3 99

- 6) Lakukan proses InvShiftRows

AddRoundKey  $\rightarrow$  InvShiftRow

95 56 76 6D	95 56 76 6D
A7 E8 C9 56	56 A7 E8 C9
51 3A 6D 39	6D 39 51 3A
0A CD F3 99	CD F3 99 0A

- 7) Lakukan proses InvSubBytes

InvShiftRows	$\rightarrow$	InvSubByte
95 56 76 6D		AD B9 0F B3
56 A7 E8 C9		B9 89 C8 12
6D 39 51 3A		B3 5B 70 A2
CD F3 99 0A		80 7E F9 A3

- 8)  $i = i - 1 = 10 - 1 = 9$ .

- 9) Lakukan proses AddRoundKey

InvSubBytes	XOR Subkunci[9]	=	AddRoundkey
AD B9 0F B3	83 B2 A5 D4		2E 0B AA 67
B9 89 C8 12	3C 8B D7 EF		85 02 1F FD

B3 5B 70 A2  
80 7E F9 A3

A2 5F 82 0C  
3C EA FA 0D

11 04 F2 AE  
BC 94 03 AE

10) Lakukan proses InvMixCol

AddRoundKey	→	InvMixCol
2E 0B AA 67		A9 18 F2 67
85 02 1F FD		64 5D FC 2A
11 04 F2 AE		9A 3E 22 81
BC 94 03 AE		51 E2 68 56

11) Lakukan proses InvShiftRows

InvMixCol	→	InvShiftRows
A9 18 F2 67		A9 18 F2 67
64 5D FC 2A		2A 64 5D FC
9A 3E 22 81		22 81 9A 3E
51 E2 68 56		E2 68 56 51

12) Lakukan proses InvSubBytes.

InvShiftRows	→	InvSubByte
A9 18 F2 67		B7 34 04 0A
2A 64 5D FC		95 8C 8D 55
22 81 9A 3E		94 91 37 D1
E2 68 56 51		3B F7 B9 70

13)  $i = i - 1 = 9 - 1 = 8$ .

14) Lakukan proses AddRoundKey

InvSubBytes	XOR	Subkunci[8]	=	AddRoundkey
B7 34 04 0A	9F 31 17 71		28 05 13 7B	
95 8C 8D 55	25 B7 5C 38		B0 3B D1 6D	
94 91 37 D1	CA FD DD 8E		5E 6C EA 5F	
B7 34 04 0A	9F D6 10 F7		A4 21 A9 87	

15) Lakukan proses InvMixCol

AddRoundKey	→	InvMixCol
28 05 13 7B		9F 00 A0 23
B0 3B D1 6D		78 3B B9 68
5E 6C EA 5F		2D 8F 40 80
A4 21 A9 87		A8 C7 D8 05

16) Lakukan proses InvShiftRows

- InvMixCol → InvShiftRow
- |             |             |
|-------------|-------------|
| 9F 00 A0 23 | 9F 00 A0 23 |
| 78 3B B9 68 | B9 68 78 3B |
| 2D 8F 40 80 | 2D 8F 40 80 |
| A8 C7 D8 05 | D8 05 A8 C7 |
- 17) Lakukan proses InvSubBytes
- |                           |
|---------------------------|
| InvShiftRows → InvSubByte |
| 9F 00 A0 23 → 6E 52 47 32 |
| B9 68 78 3B → DB F7 C1 49 |
| 2D 8F 40 80 → FA 73 72 3A |
| D8 05 A8 C7 → 2D 36 6F 31 |
- 18)  $i = i - 1 = 8 - 1 = 7.$
- 19) Lakukan proses AddRoundKey
- |  |
|--|
| InvShiftRows XOR Subkunci[7] = AddRoundkey |
| 6E 52 47 32 ⊕ 5C AE 26 66 = 32 FC 61 54    |
| DB F7 C1 49 ⊕ C8 92 EB 64 = 13 65 2A 2D    |
| FA 73 72 3A ⊕ 5E 37 20 53 = A4 44 52 69    |
| 2D 36 6F 31 ⊕ AC 49 C6 E7 = 81 7F A9 D6    |
- 20) Lakukan proses InvMixCol
- |                           |
|---------------------------|
| AddRoundKey → InvMixCol   |
| 32 FC 61 54 → 3A C6 7C FE |
| 13 65 2A 2D → 3C 8A A7 C8 |
| A4 44 52 69 → A4 21 34 54 |
| 81 7F A9 D6 → A6 CF 5F A4 |
- 21) Lakukan proses InvShiftRows
- |                           |
|---------------------------|
| InvMixCol → InvShiftRow   |
| 3A C6 7C FE → 3A C6 7C FE |
| 3C 8A A7 C8 → C8 3C 8A A7 |
| A4 21 34 54 → 34 54 A4 21 |
| A6 CF 5F A4 → CF 5F A4 A6 |
- 22) Lakukan proses InvSubBytes
- |                           |
|---------------------------|
| InvShiftRows → InvSubByte |
| 3A C6 7C FE → A2 C7 01 0C |
| C8 3C 8A A7 → B1 6D CF 89 |

	34 54 A4 21	28 FD 1D 7B
	CF 5F A4 A6	5F 84 1D C5
23)	i = i - 1 = 7 - 1 = 6.	
24)	Lakukan proses AddRoundKey	
	InvSubBytes XOR Subkunci[6] = AddRoundkey	
	A2 C7 01 0C 6F F2 88 40 CD 35 89 4C	
	B1 6D CF 89 47 5A 79 8F F6 37 B6 06	
	28 FD 1D 7B A3 69 17 73 8B 94 0A 08	
	5F 84 1D C5 A5 E5 8F 21 FA 61 92 E4	
25)	Lakukan proses InvMixCol	
	AddRoundKey → InvMixCol	
	CD 35 89 4C 94 8D 25 32	
	F6 37 B6 06 10 27 AD 5E	
	8B 94 0A 08 46 D9 F3 21	
	FA 61 92 E4 88 84 DC EB	
26)	Lakukan proses InvShiftRows	
	InvMixCol → InvShiftRow	
	94 8D 25 32 94 8D 25 32	
	10 27 AD 5E 5E 10 27 AD	
	46 D9 F3 21 F3 21 46 D9	
	88 84 DC EB 84 DC EB 88	
27)	Lakukan proses InvSubBytes	
	InvShiftRow → InvSubByte	
	94 8D 25 32 E7 B4 C2 A1	
	5E 10 27 AD 9D 7C 3D 18	
	F3 21 46 D9 7E 7B 98 E5	
	84 DC EB 88 4F 93 3C 97	
28)	i = i - 1 = 6 - 1 = 5.	
29)	Lakukan proses AddRoundKey.	
	InvSubByte XOR Subkunci[5] = AddRoundKey	
	E7 B4 C2 A1 0D 9D 7A C8 EA 29 B8 69	
	9D 7C 3D 18 04 1D 23 F6 99 61 1E EE	
	7E 7B 98 E5 47 CA 7E 64 39 B1 E6 81	
	4F 93 3C 97 4D 40 6A AE 02 D3 56 39	

30) Lakukan proses InvMixCol.

AddRoundKey	→	InvMixCol
EA 29 B8 69		53 8E 2A 2D
99 61 1E EE		A6 45 68 9D
39 B1 E6 81		10 42 58 BB
02 D3 56 39		AD A3 0C 34

31) Lakukan proses InvShiftRows

InvMixCol	→	InvShiftRow
53 8E 2A 2D		53 8E 2A 2D
A6 45 68 9D		9D A6 45 68
10 42 58 BB		58 BB 10 42
AD A3 0C 34		A3 0C 34 AD

32) Lakukan proses InvSubBytes

InvShiftRow	→	InvSubByte
53 8E 2A 2D		50 E6 95 FA
9D A6 45 68		75 C5 68 F7
58 BB 10 42		5E FE 7C F6
A3 0C 34 AD		71 81 28 18

33)  $i = i - 1 = 5 - 1 = 4$ .

34) Lakukan proses AddRoundKey.

InvSubByte	XOR	Subkunci[4]	=	AddRoundkey
50 E6 95 FA		1E 90 E7 B2		4E 76 72 48
75 C5 68 F7		A6 19 3E D5		D3 DC 56 22
5E FE 7C F6		5B 8D B4 1A		05 73 C8 EC
71 81 28 18		7A 0D 2A C4		0B 8C 02 DC

35) Lakukan proses InvMixCol

AddRoundKey	→	InvMixCol
4E 76 72 48		3F E6 3B 86
D3 DC 56 22		4E 09 E5 CC
05 73 C8 EC		49 49 46 96
0B 8C 02 DC		AB F3 76 86

36) Lakukan proses InvShiftRows.

InvMixCol	→	InvShiftRow
3F E6 3B 86		3F E6 3B 86

4E 09 E5 CC	CC 4E 09 E5
49 49 46 96	46 96 49 49
AB F3 76 86	F3 76 86 AB

37) Lakukan proses InvSubBytes.

InvShiftRow	→	InvSubByte
3F E6 3B 86		25 F5 49 DC
CC 4E 09 E5		27 B6 40 2A
46 96 49 49		98 35 A4 A4
F3 76 86 AB		7E 0F DC 0E

38)  $i = i - 1 = 4 - 1 = 3$ .

39) Lakukan proses AddRoundKey.

InvSubByte	XOR	Subkunci[3]	=	AddRoundkey
25 F5 49 DC		FF 8E 77 55		DA 7B 3E 89
27 B6 40 2A		42 BF 27 EB		65 09 67 C1
98 35 A4 A4		73 D6 39 AE		EB E3 9D 0A
7E 0F DC 0E		86 77 27 EE		F8 78 FB E0

40) Lakukan proses InvMixCol.

AddRoundKey	→	InvMixCol
DA 7B 3E 89		C0 29 EA E6
65 09 67 C1		9B F9 12 0D
EB E3 9D 0A		B2 78 37 3A
F8 78 FB E0		45 41 F0 73

41) Lakukan proses InvShiftRows.

InvMixCol	→	InvShiftRow
C0 29 EA E6		C0 29 EA E6
9B F9 12 0D		0D 9B F9 12
B2 78 37 3A		37 3A B2 78
45 41 F0 73		41 F0 73 45

42) Lakukan proses InvSubBytes.

InvShiftRow	→	InvSubByte
C0 29 EA E6		1F 4C BB F5
0D 9B F9 12		F3 E8 69 39
37 3A B2 78		B2 A2 3E C1
41 F0 73 45		F8 17 8F 68

43)  $i = i - 1 = 3 - 1 = 2$ .

44) Lakukan proses AddRoundKey.

InvSubByte	XOR	Subkunci[2]	=	AddRoundkey
1F 4C BB F5		B0 71 F9 22		AF 3D 42 D7
F3 E8 69 39		CA FD 98 CC		39 15 F1 F5
B2 A2 3E C1		AE A5 EF 97		1C 07 D1 56
F8 17 8F 68		15 F1 50 C9		ED E6 DF A1

45) Lakukan proses InvMixCol.

AddRoundkey	→	InvMixCol
AF 3D 42 D7		10 5E 67 86
39 15 F1 F5		7C 0B 2B C4
1C 07 D1 56		6D ED 95 B6
ED E6 DF A1		66 71 64 21

46) Lakukan proses InvShiftRows.

InvMixCol	→	InvShiftRow
10 5E 67 86		10 5E 67 86
7C 0B 2B C4		C4 7C 0B 2B
6D ED 95 B6		95 B6 6D ED
66 71 64 21		71 64 21 66

47) Lakukan proses InvSubBytes.

InvShiftRow	→	InvSubByte
10 5E 67 86		7C 9D 0A DC
C4 7C 0B 2B		88 01 9E 0B
95 B6 6D ED		AD 79 B3 53
71 64 21 66		2C 8C 7B D3

48)  $i = i - 1 = 2 - 1 = 1$ .

49) Lakukan proses AddRoundKey.

InvSubByte	XOR	Subkunci[1]	=	AddRoundkey
7C 9D 0A DC		92 C1 88 DB		EE 5C 82 07
88 01 9E 0B		76 37 65 54		FE 36 FB 5F
AD 79 B3 53		40 0B 4A 78		ED 72 F9 2B
2C 8C 7B D3		AC E4 A1 99		80 68 DA 4A

50) Lakukan proses InvMixCol.

AddRoundkey → InvMixCol

EE 5C 82 07	7A ED 3B ED
FE 36 FB 5F	83 00 C7 09
ED 72 F9 2B	83 23 F4 B3
80 68 DA 4A	07 BE 52 6E

51) Lakukan proses InvShiftRows.

InvMixCol	→	InvShiftRow
7A ED 3B ED		7A ED 3B ED
83 00 C7 09		09 83 00 C7
83 23 F4 B3		F4 B3 83 23
07 BE 52 6E		BE 52 6E 07

52) Lakukan proses InvSubBytes.

InvShiftRow	→	InvSubByte
7A ED 3B ED		BD 53 49 53
09 83 00 C7		40 41 52 31
F4 B3 83 23		BA 4B 41 32
BE 52 6E 07		5A 48 45 38

53)  $i = i - 1 = 1 - 1 = 0$ .

54) Lakukan proses AddRoundKey.

InvSubByte	XOR	Subkunci[0]	=	AddRoundkey
BD 53 49 53		54 53 49 53		E9 00 00 00
40 41 52 31		55 41 52 31		15 00 00 00
BA 4B 41 32		47 4B 41 32		FD 00 00 00
5A 48 45 38		41 48 45 38		1B 00 00 00

Signature = E9 15 FD 1B

h. Konversi citra RGB input ke ruang warna YCbCr.

