BAB III
METODOLOGI PENELITIAN

Penelitian ini dilakukan dengan menggunakan studi literatur, pengembangan model kriptosistem gabungan dan implementasi model ke dalam program aplikasi komputer.

3.1. Identifikasi Masalah

Algoritma kriptografi visual yang diusulkan oleh Naor dan Shamir menghasilkan ukuran gambar yang menjadi dua kali lipat yang disebabkan oleh pemisahan satu pixel menjadi dua pixel. Untuk gambar dengan resolusi kecil hal ini tidak menjadi masalah, akan tetapi beban konversi gambar akan terasa pada gambar dengan resolusi tinggi sehingga terjadi *overload* pada ruang penyimpanan data. Oleh karena itu, dibutuhkan teknik kriptografi visual yang berbeda.

Andysah (2016) dan Ratnadewi (2018) mengajukan kriptografi visual yang ditingkatkan menggunakan RSA dan perluasannya dengan memisahkan masing-masing warna dasar per pixel pada gambar aslinya. Gambar yang digunakan adalah RGB, masing-masing pixel memiliki nilai RGB tertentu yang kemudian dienkripsi menggunakan kriptografi ECC.

3.2. Model Dasar

Model dasar yang digunakan adalah kriptografi visual perluasan RGB dengan skema (2,2) dan *elliptic curve cryptography*. Dalam ECC, *plaintext* dikonversi ke dalam titik-titik pada kurva eliptik kemudian dienkripsi menghasilkan *ciphertext*. Pada kriptografi visual perluasan RGB, satu gambar dienkripsi menjadi dua buah *share image* menggunakan proses *bitxor*. Kedua model dasar memiliki keamanan masing-masing. Pada kriptografi visual, gambar asli tidak dapat diperoleh jika hanya memiliki salah satu *share*. Sedangkan pada ECC, proses dekripsi tidak akan mungkin jika tidak mengetahui fungsi kurva eliptik yang digunakan.

3.2.1. Kriptografi Visual pada Gambar Berwarna RGB

Satuan ukuran yang digunakan pada gambar digital adalah pixel (*Picture Element*) sebagai titik terkecil dalam sebuah gambar yang dihitung per inci. Pada kriptografi visual, proses enkripsi dan dekripsi pada gambar terjadi pada skala pixel. Pada tahap enkripsi, masing-masing pixel pada gambar diidentifikasi nilai RGB-nya dan melalui proses *bitxor* sehingga menghasilkan dua buah *share image*. Proses dekripsi dilakukan dengan penumpukan kedua *share* menggunakan *bitxor* juga.

3.2.2. *Elliptic Curve Cryptography*

ECC adalah kriptografi asimetris yang menggunakan kurva eliptik pada lapangan berhingga. Misalkan Alice ingin mengirimkan pesan pada Bob dengan yang diamankan menggunakan ECC. Hal pertama yang dilakukan adalah pembangkitan kunci publik dan kunci privat oleh Bob. Algoritma pembangkitan kunci adalah sebagai berikut:

1. Tentukan sebuah kurva eliptik $E_p(a,b) : y^2 = x^3 + ax + b \pmod p$ pada suatu lapangan $\mathbb{Z}_p$, dengan $a, b \in \mathbb{Z}_p$ dan $4a^3 + 27b^2 \not\equiv 0 \pmod p$.
2. Cari titik-titik pada E_p untuk $x \in \mathbb{Z}_p$ dengan menggunakan *quadratic residue*. Diperoleh n sebagai order, di mana n adalah jumlah titik pada kurva eliptik dan titik tak hingganya.
3. Pilih suatu titik $P \in E_p$ sebagai generator, kemudian hitung $Q = mP$ untuk suatu bilangan acak $m \in \mathbb{Z}_n$. m adalah kunci privat untuk Bob.
4. Bob mengirimkan kunci publik berupa (P, Q, n) pada Alice.

Langkah selanjutnya adalah membuat tabel konversi yang bergantung pada P. Alice memberitahu Bob jenis pesan yang akan disampaikan berupa alphabet dengan angka atau kombinasi lainnya. Setelah tabel konversi disetujui, Alice dapat melakukan proses enkripsi. Pesan yang akan dienkripsi terlebih dahulu dikonversi sehingga berupa titik-titik pada E dan disebut sebagai plainteks. Dengan menggunakan kunci publik, Alice melakukan proses enkripsi dan mengkonversi kembali sesuai dengan tabel menjadi cipherteks dan dikirimkan kepada Bob.

