TUGAS AKHIR -SM141501

ENKRIPSI CITRA DIGITAL MENGGUNAKAN METODE KURVA ELIPTIK DIFFIE-HELLMAN DAN TRANSFORMASI WAVELET DISKRIT

AGUS SETIAWAN
NRP 1213 100 103

Dosen Pembimbing
Dr. Dwi Ratna Sulistyaningrum, S.Si,MT
Drs. Komar Baihaqi, M.Si

DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2017

FINAL PROJECT -SM141501

DIGITAL IMAGE ENCRYPTION USING ELLIPTIC CURVE DIFFIE-HELLMAN AND DISCRETE WAVELET TRANSFORM METHOD

AGUS SETIAWAN
NRP 1213 100 103

Supervisor
Dr. Dwi Ratna Sulistyaningrum, S.Si,MT
Drs. Komar Baihaqi, M.Si

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Science
Sepuluh Nopember Institute of Technology
Surabaya 2017

LEMBAR PENGESAHAN

ENKRIPSI CITRA DIGITAL MENGGUNAKAN METODE KURVA ELIPTIK DIFFIE-HELLMAN DAN TRANSFORMASI WAVELET DISKRIT

DIGITAL IMAGE ENCRYPTION USING ELLIPTIC CURVE DIFFIE-HELLMAN AND DISCRETE WAVELET TRANSFORM METHOD

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
AGUS SETIAWAN
NRP. 1213100103

Menyetujui,

Dosen Pembimbing II,	DosenPembimbing I,
Drs. Komar Baihaqi, M.Si	Dr. Dwi Ratna S, S.Si, MT
NIP. 19600229 198803 1 001	NIP. 19690405 199403 2 003

Mengetahui,
Kepala Departemen Matematika
FMIPA ITS

Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, Agustus 2017

ENKRIPSI CITRA DIGITAL MENGGUNAKAN METODE KURVA ELIPTIK DIFFIE-HELLMAN DAN TRANSFORMASI WAVELET DISKRIT

Nama Mahasiswa : Agus Setiawan
NRP : 1213 100 103
Departemen : Matematika
DosenPembimbing : 1. Dr. Dwi Ratna S, S.Si,MT
2. Drs. Komar Baihaqi, M.Si

Abstrak

Data citra menjadi data yang paling rawan untuk disadap pada jaringan komputer. Seseorang akan cenderung meng-enkripsi data citra atau gambar tersebut untuk mengamankan data citra tersebut. Proses enkripsi yang biasa dilakukan biasanya dengan mengubah citra menjadi citra buram. Namun, hal itu akan menyebabkan citra tersebut mengundang beberapa orang untuk melakukan kriptanalisis terhadap citra tersebut, sehingga semakin banyak peluang terjadinya kriptanalisis. Pada tugas akhir ini dibahas mengenai proses enkripsi-dekripsi citra digital untuk keamanan pesan dengan ECC (*Elliptic Curve Cryptography*) dan DWT (*Discrete Wavelet Transform*). Sehingga diharapkan citra digital tersebut dapat diamankan dari kebocoran pesan yang bersifat rahasia. Tujuan dari penelitian ini diantaranya adalah untuk mengembangkan algoritma enkripsi citra digital yang tahan akan gangguan dan kriptanalisis. Setelah dilakukan uji coba, hasil yang didapatkan adalah citra enkripsi sangat berbeda dari pada citra awal dan bukan berupa citra buram. Hasil uji coba juga menunjukkan bahwa citra hasil enkripsi sangat tahan terhadap kriptanalisis berdasarkan analisis kualitas citra dan analisis sensitifitas kunci dan juga citra enkripsi tahan terhadap *noise salt and pepper* dengan keakuratan rata-rata 98,936%.

Kata kunci : Citra digital, Enkripsi, Noise, Diffie-Hellman, Wavelet

"Halaman ini sengaja dikosongkan."

DIGITAL IMAGE ENCRYPTION USING ELLIPTIC CURVE DIFFIE-HELLMAN AND DISCRETE WAVELET TRANSFORM METHOD

Name	***: Agus Setiawan***
NRP	***: 1213 100 103***
Department	***: Mathematics***
Supervisor	***: 1. Dr. Dwi Ratna S, S.Si,MT***
	2. Drs. Komar Baihaqi, M.Si

Abstract

Image data becomes the most vulnerable data to be tapped on a computer network. Someone will tend to encrypt the image or image data to secure the image data. The usual encryption process is usually done by transforming the image into blurry images. However, it will cause the image to invite some people to do the cryptanalysis of the image, so the more chances of cryptanalysis. In this final project discussed about encryption process of digital image decryption for message security with ECC (Elliptic Curve Cryptography) and DWT (Discrete Wavelet Transform). So hopefully the digital image can be secured from leakage of a secret message. The purpose of this research is to develop digital image encryption algorithms that are resistant to attack and cryptanalysis. After testing, the results obtained are the image of the encryption is very different from the initial image and not noise image. The test results also show that the encrypted image is highly resistant to cryptanalysis based on image quality analysis and key sensitivity analysis and also resistant to noise salt and pepper with average accuracy 98,936%.

Key words : *Digital Image, Encryption, Noise, Diffie-Hellman, Wavelet*

"Halaman ini sengaja dikosongkan."

KATA PENGANTAR

Segala Puji bagi Allah SWT Tuhan semesta alam yang telah memberikan karunia, rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul: **"Enkripsi Citra Digital Menggunakan Metode Kurva Eliptik Diffie-Hellman dan Transformasi Wavelet Diskrit"** yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Studi S-1 pada Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan berkat kerjasama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal itu, penulis mengucapkan terima kasih kepada:

1. Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si, MT selaku dosen pembimbing I yang senantiasa membimbing dengan sabar dan memberikan kritik dan saran dalam penyusunan Tugas Akhir ini.
2. Bapak Drs. Komar Baihaqi, M.Si selaku dosen pembimbing II yang senantiasa membimbing dengan sabar dan memberikan kritik dan saran dalam penyusunan Tugas Akhir ini.
3. Bapak Dr. Imam Mukhlas, S.Si, MT selaku Kepala DepartemenMatematika ITS.
4. Bapak Drs. Suharmadi, Dipl. Sc, M.Phil, Bapak Drs. Nurul Hidayat, M.Kom, dan Bapak Dr. Darmaji, S.Si, MT selaku dosen penguji Tugas Akhir ini.
5. Ibu Soleha, S.Si, M.Si selaku Dosen Wali.
6. Bapak Dr. Didik Khusnul Arif, S.Si, M.Si selaku kaprodi S1 departemen matematika ITS
7. Drs. Iis Herisman, M.Sc. selaku Koordinator Tugas Akhir dan Mas Ali yang selalu memberikan informasi mengenai tugas akhir.
8. Seluruh jajaran dosen dan staf jurusan Matematika ITS.

9. Ibu Suriati tersayang yang senantiasa dengan ikhlas memberikan semangat, perhatian, kasih saying, doa, motivasi dan nasihat yang sangat berarti bagi penulis.
10. Adik Sri Setyawati yang menjadi salah satu motivasi bagi penulis untuk segera menyelesaikan Tugas Akhir ini.
11. Mas Doni, Uzu, Ivan, Wawan yang telah banyak membantu penulis dalam pengerjaan tugas akhir.
12. Teman-teman pejuang 116 serta teman-teman pejuang 116 yang tidak bisa disebutkan satu per satu yang saling memotivasi satu sama lain.
13. Toem, Rozi, Gery, Romli, Bayu, Daus, Ardi, Haidar, Kunur, Satria, Gono, Fadhlan, Bhara, Firdo, Zani, Sinar, Ariel, Jojo, Asna, Yoga selaku para sahabat penulis.
14. Dulur Matematika ITS 2013 yang selalu memberi doa dan dukungan kepada penulis.
15. Teman-teman Kesma HIMATIKA ITS 2014-2016 serta BPH?DPP LDJ Ibnu Muqlah.
16. Semua pihak yang tidak bisa disebutkan satu-persatu. Terimaa kasih telah mendoakan dan mendukung penulis sampai dengan selesainya tugas akhir ini.

Penulis juga menyadari bahwa dalam tugas akhir ini masih terdapat kekurangan. Oleh sebab itu, kritik dan saran yang bersifat membangun sangat penulis harapkan demi kesempurnaan tugas akhir ini. Akhir kata, penulis berharap semoga tugas akhir ini dapat membawa manfaat bagi banyak pihak.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i
LEMBAR PENGESAHAN v
ABSTRAK vii
ABSTRACT ix
KATA PENGANTAR xi
DAFTAR ISI xiii
DAFTAR GAMBAR xvii
DAFTAR TABEL xix
DAFTAR LAMPIRAN xxi

BAB I. PENDAHULUAN
1.1 Latar Belakang Masalah 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 3
1.6 Sistematika Penulisan 3

BAB II. DASAR TEORI
2.1 Penelitian Terdahulu 7
2.2 Kriptografi 7
2.2.1 Algoritma Simetris 8
2.2.2 Algoritma Asimetris 9
2.3 Citra 11
2.4 Digitalisasi Sampling 11
2.5 *Diffie Hellman Key Exchange Algorithm* 12
2.6 *Elliptic Curve Cryptography* 11
2.6.1 Kurva eliptik pada himpunan F_p 14
2.7 Domain Parameter Kurva Eliptik 15
2.8 Wavelet 16
2.8.1 Transformasi Wavelet Diskrit 17
2.8.2 Transformasi Wavelet Diskrit Haar 21

2.8.3 Transformasi Wavelet Diskrit Daubechies 22
2.8.4 *Lifting Scheme* 24
BAB III. METODOLOGI
3.1 Objek Penelitian 27
3.2 Peralatan 27
3.3 Tahap Penelitian 27
BAB IV.PERANCANGAN DAN IMPLEMENTASI SISTEM
4.1 Perancangan Kurva Eliptik di Bidang F_p 33
4.1.1 Pembuatan semua titik (x, y) 33
4.1.2 Penjumlahan Titik pada Kurva Eliptik (x_3, y_3) 37
4.2 Representasi Piksel pada Kurva Eliptik 42
4.3 Implementasi Sistem 44
4.4 Perancangan Sistem Enkripsi ECDH 44
4.5 Perancangan Sistem Enkripsi Transformasi Wavelet Diskrit 45
4.6 Perancangan Sistem Dekripsi ECDH 48
4.7 Perancangan Sistem Dekripsi Transformasi Wavelet Diskrit 48
4.8 Matriks yang Dibentuk dari Citra dengan Format .BMP 51
4.9 Implementasi pada MATLAB 52
4.9.1 MATLAB 52
4.9.2 Pembacaan Piksel Citra pada MATLAB 52
4.9.3 Pembuatan Titik Kurva Eliptik beserta Tabel Pemetaan 54
4.9.4 Pertukaran Kunci Diffie-Hellman 55
4.9.5 Proses Pemetaan dan Enkripsi Citra 56
4.9.6 Proses Pemetaan dan Dekripsi Citra 60
4.9.7 Perancangan Antar Muka 61
BAB V. UJI COBA DAN PEMBAHASAN
5.1 Pengujian Kurva Eliptik 65
5.2 Pengujian Citra Hasil Enkripsi 67

5.3 Pengujian dengan Noise..... 71
5.4 Pengujian dengan Berbagai Ukuran dan Macam Citra 72
5.5 Pengujian dengan Parameter $a, b,$ dan p yang berbeda 73

BAB VI.PENUTUP
6.1 Kesimpulan..... 75
6.2 Saran..... 76

DAFTAR PUSTAKA..... 77
LAMPIRAN..... 79
BIODATA PENULIS.....

"Halaman ini sengaja dikosongkan."

DAFTAR GAMBAR

Halaman

Gambar 2.1 Pertukaran Kunci Diffie-Hellman 12
Gambar 2.2 Proses DWT maju 19
Gambar 2.3 Transformasi Wavelet Level 1 dan Level 2 20
Gambar 2.4 Backward DWT dua dimensi skala satu 21
Gambar 2.5 Skema kerja pada *lifting scheme* 25
Gambar 2.6 Tahap split 25
Gambar 2.7 Tahap *predict even* 26
Gambar 3.1 Tahapan Penelitian 29
Gambar 3.2 Proses Enkripsi-Dekripsi 31
Gambar 4.1 Flowchart pembuatan titik (x, y) 34
Gambar 4.2 Flowchart penjumlahan titik 41
Gambar 4.3 Diagram alir proses enkripsi 47
Gambar 4.4 Diagram alir proses dekripsi 50
Gambar 4.5 img.bmp dengan ukuran 10×10 51
Gambar 4.6 Matriks img.bmp pada MATLAB 51
Gambar 4.7 Interface MATLAB 52
Gambar 4.8 Baboon.jpg citra *grayscale* berukuran 256×256 53
Gambar 4.9 Piksel citra baboon.jpg 53
Gambar 4.10 Potongan table pemetaan 55
Gambar 4.11 Hasil penjumlahan titik dengan *shared key* 58
Gambar 4.12 Citra enkripsi awal 58
Gambar 4.13 Citra enkripsi untuk kolom 58
Gambar 4.14 Citra enkripsi akhir berukuran $2M \times 2N$ 59
Gambar 4.15 Citra enkripsi awal dari ekstraksi wavelet 60
Gambar 4.16 Citra hasil dekripsi 61
Gambar 4.17 Citra hasil dekripsi 63
Gambar 5.1 (a) Citra plain, (b) Citra plain 67
Gambar 5.2 (a) Citra Dekripsi dengan kunci $k = 15$, (b) Citra Dekripsi dengan kunci $k = 21$ 68

“Halaman ini sengaja dikosongkan”

DAFTAR TABEL

Halaman

Tabel 4.1 Tabel Hasil QR_{19} (Quadratic Residue Module) . 35
Tabel 4.2 Tabel elemen grup kurva eliptik 36
Tabel 4.3 Tabel pemetaan titik 43
Tabel 5.1 Tabel pengujian titik kurva eliptik 65
Tabel 5.2 Tabel pengujian jenis wavelet, kompleksitas citra dan kualitas citra enkripsi 70
Tabel 5.3 Tabel pengujian dengan noise 71
Tabel 5.4 Tabel hasil uji coba berbagai jenis citra dan ukuran 72
Tabel 5.5 Tabel hasil Uji Coba dengan parameter yang berbeda 73

“Halaman ini sengaja dikosongkan.”

DAFTAR LAMPIRAN

Halaman

LAMPIRAN 79
1. Kode Program titik.m 79
2. Kode Program Penjumlahan Titik (addell.m)..... 80
3. Kode Program Perkalian Titik (multell.m)..... 82
4. Kode Program Invers Modulo (invmod.m) 83
5. Kode Program Modulo Berpangkat (powermod.m)..... 84
6. Kode Program Enkripsi (enkrip.m) 85
7. Kode Program Dekripsi (dekrip.m) 87
8. Kode Program Antar Muka (enkripsi.m)..... 88
   a. Pembentukan Titik (button bentuk titik) 88
   b. Pembuatan *Public Key* (button p3)..... 89
   c. Input Citra (button input image)..... 89
   d. Enkripsi (button enkrip) 89
   a. Dekripsi (button dekrip) 90

“Halaman ini sengaja dikosongkan.”

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Keamanan adalah isu yang penting dalam transmisi data dan informasi. Pada saart ini, berjuta-juta data dan informasi telah ditransmisikan pada jaringan komputer. Permasalahan keamanan suatu data memang menjadi hal yang sensitif bagi sebagian pihak seperti pihak departemen pertahanan dan pihak medis.

Data citra menjadi data yang paling rawan untuk disadap pada jaringan komputer. Data tersebut biasanya dikirimkan dalam jaringan yang tidak aman sehingga data tersebut dapat dengan mudah ditambah dikurangi ataupun dihilangkan oleh pihak yang tidak bertanggung jawab. Dalam citra militer, segi keamanan merupakan hal yang sangat penting karena menyangkut pertahanan dan keamanan suatu negara. Dengan berkembangnya teknologi, bukan tidak mungkin data tersebut bisa di-intercept oleh pihak lain dan hal tersebut akan membuat suatu negara menjadi terancam. Sehingga kita harus bisa memastikan bahwa citra yang ditransmisikan tersebut dapat divalidasi integritas dan kerahasiaannya.

Berbagai hal telah dilakukan untuk mendapatkan jaminan keamanan informasi rahasia. Salah satu cara yang digunakan adalah dengan menyandikan isi informasi menjadi kode-kode yang tidak dimengerti. Hal ini biasa disebut dengan kriptografi. Pada Kriptografi terdapat dua konsep utama yaitu enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali hasil enkripsi menjadi bentuk informasi semula.

Dalam perkembangannya, banyak algoritma enkripsi yang ditawarkan untuk mengamankan suatu citra. Algoritma enkripsi yang biasa digunakan yaitu dengan mengganti nilai suatu piksel atau mengganti lokasi piksel tersebut. Algoritma enkripsi tersebut

dapat diklasifikasikan menjadi enkripsi citra pada domain frekuensi dan enkripsi citra pada domain spasial. Enkripsi pada domain frekuensi didesain untuk mengganti data/piksel pada citra di dalam domain frekuensi, seperti *Discrete Fourier Transform*[1], *Quantum Fourier Transform*[2], dan *Reciprocal-Orthogonal Parametic Transform*[3]. Enkripsi pada domain spasial dikenal sebagai teknik substitusi-permutasi dengan mengganti nilai piksel dari suatu citra dan lokasi piksel suatu citra. Contoh dari proses subtitusi-permutasi adalah enkripsi *P-Fibonacci Transform*[4], *Wave Transmission*[5], *Elliptic Curve Cryptography*[6], dan *Chaotic System*[7]. Kedua algoritma tersebut menghasilkan hasil enkripsi citra dengan keamanan yang tinggi. Namun, kedua algoritma tersebut hanya menghasilkan dua jenis keluaran yaitu *noise-like* dan *texture-like*. Kedua keluaran tersebut memancing orang untuk melakukan analisis terhadap hasil enkripsi tersebut.Termasuk didalamnya yaitu kriptanalisis, modifikasi dan menghapus konten citra. Sehingga meningkatkan kemungkinan hasil enkripsi tersebut untuk di-intercept oleh orang lain[8].

Berdasarkan latar belakang serta kajian dari beberapa penelitian tersebut, pada tugas akhir ini telah dikembangkan algoritma enkripsi citra digital dengan menggunakan metode kurva eliptik Diffie-Hellman dan transformasi wavelet diskrit sehingga keluaran yang dihasilkan berupa gambar bermakna atau gambar yang tidak buram

1.2 Rumusan Masalah

Rumusan masalah yang dibahas dalam Tugas Akhir, yaitu:

1. Bagaimana melakukan proses enkripsi dan dekripsi citra digital menggunakan Kurva Eliptik Diffie-Hellman dan metode *Discrete Wavelet Transform.*
2. Bagaimana hasil dan implementasi setelah dilakukan proses enkripsi dan dekripsi citra digital menggunakan Kriptografi Kurva Eliptik Diffie Hellman dan *Discrete Wavelet Transform*.

1.3 Batasan Masalah

Batasan masalah yang digunakan dalam Tugas Akhir, yaitu:

1. Citra yang digunakan adalah citra Grayscale.
2. Menggunakan software MATLAB sebagai sarana simulasi.
3. Citra harus memiliki panjang dan lebar yang sama.

1.4 Tujuan Penelitian

Tujuan yang ingin dicapai dalam Tugas Akhir, yaitu:

1. Melakukan enkripsi dan dekripsi citra digital dengan menggunakan Kurva Eliptik Diffie-Hellman dan metode *Discrete Wavelet Transform* .
2. Mendapatkan hasil enkripsi dan dekripsi citra digital sehingga hasil enkripsi tersebut tidak dikenali lagi sebagai citra awal berupa gambar bermakna bukan gambar buram.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari Tugas Akhir, yaitu:

1. Bagi institusi medis, militer dan sebagainya, penelitian ini bisa menjadi salah satu alternatif untuk mengamankan data citra digital.
2. Bagi institusi pendidikan, penelitian ini bisa menjadi rujukan bagi penelitian-penelitian yang akan dikembangkan selanjutnya terutama pada bidang pengamanan data citra digital.
3. Bagi masyarakat, penelitian ini bisa menjadikan masyarakat sedikit lebih tenang dan tidak khawatir terutama pada data citra yang akan ditransmisikan.