Citra RGB input pada tabel 3.3 dikonversi ke ruang warna YCbCr. Karena hanya nilai elemen Y yang digunakan dalam perhitungan selanjutnya, maka perhitungan hanya dilakukan untuk elemen Y saja.

Piksel (0, 0):

$$Y = 0.2990 * 88 + 0.5870 * 100 + 0.1140 * 60$$

$$Y = 91.872 \approx 92$$

Piksel (0, 1):

$$Y = 0.2990 * 60 + 0.5870 * 65 + 0.1140 * 20$$

$$Y = 58.375 \approx 58$$

Piksel (0, 2):

$$Y = 0.2990 * 96 + 0.5870 * 168 + 0.1140 * 56$$

$$Y = 40$$

Proses perhitungan akan dilakukan untuk setiap piksel. Hasil yang diperoleh dapat dilihat pada tabel 3.4.

Tabel 3.4 Elemen Nilai Y dari Citra Input pada Tabel 3.3

92	58	40	68	60	38	50	121	52	55	61	66	70	61	64	73
49	82	79	101	113	106	27	62	63	59	55	90	109	85	69	72
58	63	77	69	124	107	74	125	62	59	68	113	144	104	66	73
80	97	74	54	59	71	91	66	63	58	71	122	154	106	70	69
18	34	33	46	64	61	32	37	67	61	68	104	126	88	68	70
149	108	80	106	116	61	73	92	79	65	60	70	77	68	58	75
211	233	159	88	107	158	161	109	85	71	64	59	55	61	65	83
212	104	40	44	71	136	113	66	87	79	69	68	65	76	78	94
124	85	161	160	135	76	138	113	73	89	203	100	165	123	79	67
147	165	103	126	122	136	184	155	66	105	187	23	145	178	90	90
263	162	199	118	192	150	128	124	134	189	47	65	163	88	89	216
280	176	177	192	211	213	153	85	111	45	56	78	80	99	67	146
177	112	94	191	179	114	159	176	109	210	103	154	95	76	208	167
176	100	91	115	204	211	234	170	89	89	156	130	100	56	244	93
139	61	118	109	98	91	137	154	94	221	174	56	201	187	137	218
110	180	162	85	147	137	145	161	65	190	210	84	211	64	180	86

- i. Hitung koefisien DCT dengan nilai  $q = 50$ .

Elemen nilai Y dari citra input berukuran  $16 \times 16$  seperti terlihat pada tabel 3.4 akan dibagi menjadi 4 buah subblok berukuran  $8 \times 8$ , seperti terlihat pada tabel berikut:

Blok 1

92	58	40	68	60	38	50	121
49	82	79	101	113	106	27	62
58	63	77	69	124	107	74	125
80	97	74	54	59	71	91	66
18	34	33	46	64	61	32	37
149	108	80	106	116	61	73	92
211	233	159	88	107	158	161	109
212	104	40	44	71	136	113	66

Blok 2

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

Blok 3

124	85	161	160	135	76	138	113
147	165	103	126	122	136	184	155
263	162	199	118	192	150	128	124
280	176	177	192	211	213	153	85
177	112	94	191	179	114	159	176
176	100	91	115	204	211	234	170
139	61	118	109	98	91	137	154
110	180	162	85	147	137	145	161

Blok 4

73	89	203	100	165	123	79	67
66	105	187	23	145	178	90	90
134	189	47	65	163	88	89	216
111	45	56	78	80	99	67	146
109	210	103	154	95	76	208	167
89	89	156	130	100	56	244	93
94	221	174	56	201	187	137	218
65	190	210	84	211	64	180	86

Setelah itu, nilai 128 mengurangi data matriks citra, sehingga diperoleh:

Blok 1

36	70	88	60	68	90	78	7
79	46	49	27	15	22	101	66
70	65	51	59	4	21	54	3
48	31	54	74	69	57	37	62
110	94	95	82	64	67	96	91
-21	20	48	22	12	67	55	36
-83	-105	-31	40	21	-30	-33	19
-84	24	88	84	57	-8	15	62

Blok 2

76	73	67	62	58	67	64	55
65	69	73	38	19	43	59	56

66	69	60	15	-16	24	62	55
65	70	57	6	-26	22	58	59
61	67	60	24	2	40	60	58
49	63	68	58	51	60	70	53
43	57	64	69	73	67	63	45
41	49	59	60	63	52	50	34

Blok 3

4	43	-33	-32	-7	52	-10	15
-19	-37	25	2	6	-8	-56	-27
-135	-34	-71	10	-64	-22	0	4
-152	-48	-49	-64	-83	-85	-25	43
-49	16	34	-63	-51	14	-31	-48
-48	28	37	13	-76	-83	-106	-42
-11	67	10	19	30	37	-9	-26
18	-52	-34	43	-19	-9	-17	-33

Blok 4

55	39	-75	28	-37	5	49	61
62	23	-59	105	-17	-50	38	38
-6	-61	81	63	-35	40	39	-88
17	83	72	50	48	29	61	-18
19	-82	25	-26	33	52	-80	-39
39	39	-28	-2	28	72	-116	35
34	-93	-46	72	-73	-59	-9	-90
63	-62	-82	44	-83	64	-52	42

Kemudian, dilakukan perkalian matriks T dengan data matriks citra, hasil perkalian dengan matriks T kemudian dikalikan lagi dengan matriks  $T^t$ , sehingga menghasilkan matriks seperti tabel berikut:

Blok 1

125.2231	12.6654	-38.9151	-57.4226	-11.503	35.32662	-4.44511	7.586736
144.4996	91.88461	89.45234	90.6304	-44.7384	35.5756	-14.6629	6.097781
-78.9428	-20.9337	-52.9283	-36.9063	-8.08287	-12.0341	-11.8445	15.05902
37.37209	-14.948	13.02361	51.74496	-33.9559	-0.42611	19.13175	0.307841
145.0382	32.06024	-29.6005	-12.9175	-3.75099	-32.2951	1.708737	5.784931
-109.84	-15.024	-68.3647	27.74453	41.17039	-4.6076	19.15903	-15.161
36.76266	40.31021	-17.8433	0.419904	-72.4224	-10.5781	-5.55937	-16.4202
10.71012	42.27061	7.747972	6.633022	16.75859	19.34757	-13.7857	-3.02715

Blok 2

-162.293	-4.49104	-33.5831	-71.883	-48.2627	-39.3897	2.546769	-62.7779
----------	----------	----------	---------	----------	----------	----------	----------

-23.809	-75.7225	23.92541	2.065701	-10.3372	-29.9468	6.290881	0.086207
133.1419	40.94502	-16.3563	108.3477	58.02272	23.95572	8.768604	-12.3422
61.87257	115.4788	16.12619	12.2714	-47.0118	-55.6787	34.08594	7.662562
-15.7542	-44.8391	55.86159	33.20916	-12.7534	3.387736	12.03768	12.63529
10.21847	-66.9056	32.01399	-13.4062	-14.011	-45.4074	-57.5585	-27.7063
-58.1332	-7.33714	15.26723	-37.1887	17.90765	26.80993	-24.1351	-47.3412
71.62126	-18.1393	-11.315	74.13768	-57.5988	-7.68442	-12.5504	-19.6325

Blok 3

415.4845	30.18842	61.19879	-27.244	-56.1398	20.09456	2.384855	-0.46477
-4.46484	21.85557	60.74996	-10.253	-13.1462	7.085636	8.531416	-4.87875
46.83577	-7.36869	-77.1134	24.55962	28.91249	-9.93002	-5.41227	5.649152
48.5397	-12.0675	-34.0956	14.75881	10.24156	-6.29422	-1.82903	-1.94452
-12.1282	6.554011	13.1966	3.952046	1.875494	-1.74533	2.786686	-3.13608
7.739455	-2.90679	-2.38375	5.940574	2.379162	-0.94148	-4.3034	-1.84838
1.028516	-0.18258	-0.41332	2.413859	0.876468	3.019578	-4.11947	0.661553
0.165732	-0.1433	1.066605	4.193465	1.171441	0.097529	-0.50182	-1.67495

Blok 4

34.8842	43.00978	-10.9733	51.57235	70.64362	101.1744	38.5075	-104.34
104.3346	-14.5258	24.70445	-45.1987	-43.1172	17.08441	-106.31	35.62857
-45.3678	-44.5578	125.8027	1.007746	87.56306	70.70679	-57.6214	-105.568
-23.1784	3.608131	45.24247	32.87712	54.29843	-112.691	6.965589	-8.16681
40.88578	-37.5023	39.55656	26.48598	-17.8797	-54.7489	24.61045	3.231303
12.21023	22.26027	-19.2486	-22.9488	-68.2319	55.16528	-64.1124	31.6722
24.07136	-20.2348	-0.88339	24.97865	-43.7164	-68.9916	82.15349	-31.5063
-114.396	-10.6249	-58.128	17.34362	-17.8599	99.70104	67.41211	48.47362

Kemudian dilanjutkan dengan proses kuantisasi dengan matriks Q50, sehingga menghasilkan tabel berikut:

Blok 1

16	1	-4	-4	0	1	0	0
12	8	6	5	-2	1	0	0
-6	-2	-3	-2	0	0	0	0
3	-1	1	2	-1	0	0	0
8	1	-1	0	0	0	0	0
-5	0	-1	0	1	0	0	-2
1	1	0	0	-1	0	0	0
0	0	0	0	0	0	0	0

Blok 2

-10	0	-3	-4	-2	-1	0	-1
-----	---	----	----	----	----	---	----

-2	-6	2	0	0	-1	0	0
10	3	-1	5	1	0	0	0
4	7	1	0	-1	-1	0	0
-1	-2	2	1	0	0	0	0
0	-2	1	0	0	0	-1	-3
-1	0	0	0	0	0	0	0
1	0	0	1	-1	0	0	0

Blok 3

Blok 4

2	4	-1	3	3	3	1	-2
9	-1	2	-2	-2	0	-2	1
-3	-3	8	0	2	1	-1	-2
-2	0	2	1	1	-1	0	0
2	-2	1	0	0	-1	0	0
1	1	0	0	-1	1	-1	4
0	0	0	0	0	-1	1	0
-2	0	-1	0	0	1	1	0

Terakhir, dilakukan pembacaan bilangan secara *zig-zag scanning* pada setiap blok, sehingga menghasilkan barisan blok berikut :

Blok 4 : 2, 4, 9, -3, -1, -1, 3, 2, -3, -2, 2, 0, 8, -2, 3, 3, -2, 0, 2, -2, 1, 0, 1, 1, 1, 2, 0, 1, -2, -2, 1, 1, 0, 0, 0, -2, 0, 0, 0, 0, -1, -1, 1, -2, 0, -1, -1, 0, -1, 0, 0, 1, 0, 0, 0, -1, -1, 0, 1, 1, 4, 0, 1, 0

- j. Hitung *signature* dengan mengekstraksi ciri dari setiap subblok.

Proses kerja dari perhitungan *signature* dengan mengekstraksi ciri dari setiap subblok dapat dilihat pada rincian perhitungan berikut:

- 1) Tentukan nilai  $c_{\text{norm}}(0,0)_{\text{max}}$ .

$$\text{Subblok 1: } c_{\text{norm}}(0,0) = 16$$

$$\text{Subblok 2: } c_{\text{norm}}(0,0) = -10$$

$$\text{Subblok 3: } c_{\text{norm}}(0,0) = 26$$

$$\text{Subblok 4: } c_{\text{norm}}(0,0) = 2$$

$$c_{\text{norm}}(0,0)_{\text{max}} = 26$$

- 2) Hitung nilai *threshold*  $t_1 \dots t_5$ .

$$t_1 = C_{\text{norm}}(0,0)_{\text{max}}/4 = 26/4 = 6.5$$

$$t_2 = C_{\text{norm}}(0,0)_{\text{max}}/2 = 26/2 = 13$$

$$t_3 = 3 * C_{\text{norm}}(0,0)_{\text{max}}/4 = 3 * 26/4 = 19.5$$

$$t_4 = 0$$

$$t_5 = 3$$

- 3) Hitung nilai *signature* untuk setiap subblok:

Subblok 1:

$$c_{\text{norm}}(0,0) = 16$$

$$c_{\text{norm}}(0,0) \geq t_1 \rightarrow 16 \geq 6.5 \rightarrow s_1 = 1$$

$$c_{\text{norm}}(0,0) \geq t_2 \rightarrow 16 \geq 13 \rightarrow s_2 = 1$$

$$c_{\text{norm}}(0,0) \geq t_3 \rightarrow 16 \geq 19.5 \rightarrow s_3 = 0$$

$$c_{\text{norm}}(0,1) \geq t_4 \rightarrow 1 \geq 0 \rightarrow s_4 = 1$$

$$c_{\text{norm}}(1,0) \geq t_4 \rightarrow 12 \geq 0 \rightarrow s_5 = 1$$

$$c_{\text{norm}}(2,0) \geq t_4 \rightarrow -6 \geq 0 \rightarrow s_6 = 0$$

$$|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 1 \geq 3 \rightarrow s_7 = 0$$

$$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 12 \geq 3 \rightarrow s_8 = 1$$

$$\text{Signature} = 11011001$$

Subblok 2:

$c_{\text{norm}}(0,0) = 21$   
 $c_{\text{norm}}(0,0) \geq t_1 \rightarrow -10 \geq 6.5 \rightarrow s_1 = 0$   
 $c_{\text{norm}}(0,0) \geq t_2 \rightarrow -10 \geq 13 \rightarrow s_2 = 0$   
 $c_{\text{norm}}(0,0) \geq t_3 \rightarrow -10 \geq 19.5 \rightarrow s_3 = 0$   
 $c_{\text{norm}}(0,1) \geq t_4 \rightarrow 0 \geq 0 \rightarrow s_4 = 1$   
 $c_{\text{norm}}(1,0) \geq t_4 \rightarrow -2 \geq 0 \rightarrow s_5 = 0$   
 $c_{\text{norm}}(2,0) \geq t_4 \rightarrow 10 \geq 0 \rightarrow s_6 = 1$   
 $|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 2 \geq 3 \rightarrow s_7 = 0$