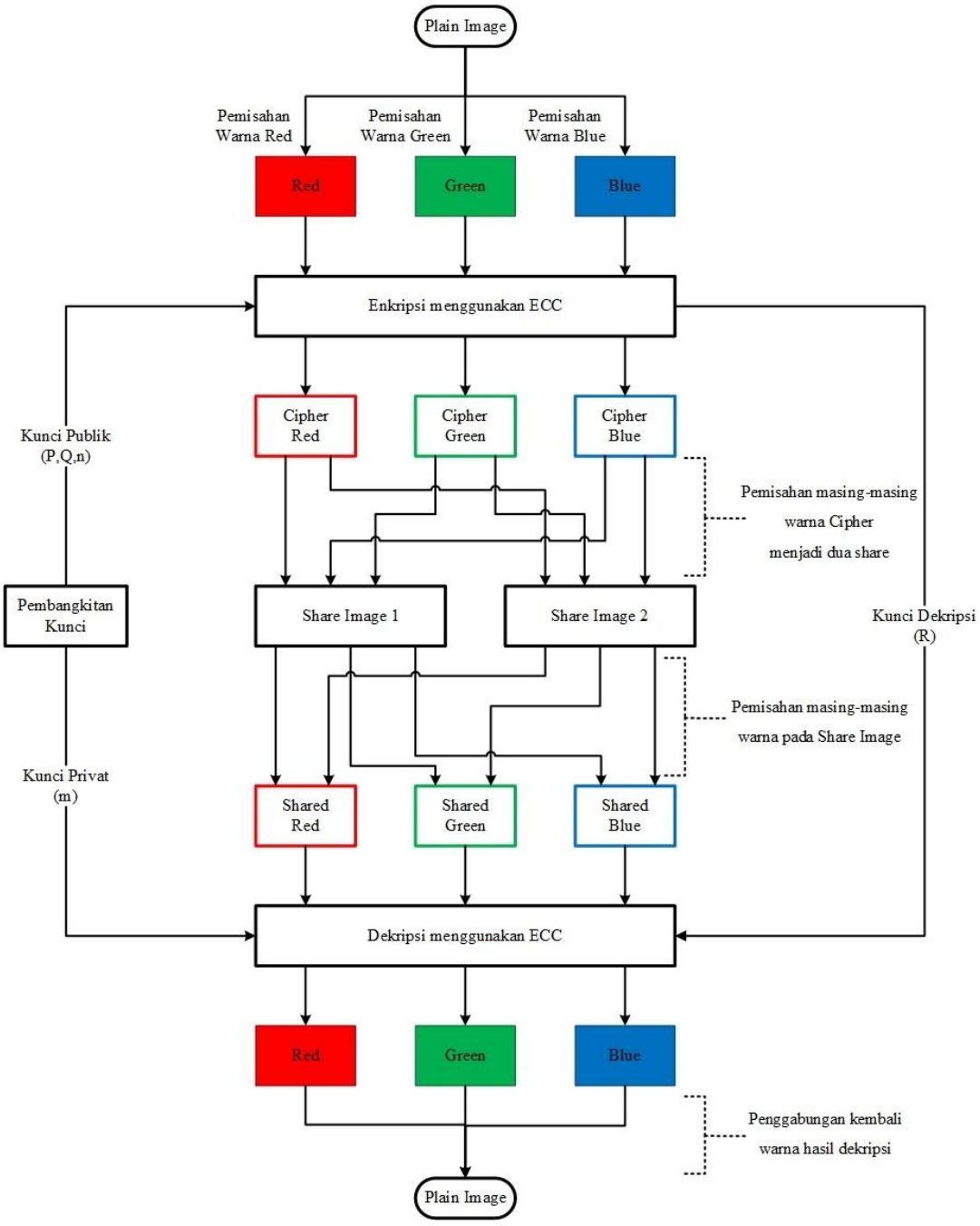
Langkah terakhir adalah proses dekripsi yang dilakukan oleh Bob. Cipherteks yang diterima harus diubah menggunakan tabel konversi. Bob mendekripsi pesan

menggunakan kunci privat m kemudian mengubah lagi plainteks menggunakan tabel, sehingga diperoleh pesan asli dari Alice.

3.3. Pengembangan Kriptografi Visual Menggunakan ECC

Pada tahap ini akan dibuat kriptosistem visual RGB yang ditingkatkan dengan menggunakan ECC. Langkah pertama adalah mengidentifikasi nilai RGB masing-masing pixel pada gambar asli menjadi sebuah matriks untuk tiap warna dasar. Kemudian, tiap anggota matriks masing-masing warna dienkripsi menggunakan kunci publik ECC. Hasil enkripsi tiap matriks digabungkan kembali dan menjadi suatu *cipher image.* Pada proses dekripsi, semua share disatukan dan nilai RGB masing-masing pixel diidentifikasi. Diperoleh matriks RGB yang terenkripsi, kemudian tiap anggota matriks didekripsi menggunakan kunci privat yang diperoleh dari ECC.

Algoritma kriptografi visual menggunakan ECC dapat digambarkan dengan skema sebagai berikut:



Gambar 3.1. Skema Alur Kriptografi Visual Menggunakan ECC

Skema pada Gambar 3.1. dapat dijelaskan sebagai berikut:

1. Penerima membangkitkan kunci publik dan kunci privat menggunakan titik generator yang disetujui.
2. Pengirim menentukan gambar yang akan dikirimkan dan membangkitkan kunci publik dan kunci privat.

3. Pengirim dan penerima saling menukar kunci publiknya.
4. Pengirim melakukan konversi nilai intensitas RGB menjadi titik kurva eliptik. Gambar yang dikonversi kemudian dienkripsi menggunakan kunci publik yang diterima dan kunci privat milik pengirim sehingga dihasilkan dua buah *share image*. Kedua *share* dikirimkan kepada penerima.
5. Penerima melakukan penumpukan kedua *share image* dengan menggunakan *bitxor*. Hasil penumpukan didekripsi menggunakan kunci dekripsi dan kunci privat milik penerima.