1.6 Sistematika Penulisan

Sistematika penulisan didalam Tugas Akhir ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakan pembuatan tugas akhir, rumusan dan batasan permasalahan yang

dihadapi dalam penelitian tugas akhir, tujuan dan manfaat pembuata tugas akhir, tujuan dan manfaat pembuatan tugas akhir dan sistematika penulisan tugas akhir.

BAB II KAJIAN TEORI

Bab ini menjelaskan tentang penelitian sebelumnya yang mengkaji metode enkripsi citra digital. Selain itu, pada bab ini akan dijelaskan kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini, meliputi Pengertian Citra Digital, Digitalisasi spasial, Kriptografi, Algoritma Simetris, Algoritma Asimetris, *Diffie-Hellman Key-Exchange Algorithm*, *Elliptic Curve Cryptography* dan *Discrete Wavelet Transform*.

BAB III METODE PENELITIAN

Bab ini berisi metodologi atau urutan pengerjaan yang dilakukan dalam menyelesaikan tugas akhir, meliputi studi literatur, analisis dan desain kriptosistem, pembuatan program, uji coba dan evaluasi program, penarikan kesimpulan, penulisan laporan tugas akhir.

BAB IV PERANCANGAN DAN IMPLEMENTASI

Bab ini menjelaskan analisi desain kriptosistem. Pada tahapan ini akan dilakukan analisis citra dan parameter-parameter yang dibutuhkan dalam pembuatan desain program. Sistem ini memiliki dua proses utama yaitu enkripsi dan dekripsi. Perancangan sistem juga terdiri dari perancangan kurva eliptik di bidang F_p, pembuatan semua titik, penambahan titik pada kurva eliptik, representasi piksel pada kurva eliptik, pertukaran kunci Diffie-

Hellman, perancangan sistem ECDH, tranformasi wavelet diskrit, matriks yang dibentuk dari citra, dan implementasi pada MATLAB.

BAB V PENGUJIAN DAN PEMBAHASAN HASIL

Pada tahap ini akan dilakukan pengujian terhadap citra yang telah dienkripsi dan sistem yang telah dibangun. Diantaranya adalah pengujian titik kurva eliptik dan pengujian citra hasil enkripsi.

BAB VI PENUTUP

Bab ini merupakan penutup, berisi tentang kesimpulan yang dapat diambil berdasarkan data yang ada dan saran yang selayaknya dilakukan bila tugas akhir ini dilanjutkan.

"Halaman ini sengaja dikosongkan."

BAB II
DASAR TEORI

Bab ini menjelasakn tentang kajian teori dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam tugas akhir ini, meliputi Penelitian Sebelumnya dan beberapa dasar-dasar teori yang akan dijelaskan pada sub bab selanjutnya.

2.1 Penelitian Sebelumnya

Ram Ratan, Manoj Ahke (2013) telah melakukan penelitian kriptografi kurva eliptik Diffie-Hellman untuk mengamankan informasi *hypertext* pada *Wide Area Network* (WAN). Berbeda dengan penelitan tersebut yang menggunakan informasi hypertext sebagai objek penelitian, pada tugas akhir ini objek penelitian yang akan digunakan adalah citra digital.

Nidhal Khdhair E. A. et al (2016) telah melakukan penelitian tentang enkripsi citra digital menggunakan algoritma Diffie-Hellman dengan *Singular Value Decomposition*. Berbeda dengan penelitian tersebut yang menggunakan metode Diffie-Hellman dan *Singular Value Decomposition* , tugas akhir ini menggunakan kurva eliptik yang akan diterapkan pada algoritma Diffie-Hellman.

Doni Rubiagatra (2016) telah melakukan penelitian terkait enkripsi citra sebelumnya yaitu dengan melakukan enkripsi citra dengan metode kurva eliptik Diffie-Hellman. Namun, pada penelitian tersebut citra yang dihasilkan masih berupa citra noise (buram), hal ini sangat mudah mengundang para ahli untuk melakukan kriptanalisis terhadap citra tersebut karena sangat terlihat mencurigakan.

2.2 Kriptografi

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan, dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Pada kriptografi terdapat dua proses utama yaitu enkripsi

dan dekripsi. Enkripsi adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa bantuan pengetahuan khusus. Dekripsi merupakan kebalikan dari enkripsi yaitu upaya pengolahan data menjadi sesuatu yang dapat diutarakan secara jelas dan tepat dengan tujuan agar dapat dimengerti oleh orang yang tidak langsung mengalaminya sendiri.

Berdasarkan jenis kunci yang digunakan, algoritma kriptografi dibagi menjadi dua bagian yaitu algoritma simetrsi dan algoritma asimetris.

2.2.1 Algoritma simetris

Algoritma simetris merupakan kriptografi konvensional dengan menggunakan satu kunci enkripsi. Algoritma ini sangat luas digunakan sebelum berkembangnya algoritma asimetris pada tahun 1970. Algoritma yang memakai kunci simetris diantaranya[9] :

1. *Data Encryption Standard* (DES) termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher* blok. DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (*internal key*) atau upa-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit.
2. *Advanced Encryption Standard* (AES) merupakan standar enkripsi dengan kunci-simetris yang diadopsi oleh pemerintah Amerika Serikat. Standar ini terdiri atas 3 blok cipher, yaitu AES-128, AES-192 and AES-256, yang diadopsi dari koleksi yang lebih besar yang awalnya diterbitkan sebagai Rijndael. Masing-masing cipher memiliki ukuran 128-bit, dengan ukuran kunci masing-masing 128, 192, dan 256 bit.

3. *International Data Encryption Algorithm* (IDEA) merupakan algoritma simetris yang beroperasi pada sebuah blok pesan terbuka dengan lebar 64-bit. Dan menggunakan kunci yang sama , berukuran 128-bit, untuk proses enkripsi dan dekripsi. Pesan rahasia yang dihasilkan oleh algoritma ini berupa blok pesan rahasia dengan lebar satu ukuran 64-bit. Pesan dekripsi menggunakan blok penyandi yang sama dengan blok proses enkripsi dimana kunci dekripsinya diturunkan dari dari kunci enkripsi.
4. A5 yakni suatu aliran kode yang digunakan untuk mengamankan percakapan telepon selular GSM. Aliran kodenya terdiri dari 3 buah Linear Feedback Shift Register(LSFR) yang dikontrol oleh blok dengan LSFR 19 bit, 22 bit, dan 23 bit. Masing-masing dari LSFR memiliki periode berturut-turut.
5. One Time Pad (OTP) yakni algoritma yang berisi deretan kunci yang dibagikan secara acak. Setiap kunci hanya digunakan sekali pakai. Pemilihan kunci harus secara acak agar tidak bisa diproduksi ulang dan membuat lawan tidak mudah menerka. Jumlah karakter kunci sama dengan jumlah karakter yang dimiliki pesan.
6. RC2, RC4, RC5, RC6 dan lainnya.

2.2.2 Algoritma asimetris

Algoritma asimetris adalah algoritma yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Pada algoritma ini, proses enkripsinya menggunakan kunci publik dimana kunci tersebut tidak perlu dijaga kerahasiaannya. Dan untuk proses dekripsinya menggunakan kunci privat dan bersifat rahasia. Algoritma yang memakai kunci asimetris diantaranya adalah[9]:

1. *Digital Signature* Algorithm (DSA) merupakan algoritma kriptografi otentikasi pesan yang menggunakan teknologi kunci publik dan Secure Hash Algorithm (SHA-1) dalam operasinya. Secara umum DSA dapat dideskripsikan sebagai algoritma kriptografi yang memproses pesan dalam sekumpulan bit (block)/ satuan waktu tertentu dengan menggunakan sepasang kunci publik dan kunci privat bagi proses pembentukan dan verifikasi tanda tangan digital.

2. RSA merupakan salah satu algoritma public key yang populer dipakai dan bahkan masih dipakai hingga saat ini. Kekuatan algoritma ini terletak pada proses eksponensial, dan pemfaktoran bilangan menjadi 2 bilangan prima yang hingga kini perlu waktu yang lama untuk melakukan pemfaktorannya. Skema RSA sendiri mengadopsi dari skema block cipher, dimana sebelum dilakukan enkripsi, plainteks yang ada dibagi – bagi menjadi blok – blok dengan panjang yang sama, dimana plainteks dan cipherteksnya berupa integer(bilangan bulat) antara 1 hingga n, dimana n berukuran biasanya sebesar 1024 bit, dan panjang bloknya sendiri berukuran lebih kecil atau sama dengan log(n) +1 dengan basis 2.

3. Diffie-Hellman yakni algoritma yang memiliki keamanan dari kesulitan untuk menghitung logaritma diskrit dalam *finite field,* dibandingkan kemudahan dalam menghitung bentuk eksponensial dalam *finite field* yang sama. Algoritma ini dapat digunakan dalam mendistribusikan kunci publik yang dikenal dengan protokol pertukaran kunci.

4. *Elliptic Curve Cryptography*(ECC) yakni algoritma yang mendasarkan keamanannya pada permasalahan matematis kurva eliptik. Tidak seperti permasalahan matematis logaritma diskrit dan pemfaktoran bilangan bulat, tidak ada algoritma waktu sub-10 eksponensial yang diketahui

memecahkan permasalahan matematis algoritma kurva eliptik.

2.3 Citra

Citra menurut Kamus Besar Bahasa Indonesia adalah rupa, gambar, atau gambaran. Sedangkan menurut kamus Webster citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda. Citra terbagi menjasi dua yaitu citra diam dan citra bergerak. Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan, citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun sehingga memberi kesan pada mata kita sebagai gambar yang bergerak.

2.4 Digitalisasi Sampling

Sampling merupakan proses pengambilan informasi dari citra analog yang memiliki panjang dan lebar tertentu untuk membaginya ke beberapa blok kecil. Blok-blok tersebut disebut piksel. Sehingga citra digital yang lazim dinyatakan dalam bentuk matriks yang memiliki ukuran $M \times N$ dengan M baris dan N kolom. Dapat disebut juga sebagai citra digital yang memiliki $M \times N$ buah piksel. Notasi matriks citra digital dapat dinyatakan sebagai persamaan 2.1 berikut[10]:

$$f(x,y) = \begin{bmatrix} f(0,0) & \cdots & f(0,N-1) \\ \vdots & \ddots & \vdots \\ f(M-1,0) & \cdots & f(M-1,N-1) \end{bmatrix} \quad (2.1)$$

2.5 *Diffie-Hellman Key Exchange Algorithm*

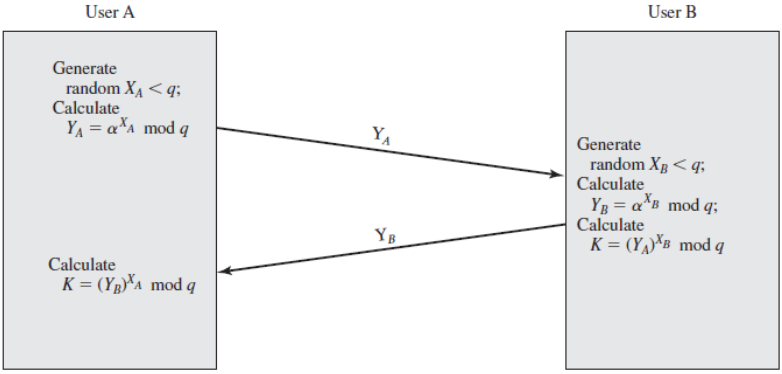
Algoritma Diffie Hellman bergantung pada kerumitannya memecahkan masalah logaritma diskrit. Sebuah akar primitif dari bilangan prima p adalah pangkat modulo p yang menggenerasi semua bilangan bulat dari 1 sampai $p-1$. Misalkan a adalah sebuah akar primitif dari p maka[9]:

$$a \bmod p, a^2 \bmod p, a^{p-1} \bmod p \quad (2.2)$$

Pada persamaan 2.2, untuk setiap bilangan bulat b dan bilangan prima p kita dapatkan eksponen i sehingga

$$b \equiv a^i (mod\ p), 0 \leq i \leq (p-1) \qquad (2.3)$$

Eksponen i dapat dikatakan sebuah logaritma diskrit b untuk basis $a\ mod\ p$ seperti pada persamaan 2.3. Kemudian akan digambarkan algoritma dari pertukaran kunci Diffie-Hellman[9].



Gambar 2.1 Pertukaran kunci Diffie-Hellman

2.6 *Elliptic Curve Cryptography*

Elliptic Curve Cryptosystem (ECC) diperkenalkan tahun 1985 oleh Neal Koblity dan Victor Miller dari Universitas Washington. Kurva eliptik mempunyai masalah logaritma yang terpisah sehingga sulit untuk dipecahkan. PadaJuni 2000 kunci enkripsi ECC yang memakai 108 bit (yang setara dengan kunci enkripsi RSA 600 bit), berhasil dipecahkan menggunakan 9500 komputer yang berjalan paralel selama 4 bulan yang dihubungkan dengan internet.

Kriptografi kurva eliptik termasuk sistem kriptografi kunci publik yang mendasarkan keamanannya pada permasalahan matematis kurva eliptik. Tidak seperti permasalahan matematis

logaritmis diskrit (*Discrete Logarithm Problem*, DLP) dan pemfaktoran bilangan bulat (*Integer Factorization Problem*, IFP), tidak ada algoritma waktu sub-eksponensial yang diketahui untuk memecahkan permasalahan matematis algoritma diskrit kurva eliptik (*Elliptic Curve Discrete Logarithm Problem*, ECDLP). Oleh karena alasan tersebut algoritma kurva eliptik mempunyai keuntungan bila dibanding algoritma kriptografi kunci publik lainnya, yaitu dalam hal ukuran kunci yang lebih pendek tetapi memiliki tingkat keamanan yang sama.

Kurva eliptik yang digunakan dalam kriptografi didefinisikan dengan menggunakan dua tipe daerah terbatas yaitu daerah karakteristik ganjil (F_p dimana $p > 3$ adalah bilangan prima yang besar) dan karakteristik dua ($F2^m$). Karena perbedaan itu tidk menjadi penting, kedua daerah terbatas tersebut dapat ditunjukkan sebagai F_p, dimana $q = p$ atau $q = 2^m$. Elemen dari F_p adalah bilangan bulat ($0 \leq x < p$) dimana eemen tersebut dapat dikombinasikan menggunakan modul aritmatik.

Pada bagian ini akan dibahas teknik dasar kurvaeliptik pada bidang terbatas F_p dimana p adalah bilangan prima lebih besar dari 3. Selanjutnya kurva eliptik secara umum didefinisikan sebagai *field* berhingga (*finite field*). Sebuah kurva eliptik E pada bidang terbatas F_p didefinisikan dalam persamaan 2.4[9]:

$$y^2 = x^3 + ax + b \tag{2.4}$$

Dimana $a, b \in F_p$ dan $4a^3 + 27b^2 \neq 0$ dan sebuah titik **O** yang disebut titik tak hingga (*infinity*). Titik tak hingg adalah identitas atau titik ideal. Himpunan $E(F_p)$ adalah semua titik (x, y)untuk $x, y \in F_p$ yang memenuhi persamaan 2.4.

Untuk menjelaskan uraian di atas, berikut ini diberikan contoh pencarian pada R dan $E(F_p)$. Diberikan persamaan kurva eliptik $E: y^2 = x^3 + x + 1$. Untuk $E(F_p)$ dipilih $p = 23$ sehingga grup $F_{23}(a = 1, b = 1)$. Maka untuk nilai

$$4a^3 + 27b^2 = 4 + 27 \neq 0$$

Membuat E ada dalam kurva eliptik.

2.6.1 Kurva eliptik pada himpunan F_p

Pada bidang terbatas F_p perhitungan dilakukan dengan menggunakan aturan-aturan aritmatika modular. Persamaan kurva eliptik pada F_p dapat dituliskan sebagai berikut :

$$y^2 mod\ p = (x^3 + ax + b) mod\ p \quad a, b \in F_p \tag{2.5}$$

Dengan p adalah bilangan prima ganjil. $F_p(a, b)$ adalah himpunan yang terdiri dari titik-titik yang memenuhi persamaan 2.5 ditambah dengan titik **O** yang disebut titik infinity. Kurva eliptik pada bidang terbatas merupakan grup abelian, apabila sisi kanan persamaan 2.5 tidak memiliki faktor yang berulang yaitu apabila koefisien-koefisiennya memenuhi persamaan $a^3 + b^2 (mod\ p) \neq 0 (mod\ p)$.

Penjumlahan dua buah titik $P(x_1, y_1)$ dan $Q(x_2, y_2)$ adalah (x_3, y_3) dengan syarat bahwa $P \neq \mathbf{O}$ dan $Q \neq P$. Secara aljabar, (x_3, y_3) diperoleh dengan rumus berikut :

$$\lambda = \left(\frac{y_2 - y_1}{x_2 - x_1}\right) mod\ p \quad \text{Jika } P \neq Q \tag{2.6}$$

$$\lambda = \left(\frac{3x_1^2 + a}{2y_1}\right) mod\ p \quad \text{Jika } P = Q \tag{2.7}$$

$$x_3 = \lambda^2 - x_1 - x_2 \tag{2.8}$$

$$y_3 = -y_2 + \lambda(x_1 - x_3) \quad (2.9)$$

Operasi penjumlahan pada kurva eliptik atas F_p didefinisikan sebagai berikut :

a. **O** adalah identitas penjumlahan, sehingga $P + \mathbf{O} = \mathbf{O} + P = P$ untuk setiap $P \in E(F_p)$

b. Jika $P = (x, y)$ maka $P + (x, -y) = \mathbf{O}$. Titik $(x, -y)$ adalah negatif P atau $(-P)$

c. Misalkan $P = (x_1, y_1) \in E(F_p)$ dan titik $P = (x_2, y_2) \in E(F_p)$ dimana $P \neq \mathbf{O}$, $Q \neq \mathbf{O}$ dan $Q \neq \pm P$. Maka $P + Q = (x_3, y_3)$ dimana :

$$x_3 = \left[\left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2\right] mod\ p \quad (2.10)$$

$$y_3 = \left[-y_1 + \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3)\right] mod\ p \quad (2.11)$$

d. Misalkan $P = (x_1, y_1) \in E(F_p)$ maka $P + P = 2P = (x_3, y_3)$ dimana :

$$x_3 = \left[\left(\frac{3x_1^2 + a}{2y_1}\right)^2 - x_1 - x_2\right] mod\ p \quad (2.12)$$

$$y_3 = \left[-y_1 + \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3)\right] mod\ p \quad (2.13)$$

2.7 Domain Paramater Kurva Eliptik

Sebelum mengimplementasikan sebuah kriptografi kurva eliptik, dipersiapkan parameter yang dibutuhkan oleh sistem kriptografi tersebut. Sehingga seluruh pengguna sistem dapat mengetahui beberapa parameter yang akan digunakan bersama. Parameter ini bersifat umum dan boleh diketahui oleh setiap pengguna sistem tersebut.

Pembuatan domain parameter tersebut tidak dilakukan oleh masing-masing pengirim atau penerima karena akan melibatkan

perhitungan jumlah titik pada kurva yang akan memakan waktu lama dan sulit untuk diterapkan. Domain parameter kurva eliptik atas F_p didefinisikan sebagai persamaan 2.14 berikut :

$$T = (p, a, b, G, n, h) \quad (2.14)$$

Dimana

p : bilangan prima

a, b : koefisien persamaan kurva eliptik

G : titik dasar (base point) yaitu elemen pembangun grup kurva eliptik

n : order dari G yaitu bilangan bulat positif terkecil $\ni n, G = 0$

H : kofaktor, $h = \frac{\#E}{n}$, $\#E$ adalah jumlah titik dalam grup eliptik $E_p(a, b)$

Kekuatan kriptografi kurva eliptik bergantung dari pemilihan parameter domain yang digunakan. Pemilihan parameter ini dilakukan sehingga dapat terhindar dari serangan terhadap kekuatan algoritma kriptografi kurva elliptik.