$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 10 \geq 3 \rightarrow s_8 = 1$

*Signature = 00010101*

Subblok 3:

$c_{\text{norm}}(0,0) = 26$   
 $c_{\text{norm}}(0,0) \geq t_1 \rightarrow 26 \geq 6.5 \rightarrow s_1 = 1$   
 $c_{\text{norm}}(0,0) \geq t_2 \rightarrow 26 \geq 13 \rightarrow s_2 = 1$   
 $c_{\text{norm}}(0,0) \geq t_3 \rightarrow 26 \geq 19.5 \rightarrow s_3 = 1$   
 $c_{\text{norm}}(0,1) \geq t_4 \rightarrow 3 \geq 0 \rightarrow s_4 = 1$   
 $c_{\text{norm}}(1,0) \geq t_4 \rightarrow 0 \geq 0 \rightarrow s_5 = 1$

$c_{\text{norm}}(2,0) \geq t_4 \rightarrow 3 \geq 0 \rightarrow s_6 = 1$   
 $|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 0 \geq 3 \rightarrow s_7 = 0$

$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 3 \geq 3 \rightarrow s_8 = 1$

*Signature = 11111101*

Subblok 4:

$c_{\text{norm}}(0,0) = 2$   
 $c_{\text{norm}}(0,0) \geq t_1 \rightarrow 2 \geq 6.5 \rightarrow s_1 = 0$   
 $c_{\text{norm}}(0,0) \geq t_2 \rightarrow 2 \geq 13 \rightarrow s_2 = 0$   
 $c_{\text{norm}}(0,0) \geq t_3 \rightarrow 2 \geq 19.5 \rightarrow s_3 = 0$   
 $c_{\text{norm}}(0,1) \geq t_4 \rightarrow 4 \geq 0 \rightarrow s_4 = 1$   
 $c_{\text{norm}}(1,0) \geq t_4 \rightarrow 9 \geq 0 \rightarrow s_5 = 1$   
 $c_{\text{norm}}(2,0) \geq t_4 \rightarrow -3 \geq 0 \rightarrow s_6 = 0$   
 $|c_{\text{norm}}(0,1)| \geq t_5 \rightarrow 4 \geq 3 \rightarrow s_7 = 1$

**UNIVERSITAS  
MIKROSKIL**

$$|c_{\text{norm}}(1,0)| \geq t_5 \rightarrow 9 \geq 3 \rightarrow s_8 = 1$$

Signature = 00011011

4) Output signature = (11011001 00010101 11111101 00011011)<sub>2</sub>  
= (D9 15 FD 1B)<sub>16</sub>

- k. Karena nilai *signature* pada langkah (j) tidak sama dengan *signature* pada langkah (g), maka output pesan bahwa citra TIDAK otentik.

Signature asli (dari dekripsi *blockchain*) = E9 15 FD 1B

Signature dari hasil ekstraksi ciri citra input (tabel 3.4) = D9 15 FD 1B

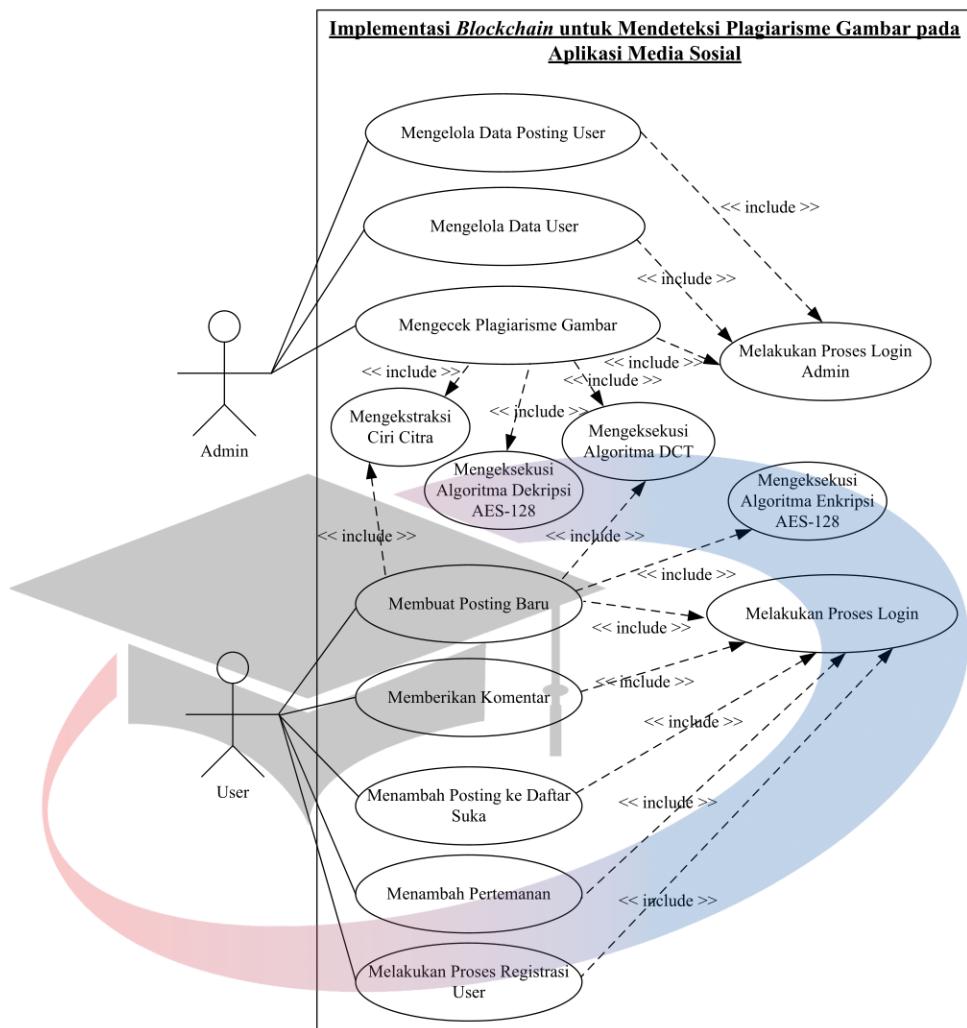
- l. Dari hasil perbandingan, diketahui bahwa perbedaan terletak pada subblok 1, berarti plagiarisme terjadi pada subblok 1.

### 3.1.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem diperlukan untuk memperoleh hasil analisis kebutuhan yang baik dengan performa sistem yang lebih efisien dan efektif. Analisis kebutuhan terhadap sistem yang akan dirancang mencakup analisis kebutuhan fungsional yang digambarkan pada diagram *use case* dan non-fungsional yang digambarkan pada kerangka PIECES.

#### 1. Analisis Kebutuhan Fungsional

Pada bagian ini dijelaskan persyaratan sistem berikut peran masing-masing pengguna dalam aplikasi berdasarkan fungsional sistem. Identifikasi persyaratan sistem dideskripsikan dalam bentuk diagram *use case*. Untuk memodelkan perangkat lunak, maka dapat digunakan *use case diagram* seperti terlihat pada gambar 3.10.

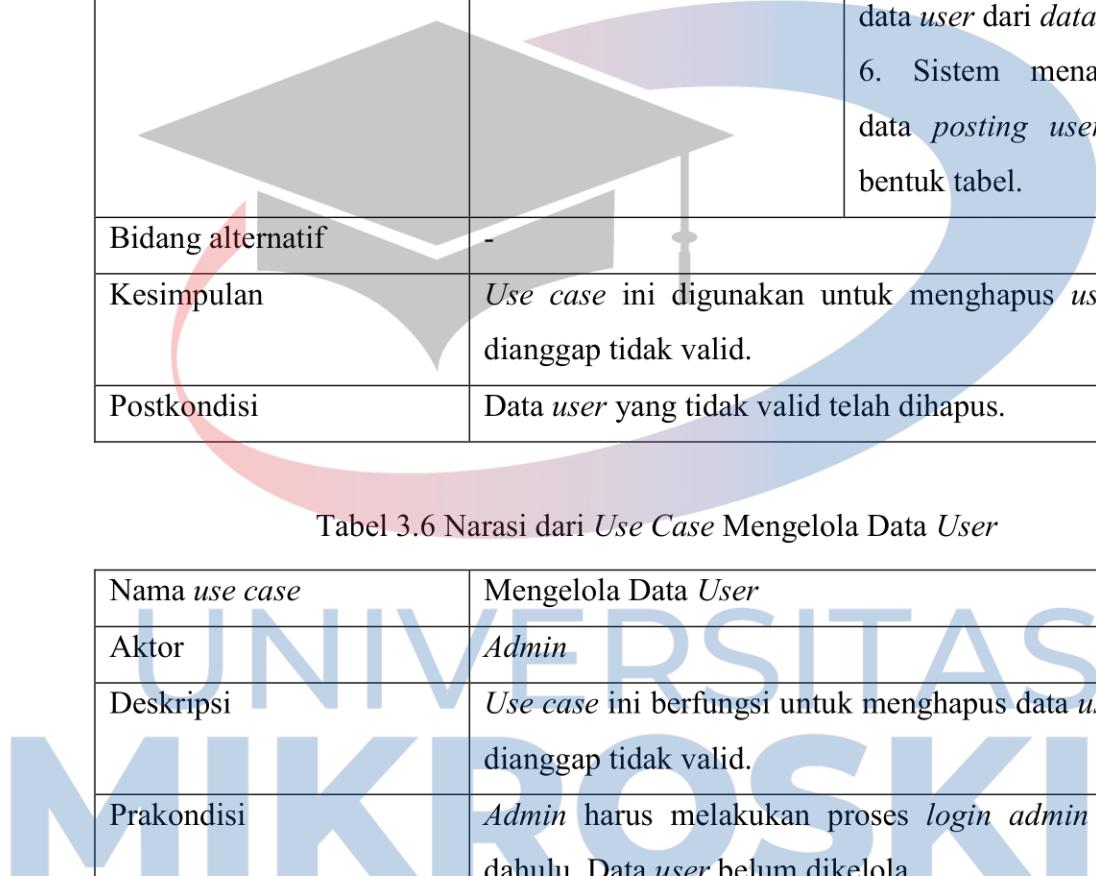


Gambar 3.10 *Use Case* dari Implementasi *Blockchain* untuk Mendeteksi Plagiarisme Gambar pada Aplikasi Media Sosial

Narasi dari *use case* diagram pada Gambar 3.9 antara lain:

Tabel 3.5 Narasi dari *Use Case* Mengelola Data *Posting User*

Nama <i>use case</i>	Mengelola Data <i>Posting User</i>	
Aktor	Admin	
Deskripsi	<i>Use case</i> ini berfungsi untuk menghapus data <i>posting user</i> yang dianggap tidak valid.	
Prakondisi	<i>Admin</i> harus melakukan proses <i>login admin</i> terlebih dahulu. Data <i>posting user</i> belum dikelola.	
Sasaran	<i>Use case</i> ini diawali saat <i>admin</i> ingin menghapus data <i>posting user</i> yang dianggap tidak valid.	
Aksi Aktor		Respons Sistem



Bidang khas suatu event	1. <i>Admin</i> mengklik <i>link Data Posting User.</i>  3. <i>Admin</i> mengklik <i>link Hapus</i> pada Tabel <i>posting User</i> pada halaman <i>home administrator.</i>	2. Sistem menampilkan data <i>posting user</i> pada tabel <i>posting user.</i>  4. Sistem menghapus data <i>posting user.</i>  5. Sistem membaca ulang data <i>user</i> dari <i>database.</i>  6. Sistem menampilkan data <i>posting user</i> dalam bentuk tabel.
Bidang alternatif	-	
Kesimpulan		<i>Use case</i> ini digunakan untuk menghapus <i>user</i> yang dianggap tidak valid.
Postkondisi		Data <i>user</i> yang tidak valid telah dihapus.