3.3.1. Proses Pembangkitan Kunci

Sebagai kriptografi asimetris, ECC memerlukan dua buah kunci, yaitu kunci privat dan kunci publik. Pihak yang melakukan pembangkitan kunci adalah pihak yang akan menerima foto yang terenkripsi. Berikut langkah pembangkitan kunci privat dan kunci publik:

1. Pilih suatu fungsi kurva eliptik E pada $\mathbb{Z}_p$ ($p > 3$ prima) kemudian pilih sebuah titik $P \in E$ sebagai titik pembangkit. Dari P diperoleh n sebagai orde titik P di mana $nP = P + P + \cdots + P = O$ (infinity).
2. Penerima memilih suatu bilangan acak $m < n$ sebagai kunci privat dan menghitung $Q = mP$.
3. Penerima mengirimkan $e_k = (P, Q)$ sebagai kunci publik dan m sebagai kunci privat.

3.3.2. Enkripsi

Pengirim pesan melakukan proses enkripsi menggunakan kunci publik yang telah diterima. Langkah-langkahnya adalah sebagai berikut:

1. Pengirim menentukan sebuah file foto berwarna sebagai plainteks.
2. Pengirim mengkonversi nilai-nilai RGB per-pixel pada foto tersebut menjadi titik pada E menggunakan tabel konversi yang dibangkitkan menggunakan P. Hasil konversi tiap pixel untuk masing-masing warna dinyatakan sebagai P_m.
3. Pengirim memilih suatu bilangan acak $k < n$ dan menghitung $R = kP$ dan $P_c = P_m + kQ$.

4. Nilai P_c dikonversikan kembali menggunakan tabel konversi menjadi nilai RGB sebagai *shared image*. Nilai R adalah kunci publik dari pengirim pesan.

3.3.3. Dekripsi

Setelah menerima *shared image* dan R, penerima melakukan proses dekripsi sebagai berikut.

1) Penerima mengkonversi nilai-nilai RGB per-pixel pada *shared image* tersebut menjadi titik pada E menggunakan tabel konversi yang dibangkitkan menggunakan P. Hasil konversi tiap pixel untuk masing-masing warna dinyatakan sebagai P_c.
2) Dengan menggunakan kunci privat m, penerima menghitung $P_d = P_c - mR$.
3) Nilai P_d dikonversi kembali menjadi nilai RGB menggunakan tabel konversi.

Contoh Enkripsi dan Dekripsi:

Misalkan Alice ingin mengirim sebuah foto dengan ukuran 4×4 pixel kepada Bob.

Gambar 3.2. Contoh Foto

Bob memilih suatu kurva eliptik $E: y^2 = x^3 + 9x + 7 \bmod 2011$, dimana $a = 9,\ b = 7, \text{dan } p = 2011$. Dipilih suatu titik secara acak $P = (257,1933)$ sebagai titik pembangkit dengan orde $n = 2027$. Kemudian Bob memilih $m = 11$ sebagai kunci privat dan menghitung $Q = mP = (1233,196)$. Diperoleh kunci publik $e_k = ((257,1933),(1233,196))$.

Alice menerima kunci publik dari Bob dan memulai proses enkripsi. Langkah pertama yang harus dilakukan oleh Alice adalah mengkonversi nilai RGB pada foto menggunakan tabel konversi yang digenerasikan dari P. Untuk membuat tabel konversi, diperlukan nilai $P, 2P, 3P, \dots, (n-1)P$ dan disusun ulang menjadi

tabel dengan baris berjumlah 256 sebagai konversi nilai RGB yang memiliki rentang dari 0 sampai 255. Jumlah kolom dari tabel adalah $\lceil n/256 \rceil = 8$. Karena n bukan kelipatan 256, maka untuk sel yang kosong diisi dengan 0. Dalam melakukan konversi, sel-sel yang bernilai 0 tidak diikutsertakan.

Tabel 3.1. Tabel Konversi

Intensitas RGB	Kolom 1	Kolom 2	Kolom 3	...	Kolom 6	Kolom 7	Kolom 8
0	(257,1933)	(1801,148)	(1539,1026)		(1649,929)	(457,931)	(1135,148)
1	(293,1719)	(1023,242)	(968,112)		(650,1627)	(1706,118)	(815,310)
2	(1006,1067)	(577,1404)	(1839,1423)		(990,879)	(1442,1987)	(1217,989)
3	(484,1401)	(588,820)	(1596,680)		(1292,1462)	(821,1488)	(219,778)
4	(1191,566)	(1349,1265)	(592,1611)		(212,757)	(402,1665)	(76,1187)
5	(966,1781)	(1738,844)	(1370,1532)		(1787,314)	(1125,656)	(718,1600)
6	(220,626)	(837,404)	(108,205)		(227,480)	(985,1668)	(518,1946)
7	(1921,650)	(751,458)	(1164,559)		(1149,179)	(332,876)	(1132,1090)
⋮							
101	(1700,459)	(822,860)	(442,1332)		(665,522)	(754,541)	(1597,871)
102	(1498,1184)	(481,1920)	(286,944)		(1286,1877)	(44,56)	(1935,1654)
103	(297,1136)	(1901,1143)	(1665,1912)	...	(1222,281)	(1444,1750)	(610,54)
104	(1076,1666)	(90,969)	(967,1514)		(122,222)	(750,10)	(437,1640)
105	(166,1389)	(1390,775)	(1139,1114)		(48,1848)	(1219,1387)	(596,2004)
⋮							
250	(1067,895)	(448,272)	(43,818)		(1163,1112)	(1646,921)	(0,0)
251	(334,1279)	(1547,1873)	(1050,128)		(1794,341)	(476,1002)	(0,0)
252	(1252,1817)	(778,149)	(1418,1265)		(1261,1254)	(2008,1250)	(0,0)
253	(127,1993)	(1953,1076)	(394,576)	...	(725,1396)	(1874,1871)	(0,0)
254	(1889,1450)	(171,1991)	(1230,57)		(289,1146)	(381,1457)	(0,0)
255	(1869,422)	(473,1767)	(1636,856)		(540,1822)	(1936,1593)	(0,0)