2.8 Wavelet

Wavelet merupakan sebuah basis yang berasal dari sebuah fungsi penskalaan atau dikatakan juga sebuah *scaling function. Scaling function* memiliki sifat yaitu dapat disusun dari sejumlah salinan dirinya yang telah didilasikan, ditranslasikan dan diskalakan. Wavelet merupakan sebuah fungsi variabel real x, diberi notasi ψ_t dalam ruang fungsi $L^2(R)$. Fungsi ini dihasilkan oleh parameter dilasi dan translasi, yang dinyatakan dengan persamaan[10] :

$$\psi_{a,b}(x) = a^{-\frac{1}{2}}\psi\left(\frac{x-b}{a}\right); a > 0, \quad a, b \in R \quad (2.15)$$

$$\psi_{a,b}(x) = 2^{\frac{a}{2}}\psi(2^a x - b); a, b \in R \quad (2.16)$$

Fungsi wavelet pada persamaan (1) diperkenalkan pertama kali oleh Grossman dan Morlet, sedangkan persamaan (2) oleh

Daubechies. Pada fungsi Grossman-Morlet, a adalah parameter dilasi dan b adalah parameter translasi, sedangkan pada fungsi Daubechies, parameter dilasi diberikan oleh Z^j dan parameter translasi oleh k. Kedua fungsi ψ dapat dipandang sebagai mother wavelet, dan harus memenuhi kondisi :

$$\int \psi(x)\, dx = 0 \qquad (2.17)$$

Berdasarkan nilai parameter translasi dan dilatasinya, transformasi wavelet dibedakan menjadi dua tipe, yaitu Continue Wavelet Transform (CWT) dan Discrete Wavelet Transform (DWT).

2.8.1 Transformasi Wavelet Diskrit

Transformasi wavelet merupakan sebuah fungsi konversi yang dapat digunakan untuk membagi suatu fungsi atau sinyal ke dalam komponen frekuensi yang berbeda, yang selanjutnya komponen-komponen tersebut dapat dipelajari sesuai dengan skalanya.

Sesuai dengan fungsi *mother wavelet* di atas, bahwa fungsi wavelet penganalisis untuk transformasi wavelet diskrit dapat didefinisikan sebagai[10] :

$$\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi\left(2^j x - k\right); j, k \in Z \qquad (2.18)$$

dengan :

Z = mengkondisikan nilai j dan k dalam nilai integer
j = parameter frekuensi atau skala
k = parameter waktu atau lokasi ruang

berdasarkan fungsi di atas, representasi fungsi sinyal $f(t) \in L^2(R)$ dalam domain wavelet diskrit didefinisikan sebagai :

$$f(t) = \sum a_{j,k}\psi_{j,k}(t) \qquad (2.19)$$

$a_{j,k}$ ini dibentuk oleh inner produk antara fungsi wavelet induk dengan $f(t)$:

$$a_{j,k} = \left(\psi_{j,k} f(t)\right) \tag{2.20}$$

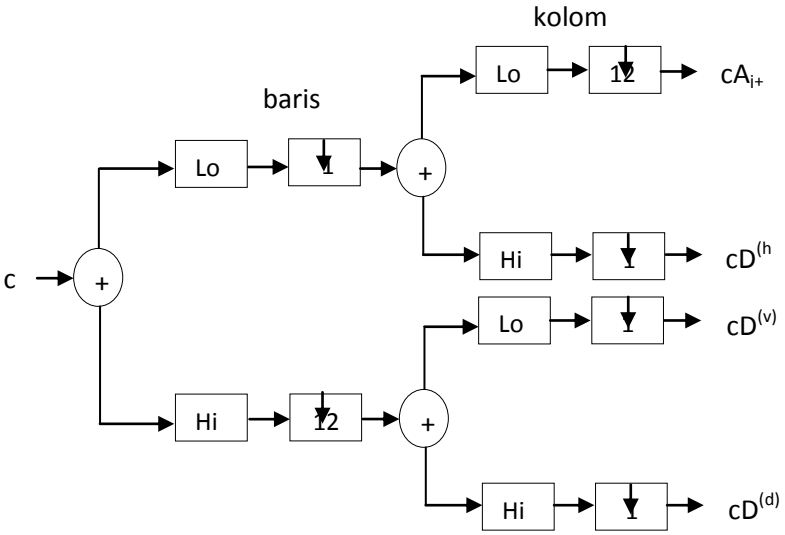
sehingga $f(t)$ dapat dinyatakan dengan :

$$f(t) = \sum \left(\psi_{j,k} f(t)\right) \psi_{j,k}(t) \tag{2.21}$$

barisan koefisien $a_{j,k}$ pada persamaan (6) merupakan Transformasi Wavelet Diskrit dari fungsi $f(t)$, sehingga $f(t)$ pada persamaan (7) disebut sebagai *inverse* Transformasi Wavelet Diskrit. Transformasi wavelet diskrit (DWT) dikelompokkan menjadi dua yaitu DWT maju dan DWT balik.

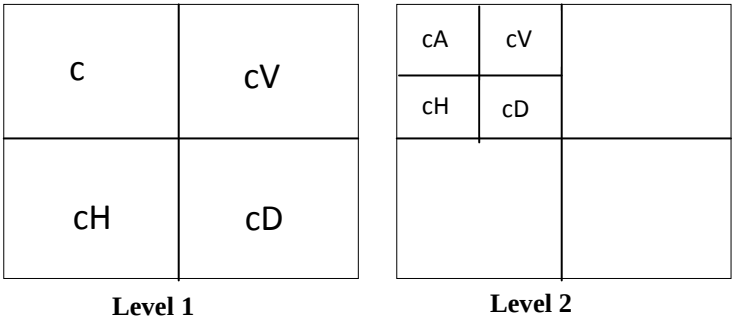
a. Transformasi Wavelet Diskrit Maju (Forward DWT)

DWT maju merupakan proses dekomposisi data citra, yang dimulai dengan melakukan dekomposisi terhadap baris data citra dan dekomposisi terhadap kolom pada koefisien citra keluaran dari tahap pertama[12]. Proses DWT maju, ditunjukkan pada Gambar 2.2 :



Gambar 2.2 Proses DWT maju

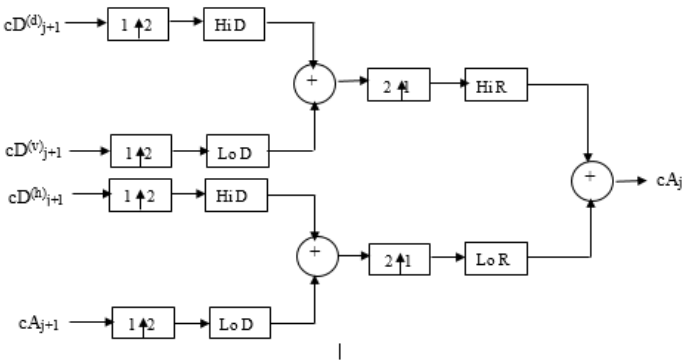
Sinyal citra masuk, didekomposisi menggunakan Lo_D (Low Pass Filter Decomposition) dan Hi_D (High Pass Filter Decomposition) dan dilakukan downsampling dua. Sinyal keluaran berfrekuensi rendah dan tinggi. Proses Lo_D dan Hi_D tersebut dilakukan terhadap baris dan terhadap kolom sebanyak dua kali, diperoleh empat subband keluaran berupa informasi frekuensi rendah dan informasi frekuensi tinggi, yaitu koefisien aproksimasi, koefisien detail horizontal, koefisien detail vertikal, dan koefisien detail diagonal[12]. Dekomposisi transfomasi wavelet , ditunjukkan pada Gambar 2.3 :



Gambar 2.3 Transformasi Wavelet Level 1 dan Level 2

b. Transformasi Wavelet Diskrit Balik (Invers DWT)

DWT balik (IDWT) adalah kebalikan dari DWT maju, yaitu proses rekonstruksi dengan arah yang berlawanan dari proses dekomposisi. Proses up-sampling dan pem-filter-an dengan koefisien-koefisien filter balik. Proses up-sampling yaitu proses mengembalikan dan menggabungkan kesinyal semula, yaitu dengan menyisipkan sebuah kolom nol di antara setiap kolom dan melakukan konvolusi pada setiap baris dengan filter berdimensi satu begitu juga menyisipkan sebuah baris nol di antara setiap baris dan melakukan konvolusi setiap baris dengan filter yang lainnya. Filter yang digunakan pada transformasi balik harus sesuai dengan filter pada sisi dekomposisi yaitu filter Lo_R (Low Pass Filter Reconstruction) dan Hi_R (High Pass Filter Reconstruction)[12]. Proses DWT balik ditunjukkan pada gambar 2.4 :



Gambar 2.4 Backward DWT dua dimensi skala satu

2.8.2 Transformasi Wavelet Diskrit Haar

Dalam matematika, wavelet Haar adalah fungsi skala "berbentuk persegi" yang bersama-sama membentuk sebuah keluarga wavelet atau dasar. analisis wavelet ini mirip dengan analisis Fourier yang memungkinkan fungsi sasaran pada interval untuk diwakili dalam hal basis ortonormal. Transformasi Haar sekarang dikenal sebagai dasar wavelet pertama dan banyak digunakan.

Transformasi Haar diusulkan pada tahun 1909 oleh Alfréd Haar. [1] Haar digunakan fungsi-fungsi ini untuk memberikan contoh dari sistem ortonormal untuk ruang fungsi persegi terintegral pada unit interval [0, 1].

Haar wavelet adalah wavelet paling sederhana. Kerugian teknis dari wavelet Haar adalah bahwa Haar wavelet tidak kontinu, dan karena itu tidak terdiferensiasi.

Adapun wavelet Haar level-1 didefinisikan sebagai berikut [14] :

$W_1^1 = (\frac{1}{\sqrt{2}}, \frac{-1}{\sqrt{2}}, 0,0, \dots \dots, 0)$

$W_2^1 = (0,0,\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}},0,0,\dots\dots,0)$

$\vdots$

$W_{\frac{N}{2}}^1 = (0,0,\dots\dots,0,\frac{1}{\sqrt{2}},\frac{-1}{\sqrt{2}})$ (2.22)

Dengan menggunakan transformasi wavelet level 1. Kita dapat menyatakan hasil fluktuasi sub-sinyal d^1 sebagai produk skalar. Misalkan :

$$d_1 = \frac{f_1 - f_2}{\sqrt{2}} = \mathrm{f} \cdot W_1^1$$

Dengan cara yang sama, $d_2 = \mathrm{f} \cdot W_2^1$ dan seterusnya. Sehingga produk skalar Haar wavelet level-1 bisa dinyatakan dengan :

$$d_m = \mathrm{f} \cdot W_m^1 \quad (2.23)$$

Adapun *scalling function* dari transformasi Haar wavelet level-1 didefinisikan sebagai :

$W_1^1 = (\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}},0,0,\dots\dots,0)$

$W_2^1 = (0,0,\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}},0,0,\dots\dots,0)$

$\vdots$

$W_{\frac{N}{2}}^1 = (0,0,\dots\dots,0,\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$ (2.24)

Dengan menggunakan *scalling function*, kecenderungan sinyal a dinyatakan dengan produk skalar :

$$a_m = \mathrm{f} \cdot V_m^1 \quad (2.23)$$

2.8.3 Transformasi Wavelet Diskrit Daubechies

Transformasi wavelet Daubechies didefinisikan dengan cara yang sama dengan transformasi wavelet Haar dengan menghitung rata-rata dan perbedaan melalui hasil skalar dengan penskalaan sinyal dan wavelet. Satu-satunya perbedaan antara transformasi Haar dan transformasi Daubechies terletak pada bagaimana penskalaan sinyal dan wavelet didefinisikan. Terdapat

banyak macam transformasi Daubechies, namun transformasi tersebut memiliki banyak kesamaan. Pada sub-bab ini akan dibahas wavelet Daub4 karena merupakan salah satu yang paling sederhana.

Perbedaan transformasi Haar dan transformasi Daub4 pada cara mendefinisikan penskalaan sinyal dan wavelet. *Scalling numbers* $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ didefinisikan sebagai berikut [14]:

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}} \quad (2.24)$$

Dengan menggunakan *scalling numbers*, penskalaaan sinyal level-1 Daub4 didefinisikan sebagai berikut :

$V_1^1 = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, 0,0, \dots \dots, 0)$

$V_2^1 = (0,0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, 0,0, \dots \dots, 0)$

$V_2^1 = (0,0,0,0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, 0,0, \dots \dots, 0)$

$\vdots$

$V_{\frac{N}{2}-1}^1 = (0,0, \dots ., 0, \alpha_1, \alpha_2, \alpha_3, \alpha_4)$

$V_{\frac{N}{2}}^1 = (\alpha_3, \alpha_4, 0,0, \dots \dots, 0, \alpha_1, \alpha_2)$ (2.25)

Sehingga dari persamaan 2.25 fungsi penskalaan level-1 bisa di-generalisasi menjadi :

$$V_m^1 = \alpha_1 V_{2m-1}^0 + \alpha_2 V_{2m}^0 + \alpha_3 V_{2m+1}^0 + \alpha_4 V_{2m+2}^0 \quad (2.26)$$

Pada wavelet Daub4 pendefinisian persamaan hampir sama dengan penskalaan sinyal. *Wavelet numbers* $\beta_1, \beta_2, \beta_3, \beta_4$ didefinisikan sebagai berikut :

$$\beta_1 = \frac{1-\sqrt{3}}{4\sqrt{2}}, \beta_2 = \frac{\sqrt{3}-3}{4\sqrt{2}}, \beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \beta_4 = \frac{-1-\sqrt{3}}{4\sqrt{2}} \quad (2.27)$$

Dengan menggunakan *wavelet numbers*, wavelet level-1 Daub4 didefinisikan sebagai berikut :

$$W_1^1 = (0,0,\beta_1,\beta_2,\beta_3,\beta_4,0,0,\dots\dots,0)$$
$$W_2^1 = (0,0,0,0,\beta_1,\beta_2,\beta_3,\beta_4,0,0,\dots\dots,0)$$
$$\vdots$$
$$W_{\frac{N}{2}-1}^1 = (0,0,\dots.,0,\beta_1,\beta_2,\beta_3,\beta_4)$$
$$W_{\frac{N}{2}}^1 = (\beta_3,\beta_4,0,0,\dots\dots,0,\beta_1,\beta_2) \quad (2.28)$$

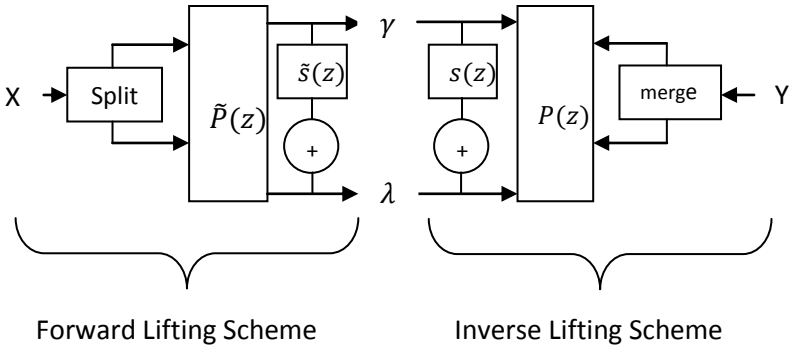
Sehingga dari persamaan 2.28 wavelet Daub4 bisa digeneralisasi menjadi :

$$W_m^1 = \beta_1 W_{2m-1}^0 + \beta_2 W_{2m}^0 + \beta_3 W_{2m+1}^0 + \beta_4 V_{2m+2}^0 \quad (2.29)$$

2.8.4 *Lifting Scheme*

Lifting Scheme adalah teknik untuk melakukan dua hal, yaitu merancang wavelet dan melakukan transformasi wavelet diskrit. *Lifting scheme* ini dimanfaatkan untuk menggabungkan langkah-langkah dan desain filter wavelet, sementara itu dapat tetap melakukan transformasi wavelet. Karena hal inilah *lifting scheme* disebut sebagai transformasi wavelet generasi kedua. Teknik ini diperkenalkan oleh Wim Sweldens [].

Teknik ini mempermudah dalam proses *Forward Lifting Scheme* (DWT) dan *Inverse Lifting Scheme* (IDWT) karena memiliki nilai integer.Gambar 2.5 menggambarkan bagaimana skema kerja pada *lifting scheme* berlangsung [15].

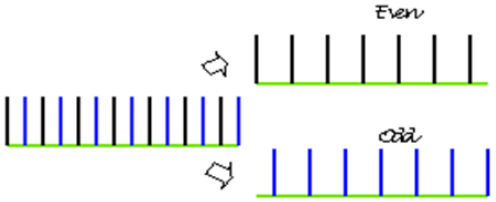


Gambar 2.5 Skema kerja pada *lifting scheme*

Pada proses Forward Lifting Scheme dibagi menjadi 3 tahapan yaitu tahap *Split*, tahap *Predict*, dan tahap *Update.*

1. *Split*

   Split adalah tahap memisahkan data sesuai dengan algoritma. Sebelum memasuki filter, data yang akan diproses dibagi-bagi menjadi 2 bagian yaitu *Even* (genap) dan *Odd* (ganjil). Tahapan split pada Gambar 2.6 ini biasa dikenal dengan istilah *Lazy Wavelet* [15].

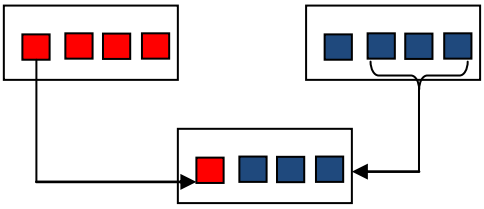


Gambar 2.6 Tahap split

2. *Predict*

   Predict merupakan tahap memprediksi. Prediksi dilakukan untuk menentukan data ganjil dari data genap. Pembuatan *predict even* dijelaskan pada Gambar 2.7. Setelah mendapatkan *predict even*, kita dapat melakukan tahap *predict* dengan menggunakan persamaan 2.30

   $$d_{n-1} = odd_n - P(even_n) \quad (2.30)$$



Gambar 2.7 Tahap *Predict even*

Keterangan : $P(even_n)$ diperoleh dari predict yang merupakan *even sample* yang diperoleh dari *odd sample* yang bersebelahan (*left neighbouring even sample*) [15].

3. *Update*

   Konsep dari *update* yaitu melanjutkan proses dari predict yang sebelumnya. Pada proses update akan dihitung nilai rata-rata dari hasil *difference* perhitungan sebelumnya. Nilai rata-rata tersebut akan dijumlahkan dengan data *even sample* (data asli). Proses *update* digambarkan pada persamaan 2.31.

   $$S_{n-1} = even_{j,i} + \left\lfloor \frac{odd_{j,i}}{2} \right\rfloor \quad (2.31)$$

BAB III
METODE PENELITIAN

Bab ini membahas mengenai metodologi sistem yang digunakan untuk menyelesaikan tugas akhir. Pembahasan metodologi sistem diawali dengan penjelasan tentang objek penelitian, peralatan yang digunakan, dan tahap penelitian.

3.1 Objek Penelitian

Objek penelitian yang akan digunakan adalah citra grayscale dari standard image test.

3.2 Peralatan

Peralatan penelitian yang akan digunakan untuk menyelesaikan tugas akhir ini adalah perangkat lunak MATLAB

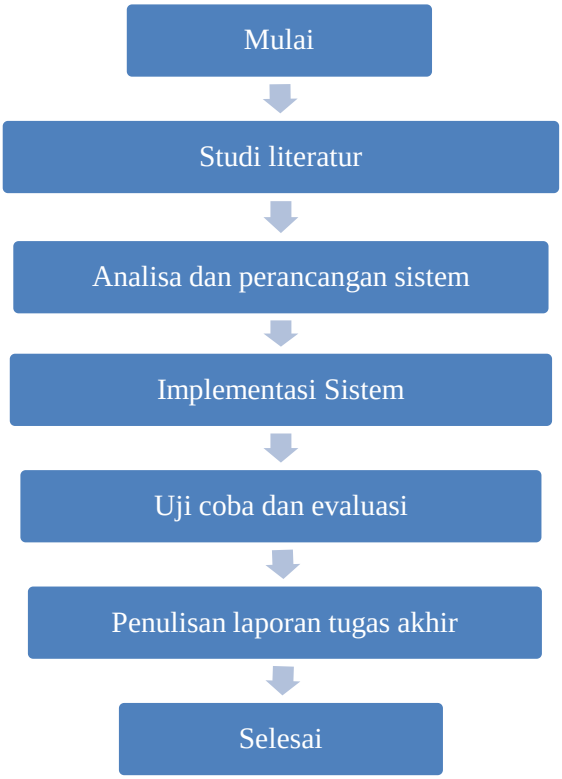
3.3 Tahap Penelitian

Adapun tahap-tahap penelitian dalam tugas akhir ini diantaranya :

1. Studi Literatur
   Pada tahap ini dilakukan pengkajian terhadap metode *elliptic curve cryptography*. Studi ini dilakukan dengan membaca jurnal- jurnal dan buku tentang enkripsi *elliptic curve cryptography* khususnya pertukaran kunci Diffie-Hellman dan *discrete wavelete transform*.
2. Analisis dan Perancangan Sistem
   Pada tahap ini dilakukan analisis citra yang akan dienkripsi dan analisa parameter-parameter diantaranya persamaan kurva eliptik, titik basis yang akan dipakai enkripsi dan bilangan prima yang sudah disepakati oleh pihak yang mengenkripsi dan mendekripsi. Sistem yang akan dirancang terdiri dari dua proses besar yaitu proses enkripsi dan dekripsi. Proses enkripsi terdiri dari dua sub proses yaitu pertukaran kunci dengan metode kurva eliptik Diffie-Hellman dan penyisipan citra hasil enkripsi menggunakan transformasi wavelet diskrit.