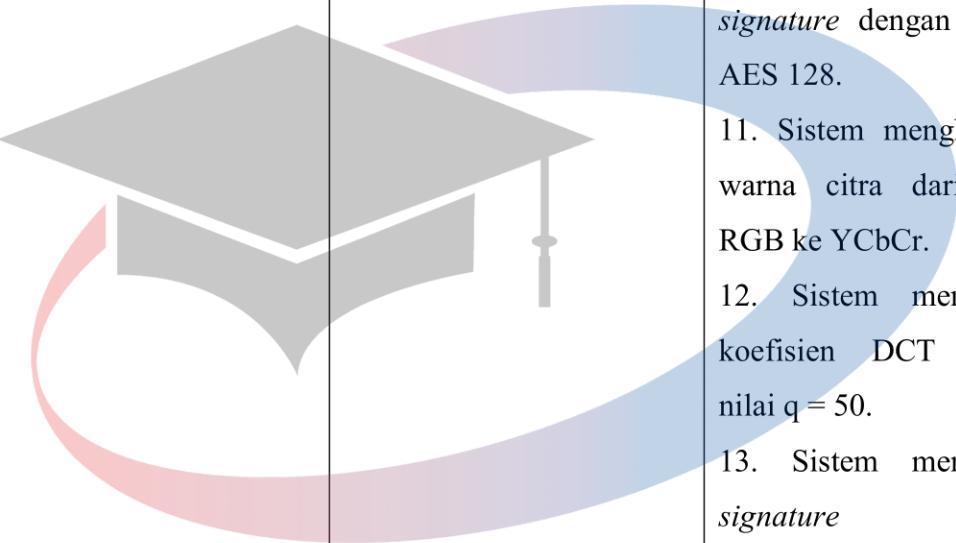
Tabel 3.6 Narasi dari *Use Case* Mengelola Data *User*

Nama <i>use case</i>	Mengelola Data <i>User</i>	
Aktor	<i>Admin</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk menghapus data <i>user</i> yang dianggap tidak valid.	
Prakondisi	<i>Admin</i> harus melakukan proses <i>login admin</i> terlebih dahulu. Data <i>user</i> belum dikelola.	
Sasaran	<i>Use case</i> ini diawali saat <i>admin</i> ingin menghapus data <i>user</i> yang dianggap tidak valid agar tidak dapat menggunakan sistem.	
	Aksi Aktor	Respons Sistem
Bidang khas suatu event	1. <i>Admin</i> mengklik <i>link Hapus</i> pada Tabel <i>User</i> pada halaman <i>home administrator.</i>	2. Sistem menghapus data <i>user</i> yang dipilih oleh staf administrator.  3. Sistem membaca ulang data <i>user</i> dari <i>database.</i>

		4. Sistem menampilkan data <i>user</i> dalam bentuk tabel.
Bidang alternatif	-	
Kesimpulan	<i>Use case</i> ini digunakan untuk menghapus <i>user</i> yang dianggap tidak valid.	
Postkondisi	Data <i>user</i> yang tidak valid telah dihapus.	

Tabel 3.7 Narasi dari *Use Case* Mengecek Plagiarisme Gambar

Nama <i>use case</i>	Mengecek Plagiarisme Gambar	
Aktor	Admin	
Deskripsi	<i>Use case</i> ini berfungsi untuk mengecek keaslian gambar. Proses ini hanya dapat dijalankan pada bagian <i>web</i> saja.	
Prakondisi	<i>Admin</i> harus melakukan proses <i>login admin</i> terlebih dahulu. Gambar belum dicek keasliannya.	
Sasaran	<i>Use case</i> ini diawali saat <i>admin</i> ingin mengecek plagiarisme gambar yang dimasukkan dalam sistem.	
	Aksi Aktor	Respons Sistem
Bidang khas suatu <i>event</i>	1. <i>Admin</i> mengklik link Cek Plagiarisme pada halaman <i>home administrator</i> . 4. <i>Admin</i> memilih gambar yang akan dicek keasliannya. 5. <i>Admin</i> mengklik tombol Cek.	2. Sistem menampilkan halaman cek plagiarisme. 3. Sistem menampilkan daftar gambar yang tersimpan dalam <i>database</i> . 6. Sistem membaca warna piksel citra. 7. Sistem mengekstrak nilai <i>hash</i> dan <i>signature</i>



	<p>terenkripsi dari <i>blockchain</i>.</p> <p>8. Sistem menghitung nilai <i>hash</i> dari <i>signature</i> terenkripsi.</p> <p>9. Sistem memvalidasi nilai <i>hash</i> terekstrak.</p> <p>10. Sistem mendekripsi <i>signature</i> dengan metode AES 128.</p> <p>11. Sistem mengkonversi warna citra dari ruang RGB ke YCbCr.</p> <p>12. Sistem menghitung koefisien DCT dengan nilai <math>q = 50</math>.</p> <p>13. Sistem menghitung <i>signature</i> dengan mengekstraksi ciri dari setiap subblok.</p> <p>14. Sistem menampilkan hasil pendekripsi keaslian citra.</p>
Bidang alternatif	-
Kesimpulan	<i>Use case</i> ini digunakan untuk mengecek plagiarisme gambar.
Postkondisi	Informasi keaslian gambar telah diperoleh.

Tabel 3.8 Narasi *Use Case* Melakukan Halaman *Login Admin*

Nama <i>use case</i>	Melakukan Halaman <i>Login Admin</i>
Aktor	<i>Admin</i>

Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan proses <i>login</i> ke dalam sistem administrator. Proses ini hanya dapat dijalankan pada bagian <i>web</i> .	
Prakondisi	<i>Admin</i> belum dapat mengakses halaman <i>dashboard</i> untuk administrator.	
Sasaran	<i>Use case</i> ini diawali saat <i>admin</i> ingin melakukan proses <i>login</i> ke dalam sistem agar dapat mengakses sistem yang dibuat.	
	Aksi Aktor	Respons Sistem
Bidang khas suatu <i>event</i>	<p>1. <i>Admin</i> mengklik link ‘<i>Login</i>’ yang terdapat pada halaman <i>home</i> sebelum <i>login</i>.</p> <p>3. <i>Admin</i> memasukkan data nama <i>user</i> dan <i>password</i>.</p> <p>4. <i>Admin</i> mengklik tombol <i>Login</i></p>	<p>2. Sistem menampilkan halaman <i>Login</i>.</p> <p>5. Sistem memvalidasi data <i>user</i> yang dimasukkan.</p> <p>6. Sistem menampilkan halaman <i>home</i> setelah <i>login</i>.</p>
Bidang alternatif	Alt. Langkah 6. Sistem menampilkan pesan kesalahan bahwa data <i>user</i> yang dimasukkan tidak valid.	
Kesimpulan	<i>Use case</i> ini untuk melakukan proses <i>login</i> ke dalam sistem	
Postkondisi	<i>User</i> dapat mengakses halaman <i>home</i> untuk administrator.	

Tabel 3.9 Narasi *Use Case* Melakukan Proses *Login User*

Nama <i>use case</i>	Melakukan Proses <i>Login User</i>
Aktor	<i>User</i>
Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan proses <i>login</i> ke dalam sistem. Proses ini hanya dapat dijalankan pada bagian <i>mobile</i> .
Prakondisi	<i>User</i> belum dapat mengakses halaman <i>home</i> setelah <i>login</i> .

Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin melakukan proses <i>login</i> ke dalam sistem agar dapat mengakses sistem yang dibuat.	
	Aksi Aktor	Respons Sistem
Bidang khas suatu <i>event</i>	1. <i>User</i> menjalankan aplikasi <i>mobile</i> . 3. <i>User</i> memasukkan data nama <i>user</i> dan <i>password</i> . 4. <i>User</i> mengklik tombol <i>Login</i>	2. Sistem menampilkan halaman <i>Login</i> .  5. Sistem memvalidasi data <i>user</i> yang dimasukkan. 6. Sistem menampilkan halaman <i>home</i> setelah <i>login</i> .
Bidang alternatif	Alt. Langkah 6. Sistem menampilkan pesan kesalahan bahwa data <i>user</i> yang dimasukkan tidak valid.	
Kesimpulan	<i>Use case</i> ini untuk melakukan proses <i>login</i> ke dalam sistem	
Postkondisi	<i>User</i> dapat mengakses halaman <i>home</i> setelah <i>login</i> .	

Tabel 3.10 Narasi dari *Use Case* Membuat *Posting* Baru

Nama <i>use case</i>	Membuat <i>Posting</i> Baru	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk membuat <i>posting</i> baru. Proses ini hanya dapat dijalankan pada bagian <i>mobile</i> .	
Prakondisi	<i>User</i> harus melakukan proses <i>login</i> <i>user</i> terlebih dahulu. <i>User</i> belum membuat <i>posting</i> baru	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin menambah <i>posting</i> baru	
	Aksi Aktor	Respons Sistem
Bidang khas suatu <i>event</i>	1. <i>User</i> mengklik link <i>Buat Posting</i> . 3. <i>User</i> memasukkan data <i>posting</i> .	2. Sistem menampilkan halaman <i>Buat Posting</i> .

	<p>4. <i>User</i> mengklik tombol Upload Gambar.</p> 	<p>5. Sistem membaca warna piksel citra.</p> <p>6. Sistem mengkonversi warna citra dari ruang RGB ke YCbCr.</p> <p>7. Sistem menghitung koefisien DCT dengan nilai <math>q = 50</math>.</p> <p>8. Sistem menghitung <i>signature</i> dengan mengekstraksi ciri dari setiap subblok.</p> <p>9. Sistem mengenkripsi <i>signature</i> dengan metode AES.</p> <p>10. Sistem mengambil data <i>blockchain</i> dari database.</p> <p>11. Sistem mengekstrak nilai <i>hash</i> dan <i>signature</i> terenkripsi dari <i>blockchain</i>.</p> <p>12. Sistem membandingkan <i>signature</i> gambar dengan <i>signature</i> yang ada di <i>database</i>.</p> <p>12. Sistem menentukan hasil plagiat atau tidak.</p> <p>14. Sistem mengirimkan respon hasil penentuan.</p> <p>16. Sistem akan menyimpan data <i>posting</i> baru ke tabel.</p>
Bidang alternatif	-	

Kesimpulan	<i>Use case ini digunakan untuk menambah posting baru.</i>
Postkondisi	<i>Posting baru telah ditambahkan.</i>

Tabel 3.11 Narasi dari *Use Case Memberikan Komentar*

Nama <i>use case</i>	Memberikan Komentar	
	Aksi Aktor	Respons Sistem
Aktor	<i>User</i>	
Deskripsi	<i>Use case ini berfungsi untuk menambah komentar. Proses ini hanya dapat dijalankan pada bagian mobile.</i>	
Prakondisi	<i>User harus melakukan proses <i>login user</i> terlebih dahulu.</i> <i>User belum menambah komentar</i>	
Sasaran	<i>Use case ini diawali saat <i>user</i> ingin menambah komentar</i>	
Bidang khas suatu <i>event</i>	1. <i>User</i> mengklik tombol Tambah Komentar pada halaman <i>Posting</i> . 3. <i>User</i> memasukkan data komentar. 4. <i>User</i> mengklik tombol Simpan.	2. Sistem menampilkan halaman Tambah Komentar. 5. Sistem akan menyimpan data komentar ke tabel.
Bidang alternatif	-	
Kesimpulan	<i>Use case ini digunakan untuk menambah komentar pada <i>posting user</i>.</i>	
Postkondisi	<i>Komentar telah ditambahkan.</i>	

Tabel 3.12 Narasi dari *Use Case Menambah Posting ke Daftar Suka*

Nama <i>use case</i>	Menambah <i>Posting</i> ke Daftar Suka
Aktor	<i>User</i>
Deskripsi	<i>Use case ini berfungsi untuk menambah <i>posting</i> ke daftar suka. Proses ini hanya dapat dijalankan pada bagian mobile.</i>
Prakondisi	<i>User</i> harus melakukan proses <i>login user</i> terlebih dahulu. <i>User</i> belum menambah <i>posting</i> ke daftar suka

Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin menambah <i>posting</i> ke daftar suka	
	Aksi Aktor	Respons Sistem
Bidang khas suatu <i>event</i>	1. <i>User</i> mengklik tombol Suka pada halaman <i>Posting</i> .	2. Sistem akan menyimpan data suka ke tabel.
Bidang alternatif	-	
Kesimpulan	<i>Use case</i> ini digunakan untuk menambah <i>posting</i> ke daftar suka.	
Postkondisi	<i>Posting</i> telah ditambahkan ke daftar suka.	

Tabel 3.13 Narasi dari *Use Case* Menambah Pertemanan

Nama <i>use case</i>	Menambah Pertemanan	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk melakukan pertemanan. Proses ini hanya dapat dijalankan pada bagian <i>mobile</i> .	
Prakondisi	<i>User</i> harus melakukan proses <i>login user</i> terlebih dahulu. <i>User</i> belum menambahkan pertemanannya	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin menambah pertemanan	
	Aksi Aktor	Respons Sistem
Bidang khas suatu <i>event</i>	1. <i>User</i> mengklik link Tambah Teman. 3. <i>User</i> memilih nama <i>user</i> yang akan ditambah pertemanannya. 4. <i>User</i> mengklik tombol Berteman.	2. Sistem menampilkan halaman Pertemanan. 5. Sistem akan menyimpan data pertemanan dari <i>user</i> .
Bidang alternatif	-	
Kesimpulan	<i>Use case</i> ini digunakan untuk menambah pertemanan <i>user</i> .	

Postkondisi	Pertemanan <i>user</i> telah ditambahkan.
-------------	---

Tabel 3.14 Narasi dari *Use Case* Melakukan Proses Registrasi *User*

Nama <i>use case</i>	Melakukan Proses Registrasi <i>User</i>	
Aktor	<i>User</i>	
Deskripsi	<i>Use case</i> ini berfungsi untuk mendaftarkan pemakai atau <i>user</i> baru yang akan menggunakan sistem. Proses ini hanya dapat dijalankan pada bagian <i>mobile</i> .	
Prakondisi	Data <i>user</i> baru belum tersimpan dalam <i>database</i>	
Sasaran	<i>Use case</i> ini diawali saat <i>user</i> ingin menyimpan data <i>user</i> baru ke dalam <i>database</i>	
	Aksi Aktor	Respons Sistem
Bidang khas suatu <i>event</i>	1. <i>User</i> mengklik tombol Registrasi pada halaman <i>login</i> . 3. <i>User</i> memasukkan data pemakai ( <i>user</i> ) baru. 4. <i>User</i> mengklik tombol Simpan.	2. Sistem menampilkan halaman registrasi. 5. Sistem mengecek data <i>input</i> . 6. Sistem menyimpan data <i>input</i> ke dalam <i>database</i> .
Bidang alternatif	Alt. Lgkh 6. Sistem menampilkan pesan kesalahan bahwa data <i>input</i> tidak lengkap	
Kesimpulan	<i>Use case</i> ini digunakan untuk melakukan pendaftaran <i>user</i> baru yang akan menggunakan sistem	
Postkondisi	Data <i>user</i> baru telah tersimpan ke dalam <i>database</i>	

## 2. Analisis Kebutuhan Non Fungsional

Persyaratan dalam kebutuhan non-fungsional dari sistem rekomendasi digambarkan dalam bentuk PIECES (*Performance, Information, Economic, Control, Efficiency* dan *Service*) seperti terlihat pada tabel 3.19.