Setelah diperoleh tabel konversi, Alice mencocokkan nilai RGB pada foto dengan titik pada tabel. Berikut adalah matriks berisi nilai warna merah dan hasil konversinya. Proses konversi dilakukan dengan memilih secara acak titik kurva pada baris dengan intensitas warna yang bersesuaian. Dengan pemilihan acak, untuk nilai warna yang sama dapat diperoleh hasil konversi yang berbeda.

Tabel 3.2. Hasil Konversi Warna

Tabel Warna Asli R					Tabel Hasil Konversi Warna R (P_m)			
43	238	2	11		(1165,1830)	(425,1411)	(1341,667)	(1618,1832)
248	229	185	32		(66,888)	(320,1780)	(239,98)	(62,427)
6	194	255	41		(135,681)	(1024,785)	(81,1523)	(1806,705)

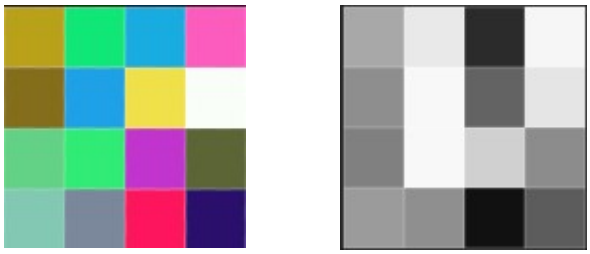
0	255	236	104		(685,1327)	(66,1123)	(1618,179)	(230,688)
Tabel Warna Asli G					Tabel Hasil Konversi Warna G (P_m)			
48	0	174	171		(1258,761)	(319,764)	(160,190)	(1715,1661)
17	83	168	30		(514,16)	(659,391)	(767,1893)	(334,206)
183	25	243	15		(1105,1053)	(940,33)	(125,1847)	(1918,733)
74	250	23	69		(112,1038)	(392,1434)	(1924,1257)	(815,1329)
Tabel Warna Asli B					Tabel Hasil Konversi Warna B (P_m)			
139	144	250	73		(111,175)	(566,1442)	(1350,218)	(160,1821)
95	26	2	33		(1940,1190)	(1193,1719)	(543,720)	(1388,996)
235	124	9	60		(1182,1145)	(666,431)	(295,1433)	(148,381)
48	222	79	39		(68,635)	(1883,504)	(54,145)	(1391,1092)

Hasil konversi warna kemudian dienkripsi menggunakan kunci publik yang diterima. Alice memilih $k = 13$ dan menghitung $kP = 13(257,1933) = (293, 1719)$. Nilai $R = kP$ kemudian disebut sebagai kunci publik dari Alice. Selanjutnya, Alice menghitung $P_c = P_m + kQ$ dimana P_m adalah nilai pixel pada tabel hasil konversi warna dan P_c sebagai hasil enkripsi tiap-tiap pixel.

Tabel 3.3. Hasil Enkripsi Foto

Tabel Hasil Enkripsi Warna R (P_c)			
47	242	6	15
252	233	189	36
10	198	2	45
4	2	240	108
Tabel Hasil Enkripsi Warna G (P_c)			
52	4	178	175
21	87	172	34
187	29	247	29
78	254	27	73
Tabel Hasil Enkripsi Warna B (P_c)			
143	148	254	77
99	30	6	37
239	128	13	64
52	226	83	43

Hasil enkripsi kemudian dikonversi kembali menjadi nilai intensitas warna RGB sehingga dihasilkan sebuah *cipher image* yang kemudian akan dipisah menjadi dua *share image* dengan menggunakan proses *bitxor*. Proses ini dilakukan dengan mengubah nilai warna RGB kedalam bentuk bit.



Gambar 3.3. Hasil Share

Bob menerima dua buah *share image* dari Alice. Langkah pertama yang dilakukan oleh Bob adalah menumpuk kedua *share* dengan menggunakan proses *bitxor*. Diperoleh sebuah gambar yang kemudian dikonversi menjadi titik kurva eliptik dan didekripsi menggunakan kunci publik dari Alice dan kunci privat milik Bob sendiri. Diperoleh hasil dekripsi yang sudah dikembalikan menjadi nilai RGB pada Tabel 3.4.