3. Implementasi Sistem
   Mengimplementasikan rancangan yang dilakukan pada tahap sebelumnya menjadi sebuah program berbasis MATLAB untuk analisa enkripsi citra tersebut.Sistem yang akan dibuat terdiri dari dua proses besar yaitu proses enkripsi dan dekripsi. Pada sistem tersebut terdapat parameter yang akan diinput oleh masing-masing user diantaranya kunci privat dan citra yang akan dienkripsi maupun didekripsi. Proses enkripsi dan dekripsi ini menggunakan metode kurva eliptik Diffie-Hellman dan transformasi wavelet diskrit.
4. Uji Coba dan evaluasi
   Pada tahap ini dilakukan uji coba dan simulasi dari program yang telah dibuat. Program tersebut akan diuji ketahanannya terhadap kriptanalisis. Dan akan dievaluasi apa saja *error* pada program tersebut sehingga implementasi yang didapatkan sesuai apa yang diinginkan.
5. Penarikan Kesimpulan
   Setelah dilakukan analisis, pembahasan dan setelah uji coba program. Maka akan dilakukan penarikan kesimpulan terhadap program yang telah dibuat. Kesimpulan pada hal ini berkaitan dengan waktu komputasi yang dilakukan program. Setelah dibuat kesimpulan yang tepat, akan dibuat saran dan rekomendasi untuk penelitian selanjutnya yang terkait dengan kriptografi kurva eliptik.
6. Penulisan Laporan Tugas Akhir
   Pada tahap ini, dilakukan penulisan Tugas Akhir setelah mendapatkan simulasi dan penarikan kesimpulan dan mendapatkan jawaban dari topik masalah.

Tahapan penelitian ditunjukkan pada gambar 3.1 :

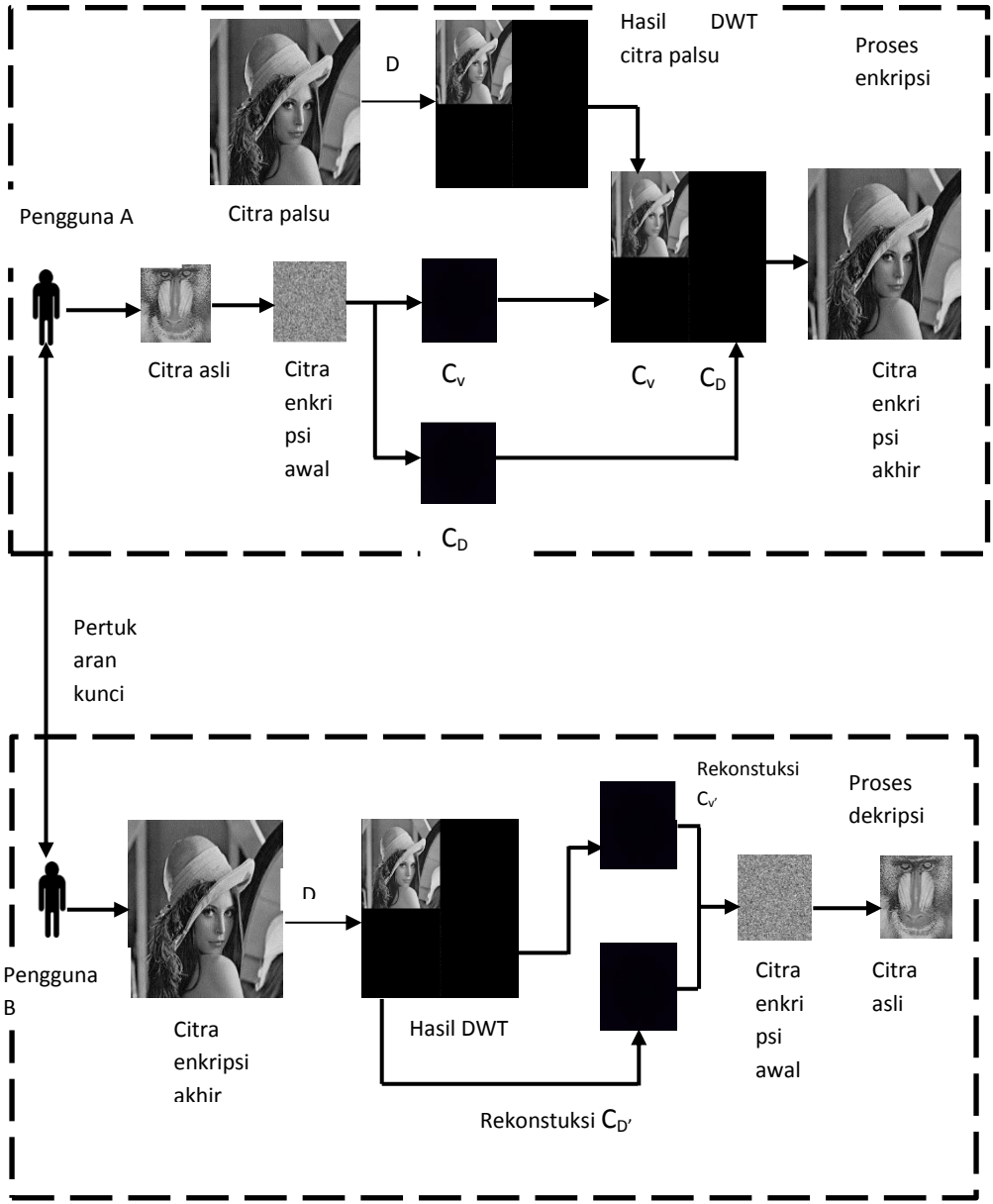


Gambar 3.1 Tahapan penelitian

Proses enkripsi citra digital pada tugas akhir ini terdiri dari dua tahapan utama yaitu:

1. Tahap Enkripsi Citra (citra buram)
   Pada tahap ini citra digital akan dienkripsi menggunakan metode kurva eliptik Diffie-Hellman sehingga didapatkan keluaran berupa citra buram/*noise*

2. Tahap Penyembunyian hasil enkripsi
   Pada tahap ini hasil enkripsi citra yang sudah didapat akan disembunyikan pada citra lain menggunakan metode transformasi wavelet diskrit



Gambar 3.2 proses enkripsi dekripsi

“Halaman ini sengaja dikosongkan.”

BAB IV
PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini akan dibahas mengenai perancangan dan implementasi sistem EC-Diffie Hellman dan Transformasi Wavelet Diskrit. Perancangan sistem meliputi perancangan kurva eliptik di bidang F_p, domain parameter, pembangkitan kunci publik dan kunci privat, perancangan sistem enkripsi dan dekripsi citra. Implementasi sistem meliputi pembuatan program secara keseluruhan dengan menggunakan MATLAB.

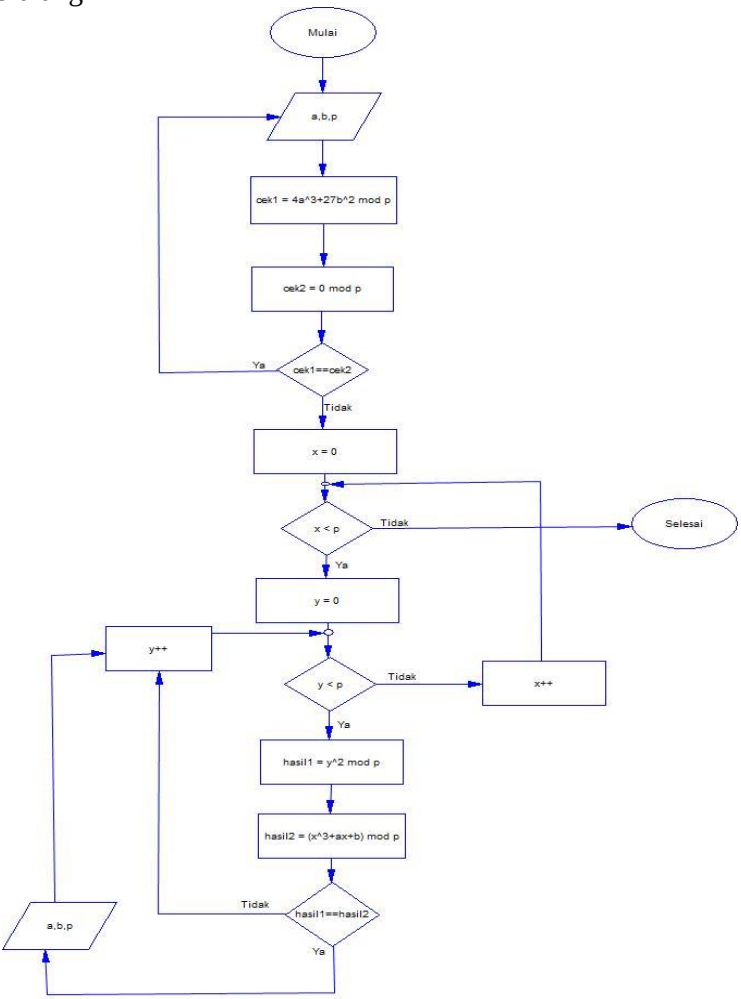
4.1 Perancangan Kurva Eliptik di Bidang F_p

Sistem kriptografi kurva eliptik tidak menggunakan kurva eliptik pada bilangan real namun menggunakan medan terbatas misalnya medan modular bilangan prima F_p. Pada bidang terbatas F_p perhitungan dilakukan dengan memperhatikan aturan-aturan aritmatika modular pada bilangan prima p. Persamaan kurva eliptik pada F_p dapat dituliskan seperti persamaan 2.5 dengan p adalah bilangan prima ganjil dan $p > 3$. $F_p(a, b)$ adalah himpunan yang terdiri atas titik-titik (x, y) yang memenuhi persamaan 2.5 ditambah dengan titik **O** yang disebut titik *infinity*. Kurva eliptik pada bidang terbatas F_p merupakan grup abelian, apabila sisi kanan persamaan 2.5 tidak memiliki faktor berulang yaitu apabila koefisien-koefisiennya memenuhi persamaan $(4a^3 + 27b^2)\ mod\ p \neq 0 (mod\ p)$.

4.1.1 Pembuatan Semua Titik (x, y)

Pada gambar 4.1 dijelaskan bahwa nilai a,b dan p berupa bilangan bulat non-negatif, untuk a dan $b \in F_p$. Kemudian nilai a dan b diproses sesuai dengan persamaan $(4a^3 + 27b^2)\ mod\ p \neq 0$, jika tidak memenuhi persamaan tersebut maka harus kembali ke awal untuk memasukkan nilai a, b dan p. Jika memenuhi persamaan tersebut maka proses akan berlanjut untuk memenuhi titik (x, y) yakni dengan syarat $x < p$, sedemikian hingga x akan

diproses sesuai persamaan $y^2 = x^3 + ax + b \ mod \ p$ sehingga akan dibagi menjadi dua proses yaitu kita misalkan $hasil\ 1 = y^2 mod\ p$ dan $hasil\ 2 = x^3 + ax + b \ mod \ p$. Jika $hasil\ 1 = hasil\ 2$ maka akan didapatkan titik (x, y), jika $hasil1 \neq hasil\ 2$ maka titik x tidak memiliki nilai y yang memenuhi dan akan berulang



Gambar 4.1 Flowchart pembuatan titik (x, y)

Contoh perhitungan secara aljabar untuk pembuatan titik kurva eliptik, diberikan persamaan kurva eliptik $E: y^2 = x^3 + x + 8$ dengan $p = 19$, yaitu grup $F_{19}(a = 1, b = 8)$. Maka untuk nilai $4a^3 + 27b^2 mod\ 19 = 4(1) + 27(64)\ mod\ 19 = 3 \neq 0$,
sehingga E telah memenuhi syarat sebagai kurva eliptik.

Untuk dapat membuat titik kurva (x, y), pertama tentukan elemen dari kurva eliptik atas $E_{19}(1,8)$ atas F_p :

$F_{19} = \{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18\}$

Sebelum membentuk semua titik (x, y) tenukan terlebih dahulu daerah elemen/ *range* kurva eliptik QR_{19} (Quadratic Residue Module)

Tabel 4.1 Hasil $\boldsymbol{QR_{19}}$ (Quadratic Residue Module)

F_p	$y^2 (mod\ 19)$	QR_{19}
0	$0^2 (mod\ 19)$	0
1	$1^2 (mod\ 19)$	1
2	$2^2 (mod\ 19)$	4
3	$3^2 (mod\ 19)$	9
4	$4^2 (mod\ 19)$	16
5	$5^2 (mod\ 19)$	6
6	$6^2 (mod\ 19)$	17
7	$7^2 (mod\ 19)$	11
8	$8^2 (mod\ 19)$	7
9	$9^2 (mod\ 19)$	5
10	$10^2 (mod\ 19)$	5
11	$11^2 (mod\ 19)$	7
12	$12^2 (mod\ 19)$	11
13	$13^2 (mod\ 19)$	17
14	$14^2 (mod\ 19)$	6
15	$15^2 (mod\ 19)$	16
16	$16^2 (mod\ 19)$	9
17	$17^2 (mod\ 19)$	4
18	$18^2 (mod\ 19)$	1

Jadi didapat $QR_{19} = \{0,1,4,5,6,7,9,11,16,17\}$

Menentukan elemen grup kurva eliptik $E_{19}(1,8)$ yang merupakan himpunan penyelesaian dari $y^2 = x^3 + x + 8(mod\ 19)$ untuk $x \in F_{19}$ dan $y^2 \in QR_{19}$.

Tabel 4.2 Elemen grup kurva eliptik

$x \in F_{19}$	$y^2 = x^3 + x + 8(mod19)$	$y^2 \in QR_{19}$	$(x, y) \in E_{19}(1,8)$
$x = 0$	$y^2 = 0^3 + 0 + 8(mod19) = 8$	$8 \notin QR_{19}$	-
$x = 1$	$y^2 = 1^3 + 1 + 8(mod19) = 10$	$10 \notin QR_{19}$	-
$x = 2$	$y^2 = 2^3 + 2 + 8(mod19) = 18$	$18 \notin QR_{19}$	-
$x = 3$	$y^2 = 3^3 + 3 + 8(mod19) = 0$	$0 \in QR_{19}$	(3,0)
$x = 4$	$y^2 = 4^3 + 4 + 8(mod19) = 0$	$0 \in QR_{19}$	(4,0)
$x = 5$	$y^2 = 5^3 + 5 + 8(mod19) = 5$	$5 \in QR_{19}$	(5,9) dan (5,10)
$x = 6$	$y^2 = 6^3 + 6 + 8(mod19) = 2$	$2 \notin QR_{19}$	-
$x = 7$	$y^2 = 7^3 + 7 + 8(mod19) = 16$	$16 \in QR_{19}$	(7,4) dan (7,15)
$x = 8$	$y^2 = 8^3 + 8 + 8(mod19) = 15$	$15 \notin QR_{19}$	-
$x = 9$	$y^2 = 9^3 + 9 + 8(mod19) = 5$	$5 \in QR_{19}$	(9,9) dan (9,10)
$x = 10$	$y^2 = 10^3 + 10 + 8(mod19) = 8$	$8 \notin QR_{19}$	-
$x = 11$	$y^2 = 11^3 + 11 + 8(mod19) = 8$	$8 \notin QR_{19}$	-
$x = 12$	$y^2 = 12^3 + 12 + 8(mod19) = 0$	$0 \in QR_{19}$	(12,0)
$x = 13$	$y^2 = 13^3 + 13 + 8(mod19) = 14$	$14 \notin QR_{19}$	-
$x = 14$	$y^2 = 14^3 + 14 + 8(mod19) = 11$	$11 \in QR_{19}$	(14,7) dan (14,12)
$x = 15$	$y^2 = 15^3 + 15 + 8(mod19) = 16$	$16 \in QR_{19}$	(15,4) dan (15,15)
$x = 16$	$y^2 = 16^3 + 16 + 8(mod19) = 16$	$16 \in QR_{19}$	(16,4) dan (16,15)

$x = 17$	$y^2 = 17^3 + 17 + 8(mod19)$ $= 17$	$17 \in QR_{19}$	(17,6) dan (17,13)
$x = 18$	$y^2 = 18^3 + 18 + 8(mod19)$ $= 6$	$6 \in QR_{19}$	(18,5) dan (18,14)

4.1.2 Penjumlahan Titik pada Kurva Eliptik (x_3, y_3)

Pada gambar 4.2 dan 4.3 menjelaskan proses pembuatan titik ketiga pada bidang terbatas F_p dengan penjumlahan antara dua titik pada kurva eliptik. Adapun penjumlahan titik-titik pada kurva eliptik bisa dibagi 3 kategori penjumlahan :

1. Penjumlahan dua titik yang sama

Pada kategori ini pembuatan titik ketiga didapatkan dari dua titik yang sama yakni $P = (x_1, y_1)$ dan $Q = (x_2, y_2)$, untuk nilai $x_1, y_1, x_2, y_2, x_3, y_3 \in E(F_p)$. Syarat awal untuk proses ini adalah $P = Q$, jika syarat tersebut tidak terpenuhi maka kembali ke awal untuk x_1, y_1, x_2, y_2. Jika syarat terpenuhi maka akan berlanjut ke proses selanjutnya yakni melakukan penghitungan menggunakan rumus lamda pada persamaan 2.7. Setelah nilai lamda ditemukan, kemudian x_3 dapat dihitung dengan menggunakan persamaan 2.8. Selanjutnya y_3 dapat dihitung menggunakan persamaan 2.9. Titik ketiga (x_3, y_3) telah didapatkan dari proses diatas dan diagram alir proses pembuatan titik ketiga dengan penjumlahan dua titik yang sama dapat dilihat pada gambar 4.2.

Contoh perhitungan secara aljabar untuk pembuatan titik ketiga dengan titik awal yang sama, untuk persamaan kurva eliptik dengan persamaan $E: y^2 = x^3 + x + 8$ dengan $a = 1$ $b = 8$ dan $p = 19$ didapatkan titik kurva eliptik.

$$(x, y) = \left\{ \begin{matrix} (3,0), (4,0), (5,9), (5,10), (7,4), (7,15), (9,9), (9,10), (12,0), (14,7), \\ (14,12), (15,4), (15,15), (16,4), (16,15)(17,6), (17,13)(18,5), (18,14) \end{matrix} \right\}$$

Untuk $P = (x_1, y_1)$ dan $Q = (x_2, y_2)$ maka $P = Q = P + P = 2P = (x_3, y_3)$ dimana :

$$x_3 = \left[\left(\frac{3x_1^2 + a}{2y_1} \right)^2 - 2x_1 \right] mod\ p$$

$$y_3 = \left[\left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1\right] mod\ p$$

Ambil $P = (5,9)$ maka $2P = P + P = (x_3, y_3)$, perhitungannya seperti dibawah ini

$$x_3 = \left[\left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1\right] mod\ p$$

$$= \left[\left(\frac{3(5)^2 + 1}{2(10)}\right)^2 - 2(5)\right] mod\ 19$$

$$= \left[\left(\frac{76}{20}\right)^2 - 10\right] mod\ 19$$

$$= \left[\left(\frac{5776}{400}\right) mod\ 19 - 10\ mod\ 19\right]$$

$$= \left[\left(\frac{0}{400}\right) mod\ 19 - 10\ mod\ 19\right]$$

$$= [0 + 9]$$

$$x_3 = 9$$

$$y_3 = \left[\left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1\right] mod\ p$$

$$y_3 = \left[\left(\frac{3(5)^2 + 1}{2(10)}\right)(5 - 10) - 10\right] mod\ 19$$

$$= \left[\left(\frac{76}{20}\right)(-5) - 10\right] mod\ 19$$

$$= \left[\left(\frac{76}{20}\right)(-5) mod\ 19 - 10\ mod\ 19\right]$$

$$= \left[\left(\frac{-380}{1}\right) mod\ 19 - 9\ mod\ 19\right]$$

$$y_3 = [0 + 10]$$

$$y_3 = 10$$

Sehingga didapat $(x_3, y_3) = (9,10)$

2. Penjumlahan dua titik yang berbeda

Untuk mencari titik ketiga, tidak hanya menggunakan penjumlahan dua titik yang sama namun bisa menggunakan dua titik yang berbeda. Pada gambar 4.2 menjelaskan proses pembuatan titik ketiga di bidang terbatas F_p dengan penjumlahan dua titik yang berbeda. $P = (x_1, y_1)$ dan $Q = (x_2, y_2)$, untuk nilai $x_1, y_1, x_2, y_2, x_3, y_3 \in E(F_p)$. Syarat awal untuk proses ini adalah $P \neq Q$, jika syarat tersebut tidak terpenuhi maka kembali ke awal untuk $x_1, y_1, x_2 y_2$. Jika syarat terpenuhi maka akan berlanjut ke proses selanjutnya yaitu melakukan penghitungan menggunakan rumus lamda pada persamaan 2.6. Setelah nilai lamda ditemukan, kemudian x_3 dapat dihitung dengan menggunakan persamaan 2.10. Selanjutnya y_3 dapat dihitung dengan menggunakan persamaan 2.11. Titik ketiga (x_3, y_3) telah didapatkan dari proses diatas.