Tabel 3.15 Analisis Kebutuhan Non-Fungsional dengan PIECES

Aspek	Sistem
-------	--------

<i>Performance</i>	Sistem menyediakan fitur pendekripsi plagiarisme agar <i>administrator</i> dapat melakukan pengujian untuk gambar yang berbeda.
<i>Information</i>	Sistem akan memberikan informasi mengenai plagiarisme gambar dengan menggunakan algoritme <i>blockchain</i> .
<i>Economic</i>	Tidak harus melakukan pembayaran untuk menjadi <i>member</i> dan menggunakan sistem.
<i>Control</i>	Sistem dapat mengenali antara <i>user</i> dan <i>admin</i> dengan hak akses yang berbeda terhadap sistem.
<i>Efficiency</i>	Efektif dan efisien karena sistem berbasis <i>mobile</i> sehingga dapat di akses dimana saja.
<i>Service</i>	Sistem mudah digunakan karena fitur yang tersedia di dalam sistem dirancang sederhana dan sistem mampu mendekripsi plagiarisme.

### 3.2. Perancangan

Didalam membangun sebuah sistem diperlukan adanya suatu perancangan untuk mendeskripsikan tiap fitur yang ada pada sistem. Adapun perancangan terdiri dari perancangan tampilan sistem dan perancangan basis data.

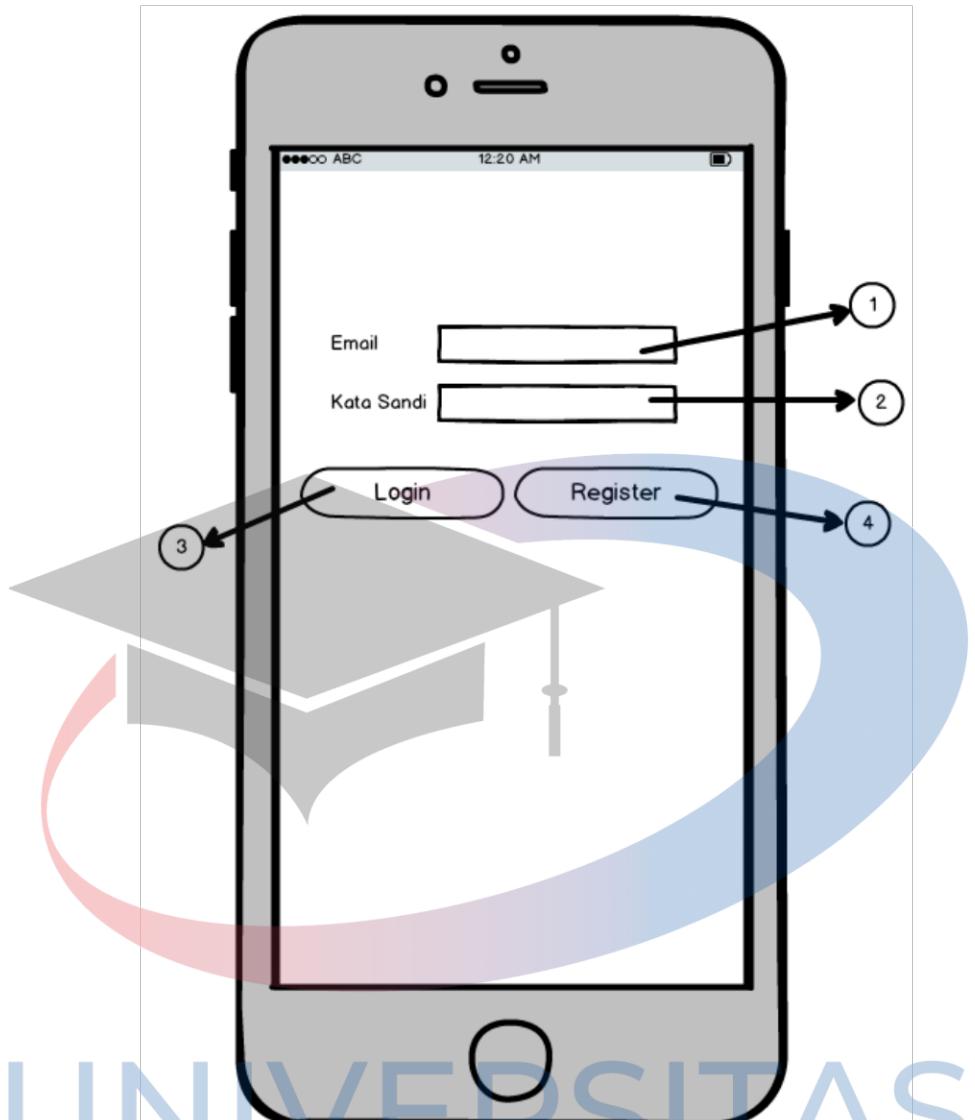
#### 3.2.1. Perancangan Tampilan

Perancangan tampilan *interface* Implementasi *Blockchain* untuk Mendekripsi Plagiarisme Gambar pada Aplikasi Media Sosial ini dibagi ke dalam dua bagian, yaitu perancangan tampilan antarmuka untuk *user* (berbasis *mobile*) dan tampilan antarmuka untuk *administrator* (berbasis *web*).

##### 1. Perancangan tampilan antarmuka untuk *user* (*mobile*)

###### a. Halaman *Login*

Halaman ini merupakan tampilan jika *user* ingin *login* ke sistem. Rancangan tampilan halaman ‘*Login*’ dapat dilihat pada gambar 3.11.



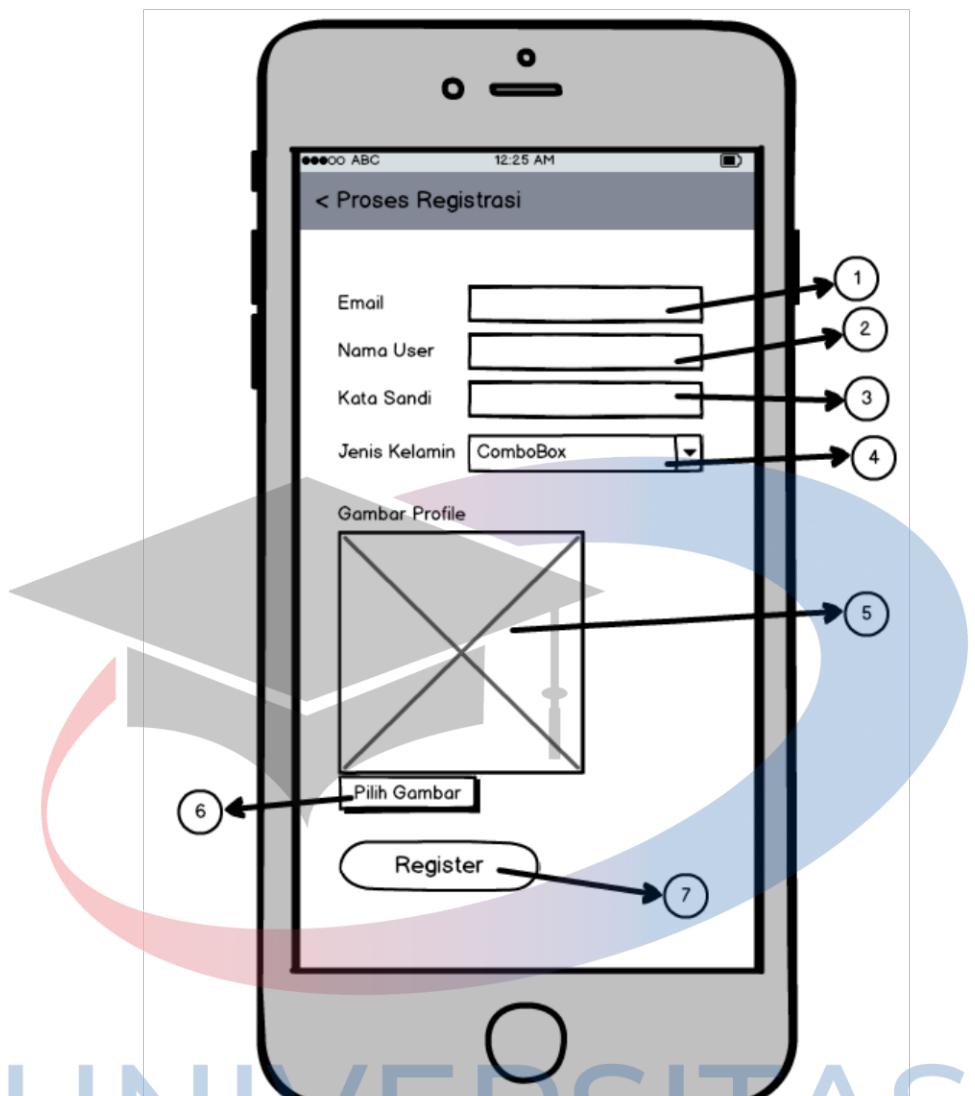
Gambar 3.11 Rancangan Halaman *Login*.

Keterangan :

- 1) *Text input* ‘Email’ berfungsi sebagai tempat pengisian data *email user*.
- 2) *Text input* ‘Kata Sandi’ berfungsi sebagai tempat pengisian data kata sandi *user*.
- 3) *Button login* berfungsi untuk memulai proses *login*.
- 4) *Button register* berfungsi untuk membuat *user* baru.

b. *Halaman Register*

Halaman ini merupakan tampilan jika *user* ingin mendaftar agar dapat menggunakan aplikasi. Rancangan tampilan halaman ‘*Register*’ dapat dilihat pada gambar 3.12.



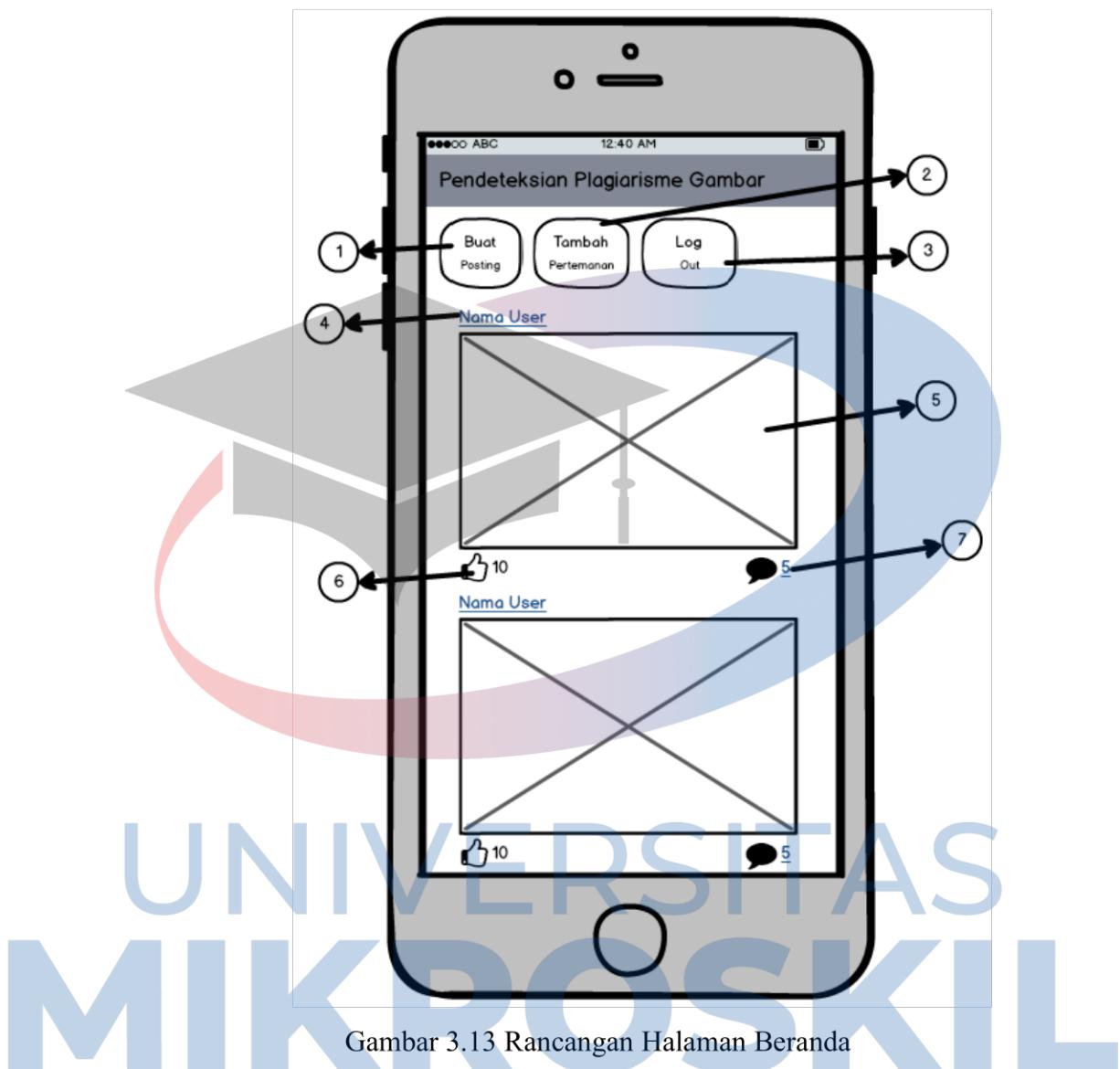
Gambar 3.12 Rancangan Halaman Register

Keterangan :

- 1) *Text input ‘Email’* berfungsi sebagai tempat pengisian data *email user*.
- 2) *Text input ‘Nama’* berfungsi sebagai tempat pengisian data *nama user*.
- 3) *Text input ‘Kata Sandi’* berfungsi sebagai tempat pengisian data *kata sandi user*.
- 4) *ComboBox ‘Jenis Kelamin’* berfungsi untuk menyediakan pilihan jenis kelamin *user*, yaitu Laki-laki atau Perempuan.
- 5) *Picturebox*, berfungsi untuk menampilkan gambar profil *user*.
- 6) *Button ‘Pilih Gambar’*, berfungsi untuk memilih gambar yang diinginkan *user*.
- 7) *Button ‘Registrasi’*, berfungsi untuk menyimpan data *user* ke dalam *database*.

c. Halaman Beranda

Halaman ini merupakan *form* awal dari sistem yang berfungsi untuk menghubungkan *form* lainnya yang ada pada sistem. Rancangan tampilan dari halaman ‘Beranda’ dapat dilihat pada gambar 3.13.

