BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Sebagai bahan pertimbangan dalam penelitian ini akan dicantumkan hasil penelitian terdahulu oleh peneliti yang pernah penulis baca diantaranya penelitian yang dilakukan oleh :

Peneliti yang dilakukan oleh (Studi et al., 2018)dengan judul "Analisis dan Desain Algoritma Hybrid Kriptografi untuk Manajemen Strategi Pengamanan Data Perusahaan" Peneliti ini menggunakan metode RSA dan ElGamal dengan aplikasi pengaman dokumen menggunakan algoritma kriptografi asimetris RSA dan ElGamal dapat melakukan enkripsi dan melakukan dekripsi (pengembalian dokumen seperti semula) dengan keakuratan rata-rata sebesar 85,57%.

Penelitian yang dilakukan oleh (Allan et al., n.d.) dengan judul "Implementasi Algoritma AES, ElGamal, dan SHA3 untuk Keamanan File Digital" Peneliti ini menggunakan metode algoritma AES, algoritma ElGamal, dan fungsi hash SHA3 agar dapat digunakan secara bersamaan menjadi suatu sistem enkripsi yang aman. Pengujian dalam penelitian ini terbagi menjadi tiga tahap, yaitu pengujian kemampuan sistem dalam melakukan proses enkripsi dan dekripsi, pengujian kecepatan sistem, dan pengujian web platform. Dari hasil pengujian dapat dibuktikan bahwa program dapat melakukan enkripsi dan dekripsi terhadap semua jenis file dengan baik dan dapat mendeteksi keaslian file tersebut.

Pada pengujian web, dapat diketahui bahwa web mampu mengirimkan file yang terenkripsi kepada penerima dengan baik tanpa ada satupun yang hilang.

Penelitian ini menghasilkan sebuah perangkat lunak yang dapat melakukan enkripsi dan dekripsi terhadap berbagai jenis file dan dapat memverifikasi jika terdapat perubahan/modifikasi terhadap file tersebut. Penelitian ini juga menghasilkan sebuah web platform yang dapat digunakan untuk melakukan pengiriman file antar pengguna dengan benar. Dari hasil pengujian, program dapat melakukan enkripsi pada file dengan kecepatan rata-rata yaitu 999.43 bytes/ms dan melakukan dekripsi pada file dengan kecepatan rata-rata yaitu 450.38 bytes/ms. Dan adapun saran untuk penelitian selanjutnya diantaranya yaitu program dapat dikembangkan agar bisa dijalankan pada perangkat smartphone. Kemudian dalam melakukan proses algoritma, program dapat dikembangkan agar bisa berjalan pada multiple thread untuk meningkatkan efisiensi program. Dan yang terakhir, tampilan pada program dapat dibuat lebih menarik agar tampilan program tidak terkesan tua/klasik.

Penelitian yang dilakukan oleh (Irawan & Rachmawanto, n.d.)pada tahun 2021 dengan judul “Keamanan Data Menggunakan Gabungan Kriptografi AES Dan RSA” peneliti ini menggunakan metode Algoritma AES, algoritma RSA Tujuan gabungan ini adalah meningkatkan nilai Avalanche Effect pada hasil implementasi serta untuk menganalisa waktu kebutuhan untuk proses enkripsi dan dekripsi. Pada penelitian ini, digunakan sejumlah data teks, audio dan video dengan jumlah byte antara 100 smapai 4000 bytes. Telah dilakukan perbandingan hasil perhitungan Avalanche effect (AE) pada AES, modified AES dan AES-RSA

melalui 15 kali percobaan. Hasil perbandingan menunjukkan bahwa AES-RSA menghasilkan AE paling tinggi yaitu 44,74%. Adapun saran untuk pengembangan selanjutnya adalah menggunakan hardware yang lebih baru untuk mempercepat proses enkripsi dan dekripsi. File yang dapat dienkripsi dan dekripsi tidak hanya file pdf, word, ppt, excel, mp3, gambar dan video tetapi dapat dikembang untuk file yang lain.

Penelitian (*15_135_Naskah_Publikasi_130-139 (1)*, n.d.) dengan judul " Implementasi Kriptografi File Ujian Siswa Dengan Metode Rsa Berbasis Website Di SMAN 84 Jakarta" peneliti menggunakan metode Kriptografi RSA untuk mengamankan file soal ujian siswa berekstensi .txt. Dokumen yang dipakai pada proses penelitian ini adalah dokumen soal ujian siswa milik SMA Negeri 84 Jakarta. Hasil penelitian ini berupa suatu sistem aplikasi kriptografi pada file berekstensi .txt menggunakan algoritme kriptografi RSA. Berdasarkan hasil pengujian yang dilakukan, terdapat perubahan besar ukuran file enkripsi, yaitu peningkatan ukuran dari ukuran semula, namun tidak terjadi perubahan besar file hasil dekripsi dengan ukuran file awal. Kecepatan rata-rata saat melakukan enkripsi file adalah 46.8 ms, dan 56.2 ms untuk melakukan dekripsi. Tujuan dalam melakukan penelitian ini adalah membangun aplikasi untuk menjaga serta meningkatkan keamanan maupun kerahasiaan data dengan enkripsi dan dekripsi file soal ujian siswa menggunakan metode RSA, yang memiliki tiga proses dalam mengamankan data, yaitu proses generate key, proses generate key, proses enkripsi, dan proses dekripsi. Selain itu, aplikasi juga dibuat dengan web-based dengan otorisasi user sehingga memudahkan untuk diakses oleh banyak user.