Contoh perhitungan secara aljabar untuk pembuatan titik ketiga dari titik awal yang berbeda, untuk persamaan kurva eliptik dengan persamaan $E: y^2 = x^3 + x + 8$ dengan $a = 1, b = 8$, $p = 19$ didapatkan titik kurva eliptik

$$(x, y) = \{ (3,0), (4,0), (5,9), (5,10), (7,4), (7,15), (9,9), (9,10), (12,0), (14,7), (14,12), (15,4), (15,15), (16,4), (16,15)(17,6), (17,13)(18,5), (18,14) \}$$

Untuk $P = (x_1, y_1)$ dan $Q = (x_2, y_2)$ maka $P \neq Q = P + Q = R = (x_3, y_3)$ dimana :

$$x_3 = \left[\left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2 \right] mod\ p$$

$$y_3 = \left[\left(\frac{y_2 - y_1}{x_2 - x_1} \right) (x_1 - x_3) - y_1 \right] mod\ p$$

Ambil $P = (4,0)$ dan $Q = (5,10)$ maka $P + Q = R = (x_3, y_3)$, sehingga perhitungannya seperti dibawah ini :

$$x_3 = \left[\left(\frac{y_2 - y_1}{x_2 - x_1} \right)^2 - x_1 - x_2 \right] mod\ p$$

$$= \left[\left(\frac{10-0}{5-4}\right)^2 - 4 - 5\right] \; mod \; 19$$
$$= \left[\frac{100}{1} - 4 - 5\right] \; mod \; 19$$

$$= 91 \; mod \; 19$$
$$x_3 = 15$$
$$y_3 = \left[\left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1\right] \; mod \; p$$
$$= \left[\left(\frac{10-0}{5-4}\right)(4 - 15) - 0\right] \; mod \; 19$$
$$= \left[\frac{10}{1}(-11) - 0\right] \; mod \; 19$$
$$= \left[\frac{-110}{1} - 0\right] \; mod \; 19$$
$$= \left[\frac{-110}{1} - 0\right] \; mod \; 19$$
$$y_3 = 4$$

Sehingga didapat $(x_3, y_3) = (15,4)$

3. Penjumlahan titik dengan elemen negatifnya

Untuk mencari titik ketiga, dapat juga dengan menjumlahkan titik awal dengan elemen negatifnya. Pada gambar 4.2 menjelaskan proses pembuatan titik ketiga di bidang terbatas F_p dengan $P = (x1, y1)$ dan $Q = (x2, y2)$ dimana $x1 = x2$ dan $y1 = -y2$, untuk nilai $x_1, y_1, x_2, y_2, x_3, y_3 \in E(F_p)$.

Contoh perhitungan secara aljabar untuk pembuatan titik ketiga dari titik awal yang berbeda, untuk persamaan kurva eliptik dengan persamaan $E: y^2 = x^3 + x + 8$ dengan $a = 1, b = 8$, $p = 19$ didapatkan titik kurva eliptik

$$(x, y)$$
$$= \left\{ \begin{matrix} (3,0), (4,0), (5,9), (5,10), (7,4), (7,15), (9,9), (9,10), (12,0), (14,7), \\ (14,12), (15,4), (15,15), (16,4), (16,15)(17,6), (17,13)(18,5), (18,14) \end{matrix} \right\}$$

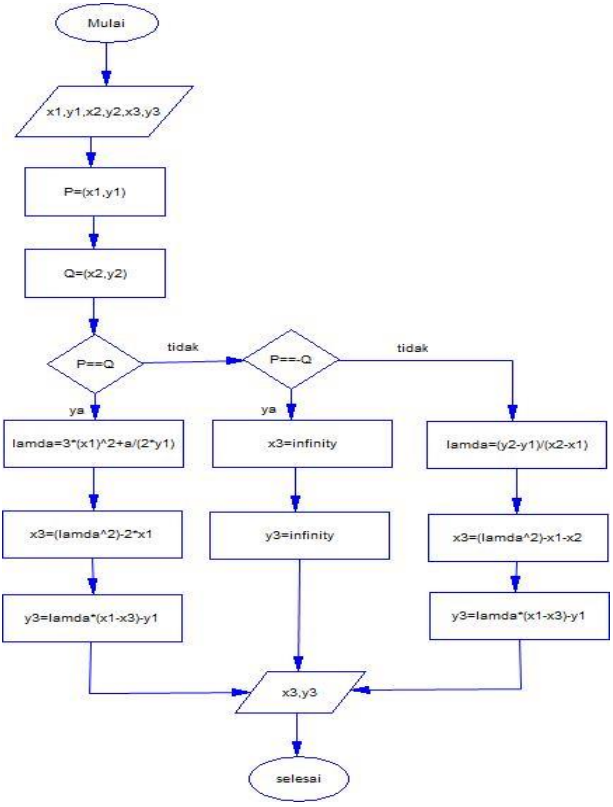
Untuk $P = (x_1, y_1)$ dan $Q = (x_1, -y_1)$ maka $P = -Q = P + Q = R = (x_3, y_3)$ dimana :

$$\lambda = \left(\frac{y_2 - y_1}{x_2 - x_1}\right) mod\ p$$

Ambil $P = (5,9)$ dan $Q = (5,10)$ maka $P + Q = R = (x_3, y_3)$, sehingga perhitungannya seperti dibawah ini :

$$\lambda = \left(\frac{10 - 9}{5 - 5}\right) mod\ p$$

Terlihat bahwa nilai λ adalah tidak terdefinisi sehinggga bisa kita asumsikan jika $P + Q = P + (-P) = O$ dimana O adalah titik infinity dan titik infinity ini adalah elemen netral dari kurva eliptik $E(F_P)$.



Gambar 4.2 Flowchart penjumlahan titik

4.2 Representasi Piksel pada Kurva Eliptik

Setiap citra terdiri dari piksel. Dalam citra *grayscale*, setiap piksel memiliki nilai 8-bit yang terdiri dari 0 sampai 255. Untuk mengenkripsi menggunakan sebuah citra dengan menggunakan ECC, setiap piksel dianggap sebuah pesan dan harus dipetakan ke dalam setiap titik pada kurva eliptik yang sudah didefinisikan sebelumnya. Pada tugas akhir ini, pemetaan yang digunakan adalah pemetaan berdasarkan sebuah tabel. Untuk membuat tabel ini, Untuk membuat tabel ini, sebuah kurva eliptik $E_p(a, b)$ akan membentuk semua titik yang memungkinkan, kemudian titik-titik tersebut akan dipetakan menjadi 256 grup. Setiap grup mempunyai $N = [\#E(F_p)/256]$ anggota. Setiap baris merepresentasikan sebuah nilai piksel. Jika N tidak mencapai 256 anggota, maka sisa kolom tidak diisi/dibiarkan kosong sampai akhir.

Dari nilai piksel pertama dari sebuah citra plain, titik pada kurva eliptik yang berkorespondensi dengan piksel tersebut kemudian dipetakan hingga piksel terakhir. Misalkan pengirim dan penerima menentukan sebuah kurva eliptik $E_{8209}(1{,}7710)$ yang direpresentasikan oleh persamaan 4.1:

$$y^2 mod\ 8209 = x^3 + x + 7710\ mod\ 8209 \qquad (4.1)$$

Tabel 4.3 menunjukkan titik-titik yang dihasilkan dari persamaan 4.1. Titik pertama disini adalah titik ideal/ *infinity* yang akan berkorespondensi dengan nilai piksel 0, kemudian dilanjutkan dengantitik dan nilai piksel selanjutnya. Setelah 256 titik dimuat pada kolom pertama, 256 titik selanjutnya akan diletakkan pada kolom kedua, demikian selanjutnya sampai kolom yang terakhir. Dalam contoh ini terdapat 8449 titik pada kurva.Tiap-tiap titik tersebut mengisi tabel pemetaan dengan 255 baris dan 33 kolom.

Table 4.3 Tabel pemetaan titik

Indeks	Pemetaan Ke-1	Pemetaan Ke-2	Pemetaan Ke-3	--	Pemetaan Ke-33
0	Ideal	(292,7217)	(547,7960)	--	(8143,1064)
1	(0,2056)	(294,3125)	(550,1822)	--	(8143,7145)
2	(0,6153)	(294,5084)	(550,6387)	--	(8144,766)
3	(1,520)	(297,3077)	(552,1077)	--	(8144,7443)
4	(1, 7689)	(297,5132)	(552,7132)	--	(8145,536)
5	(4,3340)	(298,1261)	(553,3973)	--	(8145,7673)
6	(4,4869)	(298,6948)	(553,4236)	--	(8149,266)
7	(5,3482)	(299,102)	(554,1000)	--	(8149,7943)
--	--	--	--	--	--
154	(185,4709)	(449,7134)	(730,4820)	--	--
155	(186,2564)	(455,1610)	(731,2092)	--	--
156	(186,5645)	(455,6599)	(731,6117)	--	--
157	(187,3003)	(456,1607)	(732,468)	--	--
158	(187,5206)	(456,6602)	(732,7741)	--	--
159	(188,2264)	(457,1728)	(734,1476)	--	--
160	(188,5945)	(457,6481)	(734,6733)	--	--
161	(191,3274)	(458,2370)	(735,2853)	--	--
--	--	--	--	--	--
248	(284,5136)	(541,6439)	(816,8170)	--	--
249	(287,1825)	(542,3553)	(818,2294)	--	--
250	(287,6384)	(542,4656)	(818,5915)	--	--
251	(288,3552)	(545,3950)	(819,3816)	--	--
252	(288,4657)	(545,4259)	(819,4393)	--	--
253	(291,3457)	(546,378)	(820,2708)	--	--
254	(291,4752)	(546,7831)	(820,5501)	--	--
255	(292,992)	(547,249)	(822,4061)	--	--

4.3 Implementasi Sistem

Dari parameter yang telah dibuat di subbab sebeumnya, beberapa parameter akan diambil untuk membangkitkan kunci publik dan kunci privat tersebut. Untuk membangkitkan kunci tersebut dapat menggunakan perhitungan $Q = d.G$ yang sesuai dengan aturan kurva eliptik. Jika kunci sudah dibangkitkan, langkah selanjutnya yaitu proses enkripsi dan dekripsi citra. Algoritma pembangkit kunci Diffi-Hellman dengan kurva eliptik yaitu [13]

INPUT : Domain parameter $T(p, a, b, G, n, h)$
OUTPUT : $K_{publik} = Q_1, Q_2\ K_{privat} = d_1, d_2$
Pilih $G = (x_1, y_1)$ sebagai titik pembangkit grup
Kurva eliptik $E(a, b)$
Hitung $Q_1 = d_1.G$ dan $Q_2 = d_2.G$
$K_{publik} = Q_1, Q_2\ K_{privat} = d_1, d_2$

4.4 Perancangan Sistem Enkripsi ECDH

Di dalam perancangan sistem ini, citra asli *grayscale* akan di enkrip dengan menggunakan algoritma ECDH. Langkah awal untuk memulai proses ini adalah menentukan persamaan kurva eliptik $(mod\ p)$, dengan memasukkan nilai koefisien x dan konstanta serta pemodulo(p). Maka dari nilai tersebut dapat dihasilkan beberapa titik-titik yang sesuai dengan input yang dimasukkan.Selanjutnya titik tersebut akan dipetakan pada tabel pemetaan

Setelah menghasilkan beberapa titik tersebut, proses selanjutnya yaitu membangkitkan kunci (*generate key*). Pada tahap ini, kedua pihak membangkitkan kunci publik masing-masing pihak dengan mengalikan nilai G(titik pembangkit) yang telah disepakati oleh masing-masing pihak dengan kunci privat

masing-masing pihak. Selanjutnya kunci publik yang telah didapatkan akan ditukarkan ke pihak yang berlawanan, setelah itu, kunci publik dari pihak berlawanan akan dikalikan dengan kunci privat masing-masing pihak sehingga akan dihasilkan kunci bersama.

Proses selanjutnya adalah memasukkan citra asli *grayscale* ke dalam sistem. Setelah itu, tahapan akan berlanjut ke proses enkripsi yaitu mengambil setiap titik piksel pada citra dan setiap piksel itu akan terpetakan pada satu titik, setiap titik tersebut akan ditambah dengan kunci rahasia bersama untuk mencari titik ketiga (x_3, y_3), dimana titik ketiga ini merupakan *chiper point* pada kurva eliptik. Dengan menggunakan pemetaan tabel, kembali didapat nilai piksel dari indeks baris dan indeks kolom. Indeks baris dan indeks kolom tersebut akan dijadikan *chiper image*. Sehingga *chiper image* yang dihasilkan berjumlah dua.

Citra asli (M) sebagai masukan algoritma enkripsi sistem kriptografi ECDH. Pengenkripsi memilih secara acak integer dan kemudian menghitungnya. Berikut algoritma enkripsinya[13].

INPUT : Domain parameter $T(p, a, b, G, n, h)$, Kunci publik Q_1,Q_2. Kunci Privat d_1, d_2 Titik Plain M
OUTPUT : Titik Chiper C
Hitung $SK = d_2.Q_1 = d_1.Q_2$
Hitung $M = C - SK$
Titik Chiper C

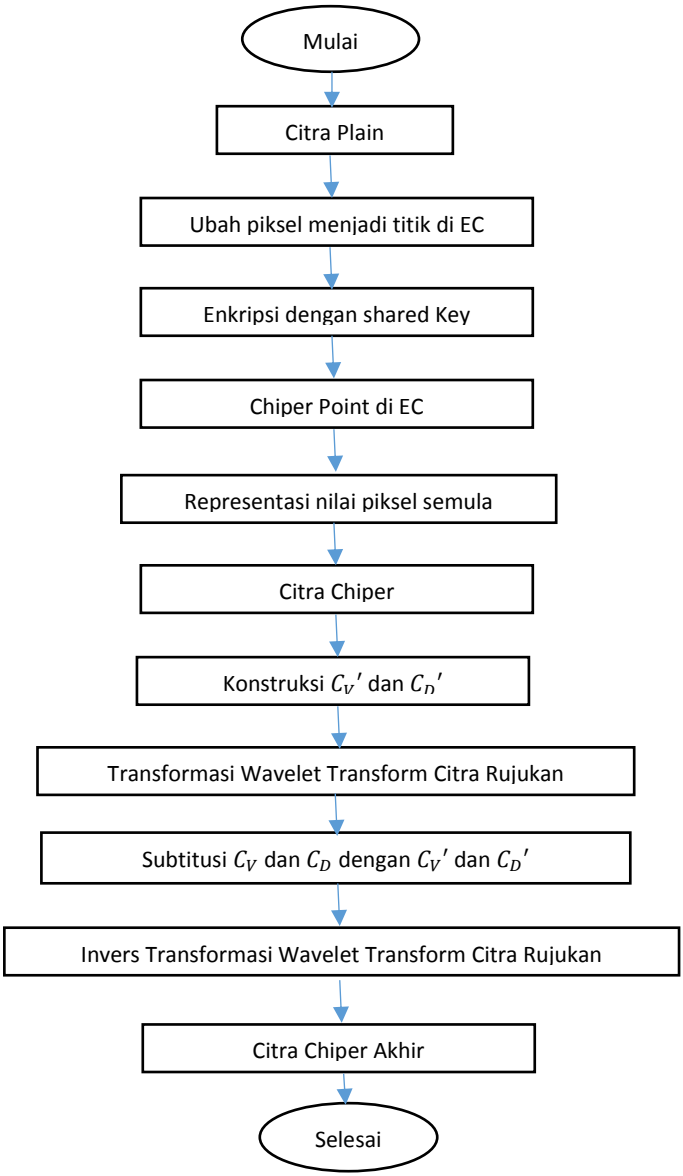
4.5 Perancangan Sistem Enkripsi Transformasi Wavelet Diskrit

Di dalam perancangan sistem ini, citra hasil enkripsi ECDH akan disembunyikan menggunakan transformasi wavelet diskrit. Langkah awal untuk memulai proses ini adalah membagi citra hasil enkripsi menjadi dua citra. Dua citra tersebut bisa disebut $C_V{}'$ dan $C_D{}'$, dimana $C_V{}'(m,n) = \left\lfloor \frac{P(m,n)}{10} \right\rfloor$ dan $C_D =$

$P(m,n)\ mod\ 10$. Setelah itu, dilanjutkan proses transformasi wavelet diskrit citra *grayscale* yang baru sehingga didapat nilai C_A, C_H, C_V, C_D. Selanjutnya,nilai C_V akan disubtitusi dengan $C_V{}'$ dan C_D akan disubtitusi dengan $C_D{}'$. Setelah disubtitusi maka akan dilanjutkan proses invers transformasi wavelet diskrit, sehingga akan didapatkan citra enkripsi. Berikut algoritma enkripsinya[8].