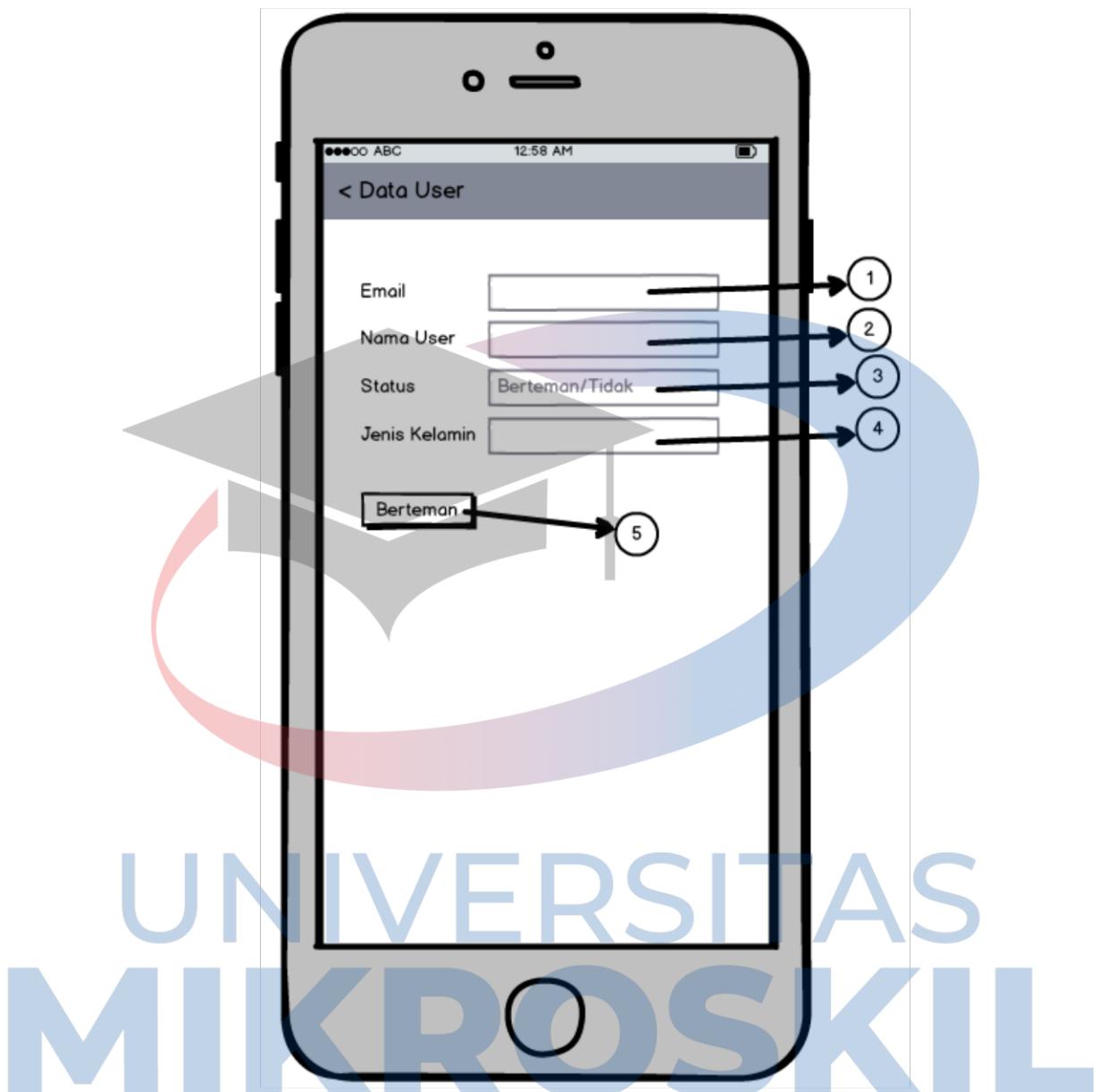


Keterangan:

- 1) Button ‘Buat Posting’ menampilkan halaman Buat Posting.
- 2) Button ‘Tambah Pertemanan’ menampilkan halaman Tambah Teman.
- 3) Button ‘Log Out’ berfungsi untuk keluar dari aplikasi.
- 4) Link ‘Nama User’ menampilkan halaman Detail User.
- 5) Picturebox untuk menampilkan gambar yang di-posting oleh user yang bersangkutan.
- 6) Link ‘Suka’ untuk menambah posting ke daftar suka.
- 7) Link ‘Komentar’ untuk membuka halaman Tambah Komentar.

d. Halaman Detail User

Halaman ini berfungsi untuk menampilkan rincian data pribadi dari *user* yang dipilih. Rancangan tampilan halaman ‘Detail User’ dapat dilihat pada gambar 3.14.



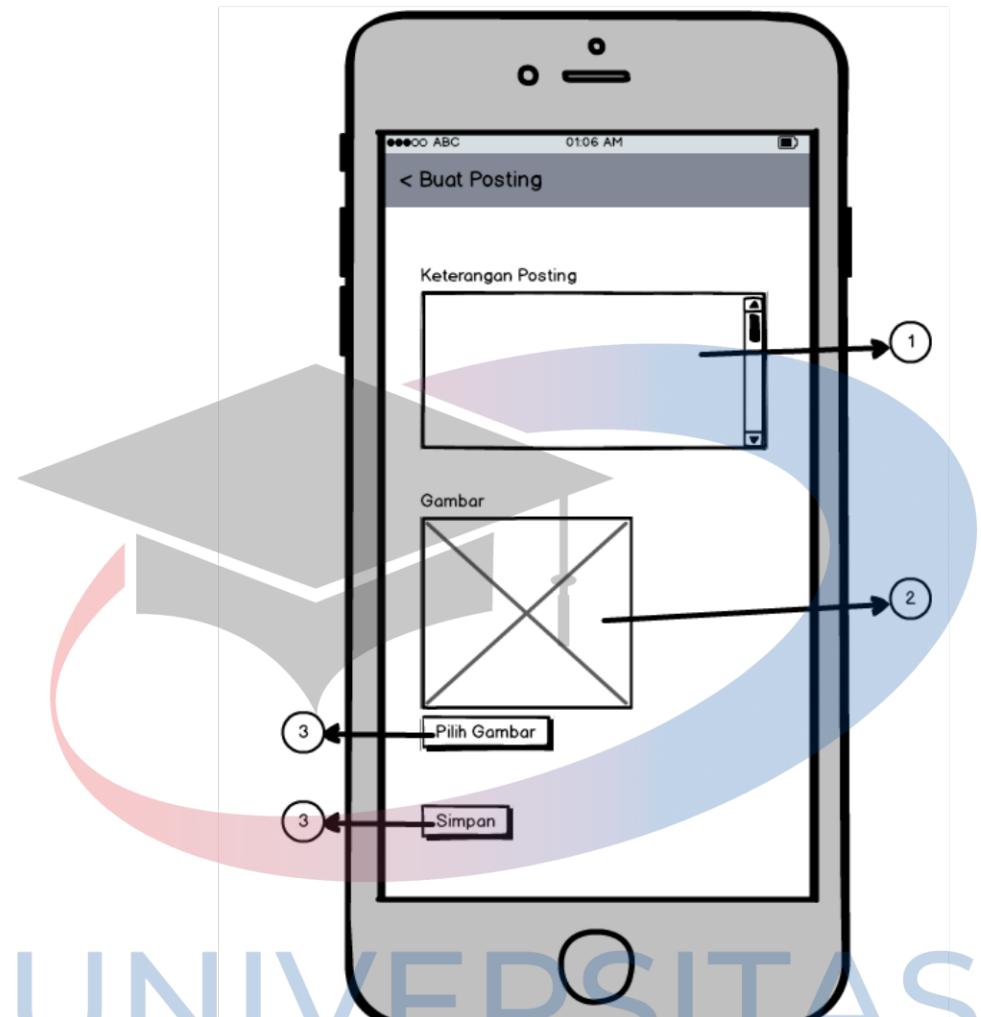
Gambar 3.14 Rancangan Halaman Detail *User*

Keterangan :

- 1) *Label Email* berfungsi untuk menampilkan informasi *email user*.
- 2) *Label Nama User* berfungsi untuk menampilkan informasi nama *user*.
- 3) *Label Status* berfungsi untuk menampilkan informasi status pertemanan *user*.
- 4) *Label Jenis Kelamin* berfungsi untuk menampilkan informasi jenis kelamin *user*.
- 5) *Button Berteman* berfungsi untuk menambah pertemanan *user*.

#### e. Halaman Buat *Posting*

Halaman ini merupakan tampilan jika *user* yang sudah *login* ke sistem mengklik *link* Buat *Posting*. Rancangan tampilan halaman ‘Buat *Posting*’ dapat dilihat pada gambar 3.15.



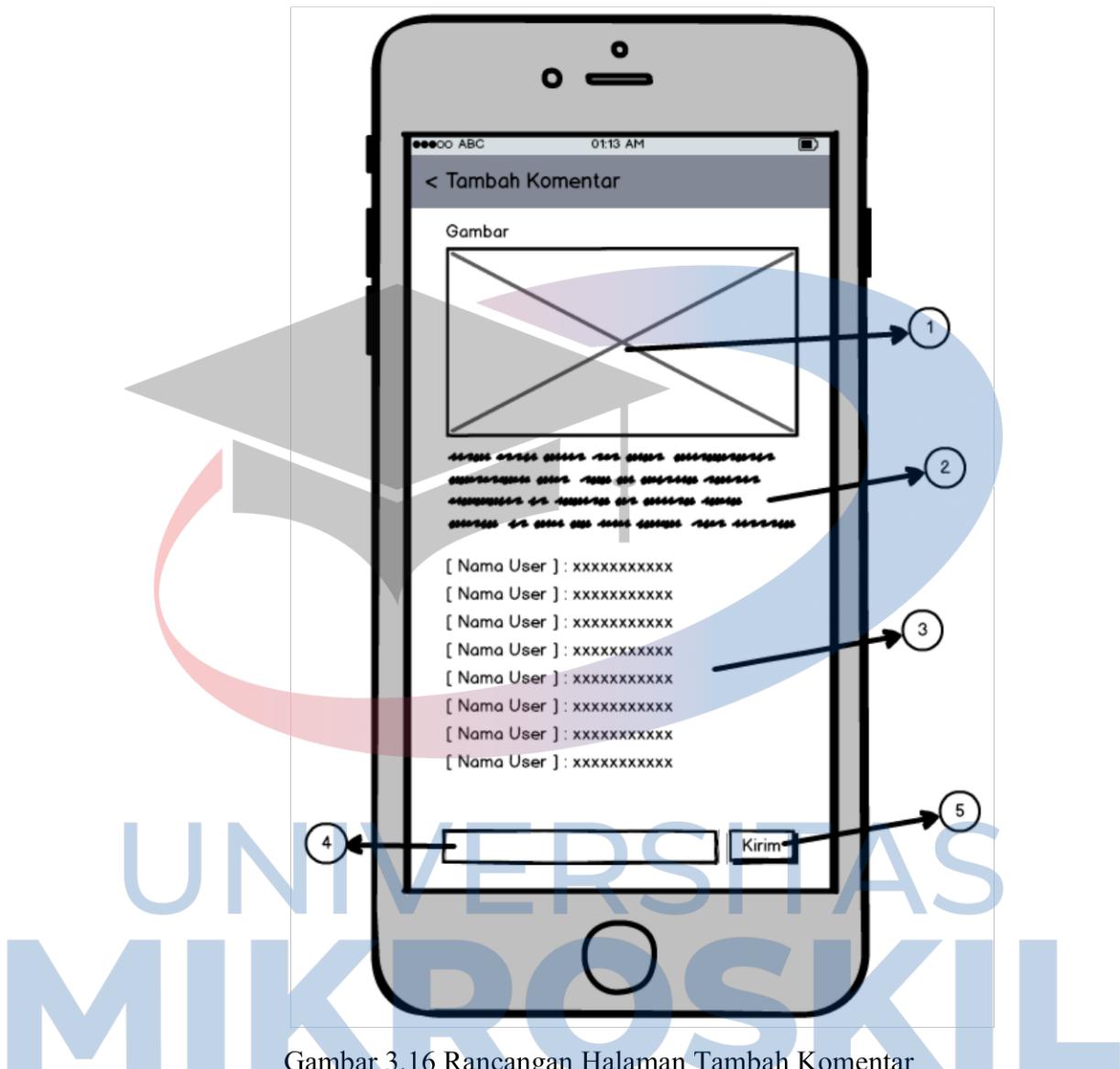
Gambar 3.15 Rancangan Halaman Buat *Posting*

Keterangan :

- 1) *Text area* berfungsi untuk pengisian keterangan *posting*.
- 2) *Picture box* berfungsi untuk menampilkan gambar yang dipilih.
- 3) *Button* ‘Pilih Gambar’ berfungsi untuk memilih gambar yang akan dimasukkan ke dalam *posting*.
- 4) *Button Simpan* berfungsi untuk menyimpan data *posting* yang dimasukkan ke dalam *database*. Namun, sebelum disimpan, data gambar yang dimasukkan akan dicek plagiarisme terlebih dahulu. Apabila gambar tidak autentik, maka data *posting* tidak akan dapat disimpan.