Aplikasi ini masih memiliki kekurangan sehingga memerlukan perbaikan dan pengembangan lebih lanjut. Beberapa masukan yang dapat ditambahkan untuk mengembangkan dan menyempurnakan aplikasi ini, yaitu : aplikasi ini diharapkan untuk dapat disempurnakan lebih jauh dengan memperbanyak ekstensi atau format file yang dapat dienkripsi dan didekripsi, seperti .pdf, .doc, .xls, dan ekstensi lainnya, serta dapat melakukan enkripsi dan dekripsi file berupa gambar dan simbol.

Penelitian yang dilakukan oleh (Pramitasari, n.d.) dengan judul "Algoritma Optimasi Chaos Pada Ridge Polynomial Neural Network Untuk Kriptanalisis Kunci Publik Elgamal" pada bulan November 2022 mengkriptanalisis kunci public ElGamal dengan menggunakan Algoritma Optimasi Chaos pada Ridge Polynominal Neural Network. Nilai awal Penerimaot Ridge Polynomial Neural Network menggunakan hasil dari Algoritma Optimasi Chaos karena untuk mengatasi iterasi yang terjebak ke dalam minimum local. Variabel chaos di dalam range tertentu memiliki beberapa fitur yaitu Randomness, property ergodis dan regularity. Fungsi chaos sederhana adalah Logistic Map. Hasil menunjukkan bahwa penelitian menghasilkan kunci privat dengan akurat dan baik dimana menghasilkan iterasi yang konvergen.

Penelitian yang dilakukan oleh (Pendahuluan et al., n.d.)dengan judul " Manajemen Kunci Menggunakan Kombinasi dari Pertukaran Kunci Diffie–Hellman dengan Enkripsi AES" dilakukan aplikasi kriptografi untuk keamanan data oleh menggunakan pertukaran kunci Diffie-Hellman dan algoritma AES adalah untuk mengamankan data saat mengirim, pengirim pertama penerima

menukar kunci atau menghasilkan kunci untuk file yang dienkripsi oleh menggunakan pertukaran kunci Diffie-Hellman, setelah mendapatkan kunci file akan dienkripsi dengan menggunakan algoritma AES untuk enkripsi dan file dekripsi, pengirim akan mengirim file ke pihak penerima melalui server terlebih dahulu untuk menyimpan data, kemudian penerima untuk mendekripsi file membaca file agar dapat dibaca kembali oleh penerima. Di dalam tugas, aplikasi yang menggabungkan Diffie-Hellman. Pertukaran Kunci dengan algoritma enkripsi AES. Aplikasi ini yang pertama kali menggunakan Pertukaran Kunci Diffie-Hellman untuk mencapai tujuan bertukar kunci rahasia bersama. Hasil menunjukkan bahwa dengan menggunakan Diffie-Hellman dengan kunci 1024 bit keamanannya setara dengan algoritma kunci 3DES 2, jika Diffie-Hellman dengan 2048 kunci bit maka keamanannya sama dengan kunci 3DES 3, jika Diffie-Hellman dengan kunci 3072 bit maka keamanan yang sama dengan AES-128 bit, jika Diffie-Hellman dengan kunci 7680 bit maka keamanannya setaraAES-182 bit, dan jika Diffie-Hellman dengan kunci 15360 bit maka keamanan setara dengan AES-256 bit.

Penelitian yang dilakukan (Nisa et al., 2020)dengan judul “ Aplikasi Enkripsi Citra Dan Teks Menggunakan Algoritma Diffie-Hellman Dan ElGamal” Pada Januari 2020, Keamanan data menjadi hal terpenting dalam penyimpanan informasi. Salah satunya adalah keamanan dalam bertukar pesan. Oleh karena itu, diperlukan metode enkripsi. Dalam enkripsi, ada enkripsi simetris dan asimetris. Enkripsi asimetris dianggap lebih aman karena menggunakan dua jenis kunci yang berbeda dibandingkan dengan enkripsi simetris yang hanya menggunakan