Tabel 3.4. Hasil Dekripsi Foto

Tabel Hasil Dekripsi Warna R (P_d)			
43	238	2	11
248	229	185	32
6	194	254	41
0	254	236	104
Tabel Hasil Dekripsi Warna R (P_d)			
48	0	174	171
17	83	168	30
183	25	243	15
74	250	23	69
Tabel Hasil Dekripsi Warna R (P_d)			
139	144	250	73
95	26	2	33
235	124	9	60
48	222	79	39

3.4. Konstruksi Program Komputer

Konstruksi program komputer yang dilakukan menggunakan MATLAB terdiri dari bagian yaitu pembangkitan kunci, enkripsi, dan dekripsi data.

3.4.1. Input dan Output

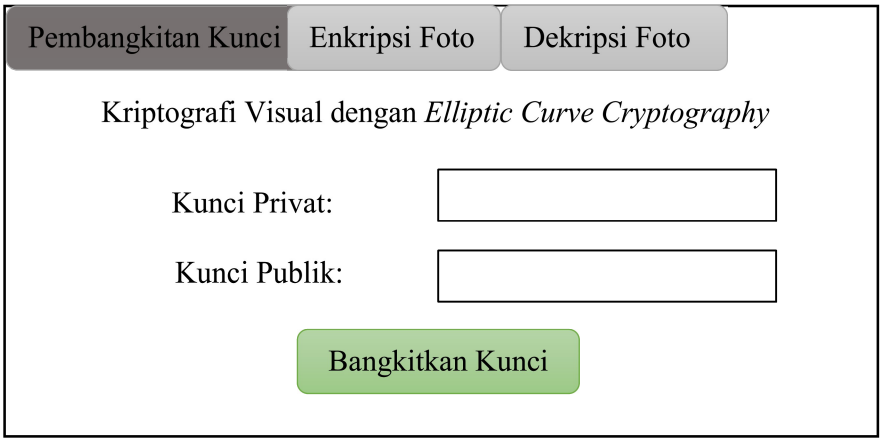
Data yang diproses adalah gambar dengan format JPEG. Proses pembangkitan kunci dilakukan dengan menggunakan MATLAB dan hasilnya berupa teks.

Sebelum melakukan proses enkripsi-dekripsi dibutuhkan sepasang kunci publik dan kunci privat yang dibuat oleh Bob. Program pembangkitan kunci ECC dibuat sebagai sub-program dengan kurva eliptik yang sudah ditentukan dan memberi output berupa titik generator, kunci publik dan kunci privat. Titik generator dan kunci publik kemudian dikirimkan pada Alice.

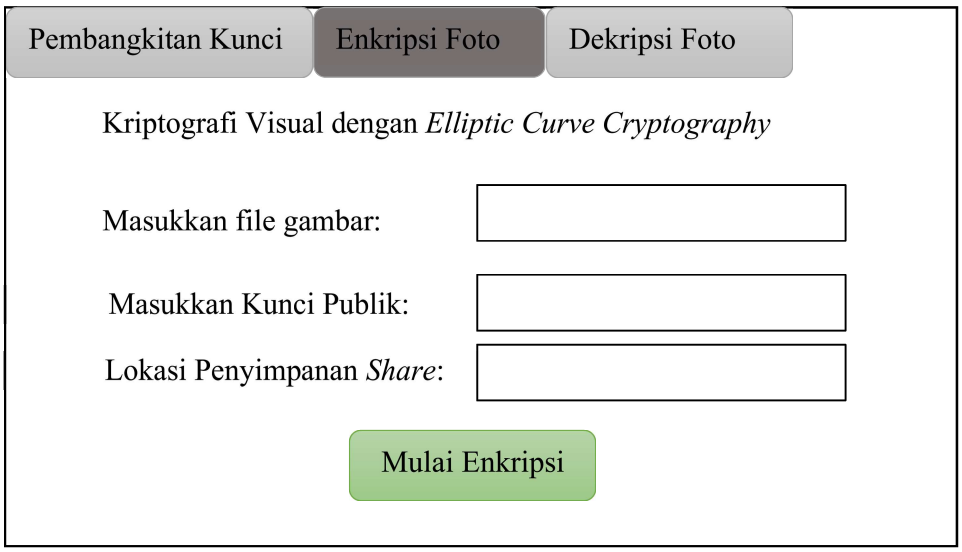
Data yang diinputkan pada program enkripsi adalah file gambar dan kunci publik untuk proses enkripsi. Output dari program enkripsi adalah sebuah *share image* yang akan dikirim kepada Bob. Proses dekripsi dilakukan dengan menginputkan kunci privat dan *share image* pada program. Hasil dekripsi berupa sebuah file gambar asli.