#### f. Halaman Tambah Komentar

Halaman ini merupakan tampilan jika *user* yang sudah *login* ke sistem mengklik *link* Tambah Komentar. Rancangan tampilan halaman ‘Tambah Komentar’ dapat dilihat pada gambar 3.16.

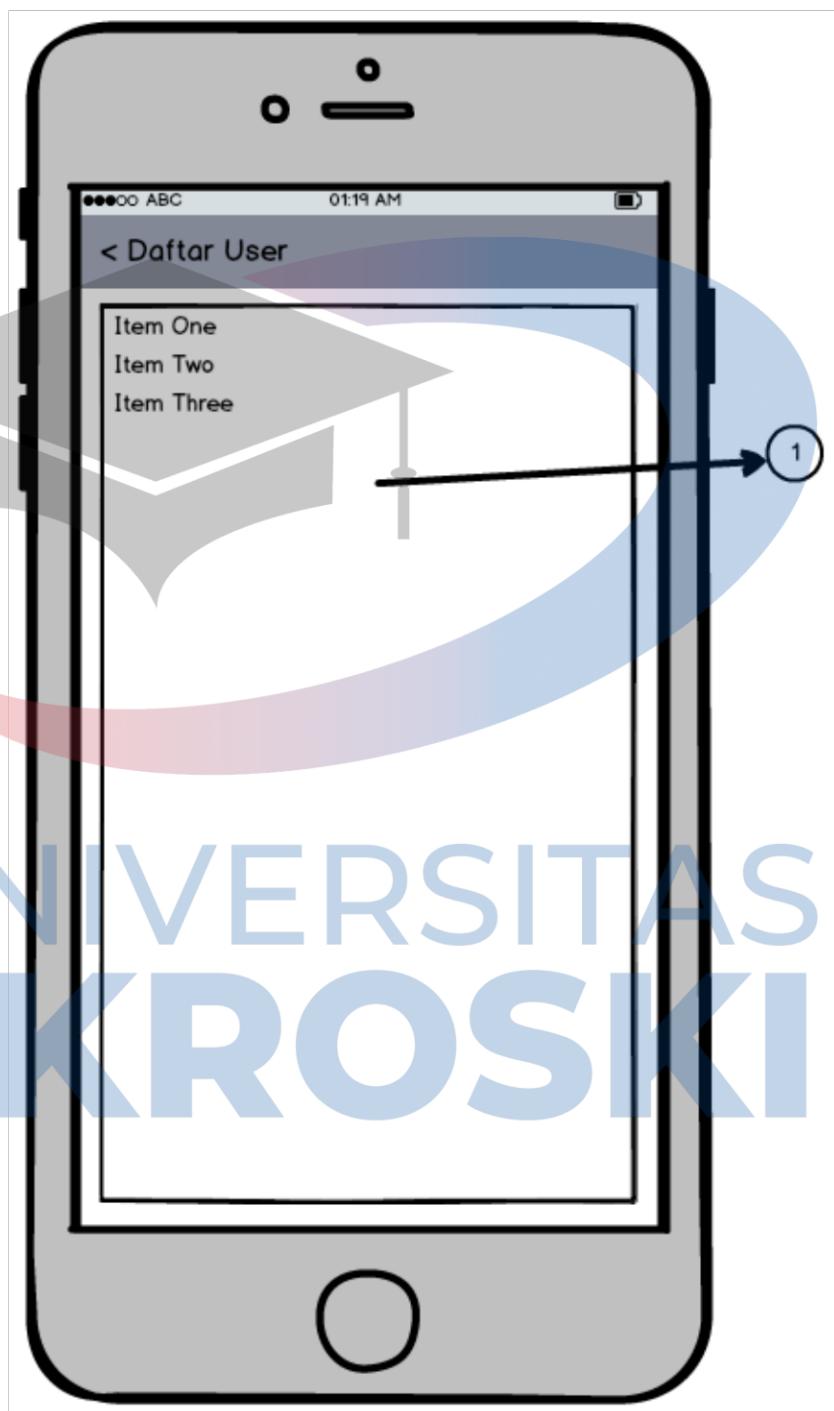


Keterangan :

- 1) *Picture box* berfungsi untuk menampilkan gambar yang di-posting.
- 2) *Label* berfungsi untuk menampilkan informasi yang di-posting.
- 3) *Label* berfungsi untuk menampilkan informasi mengenai komentar dari *user* lain.
- 4) *Textbox Komentar*, untuk memasukkan komentar yang akan dikirimkan.
- 5) Tombol Kirim, untuk menyimpan komentar yang dimasukkan oleh *user* ke dalam database.

g. Halaman Daftar *User*

Halaman ini merupakan tampilan jika *user* yang sudah *login* ke sistem ingin melihat daftar *user* yang tersimpan dalam sistem. Rancangan tampilan halaman ‘Daftar *User*’ dapat dilihat pada gambar 3.17.



Gambar 3.17 Rancangan Halaman Daftar *User*

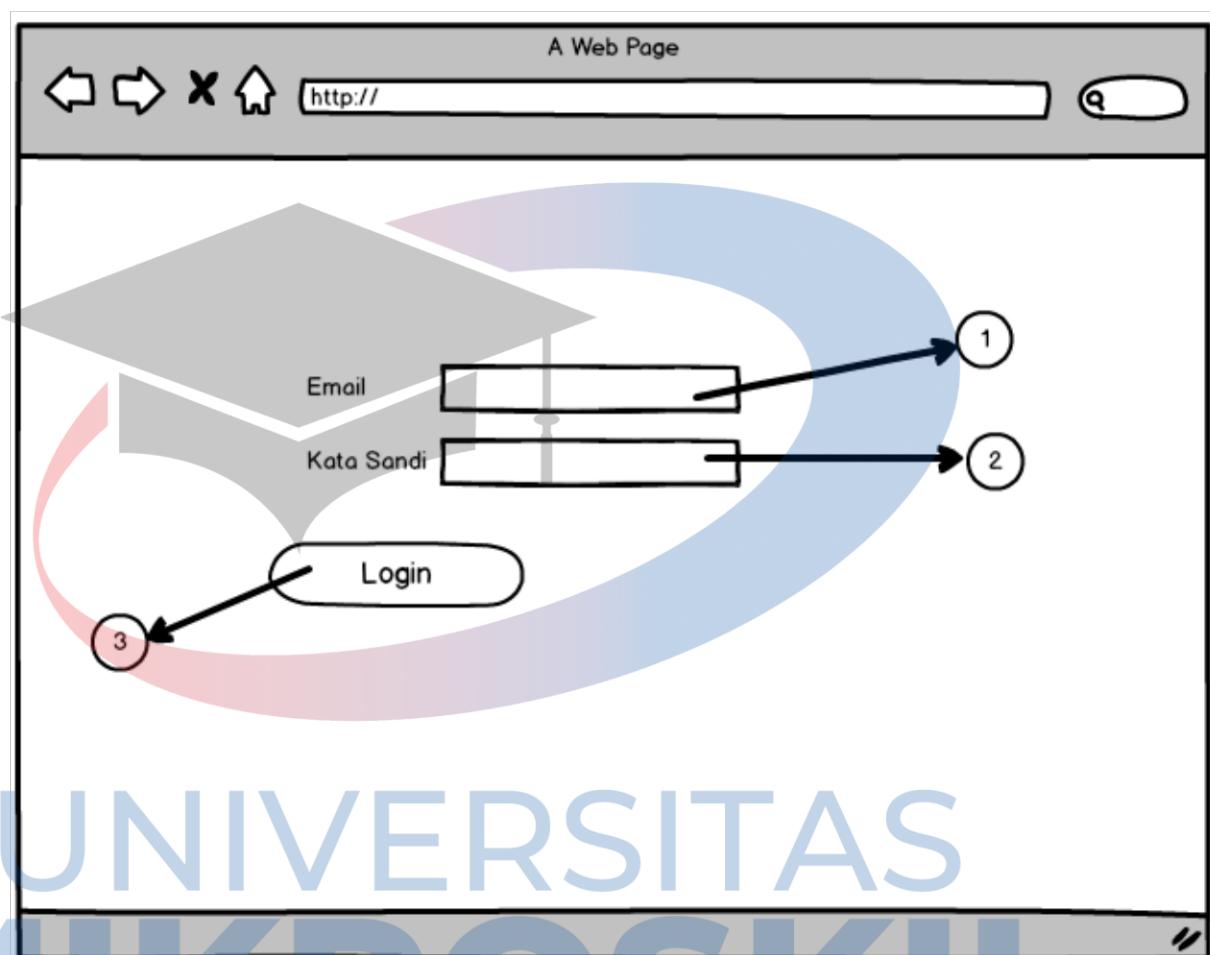
Keterangan :

- 1) *Listview User*, untuk menampilkan daftar *user* yang tersimpan dalam *database*.

2. Perancangan tampilan antarmuka untuk administrator (*website*)

a. Halaman *Login Administrator*

*Form* ini merupakan halaman *login* administrator, sebelum *login* administrator tidak dapat melakukan aktifitas di dalam sistem. Rancangan tampilan ‘Halaman *Login Administrator*’ dapat dilihat pada Gambar 3.18.



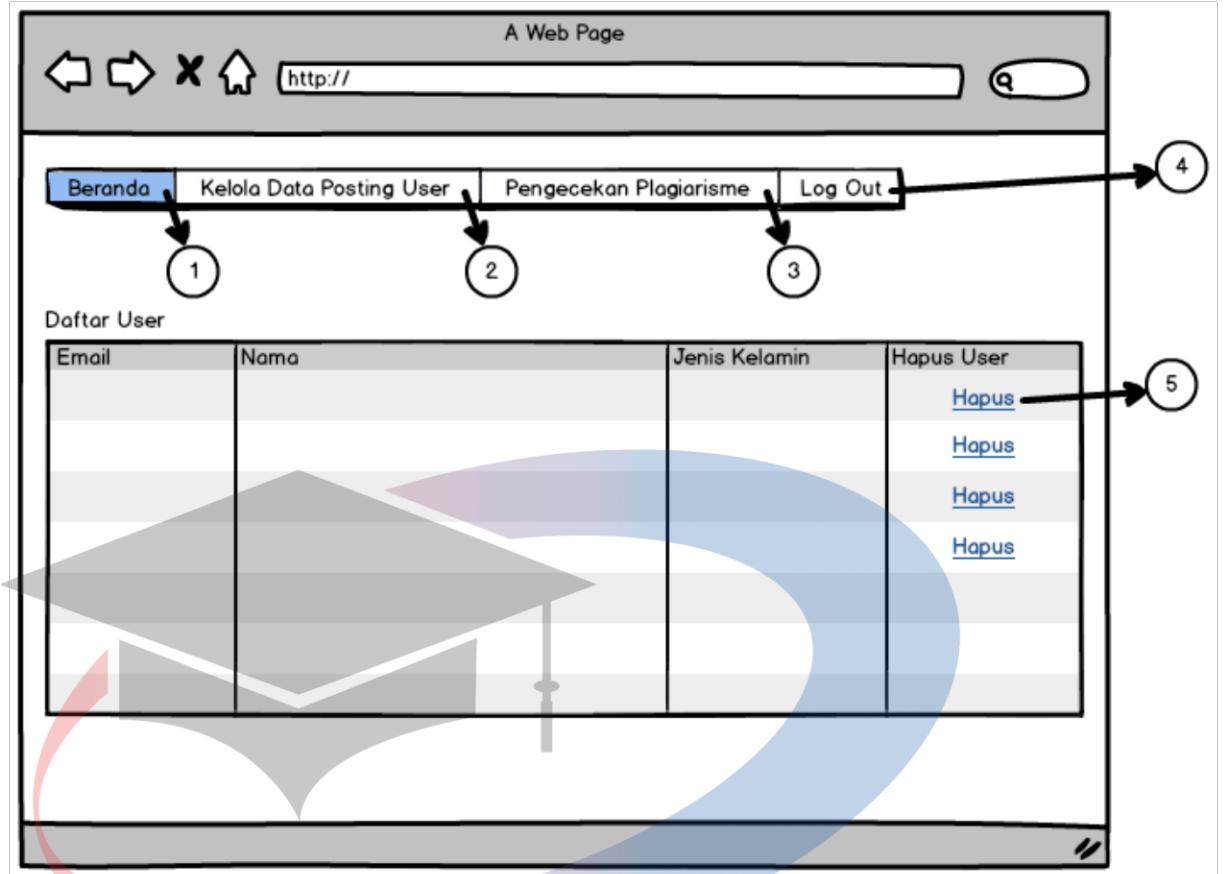
Gambar 3.18 Rancangan Halaman *Login Administrator*.