INPUT : Citra hasil enkripsi ECDH dengan ukuran $M \times N$, citra rujukan R dengan ukuran $2M \times 2N$
OUTPUT : Citra enkripsi akhir E dengan ukuran $2M \times 2N$
Lakukan Transformasi Wavelet Diskrit Citra R
for $m = 1$ to M **do**
  for $n = 1$ to N **do**
    $C_v{}'(m,n) = \left\lfloor \frac{P(m,n)}{10} \right\rfloor$
    $C_D{}'(m,n) = P(m,n)\ mod\ 10$
  end for
end for
Lakukan Invers Transformasi Wavelet Diskrit dengan sub-bands $C_A, C_H, C_V{}', C_D{}'$

Diagram alir proses enkripsi bisa dilihat pada gambar 4.3



Gambar 4.3 Diagram alir proses enkripsi

4.6 Perancangan Sistem Dekripsi ECDH

Pada perancangan sistem dekripsi ini, citra enkripsi dari hasil dekripsi transformasi wavelet akan dikembalikan menjadi sebuah citra plain kembali. Citra cipher akan diubah kembali menjadi titik-titik yang ada di kurva eliptik. Citra cipher tersebut mempresentasikan indeks kolom dan indeks baris. Citra cipher tersebut akan diubah kembali menjadi titik-titik yang ada di kurva eliptik. Selanjutnya dilakukan proses dekripsi dengan mengurangi titik cipher dengan kunci rahasia bersama. Setelah itu digunakan tabel pemetaan kembali untuk mengembalikan ke citra plain. Berikut adalah algoritma dekripsinya.[13]

INPUT : Domain parameter $T(p, a, b, G, n, h)$, Kunci rahasia bersama SK, Titik Chiper C
OUTPUT: Titik Plain M
Hitung $M = C - SK$
Titik Plain M

4.7 Perancangan Sistem Dekripsi Transformasi Wavelet Diskrit

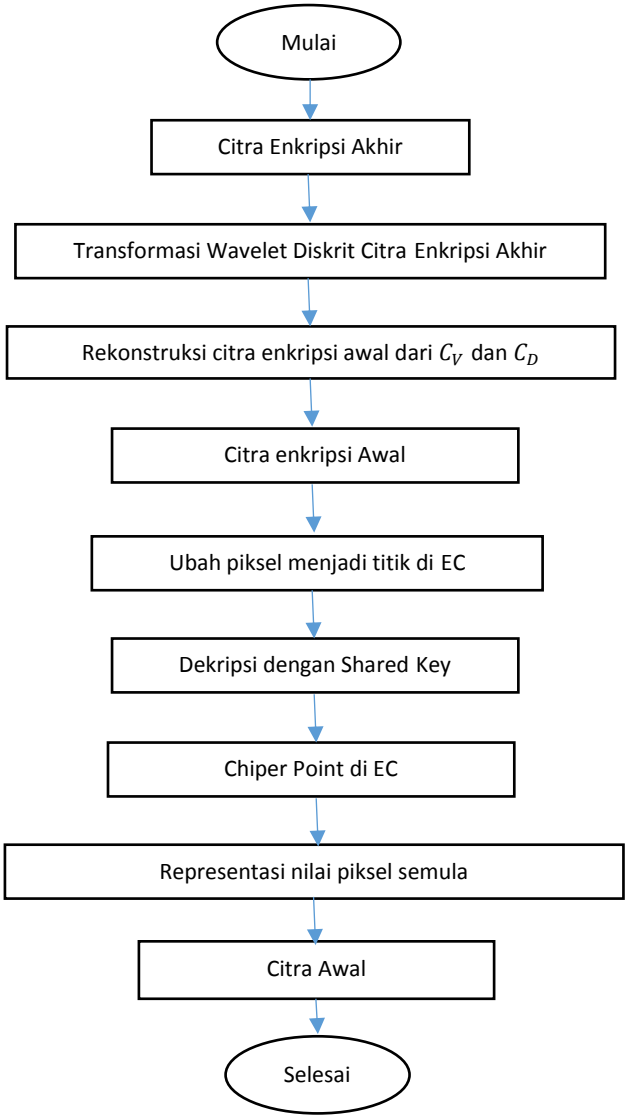
Pada perancangan sistem dekripsi ini, citra hasil enkripsi akhir akan dikembalikan ke citra enkripsi awal/ citra chiper. Citra enkripsi akhir akan dilakukan transformasi wavelet untuk mendapatkan nilai *sub-bands*, yaitu C_A, C_H, C_V, C_D. Setelah itu, nilai C_V dan C_D akan diambil untuk menghitung citra enkripsi awal dengan persamaan :

$$P(m,n) = 10C_V(m,n) + C_D(m,n)$$

Dimana C_V dan C_D adalah *sub-bands* dari citra enkripsi, dan $P(m,n)$ adalah citra enkripsi awal yang telah di rekonstrusi. Berikut adalah algoritma dekripsinya.[8].

INPUT : Citra hasil enkripsi akhir dengan ukuran $2M \times 2N$
OUTPUT : Citra enkripsi awal P dengan ukuran $2M \times 2N$
Lakukan Transformasi Wavelet Diskrit Citra hasil enkripsi akhir
for $m = 1$ to M **do**
  for $n = 1$ to N **do**
    $P(m,n) = 10C_V(m,n) + C_D(m,n)$
  end for
end for

Diagram alir proses dekripsi bisa dilihat pada gambar 4.4

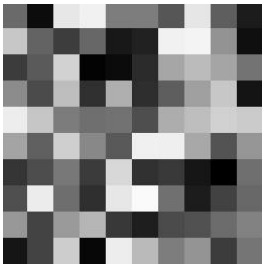


Gambar 4.4 Diagram alir proses dekripsi

4.8 Matriks yang Dibentuk dari Citra dengan Format .BMP

File citra dengan format .bmp termasuk format citra raster yang sering dipakai. Citra raster adalah citra yang bergantung pada resolusi. Resolusi merupakan sebuah ekspresi $m \times n$ dimana m adalah jumlah baris dan n adalah jumlah kolom. Resolusi juga mengacu pada jumlah piksel di dalam sebuah citra. Pada tugas akhir ini citra raster yang digunakan adalah sebuah citra *grayscale.*

Berikut akan diberikan contoh citra *grayscale* format .bmp dengan ukuran 10×10 yang dibentuk dari citra tersebut.



Gambar 4.5 img.bmp dengan ukuran $\mathbf{10 \times 10}$

Dengan menggunakan MATLAB, Citra tersebut dapat terbaca sebagai sebuah matriks $m \times n$ seperti dibawah ini

```
img =

  10×10 uint8 matrix

   106    12   231   241   125   125    86   230    94    28
   199    99    61   103    24    33   241   244   147    15
    60    90   210     3    11    43   166   187   165   115
   140    75   190    48   175    46    94   160   199    20
   237   198   124   111   114    78   130   130   209   203
   164    96   207   136    89   240   224   140   159   150
    53    77   120    59   216    49    57    43    58   111
    79   236   110    47   231   250   112    28    66   104
   152    67   154   182    56    30    75    81   108   130
    21    67   205     7   237   186   125   148    60   117
```

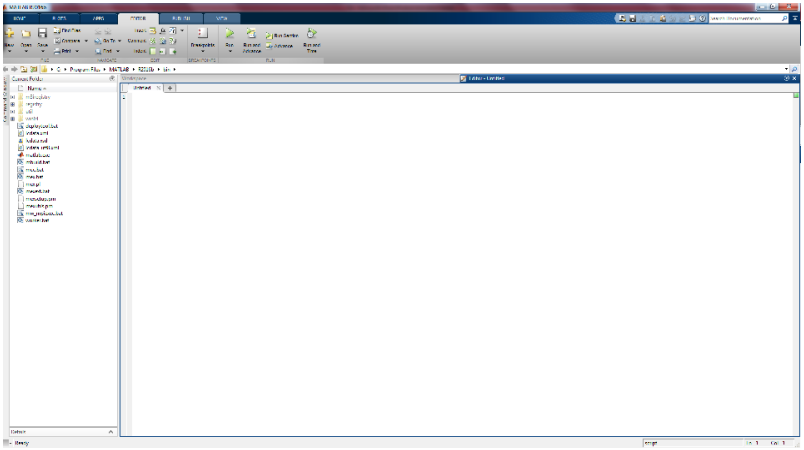
Gambar 4.6 Matriks img.bmp pada MATLAB

4.9 Implementasi pada MATLAB

4.9.1 MATLAB

MATLAB (*Matrix Laboratory*) adalah lingkungan komputasi numerikal dan bahasa pemrograman computer generasi keempat. Dikembangkan oleh The MathWorks, MATLAB memungkinkan manipulasi matriks, pem-plot-an fungsi dan data dan implementasi algoritma.

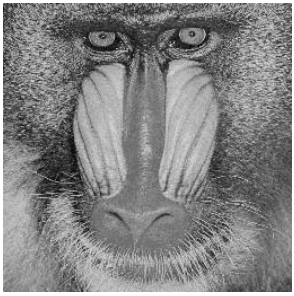
Meskipun hanya bernuansa numerik, sebuah kotak kakas (*toolbox*) yang menggunakan mesin simbolik MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer.



Gambar 4.7 Interface MATLAB

4.9.2 Pembacaan Piksel Citra pada MATLAB

Pada tahap ini, citra asli grayscale pada gambar 4.8 sebagai file input akan dibaca oleh MATLAB yang hasilnya sebuah matriks berukuran $m \times n$ pada gambar 4.9. Kemudian matriks itu akan diubah menjadi sebuah list agar dapat dengan mudah untuk diolah. Kemudian tipe data dari cita tersebut kita ubah menjadi *double* agar bisa *diolah*. *Source code* dibawah adalah proses mengubah citra menjadi matriks



Gambar 4.8 Baboon.jpg citra *grayscale* berukuran $\mathbf{256 \times 256}$

```
baboon=imread('baboon.jpg');
baboon
x=double(baboon(:)');
x
```

baboon

256x256 uint8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	145	39	143	43	60	118	140	72	168	96	54	112	59	133
2	76	62	78	48	115	42	69	36	108	100	78	48	102	196
3	101	101	88	74	54	137	71	23	65	132	34	49	26	189
4	45	143	71	51	65	77	102	88	37	101	53	68	66	116
5	43	69	140	52	80	173	114	161	74	111	128	91	102	161
6	38	57	76	74	83	183	165	102	104	59	87	152	183	120
7	30	100	50	53	158	93	137	101	94	87	185	147	112	176
8	35	77	76	37	102	165	106	60	70	125	145	78	198	63
9	90	125	106	71	31	52	81	80	52	98	173	194	101	91
10	52	67	83	68	122	98	116	44	154	62	67	133	65	162
11	148	45	68	79	41	151	65	77	90	81	45	160	160	41
12	48	47	33	87	74	92	103	45	81	152	76	143	180	198
13	73	153	27	62	96	54	71	52	48	131	170	163	140	178
14	110	69	141	29	87	167	137	65	73	57	120	56	155	90
15	114	57	45	61	160	134	58	147	130	144	69	156	127	73
16	84	49	84	72	157	57	148	67	53	182	54	72	165	106
17	52	65	145	83	91	144	90	76	137	62	68	71	150	148
18	72	153	42	85	81	58	153	68	115	127	109	109	202	187
19	53	76	93	155	32	35	87	164	187	152	150	143	100	167
20	66	45	90	59	110	92	113	76	63	153	161	190	50	120
21	57	136	94	64	75	43	97	111	124	129	143	115	176	68
22	41	104	125	185	160	37	125	67	62	134	136	91	116	162
23	99	70	131	145	82	164	113	69	38	144	171	118	62	37
24	65	128	99	92	88	71	71	78	47	124	168	173	136	123
25	69	96	60	43	148	88	139	118	67	93	127	201	118	170
26	48	40	78	158	88	105	121	98	58	108	65	86	156	155
27	141	60	158	57	105	72	127	114	92	57	117	194	155	180
28	73	126	100	184	129	158	140	140	163	64	132	160	88	104
29	158	60	109	132	136	126	175	87	89	201	116	188	183	118
30	133	156	51	51	99	168	120	120	151	58	179	141	179	109

Gambar 4.9 Piksel citra baboon.jpg

4.9.3 Pembuatan Titik Kurva Eliptik beserta Tabel Pemetaan

Pada proses enkripsi dan dekripsi citra dengan menggunakan algoritma ECDH yang mana dibutuhkan sebuah tabel pemetaan untuk mengubah piksel menjadi titik-titik pada kurva eliptik, sehingga semua titik harus dicari terlebih dahulu. Pada program enkripsi dan dekripsi citra ini menggunakan metode kriptografi kurva eliptik pada bidang terbatas F_p yang perhitungannya sesuai dengan persamaan 2.5 yang mana $a, b \in F_p$, dan p adalah bilangan prima ganjil dengan $p > 3$. Untuk perhitungan tersebut, penerapannya pada MATLAB adalah sebagai berikut

```
.......................
for i=1:length(nilaix)
    if(any(nilaix(i)==z)==1)
        cari(:,idx)=find(nilaix(i)==z);
        [pan leb]=size(cari);
        for j=1:pan
            if(yy(cari(j,idx)==0))
                j=2;
            titikxy(id,:)=[xx(i)
yy(cari(j,idx))];
            else
            titikxy(id,:)=[xx(i)
yy(cari(j,idx))];
            end
        id=id+1;
        end
        idx=idx+1;
    end

end
.........
```

Untuk lebih lengkapnya bisa dilihat di lampiran.

Misalkan titik-titik kurva eliptik didefinisikan oleh $E_{8209}(1{,}7710)$ sehingga jumlah titik-titik tang memenuhi mencapai 8449 titik, maka terdapat 33 grup pemetaandengan piksel lengkap yaitu 256 buah. Dapat dilihat hasilnya pada gambar

```
val(:,:,1) =

  Columns 1 through 18

       Inf          0          0          1          1          4          4
       Inf       2056       6153        520       7689       3340       4869

  Columns 19 through 36

        22         23         23         24         24         26         26
      6107       2403       5806       3585       4624       3522       4687

  Columns 37 through 54

        45         51         51         52         52         53         53
      6127       1430       6779       3253       4956       1857       6352

  Columns 55 through 72

        64         68         68         72         72         73         73
      5421       3892       4317        576       7633       3411       4798
```

Gambar 4.10 Potongan tabel pemetaan

4.9.4 Pertukaran Kunci Diffie-Hellman

Pada pertukaran kunci Diffie-Hellman, kita misalkan terdapat dua orang A dan B. Mereka berdua sama-sama mengetahui parameter-parameter yang akan digunakan dalam algoritma ECDH. Kemudian mereka memilih kunci privat masing-masing yaitu K_A dan K_B. Untuk mendapatkan kunci publik, masing-masing pengguna mengalikan kunci privat masing-masing dengan G (titik pembangkit) yang sudah disepakati kedua pengguna, kemudian kedua pengguna saling bertukar kunci publik. Setelah bertukar, mereka kembali mengalikan kunci privat masing-masing dengan kunci publik yang telah ditukar oleh pengguna lainnya.

Pertukaran kunci seperti ini membuat dua orang A dan B dapat berkomunikasi walaupun di dalam jaringan yang ridak aman. Untuk mengaplikasikan sebuah pertukaran kunci Diffie-

Hellman, kita harus menerapkan sebuah multiplikasi skalar pada kurva eliptik. Jika P adalah sebuah titik pada kurva eliptik, dan n adalah sebuah integer maka $nP = P + P + P + \cdots$ hingga n kali Implementasi algoritma Diffie-Hellman bisa dilihat dibawah ini

```
...........
while (z1 ~=0),
    while (mod(z1,2) ==0),
        z1=(z1/2);
        p=addell(p,p,a,b, n)
        if (length(p)==0),
           y=[];
           disp('Multell found a factor of n
and exited');
           z1
           return;
        end;
    end;  %end while
    z1=z1-1;
    y=addell(y,p,a,b,n)
    if (length(y)==0),
       disp('Multell found a factor of n and
exited');
       z1
       return;
     end;
 end;
.....................
```

4.9.5 Proses Pemetaan dan Enkripsi Citra

Setelah mendefinisikan kurva eliptik dan membuat tabel pemetaan, maka selanjutnya adalah proses enkripsi pada citra. Sebelum dilakukan proses enkripsi, setiap piksel akan dianggap sebagai pesan untuk dienkripsi. Sehingga setiap pksel yang ada akan direpresentasikan pada titik pada kurva eliptik. Berikut adalah implementasinya dibawah ini

```
......
for i=1:length(gray)
    r(i) = randi([1 panjang/256]);
    abu(i,:)=mapped(:,gray(i)+1,r(i));
    while abu(i,:)==[0 0]
        r(i) = randi([1 panjang/256]);
        abu(i,:)=mapped(:,gray(i)+1,r(i));
    end
    result(i,:)=addell(abu(i,:),p3,a,b,n);
end
.......
```

Setelah dilakukan pemetaan, maka setiap piksel akan menjadi sebuah titik yang berada pada kurva eliptik. Kemudian setiap titik yang ada akan dienkripsi menggunakan kunci rahasia bersama yang telah dibuat sebelumnya. Titik enkrpsi bisa didapatkan dengan menggunakan operasi penjumlahan dari titik awal dengan kunci rahasia bersama. Untuk penjumlahan titik pada kurva eliptik implementasinya bisa dilihat pada *source code* dibawah ini

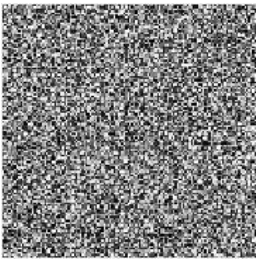
```
.........
if all(p1==p2),
   temp=mod(2*y1,n);
   if temp==0,
      p3(1)=Inf;
      p3(2)=Inf;
      return;
   end;
   den=powermod(2*y1, -1, n);
   num=mod(x1*x1,n);
   num=mod(mod(3*num,n) + a,n);
   m=mod(num*den,n);
   temp=mod(m*m,n);
   x3=mod(temp-x1-x2, n);
   temp=x1-x3;
   y3=mod(m*temp,n);
   y3=mod(y3-y1,n);
........
```

Pada gambar 4.11 dapat kita lihat bahwa titik awal (titik plain) ditambahkan dengan kunci bersama menghasilkan titik chiper atau titik yang telah terenkripsi. Kemudian untuk mengembalikan titik menjadi sebuah piksel menjadi piksel prosesnya dapat dilihat pada *source code* dibawah ini dan citra enkripsi awal bisa dilihat pada gambar 4.12 dan 4.13. Citra enkripsi yang dihasilkan adalah dua buah karena kita butuh nilai kolom tabel untuk mendekrip citra tersebut.

16384x2 double

	1	2
1	5282	3009
2	5491	5834
3	929	6808
4	1350	5796
5	1356	6257
6	2156	4014
7	1079	2241
8	3247	6874
9	1098	2252
10	7130	6890

Gambar 4.11 Hasil penjumlahan titik dengan *shared key*



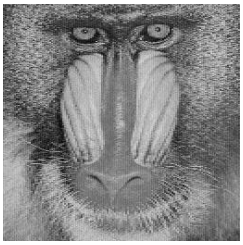
Gambar 4.12 Citra enkripsi awal

Gambar 4.13 Citra enkripsi untuk kolom

Setelah citra tersebut dienkripsi oleh algoritma kurva eliptik Diffie-Hellman, citra hasil enkripsi akan menuju pada proses selanjutnya yaitu penyembunyian citra hasil enkripsi dengan transformasi wavelet diskrit. Pada proses transformasi wavelet ini diperlukan citra baru (Citra R) yang berbeda dengan citra awal yang berukuran $2M \times 2N$ atau 2 kali ukuran citra awal. Transformasi wavelet yang akan digunakan adalah transformasi wavelet integer dengan menggunakan jenis wavelet yaitu 'haar'. Setiap piksel dari citra enkripsi awal akan diubah menjadi dua citra yang selanjutnya akan disubtitusikan pada sub-bands C_V dan C_D citra R sehingga citra R akan menyimpan data piksel dari citra enkripsi awal. Proses penyembunyian citra enkripsi awal bisa dilihat pada source code dibawah ini

```
cv=floor(ro/10);
 cd=mod(ro,10);
LS = liftwave('haar','Int2Int');
[CA,CH,CV,CD] = lwt2(double(citra),LS);
X=ilwt2(CA,CH,cv,cd,LS);
X=uint8(X);
imshow(X);
```

Pada source code diatas bisa kita lihat bahwa citra tersebut ditransformasi wavelet kemudian disubtitusikan nilai C_V dan C_D yang baru dan selanjutnya akan di invers wavelet untuk mendapatkan citra yang baru. Citra enkripsi akhir bisa dilihat pada gambar 4.14



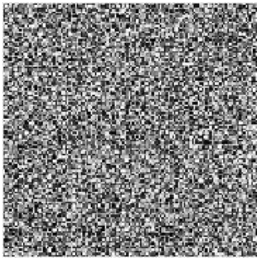
Gambar 4.14 Citra enkripsi akhir berukuran $\boldsymbol{2M \times 2N}$

4.9.6 Proses Pemetaan dan Dekripsi Citra

Pada bagian ini, citra hasil enkripsi akan dilakukan proses dekripsi atau pengembalian ke bentuk semula. Awalnya, citra enkripsi akan di transformasi wavelet diskrit untuk mendapatkan nilai C_V dan C_D sehingga bisa didapatkan citra enkripsi awal. Implementasi ekstraksi citra awal menggunakan transformasi wavelet diskrit pada MATLAB bisa dilihat pada *source code* dibawah ini

```
LS = liftwave('haar','Int2Int');
[aa,bb,cc,dd]=lwt2(double(image),LS);
h=10*cc+dd;
h=uint8(h);
```

Dapat dilihat pada gambar 4.15 didapatkan citra enkripsi awal yang kemudian akan didekripsi menggunakan algoritma kurva eliptik Diffie-Hellman.