3.4.2. Rancangan Tampilan

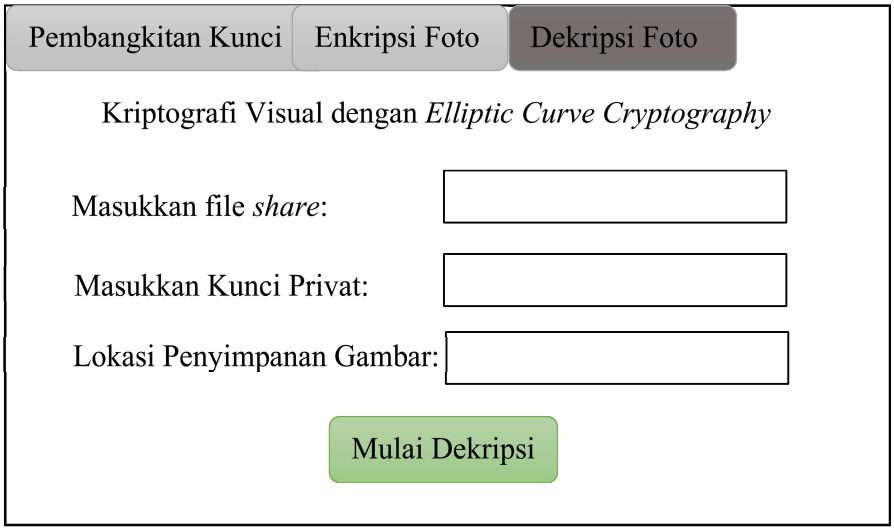
Berikut adalah rancangan tampilan dari program menggunakan MATLAB:



Gambar 3.4. Rancangan Aplikasi Pembangkit Kunci ECC



Gambar 3.5. Rancangan Aplikasi Enkripsi Foto



Gambar 3.6. Rancangan Aplikasi Dekripsi Foto

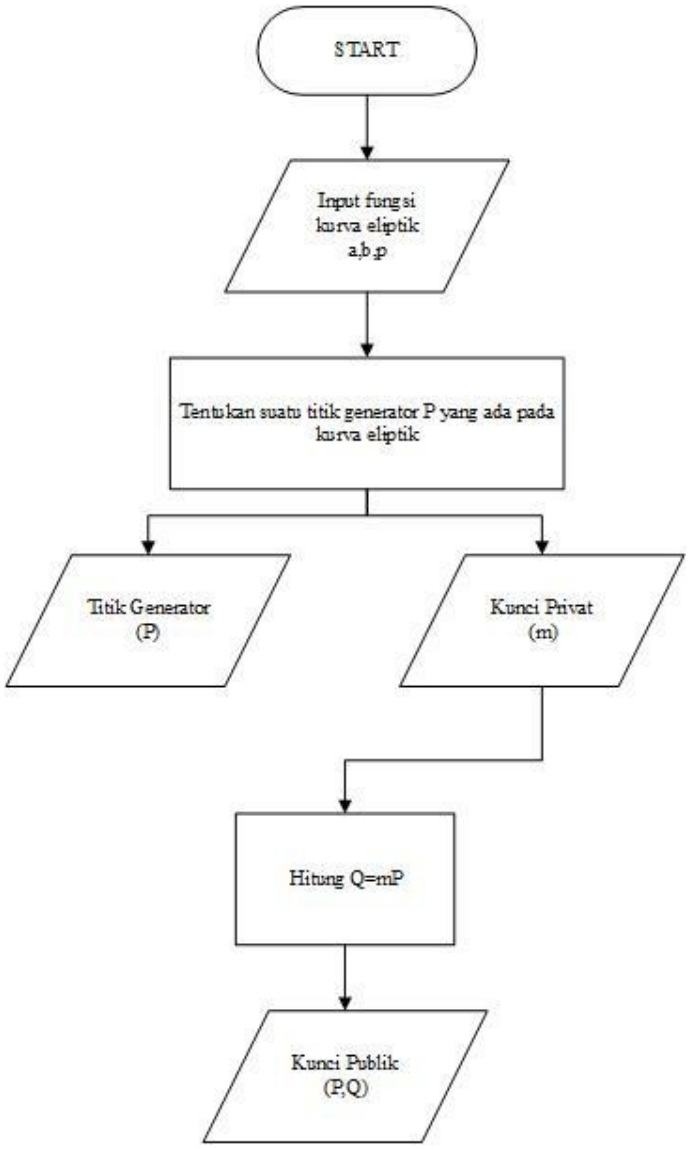
3.4.3. Algoritma

Algoritma program pengembangan kriptografi visual menggunakan ECC dapat diuraikan sebagai berikut:

1. Algoritma Pembentukan Kunci

   Input : $E_p(a, b) : y^2 = x^3 + ax + b \pmod{p}$

   Output : Kunci publik (P, Q) dan kunci privat m

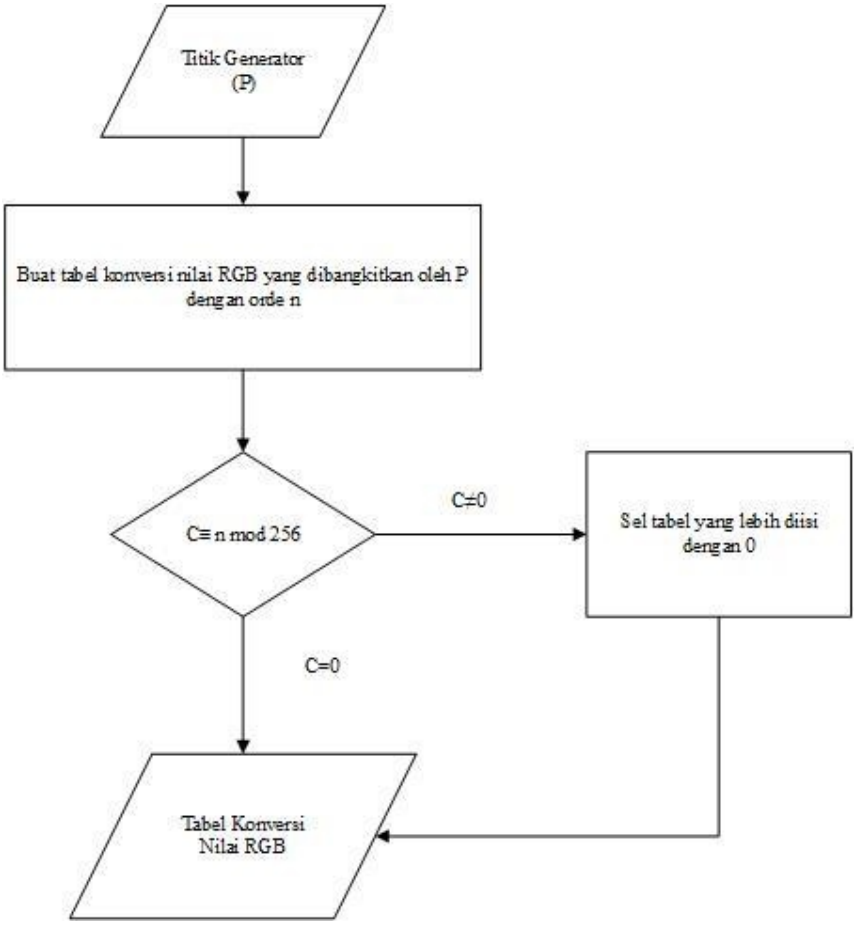


Gambar 3.7. Diagram Alir Algoritma Pembentukan Kunci

2. Algoritma Pembentukan Tabel Konversi RGB

Input : Titik generator *P*

Output : Tabel Konversi Nilai RGB

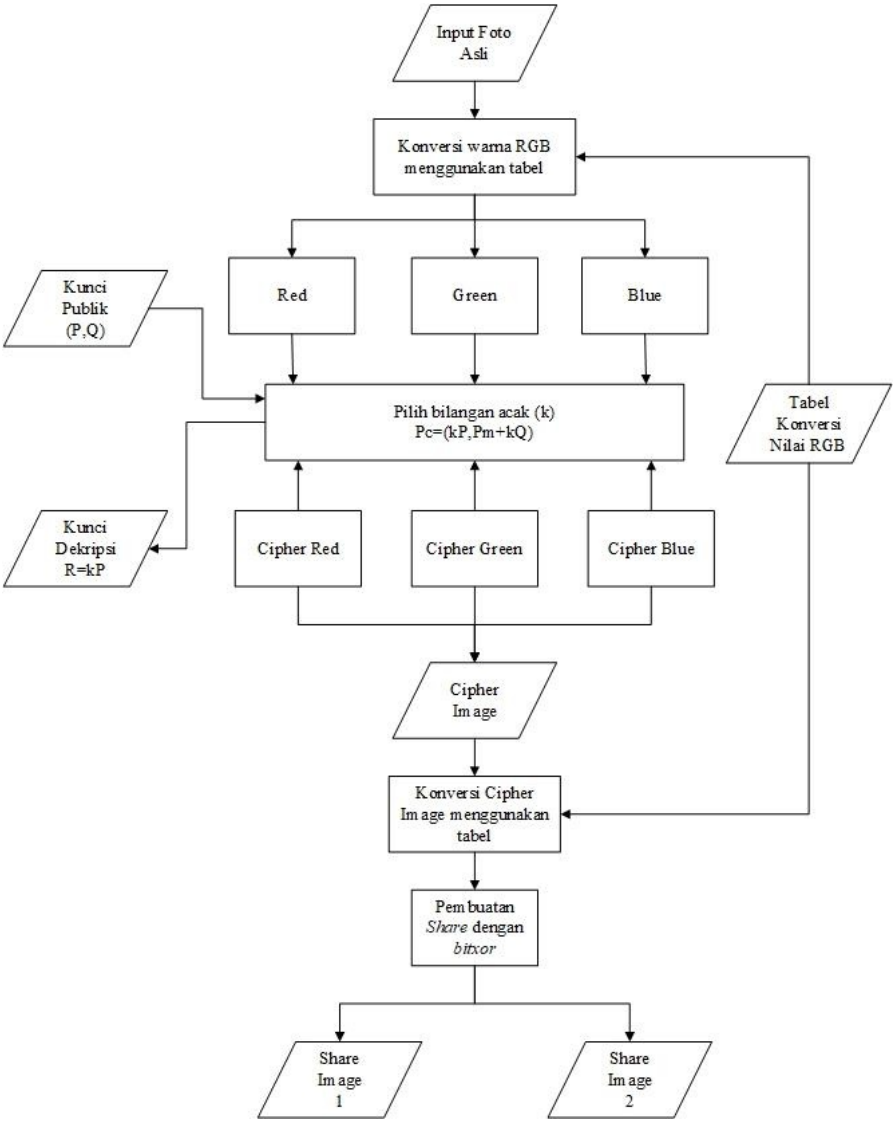


Gambar 3.8. Diagram Alir Algoritma Pembentukan Tabel Konversi

3. Algoritma Enkripsi

Input : Foto, Kunci Publik (P, Q), Tabel Konversi RGB

Output : *Share Image* 1 & *Share Image* 2

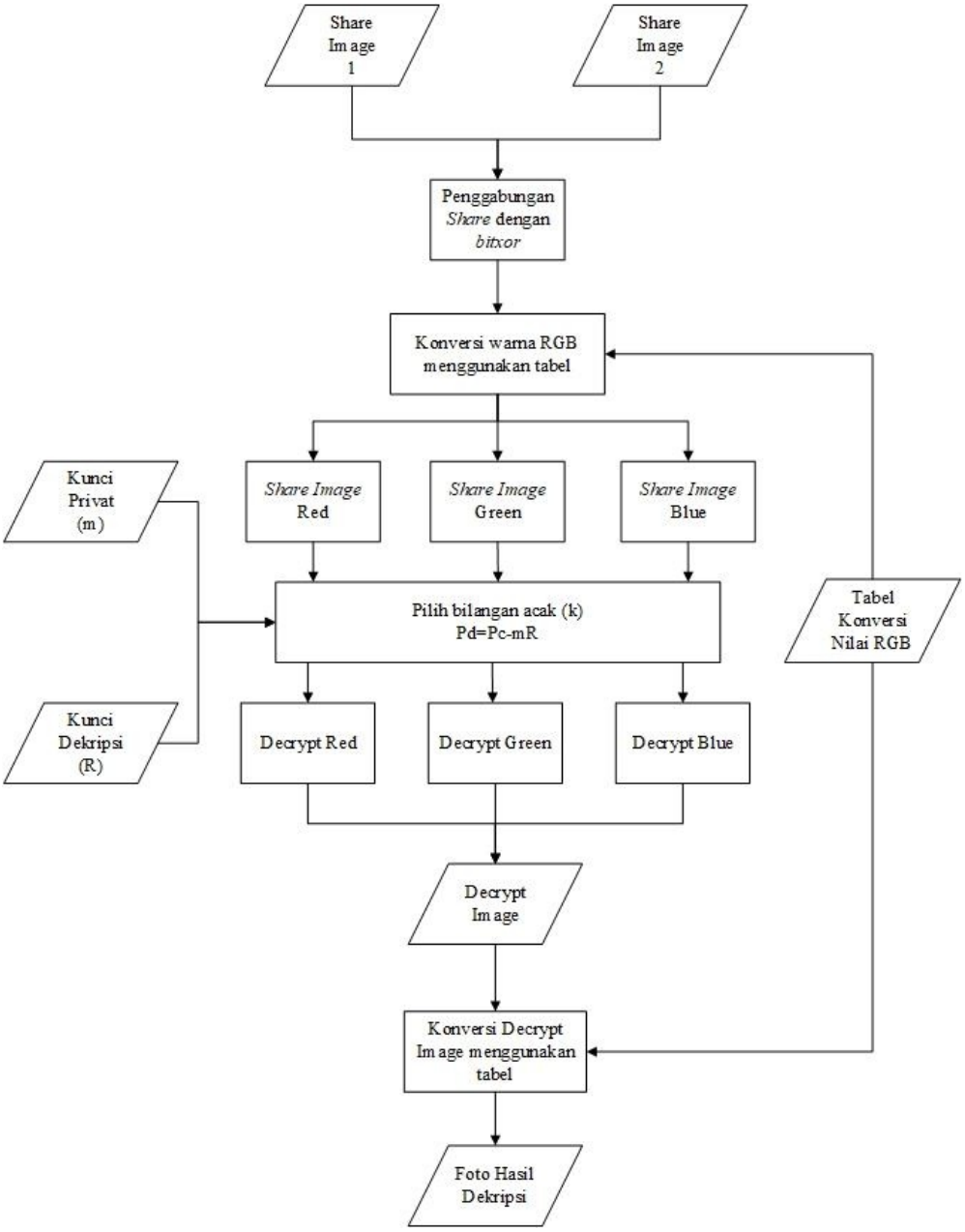


Gambar 3.9. Diagram Alir Algoritma Enkripsi

4. Algoritma Dekripsi

Input : *Share Image* 1, *Share Image* 2, Kunci Privat m, Kunci Dekripsi R

Output : Gambar hasil dekripsi



Gambar 3.10. Diagram Alir Algoritma Dekripsi

3.5.Validasi

Model algoritma kriptografi visual yang digabungkan dengan ECC yang diperoleh akan diperiksa validasinya terhadap program yang dikonstruksi. Validasi dilakukan untuk mengetahui apakah model yang dirancang sudah sesuai atau belum.

Setelah model divalidasi, maka kriptosistem visual yang ditingkatkan menggunakan ECC akan lebih sulit dipecahkan jika hanya memperoleh sebagian share atau semua share tanpa kunci publik.