Keterangan :

- 1) *Textbox User ID*, berfungsi sebagai tempat pengisian *user ID* dari administrator.
- 2) *Textbox Password*, berfungsi sebagai tempat pengisian *password* dari administrator.
- 3) *Button login* untuk memulai proses *login*.

b. Halaman *Home Administrator*

*Form* ini merupakan halaman untuk menampilkan informasi *user*, *posting* dan pengecekan plagiarisme. Rancangan tampilan halaman ‘*Home Administrator*’ dapat dilihat pada gambar 3.19.



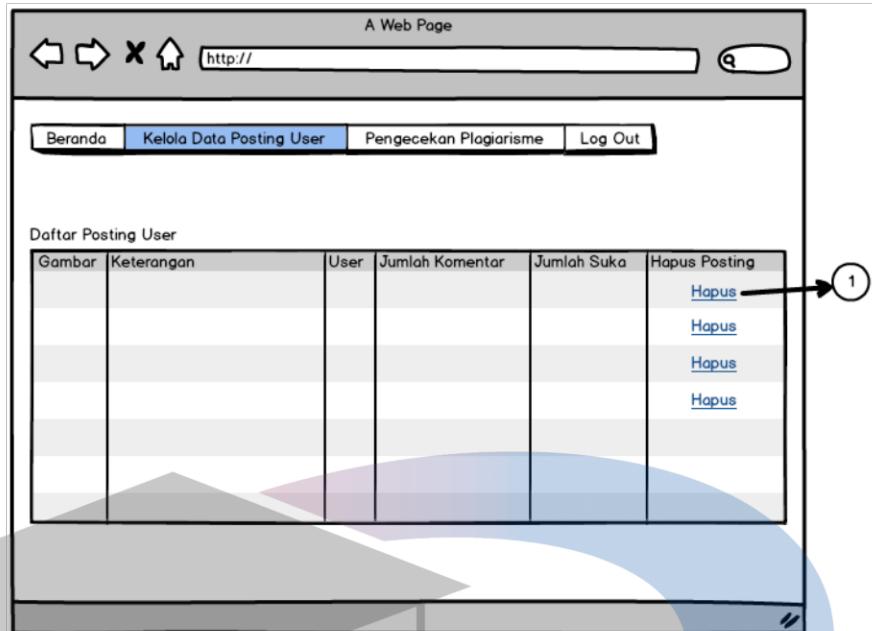
Gambar 3.19 Rancangan Halaman *Home Administrator*.

Keterangan :

- 1) Link menu *home administrator*
- 2) Link menu *kelola data posting user*.
- 3) Link menu *pengecekan plagiarisme*.
- 4) Link menu *logout*.
- 5) Link *Hapus* untuk menghapus data *user* yang dipilih.

#### c. Halaman Kelola Data *Posting User*

Halaman ini berfungsi untuk mengelola data *posting user*, dimana administrator dapat menghapus data *posting user* yang dianggap tidak pantas. Rancangan tampilan halaman ‘*Kelola Data Posting User*’ dapat dilihat pada gambar 3.20.



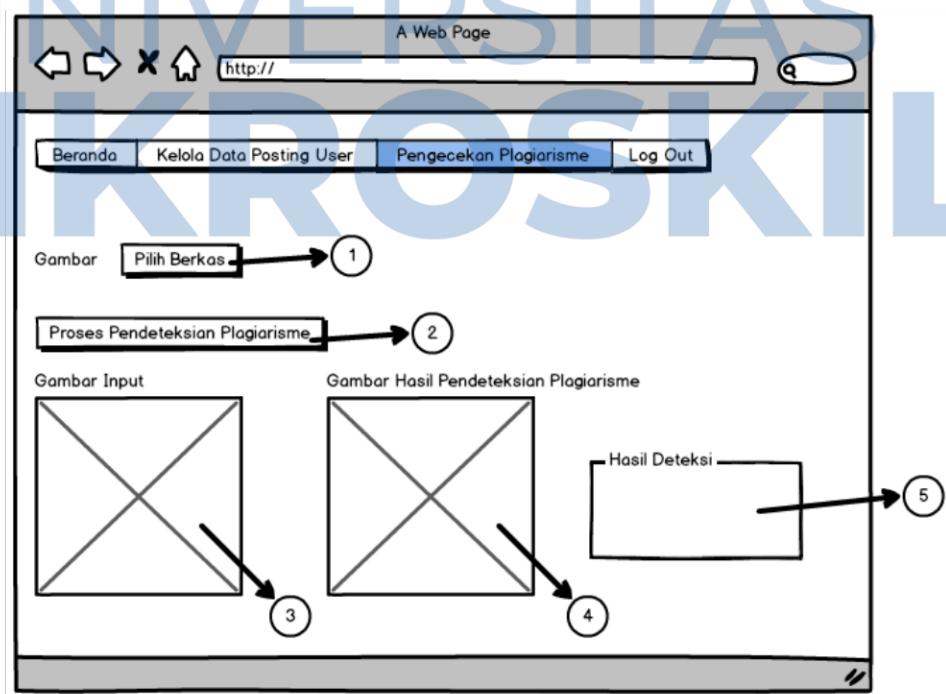
Gambar 3.20 Rancangan Halaman Kelola Data *Posting User*

Keterangan :

- 1) *Link* hapus untuk menghapus data *posting user*.

#### d. Halaman Pendekripsi Plagiarisme

Tampilan ini merupakan halaman untuk melakukan pendekripsi plagiarisme gambar dengan mengimplementasikan *blockchain*. Rancangan tampilan halaman ‘Pendekripsi Plagiarisme’ dapat dilihat pada gambar 3.21.



Gambar 3.21 Rancangan Halaman Pendekripsi Plagiarisme

Keterangan :

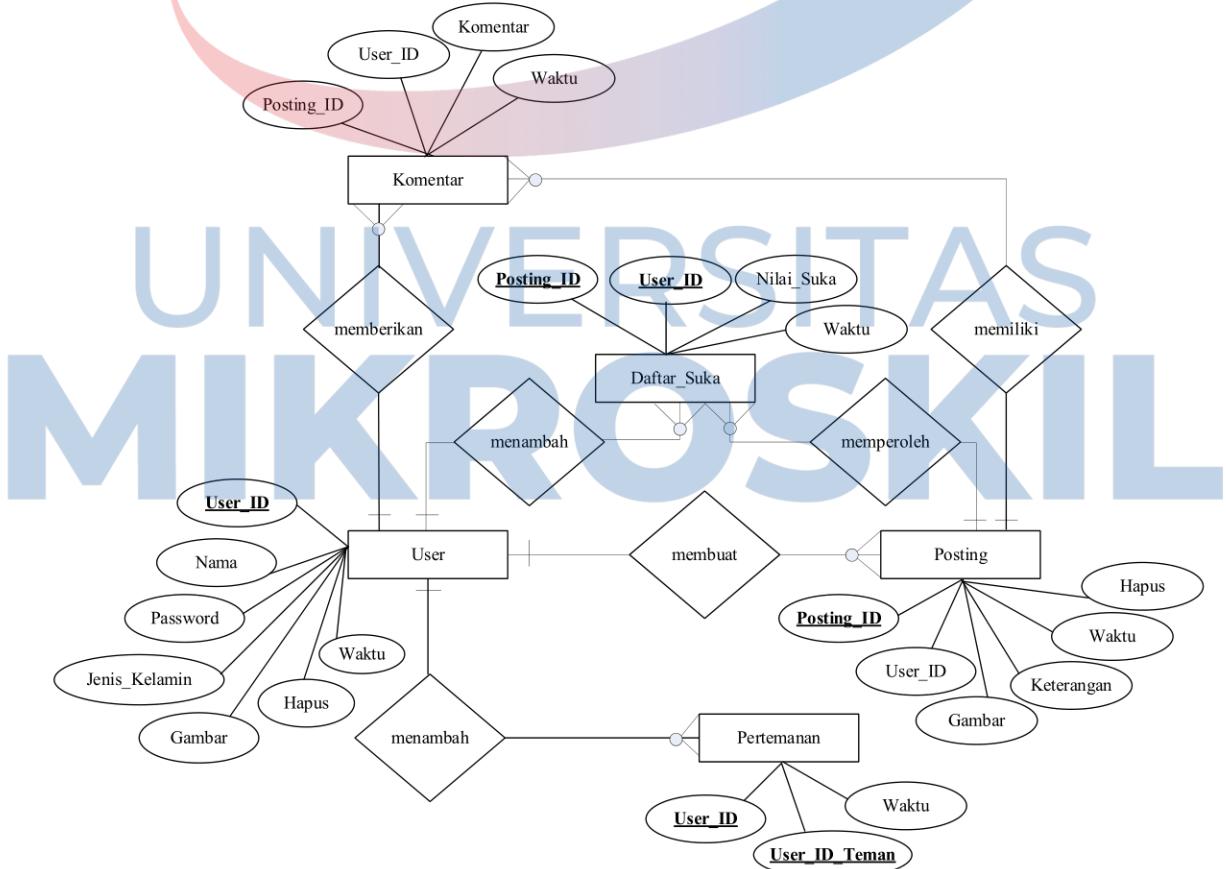
- 1) *Button* untuk memilih gambar yang akan dideteksi plagiarisme.
- 2) *Button* untuk memulai proses pendekripsi.
- 3) *Picture box* untuk menampilkan gambar yang dimasukkan.
- 4) *Picture box* untuk menampilkan gambar hasil pendekripsi.
- 5) Daerah tampilan informasi mengenai hasil pendekripsi.

### 3.2.2. Perancangan Basis Data

Pada perancangan basis data ini akan ditampilkan mengenai *Entity Relationship Diagram* (ERD) dan struktur tabel.

#### 1. Entity Relationship Diagram (ERD)

*Entity Relationship Diagram* (ERD) digunakan untuk menunjukkan hubungan antara entitas pada masing-masing tabel dan objek-objek (himpunan entitas) apa saja yang ingin dilibatkan dalam sebuah data dan bagaimana hubungan yang terjadi diantara objek-objek tersebut. Adapun *Entity Relationship Diagram* dapat dilihat dari gambar 3.22.



Gambar 3.22 *Entity Relationship Diagram (ERD)* Sistem

## 2. Struktur Tabel Basis Data

Pada sistem yang dibangun memiliki beberapa tabel seperti perincian berikut:

### a. Tabel *User*

Tabel *User* menyimpan data *user* yang memiliki atribut-atribut yang dapat dilihat pada tabel di bawah ini:

*Primary Key*: User\_Id

Tabel 3.16 Atribut pada Tabel *User*

Field	Type	Size	Keterangan
Email	VARCHAR	100	Email user
Nama	VARCHAR	255	Nama
Password	VARCHAR	255	Password
Jenis Kelamin	VARCHAR	1	Jenis Kelamin
Hapus	BIT	1	Nilai = 0 berarti tidak dihapus/diblokir oleh <i>admin</i> , nilai = 1 berarti telah dihapus/diblokir oleh <i>admin</i>
Waktu	TIMESTAMP	-	Waktu daftar

### b. Tabel *Posting*

Tabel *Posting* menyimpan data *posting* yang memiliki atribut-atribut yang dapat dilihat pada tabel di bawah ini:

*Primary Key*: Posting\_Id

Tabel 3.17 Atribut pada tabel *Posting*

Field	Type	Size	Keterangan
Posting_Id	INT	10	ID/ <i>Primary Key</i> (PK)
User_Id	INT	10	Foreign Key
Keterangan	TEXT	-	Deskripsi mengenai <i>posting</i>
Gambar	TEXT	-	Lokasi URL dari gambar
Blockchain_Info	TEXT	-	Informasi blockchain
Waktu	TIMESTAMP	-	Waktu <i>posting</i>
Hapus	BIT	1	Nilai = 0 berarti tidak dihapus oleh <i>admin</i> , nilai = 1 berarti telah dihapus oleh <i>admin</i>

### c. Tabel Komentar

Tabel Komentar menyimpan data komentar dari *user* terhadap *posting* yang memiliki atribut-atribut yang dapat dilihat pada tabel di bawah ini:

Tabel 3.18 Atribut pada tabel Komentar

Field	Type	Size	Keterangan
Posting_Id	INT	10	ID Posting
User_Id	INT	10	Foreign Key
Komentar	VARCHAR	255	Komentar terhadap posting
Waktu	TIMESTAMP	-	Waktu posting

### d. Tabel Daftar\_Suka

Tabel Daftar Suka menyimpan data daftar suka dari *user* terhadap *posting* yang memiliki atribut-atribut yang dapat dilihat pada tabel di bawah ini:

*Foreign Key:* posting\_id dan user\_id

Tabel 3.19 Atribut pada tabel Daftar\_Suka

Field	Type	Size	Keterangan
Posting_Id	INT	10	Foreign Key (FK)
User_Id	INT	10	Foreign Key (FK)
Waktu	TIMESTAMP	-	Waktu suka

### e. Tabel Pertemanan

Tabel Pertemanan menyimpan data pertemanan *user* yang memiliki atribut-atribut yang dapat dilihat pada tabel di bawah ini:

*Foreign Key:* user\_id dan user\_id\_teman

Tabel 3.20 Atribut pada tabel Pertemanan

Field	Type	Size	Keterangan
user_id	INT	10	Foreign Key (FK)
user_id_teman	INT	10	Foreign Key (FK)
Waktu	TIMESTAMP	-	Waktu mulai berteman

#### f. Tabel Admin

Tabel *admin* menyimpan data *admin* yang memiliki atribut-atribut yang dapat dilihat pada tabel di bawah ini:

Tabel 3.21 Atribut pada tabel *Admin*

Field	Type	Size	Keterangan
Admin_id	INT	10	ID/ <i>Primary Key</i> (PK)
Password	VARCHAR	255	<i>Password</i>



UNIVERSITAS  
**MIKROSKIL**