Gambar 4.15 Citra enkripsi awal dari ekstraksi wavelet

Citra enkripsi awal tersebut akan dirubah menjadi titik-titik pada kurva eliptik melalui piksel-pikselnya. Kemudian titik-titik yang sudah didapatkan (titik chiper) akan dilakukan operasi pengurangan dengan kunci bersama yang telah didapatkan. Proses akan dilanjutkan dengan memetakan setiap titik hasil dari pengurangan ke tabel pemetaan yang telah dibuat sehingga titik-titik tersebut akan diubah menjadi piksel kembali. Proses dekripsi ECDH bisa dilihat pada *source code* dibawah ini

```
for i=1:a1
    row=find(haha==result(i,1));
    column=find(haha==result(i,2));
    pos(i)=0;
    ukur=length(column);
    idx=1;
    ketemu=any(bsxfun(@minus, row,
column(idx))==-1);
    while ketemu==0
        idx=idx+1;
        ketemu=any(bsxfun(@minus, row,
column(idx))==-1);
    end
    rr=find(bsxfun(@minus, row,
column(idx))==-1);
    pos(i)=row(rr);
end
```

Citra hasil dekripsi ditunjukkan pada gambar 4.16

Gambar 4.16 Citra hasil dekripsi

4.9.7 Perancangan Antar Muka

Salah satu aspk penting dalam pembuatan program adalah perancangan antar muka, karena perancangan antar muka yang baik berbanding lurus dengan tingkat *user friendly* sebuah program. Artinya perangkat lunak dirancang dengan sedemikian

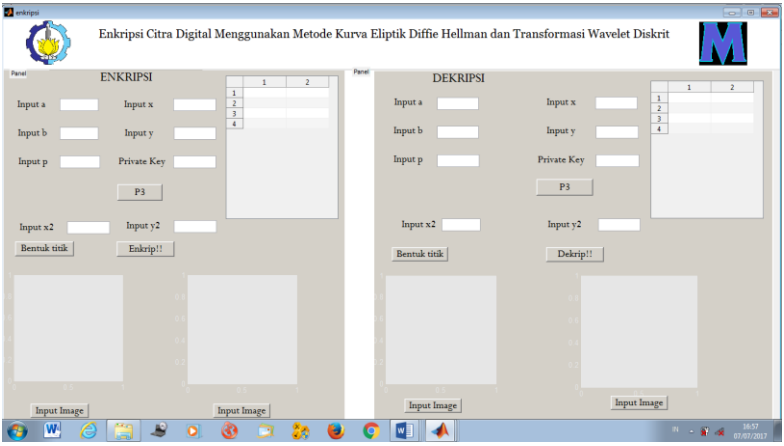
rupa agar pemakai dapat beradaptasi dengan mudah dalam pemakaian program tersebut. Berikut adalah desain antar muka halaman utama :

Perancangan antar muka halaman utama ditunjukkan pada Gambar 4.17. Form ini adalah form penggerjaan metode Kurva Eliptik Diffie-Hellman dan Transformasi Wavelet Diskrit. Adapun pada form ini terdapat dua proses besar yaitu proses enkripsi dan dekripsi.

Pada bagian enkripsi dilakukan proses pengisian parameter, pembuatan titik, pemilihan cita serta proses enkripsi. Pada halaman dekripsi dilakukan proses pengisian parameter, pembuatan titik, pemilihan citra seta proses dekripsi. Antar muka halaman utama terdiri dari :

1. *Edit text* **Input a,** berfungsi agar pengguna dapat menentukan nilai parameter a pada kurva eliptik.
2. *Edit text* **Input b,** berfungsi agar pengguna dapat menentukan nilai parameter b pada kurva eliptik.
3. *Edit text* **Input p,** berfungsi agar pengguna dapat menentukan nilai parameter p (bilangan prima) pada kurva eliptik.
4. *Edit text* **Private key,** berfungsi agar pengguna dapat menentukan kunci privat rahasia untuk proses enkripsi dan dekripsi.
5. *Edit text* **Input** $\boldsymbol{x_2,y_2}$ berfungsi agar pengguna dapat memasukkan nilai kunci publik dari user yang yang lain.
6. ***Axes1***, berfungsi untuk menampilkan citra plain yang akan dienkripsi.
7. ***Axes4***, berfungsi untuk menampilkan citra rujukan yang akan digunakan sebagai enkripsi citra plain.
8. ***Axes3***, berfungsi untuk menampilkan citra hasil enkripsi.
9. ***Axes5***, berfungsi untuk menampilkan citra kolom/ citra bantuan untuk mendekrip citra hasil enkripsi.
10. *Push button* **Bentuk titik,** berfungsi membentuk semua titik yang mungkin dari inputan persamaan kurva eliptik.

11. *Push button* **Input image(1)**, berfungsi mengambil file citra yang akan dienkripsi.
12. *Push button* **Input image(2)**, berfungsi mengambil file citra rujukan yang akan digunakan sebagai enkripsi.
13. *Push button* **Input image(3)**, berfungsi mengambil file citra enkripsi yang akan didekripsi.
14. *Push button* **Input image(4)**, berfungsi mengambil file citra kolom yang akan digunakan sebagai dekripsi.
15. *Push button* **Enkrip,** berfungsi mengenkripsi citra yang telah diinputkan
16. *Push button* **Dekrip,** berfungsi mendekripsi citra yang telah diinputkan.



Gambar 4.17 Perancangan Form Enkripsi-Dekripsi

“Halaman ini sengaja dikosongkan.”

BAB V
UJI COBA DAN PEMBAHASAN

Pada bab ini berisi tentang hasil uji coba yang dihasilkan oleh implementasi sistem dan melakukan pembahasn terhadap hasil uji coba yang dilakukan.

5.1 Pengujian Titik Kurva Eliptik

Setelah dilakukan pembuatan titik kurva eliptik pada program, maka selanjutnya yang dilakukan adalah pengujian terhadap titik dengan data yang akan diambil berbeda-beda untuk nilai $a, b,$ dan p nya. Pada tabel 5.1 akan ditampilkan hasil pengujian titik kurva eliptik dengan nilai $a, b,$ dan p yang berbeda

Tabel 5.1 Pengujian titik Kurva eliptik.

No	**Nilai a, b, p**	**Hasil Titik dalam Program**	**No**	**Nilai a, b, p**	**Hasil Titik dalam Program**
1	$a = 1$ $b = 1$ $p = 7$	$x = 4, y = 1$ $x = 4, y = 6$ $x = 5, y = 0$ $x = 6, y = 1$ $x = 6, y = 6$	2	$a = 3$ $b = 4$ $p = 17$	$x = 0, y = 2$ $x = 0, y = 11$ $x = 3, y = 1$ $x = 3, y = 12$ $x = 5, y = 1$ $x = 5, y = 12$ $x = 6, y = 2$ $x = 6, y = 11$ $x = 7, y = 2$ $x = 7, y = 11$ $x = 11, y = 4$ $x = 11, y = 9$ $x = 12, y = 0$
3	$a = 5$ $b = 6$ $p = 17$	$x = 9, y = 7$ $x = 9, y = 10$ $x = 10, y = 6$ $x = 10, y = 11$ $x = 11, y = 7$ $x = 11, y = 10$	4	$a = 3$ $b = 15$ $p = 23$	$x = 2, y = 11$ $x = 2, y = 12$ $x = 9, y = 9$ $x = 9, y = 14$ $x = 12, y = 10$ $x = 12, y = 13$

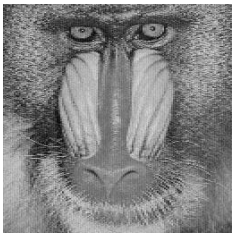
		$x = 12, y = 3$ $x = 12, y = 14$ $x = 14, y = 7$ $x = 14, y = 10$ $x = 16, y = 0$			$x = 14, y = 8$ $x = 14, y = 15$ $x = 15, y = 10$ $x = 15, y = 13$ $x = 18, y = 6$ $x = 18, y = 17$ $x = 19, y = 10$ $x = 19, y = 13$ $x = 20, y = 5$ $x = 20, y = 18$ $x = 21, y = 1$ $x = 21, y = 22$
5	$a = 5$ $b = 6$ $p = 17$	$x = 0, y = 7$ $x = 0, y = 16$ $x = 2, y = 6$ $x = 2, y = 17$ $x = 4, y = 5$ $x = 4, y = 18$ $x = 5, y = 8$ $x = 5, y = 15$ $x = 6, y = 8$ $x = 6, y = 15$ $x = 7, y = 10$ $x = 7, y = 13$ $x = 10, y = 1$ $x = 10, y = 22$ $x = 12, y = 8$ $x = 12, y = 15$ $x = 14, y = 1$ $x = 14, y = 22$ $x = 15, y = 9$ $x = 15, y = 14$ $x = 19, y = 2$ $x = 19, y = 21$ $x = 21, y = 4$ $x = 21, y = 19$ $x = 22, y = 1$ $x = 22, y = 22$	6	$a = 7$ $b = 13$ $p = 31$	$x = 2, y = 2$ $x = 2, y = 29$ $x = 5, y = 7$ $x = 5, y = 24$ $x = 7, y = 8$ $x = 7, y = 23$ $x = 13, y = 10$ $x = 13, y = 21$ $x = 16, y = 6$ $x = 16, y = 25$ $x = 18, y = 9$ $x = 18, y = 22$ $x = 20, y = 0$ $x = 21, y = 11$ $x = 21, y = 20$ $x = 26, y = 15$ $x = 26, y = 16$ $x = 27, y = 13$ $x = 27, y = 18$ $x = 30, y = 6$ $x = 30, y = 25$

5.2 Pengujian Citra Hasil Enkripsi

Untuk menentukan apakah sebuah kriptosistem kuat atau tidak, maka akan dilakukan beberapa bentuk analisis dan uji coba terhadap citra hasil enkripsi. Terdapat beberapa analisi dengan berbagai cara dari serangan kriptanalisis.

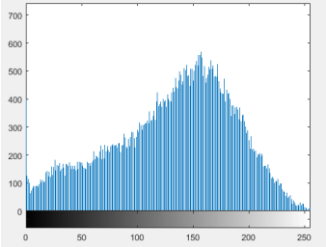
Analisis Histogram : Sebuah histogram adalah sebuah distribusi grafis dari intensitas setiap nilai piksel. Untuk mencegah bocornya informasi dari citra chiper, sebuah histogram citra chiper harus berbeda secara signifikan dengan citra plainnya. Pengujian menggunakan citra Hat ukuran 256×256 dan menggunakan persamaan 4.1

(a)

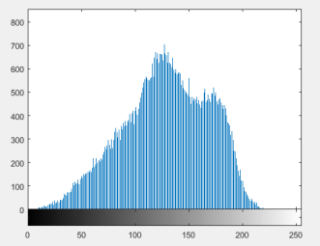


(b)

Gambar 5.1 (a) Citra plain, (b) Citra cipher



(a)

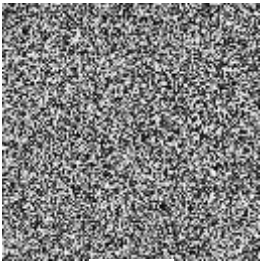


(b)

Gambar 5.2 (a) Histogram citra plain, (b) Histogram citra cipher

Analisis Sensitifitas Kunci : Sebuah algoritma yang aman memiliki tingkat sensitifitas yang tinggi terhadap kunci. Ini berarti bahwa citra yang telah terenkripsi tidak dapat didekripsi walaupun kuncinya diubah dengan nilai yang kecil. Pada gambar 5.3 menunjukkan bahwa citra yang telah terenkripsi akan diubah nilai kuncinya pada saat proses dekripsi sehingga menghasilkan perubahan yang signifikan. Hal ini membuat sebuah citra enkripsi tahan terhadap serangan *brute-force attack.*

(a)



(b)

Gambar 5.2 (a) Citra Dekripsi dengan kunci $\boldsymbol{k = 15}$,
(b) Citra Dekripsi dengan kunci $\boldsymbol{k = 21}$

Analisis Jenis Wavelet : Pada transformasi wavelet diskrit, terdapat beberapa jenis wavelet yang mempunyai karakteristik dan fungsi yang berbeda-beda. Pemilihan jenis wavelet yang tepat akan mempengaruhi citra hasil enkripsi dan dekripsi secara umum. Pada uji coba ini, wavelet yang akan digunakan diantaranya *'Lazy','Haar'*, dan *'Daubechies'*. Citra diuji pada ukuran yang sama yaitu (256×256). Pada Tabel 5.2 terlihat bahwa jenis wavelet yang paling baik untuk enkripsi adalah jenis 'Lazy' karena perbedaan piksel untuk uji coba dengan menggunakan wavelet jenis 'Lazy' lebih sedikit daripada 'Haar' dan 'Daubechies'. Karena semakin kecil perbedaan piksel maka semakin tinggi tingkat keakuratan sebuah proses enkripsi dan dekripsi citra.

Analisis Kompleksitas Citra : Pada transformasi wavelet diskrit, kompleksitas citra sangat mempengaruhi proses transformasi dan proses invers transformasi wavelet sehingga pemilihan jenis citra yang tepat akan mempengaruhi hasil enkripsi dan dekripsi secara umum. Pada uji coba ini, dipilih tiga citra dimana terdapat satu citra yang memiliki kompleksitas tinggi dan 2 citra yang memiliki kompleksitas rendah. Citra diuji pada ukuran yang sama yaitu (256×256). Uji coba dilakukan dengan menggunakan parameter PSNR (*Peak Signal-to-Noise Ratio*). Parameter PSNR (*Peak Signal-to-Noise Ratio*) memiliki nilai minimum 0 dB dan mempunyai nilai maksimum *infinity*. Citra dikatakan memiliki keakuratan yang tinggi terhadap citra acuan jika memiliki nilai PSNR yang tinggi pula. Citra dapat dikenali sebagai citra awal jika nilai PSNR lebih dari 20 dB. Pada Tabel 5.2 terlihat bahwa kompleksitas citra yang paling baik untuk enkripsi adalah citra dengan kompleksitas tinggi karena uji coba dengan menggunakan citra dengan kompleksitas tinggi memiliki PSNR (*Peak Signal-to-Noise Ratio*) yang tinggi.

Analisis Kualitas Citra Enkripsi: Kualitas citra enkripsi akan mempengaruhi apakah citra enkripsi tersebut dapat dengan mudah teridentifikasi sebagai citra enkripsi atau tidak. Apabila citra tersebut sulit dikenali sebagai citra enkripsi maka citra tersebut akan memiliki kemungkinan kriptanalisis yang semakin sedikit. Pada analisis ini nilai kualitas citra ditunjukkan dengan *Universal Image Quality Index*. Indeks tersebut memiliki nilai dari -1 sampai 1 dan tergantung pada citra yang digunakan sebagai citra rujukan. Jika citra tersebut memiliki nilai yang mendekati 1 maka kualitas citra tersebut semakin baik.

Misalkan $x = \{x_i | i = 1,2,..,..,N\}$ dan $y = \{y_i | i = 1,2,..,..,N\}$ berturut-turut adalah citra asli dan citra pengujian. Indeks kualitas citra bisa dihitung dengan

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2+\sigma_y^2)[(\bar{x})^2+(\bar{y})^2]} \quad (5.1)$$

Dimana

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \qquad \bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{5.2}$$

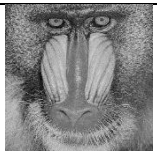
$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2, \qquad \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{5.3}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}) \tag{5.4}$$

Hasil terbaik Q adalah 1 didapat jika $y_i = x_i$ untuk semua $i = 1,2,\dots,\dots,N$. Hasil terendah yaitu -1 didapt jika $y_i = 2\bar{x} - x_i$ untuk semua $i = 1,2,\dots,\dots,N$. Indeks kualitas citra adalah kombinasi dari 3 faktor yaitu : korelasi yang hilang, distorsi pencahayaan dan distorsi kontras.

Pada Tabel 5.2 terlihat bahwa nilai indeks terbesar terletak pada citra enkripsi yang menggunakan jenis wavelet 'Daubechies'. Pengujian dilakukan dengan menggunakan citra plain yang sama yaitu hat.jpg dengan ukuran (128×128).

Tabel 5.2 Tabel pengujian jenis wavelet, kompleksitas citra dan kualitas citra enkripsi

Jenis Citra	Jenis Wavelet	PSNR	*Universal Image Quality Index*
 Baboon.jpg (Kompleksitas tinggi)	Lazy	60.7665 dB	0,6862
	Haar	45.5634 dB	0,9999
	Daubechies	43.7539 dB	1,0000
	Lazy	63.8110 dB	0,7081
	Haar	37.2925 dB	0,9991

Lena.jpg (Kompleksitas rendah)	Daubechies	29.7513 dB	0,9993
 Cameraman.tiff (Kompleksitas rendah)	Lazy	82.4935 dB	0,6423
	Haar	25.8162 dB	0,8388
	Daubechies	20.1946 dB	0,9793

Pada Tabel 5.2 terlihat bahwa keakuratan terbesar dan mempunyai kualitas tinggi dicapai oleh citra baboon.jpg (kompleksitas citra) yang menggunakan jenis wavelet 'Haar' atau Daubechies.

5.3 Pengujian dengan Noise

Tidak dipungkiri dalam proses transmisi terdapat kemungkinan citra yang ditransmisikan tersemat noise dan akan membuat nilai piksel akan berubah. Pada uji coba ini, citra yang telah dienkripsi akan diberikan noise diantaranya '*Salt and Pepper noise*' dan '*Speckle noise'* dengan tingkat densitas masing masing adalah 0.001. Pengujian dilakukan dengan citra baboon.jpg sebagai citra rujukan.

Tabel 5.3 Tabel pengujian dengan noise

Citra awal	Jenis Wavelet	Hasil Pengujian Citra Dekripsi (Tingkat keakuratan)	
		Salt and Pepper noise	*Speckle noise*
	Lazy	99,768%	74,475%
	Haar	99,609%	2%

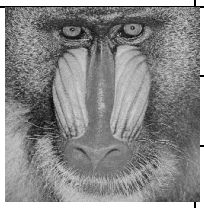
Hat.jpg	Daubechies	97,339%	1,727%
 Fruit.jpg	Lazy	99,762%	74,261%
	Haar	99,493%	1,849%
	Daubechies	97,107%	1,837%
 Lena.jpg	Lazy	99,768%	73,309%
	Haar	99,536%	1,764%
	Daubechies	98,047%	1,697%

Pada Tabel 5.3 terlihat bahwa citra dekripsi pada pengujian dengan *salt and pepper noise* dengan berbagai jenis memiliki tingkat keakuratan yang tinggi. Pada pengujian dengan *speckle noise* hanya pada pengujian dengan jenis wavelet 'lazy' menghasilkan tingkat keakuratan yang paling besar dan dapat dikenali sebagai citra awal.

5.4 Pengujian dengan Berbagai Ukuran dan Macam Citra

Tabel 5.4 Tabel hasil uji coba berbagai jenis citra dan ukuran

Nama Citra	Ukuran Citra	Waktu Enkripsi	Waktu Dekripsi
	64 × 64	1,130757 detik	1,065588 detik
	128 × 128	3,508605 detik	3,402906 detik
	256 × 256	17,32388 detik	16,882179 detik

Hat.jpg	512 × 512	153,049086 detik	150.112889 detik
 Lena.jpg	64 × 64	1,144946 detik	1,074316 detik
	128 × 128	3,538568 detik	3,362111 detik
	256 × 256	16,8987 detik	16,28628 detik
	512 × 512	150,919467 detik	158,100866 detik
 Baboon.jpg	64 × 64	1,153146 detik	1,057974 detik
	128 × 128	3,500165 detik	3,411043 detik
	256 × 256	17,101506 detik	17,27205 detik
	512 × 512	157,599181 detik	154,352987 detik

Dapat dilihat pada Tabel 5.4, uji coba dilakukan dengan total 12 kali percobaan dengan ukuran dan macam citra yang berbeda-beda. Untuk proses enkripsi sengan ukuran citra yang kecil menghasilkan waktu enkripsi yang memiliki yang kecil pula. Sedangkan untuk proses enkripsi dengan ukuran citra yang cukup besar mempunyai rata-rata waktu yang besar pula yang artinya proses tersebut membutuhkan waktu yang cukup lama. Kasus yang sama terjadi juga pada proses dekripsi.

5.5 Pengujian dengan Parameter a, b, p yang berbeda

Tabel 5.5 Tabel hasil Uji Coba dengan parameter yang berbeda

a, b, p	Titik yang dihasilkan	Waktu Enkripsi	Waktu Dekripsi
$a = 1 \;\; b = 7206$ $p = 7211$	7223	3,453491 detik	3,392608 detik

$a = 1 \ \ b = 7710$ $p = 8209$	8256	3,727386 detik	3,552373 detik
$a = 1 \ \ b = 8091$ $p = 10979$	10828	4,181869 detik	4,054151 detik

Pada percobaan diatas terlihat bahwa semakin besar nilai $a, b,$ dan p maka semakin banyak pula titik yang dihasilkan sehingga semakin luas pula lapangan terbatas (*finite field*) yang dihasilkan. Pada percobaan diatas juga dapat dilihat bahwa semakin besar lapangan terbatas (*finite field*) yang dihasilkan maka semakin lama pula proses enkripsi dan dekripsi dari citra tersebut.

BAB VI
PENUTUP

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. Di samping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut:

1. Tugas akhir ini telah berhasil mengimplementasikan enkripsi citra digital menggunakan kurva eliptik Diffie-Hellman dan transformasi wavelet diskrit dengan langkah-langkah sebagai berikut:
   a. Representasikan piksel ke kurva eliptik dengan tabel pemetaan yang telah dibuat
   b. Enkripsi citra dengan menambahkan titik awal dengan kunci yang telah disepakati
   c. Dekomposisi citra rujukan dengan transformasi wavelet diskrit
   d. Subtitusi hasil dekomposisi citra rujukan dengan citra enkripsi awal
2. Hasil pengujian titik kurva eliptik menunjukkan bahwa titik-titik yang dihasilkan dipengaruhi oleh parameter bilangan prima. Semakin besar nilai bilangan prima maka titik yang dihasilkan pun semakin banyak.
3. Semakin besar lapangan terbatas (*finite field*) yang digunakan maka elemen-elemen pada lapangan (*field*) tersebut pun semakin panjang, sehingga membuat tabel pemetaan akan semakin besar dan memiliki tingkat keamanan yang semakin tinggi juga, namun memiliki kelemahan waktu komputasi yang cukup lama.
4. Hasil citra dekripsi sangat dipengaruhi oleh jenis wavelet serta tingkat kompleksitas citra. Tingkat keakuratan wavelet tertinggi dicapai dengan jenis ‘lazy’ wavelet. Semakin tinggi

kompleksitas citra maka akan semakin tinggi keakuratan citra plain terhadap citra hasil dekripsi.
5. Citra enkripsi kuat terhadap beberapa serangan kriptografi. Hal ini dapat diketahui pada sensitifitas kunci dan pengujian dengan *salt and pepper* noise.
6. Hasil citra enkripsi dengan kualitas tertinggi menggunakan jenis wavelet ‘Haar’ atau ‘Daubechies’.

6.2 Saran

Dengan melihat hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan selanjutnya yaitu:

1. Mengembangkan enkripsi citra dengan citra berwarna.
2. Mengembangkan cara untuk mengembalikan citra enkripsi ke citra semula dengan menghasilkan hanya satu hasil enkripsi dikarenakan hasil dari program enkripsi-dekripsi ini terdapat 2 citra.
3. Mengembangkan sistem dengan output citra berbagai format.
4. Karena pada skema kriptosistem ini mengharuskan mengkomputasi tiap pikselnya untuk dienkripsi, maka untuk penelitian selanjutnya diharapkan ada proses grouping beberapa piksel menjadi satu untuk membuat proses komputasi lebih efisien.

DAFTAR PUSTAKA

[1] Guo, C,. Et al. 2014. "Optical Double Image Encryption Employing A Pseudo Technique In The Fourier Domain". **Opt.Commun, Vol. 321,61-72**

[2] Yang, Y.G,. et al. 2014. "Quantum Cryptographic Algorithm For Color Images Using Quantum Fourier Transform and Double Random-Phase Encoding". **Information Science, Vol.277,445-457**

[3] Bouguezel, S. 2012. "A Reciprocal-Orthogonal Parametric Transform And Its Fast Algorithm"**, IEEE Signal Process, Lett.19(11), 769-772**

[4] Zhou, Y., et al. 2012. "Image Encryption using P-Fibonacci Transform and IJADecomposition". **Optics Communications Vol.285, 594-608**

[5] Liao, X., et al. 2010. "A Novel Image Encryption Algorithm based Onself-adaptive Wave Transmission". **Signal Process, Vol. 90, 2714-2722**

[6] Singh, L.D & Singh, K.M. 2015. "Image Encryption pusing Elliptic Curve Cryptography". **Procedia Computer Science, Vol 54, 472-481.**

[7] Liu, Hong & Liu, Yanbing. 2014. "Criptanalyzing an Image Encryption Scheme based on Hybrid Chaotic System Ana Cyclic Elliptic Curve". **Optics & Laser Technology, Vol. 56, 15-19**

[8] Bao, Long& Zhou Yicong. 2015. "Image Encryption:Generating Visually Meaningful Image". **Information Science, Vol. 324, 197-207**

[9] Stallings, William.2011. "Cryptography And Network Security Principles and Practice Fifth Edition". **Pearson. USA.**

[10] Gonzales, R.C., et al. 2001. “Digital Image Processing Second Edition”. **Gatesmark. USA.**

[11] Nidhal Khdhair El Abbadi, et.al. 2016. “New Image Encryption Algorithm Based on Diffie-Hellman and Singular Value Decomposition”. **IJARCCE. Vol. 5, Issue 1**

[12] S., William, (2002), “Komunikasi Data dan Komputer:Jaringan Komputer”, Thamir Abdul Hafedh Al_Hamdany (Penterjemah). **Salemba Teknik. Jakarta.**

[13] Cohen, Henri & Frey, Gerhard. 2006. “Handbook of Elliptic And Hyperelliptic Curve Cryptography”. **Chapman & Hall/CRC. USA**

[14] Walker, James S. 1999. “A Primer on Wavelets and their Scientific Applications” . **Chapman & Hall/CRC. USA**

[15] Susanti Pungki. 2012. “Implementasi Watermarking Citra Digital Berbasis *Lifting Scheme* Menggunakan Permutasi RC-4 dan Linear Congruential Generator (LCG)**”. Tugas Akhir Politeknik Negeri Bandung**

LAMPIRAN

1. Kode Program titik.m

```
function mapping = titik(a,b,n);
x=0;
x=0;
y=0;
i=1;
while y<=n-1
    z(i)=mod(y^2,n);
    yy(i)=y;
    y=y+1;
    i=i+1;
end
i=1;
w=1;
while x<=n-1
    nilaix(i)=mod(x^3+a*x+b,n);
     xx(i)=x;
    i=i+1;
    x=x+1;
end
idx=1;
id=1;
for i=1:length(nilaix)
   if(any(nilaix(i)==z)==1)
       cari(:,idx)=find(nilaix(i)==z);
       [pan leb]=size(cari);
       for j=1:pan
         if(yy(cari(j,idx)==0))
                j=2;
          titikxy(id,:)=[xx(i) yy(cari(j,idx))];
         else
          titikxy(id,:)=[xx(i) yy(cari(j,idx))];
            end
        id=id+1;
        end
        idx=idx+1;
    end
```

LAMPIRAN (LANJUTAN)

```
end
mapping=unique(titikxy,'rows');
map_0=[Inf Inf];
mapping=[map_0;mapping];
```

2. Kode Program Penjumlahan Titik (addell.m)

```
function p3 = addell(p1,p2,a,b,n);
% This function add points on the elliptic curve
%    y^2 = x^3 + ax +b mod n
% The points are represented by
% p1(1) = x1    p1(2) = y1
% p2(1) = x2    p2(2) = y2
if (any(p1==Inf)),
   p3=p2;
   return;
end;

if (any(p2==Inf)),
   p3=p1;
   return;
end;

x1=p1(1);
x2=p2(1);
y1=p1(2);
y2=p2(2);
z1=1;  % this will store the gcd incase the
addition produced a factor of n

if ( (x1==x2) & (y1==y2) & (y1==0)),  % an
infinity case
   p3(1)=inf; p3(2)=inf;
   return;
end;

if ( (x1==x2) & (y1 ~= y2)),           % an
infinity case
```

LAMPIRAN (LANJUTAN)

```
   p3(1)=inf; p3(2)=inf;
   return;
end;

if (all(p1==p2) & (gcd(y1,n)~=1) & (gcd(y1,n)
~=n)),
   z1=gcd(y1,n);
   p3=[];
   disp(['Elliptic Curve addition produced a
factor of n, factor = ',num2str(z1)]);
   return;
end;
if all(p1==p2),
   temp=mod(2*y1,n);
   if temp==0,
      p3(1)=Inf;
      p3(2)=Inf;
      return;
   end;
   den=powermod(2*y1, -1, n);
   num=mod(x1*x1,n);
   num=mod(mod(3*num,n) + a,n);
   m=mod(num*den,n);
   temp=mod(m*m,n);
   x3=mod(temp-x1-x2, n);
   temp=x1-x3;
   y3=mod(m*temp,n);
   y3=mod(y3-y1,n);
else   % case p1 ~= p2
   if (gcd(x2-x1,n) ~= 1),
      z1=gcd(x2-x1,n);
      p3=[];
      disp(['Elliptic Curve addition produced a
factor of n, factor= ',num2str(z1)]);
      return;
    end; % end if gcd
   temp=mod(x2 - x1,n);
```

LAMPIRAN (LANJUTAN)

```
    if (mod(n,temp)==0),    % Infinity case
       p3(1)=Inf;
       p3(2)=Inf;
       return;
    end;
    den=powermod(temp,-1,n);
    num=y2-y1;
    num=mod(y2-y1,n);
    m=mod(num*den,n);
    temp=mod(m*m,n);
    x3=mod(temp-x1-x2, n);
    temp=x1-x3;
    y3=mod(m*temp,n);
    y3=mod(y3-y1,n);
end;

p3(1)=x3;
p3(2)=y3;
```

3. Kode Program Perkalian titik(multell.m)

```
function y = multell(p,M,a,b,n);
% This function prints the Mth multiple of p on
the elliptic
%  curve with coefficients a and b mod n.
z1=M;
y=[inf inf];
while (z1 ~=0),
    while (mod(z1,2) ==0),
        z1=(z1/2);
        p=addell(p,p,a,b, n)
        if (length(p)==0),
           y=[];
           disp('Multell found a factor of n and
exited');
           z1
           return;
        end;
    end;  %end while
```

LAMPIRAN (LANJUTAN)

```
    z1=z1-1;
    y=addell(y,p,a,b,n)
    if (length(y)==0),
       disp('Multell found a factor of n and
exited');
       z1
       return;
     end;
 end;
```

4. Kode Program Invers Modulo (invmod.m)

```
function y = invmodn( b,n);
% This function calculates the inverse of an
element b mod n
% It uses the extended euclidean algorithm

n0=n;
b0=b;
t0=0;
t=1;

q=floor(n0/b0);
r=n0-q*b0;
while r>0,
   temp=t0-q*t;
   if (temp >=0),
      temp=mod(temp,n);
   end;
   if (temp < 0),
      temp= n - ( mod(-temp,n));
   end;
   t0=t;
   t=temp;
   n0=b0;
   b0=r;
   q=floor(n0/b0);
```

LAMPIRAN (LANJUTAN)

```
    r=n0-q*b0;
end;

if b0 ~=1,
    y=[];
    disp('No inverse');
else
    y=mod(t,n);
end;
```

5. Kode Program Modulo Berpangkat (powermod.m)

```
function y = powermod(a,z,n)
% This function calculates y = a^z mod n
% If a is a matrix, it calculates a(j,k)^z mod
for every element in a
[ax,ay]=size(a);


% If a is negative, put it back to between 0 and
n-1
a=mod(a,n);

% Take care of any cases where the exponent is
negative
if (z<0),
    z=-z;
    for j=1:ax,
        for k=1:ay,
            a(j,k)=invmodn(a(j,k),n);
        end;
    end;
end;


for j=1:ax,
for k=1:ay,
  x=1;
```

LAMPIRAN (LANJUTAN)

```
  a1=a(j,k);
  z1=z;
  while (z1 ~= 0),
     while (mod(z1,2) ==0),
        z1=(z1/2);
        a1=mod((a1*a1), n);
     end;  %end while
     z1=z1-1;
     x=x*a1;
     x=mod(x,n);
  end;
  y(j,k)=x;
end; %end for k
end; %end for j
```

6. Kode Program Enkripsi (enkrip.m)

```
function enkrip(im,citra,mapping,a,b,n,p3);
im=imresize(im,[128 128]);
im=double(im);
[x2,y2]=size(im);
citra=imresize(citra,[2*x2 2*y2]);
gray=double(im(:)');
[panjang,lebar]=size(mapping);
tambah=[0.5 0.5];
while(ceil(panjang/256)~=floor(panjang/256))
    mapping=[mapping;tambah];
    panjang=panjang+1;
end
mapped=reshape(mapping',
[lebar,256,panjang/256]);
haha= mapped(:,:);
for i=1:length(gray)
    r(i) = randi([1 panjang/256]);
    abu(i,:)=mapped(:,gray(i)+1,r(i));
    while abu(i,:)==[0.5 0.5]
        r(i) = randi([1 panjang/256]);
        abu(i,:)=mapped(:,gray(i)+1,r(i));
```

LAMPIRAN (LANJUTAN)

```
    end
    result(i,:)=addell(abu(i,:),p3,a,b,n);
end
[a1,b1]=size(result);
for i=1:a1
    row=find(haha==result(i,1));
    column=find(haha==result(i,2));
    pos(i)=0;
    ukur=length(column);
    idx=1;
    ketemu=any(bsxfun(@minus, row,
column(idx))==-1);
    while ketemu==0
        idx=idx+1;
        ketemu=any(bsxfun(@minus, row,
column(idx))==-1);
    end
    rr=find(bsxfun(@minus, row, column(idx))==-
1);
    pos(i)=row(rr);
end
pos=ceil(pos/2);
col=ceil((pos/256));
ro=mod(pos,256);
col=reshape(col,sqrt(length(col)),sqrt(length(co
l)));
ro=reshape(ro,sqrt(length(ro)),sqrt(length(ro)))
;
figure
col=uint8(col);
 cv=floor(ro/10);
 cd=mod(ro,10);
LS = liftwave('lazy','Int2Int');
[CA,CH,CV,CD] = lwt2(double(citra),LS);
X=ilwt2(CA,CH,cv,cd,LS);
X=uint8(X);
imshow(X);
imwrite(X,'hasilenkrip.bmp');
```

LAMPIRAN (LANJUTAN)

```
imwrite(col,'kolom.bmp');
```

7. Kode Program Dekripsi (dekrip.m)

```
function dekrip(im,image,mapping,a,b,n,p3);
LS = liftwave('lazy','Int2Int');
[aa,bb,cc,dd]=lwt2(double(image),LS);
h=10*cc+dd;
h=uint8(h);
[x2,y2]=size(h);
gray=double(h(:)');
[x3,y3]=size(im);
gray1=double(im(:)');
ketemu=find(gray==0);
gray(ketemu)=256;
[panjang,lebar]=size(mapping);
tambah=[0.5 0.5];
p4=[p3(1) n-p3(2)];
while(ceil(panjang/256)~=floor(panjang/256))
    mapping=[mapping;tambah];
    panjang=panjang+1;
end
mapped=reshape(mapping',
[lebar,256,panjang/256]);
haha= mapped(:,:);
for i=1:length(gray)
    abu(i,:)=mapped(:,gray(i),gray1(i));
    while abu(i,:)==[0.5 0.5]
        gray(i)=gray(i)-1;
        abu(i,:)=mapped(:,gray(i),gray1(i));
    end
    result(i,:)=addell(abu(i,:),p4,a,b,n);
end
[a1,b1]=size(result);
for i=1:a1
    row=find(haha==result(i,1));
    column=find(haha==result(i,2));
    pos(i)=0;
```

LAMPIRAN (LANJUTAN)

```
    ukur=length(column);
    idx=1;
    ketemu=any(bsxfun(@minus, row,
column(idx))==-1);
    while ketemu==0
        idx=idx+1;
        ketemu=any(bsxfun(@minus, row,
column(idx))==-1);
    end
    rr=find(bsxfun(@minus, row, column(idx))==-
1);
    pos(i)=row(rr);
end
pos=floor(pos/2);
col=ceil((pos/256));
ro=mod(pos,256);
col=reshape(col,sqrt(length(col)),sqrt(length(co
l)));
ro=reshape(ro,sqrt(length(ro)),sqrt(length(ro)))
;
figure
ro=uint8(ro);
imshow(ro);
col=uint8(col);
imwrite(ro,'hasildekrip.bmp');
```

8. Kode Program Antar Muka (enkripsi.m)

a. Pembentukan Titik (button titik)

```
a = get(handles.edittext_a,'String');
a=str2num(a);
b = get(handles.edittext_b,'String');
b=str2num(b);
n= get(handles.edittext_p,'String');
n=str2num(n);
while mod((4*a^3+27*b^2),n)==0
   set(handles.txtpoint, 'String', 'Inputan
salah,input kembali nilai!');
end
```

LAMPIRAN (LANJUTAN)

```
mapping=titik(a,b,n);
set(handles.uitable1,'Data',mapping);
```

b. Pembuatan *Public Key* (button p3)

```
a = get(handles.edittext_a,'String');
a=str2num(a);
b = get(handles.edittext_b,'String');
b=str2num(b);
n = get(handles.edittext_p,'String');
n=str2num(n);
x = get(handles.edittext_x,'String');
x=str2num(x);
y = get(handles.edittext_y,'String');
y=str2num(y);
pk = get(handles.edittext_pk,'String');
titik_a=[x y];
pk=str2num(pk);
p3=multell(titik_a,pk,a,b,n);
txtp3=num2str(p3);
set(handles.textp32, 'String', txtp3);
```

c. Input Citra (button input image)

```
[aa ba]=uigetfile('*.*','All Files');
img=imread([ba aa]);
imshow(img,'Parent',handles.axes1);
```

d. Enkripsi (button enkrip)

```
a = get(handles.edittext_a,'String');
a=str2num(a);
b = get(handles.edittext_b,'String');
b=str2num(b);
n = get(handles.edittext_p,'String');
n=str2num(n);
x2 = get(handles.edittext_x2,'String');
x2=str2num(x2);
y2 = get(handles.edittext_y2,'String');
```

LAMPIRAN (LANJUTAN)

```
y2=str2num(y2);
pk = get(handles.edittext_pk,'String');
titik_a1=[x2 y2];
pk=str2num(pk);
p3=multell(titik_a1,pk,a,b,n);
mapping=titik(a,b,n);
im=getimage(handles.axes1);
citra=getimage(handles.axes4);
enkrip(im,citra,mapping,a,b,n,p3);
```

e. Dekripsi (button dekrip)

```
a = get(handles.edittext_a1,'String');
a=str2num(a);
b = get(handles.edittext_b1,'String');
b=str2num(b);
n = get(handles.edittext_p1,'String');
n=str2num(n);
x2 = get(handles.edittext_x12,'String');
x2=str2num(x2);
y2 = get(handles.edittext_y12,'String');
y2=str2num(y2);
pk = get(handles.edittext_pk1,'String');
titik_a1=[x2 y2];
pk=str2num(pk);
p3=multell(titik_a1,pk,a,b,n);
mapping=titik(a,b,n);
image=getimage(handles.axes3);
im=getimage(handles.axes5);
dekrip(im,image,mapping,a,b,n,p3);
```

BIODATA PENULIS

Penulis bernama Agus Setiawan, lahir di kota Lamongan, 25 Agustus 1995. Penulis berasal dari Kota Surabaya, bertempat tinggal di Jalan Simo Katrungan Kidul nomor 35 Surabaya. Pendidikan formal yang pernah ditempuh yaitu pendidikan dasar di SDN Banyu Urip VI Surabaya sepanjang tahun 2001-2007, dan pendidikan menengah pertama di SMP Negeri 3 Surabaya pada tahun 2007-2010. Dan melanjutkan pendidikan menengah atas di SMA Negeri 1 Surabaya sejak tahun 2010-2013. Mulai menyandang status mahasiswa di Jurusan Matematika FMIPA ITS pada tahun 2013 melalui jalur penerimaan SBMPTN dan menyelesaikan studi S1 nya di tahun 2017. Semasa di bangku perkuliahan, Penulis mengikuti beberapa organisasi kemahasiswaan. Diantaranya adalah Himpunan Mahasiswa Matematika (Himatika) ITS, Lembaga Dakwah Jurusan (LDJ) Ibnu Muqlah Matematika ITS serta event Olimpiade Matematika ITS. Selama masa perkuliahan, penulis aktif di Lab. Ilmu Komputer Matematika untuk memperdalam skill berbasis keilmuan *programming*. Penulis dapat dihubungi lebih lanjut di agus103setiawan@gmail.com.

“Halaman ini sengaja dikosongkan”