satu jenis kunci. ElGamal merupakan salah satu algoritma enkripsi asimetris yang keunggulannya terletak pada perhitungan logaritma diskrit yang sulit dipecahkan Sedangkan Diffie-Hellman merupakan salah satu algoritma proses pertukaran kunci yang menghasilkan kunci rahasia selama proses komunikasi untuk menjaga kerahasiaan kunci tersebut. Pada penelitian ini, kombinasi algoritma Diffie-Hellman dan ElGamal digunakan untuk mengamankan pesan teks dan gambar. gabungan dari kedua algoritma ini terdiri dari 4 proses yaitu proses pertukaran kunci, pembangkitan kunci, enkripsi, dan dekripsi. Dari hasil pengujian diperoleh rata-rata waktu enkripsi file teks sebesar 119,9 ms, dekripsi 248,3 ms, dan throughput 600,96 Kbps. Ratarata waktu enkripsi file citra sebesar 2623,4 ms dan waktu dekripsi 4349,5 ms. Dari hasil pengujian pada beberapa dimensi citra yang berbeda, rata-rata MSE pada citra sebesar 213,9 dan PSNR sebesar 173,27 dB. Nilai ini menunjukkan kualitas citra cukup baik (diatas 40 dB). Semakin besar dimensi citra, maka nilai MSE tiap intensitas piksel akan semakin kecil, dengan rata-rata penurunan sebesar 27.67%. Semakin besar dimensi citra, maka nilai PSNR akan semakin besar, dengan kenaikan rata-rata sebesar 1,94%. Kombinasi algoritma Diffie Hellman dan ElGamal menghasilkan nilai avalanche effect sebesar 85,18% untuk file teks dan 84,46% untuk file citra. Nilai ini menunjukkan kombinasi algoritma cukup baik dengan hasil penyelesaian bit di atas 60%.

Penelitian yang dilakukan (Elgamal dengan Pertukaran Kunci Diffie Hellman pada Aplikasi Keamanan Citra Sidik Jari Berbasis Android Nurul Yalisa & Arhami, 2018) Aplikasi pengamanan data pada citra sidik jari berbasis Android dengan menggunakan algoritma *Elgamal* serta *Diffie Hellman* sebagai metode

pertukaran kunci. Algortima *Diffie Hellman* digunakan untuk proses pertukaran dan pembangkitan kunci sedangkan algoritma *elgamal* digunakan untuk proses enkripsi dan dekripsi. Citra yang akan dienkripsi pada aplikasi ini adalah citra sidik jari berukuran 180 x 230 piksel. Proses enkripsi akan menghasilkan *chiper* citra berukuran 460 x 360 piksel atau berukuran 2 kali dari ukuran citra aslinya, selanjutnya di dekripsi untuk menghasilkan *plain* citra (citra asli). Dari 30 sampel yang di uji dengan menggunakan kunci 16 bit, hasil enkripsi dan dekripsi berhasil dilakukan dengan waktu rata-rata waktu enkripsi 102,482 detik dan rata-rata waktu dekripsi 34,151 detik.

Tabel II. 1 Penelitian Terkait

Judul	Metode	Proses Kerja	Hasilnya/akurasi
1. Analisis dan Desain Algoritma Hybrid Kriptografi untuk Manajemen Strategi Pengamanan Data Perusahaan (Studi et al., 2018)	Algoritma hybrid (RSA dan ElGamal).	A. Pembuatan Kunci Pembuatan kunci ini digunakan untuk melakukan membuat file kunci publik dan file kunci privat yang nantinya akan digunakan untuk melakukan proses enkripsi dan proses dekripsi dokumen. B. Halaman Enkripsi Dokumen Halaman enkripsi dokumen ini merupakan halaman yang digunakan untuk melakukan enkripsi terhadap dokumen. Dibutuhkan file kunci public yang dibuat pada halaman pembuatan kunci untuk melakukan enkripsi dokumen. Dokumen hasil dari enkripsi dokumen mempunyai ekstensi yang sama dengan dokumen	Dengan aplikasi pengaman dokumen menggunakan algoritma kriptografi asimetris *RSA* dan *ElGamal* dapat melakukan enkripsi dan melakukan dekripsi (pengembalian dokumen seperti semula) dengan keakuratan rata-rata sebesar 85,57%.

		aslinya C. Halaman Dekripsi Dokumen Halaman dekripsi dokumen ini merupakan halaman yang digunakan untuk mendekripsi atau mengembalikan dokumen yang telah terenkripsi menjadi dokumen aslinya atau seperti d okumen sebelum dilakukan proses enkripsi. D. Pengujian Sistem Pengujian sistem dilakukan dengan cara menguji keakuratan dalam pengembalian dokumen seperti aslinya setelah dilakukan enkripsi. Dokumen yang diujikan adalah dokumen yang mempunyai ekstensi doc dan docx.	

2. Implementasi Algoritma AES, ElGamal, dan SHA3 untuk Keamanan File Digital (Allan et al., n.d.)	Algoritma AES, Algoritma ElGamal. Algoritma SHA3	1. Desain Sistem Penelitian ini menggunakan model skema hybrid yang menggabungkan algoritma AES, ElGamal, dan SHA3. Algoritma AES digunakan dalam proses enkripsi dan dekripsi data (data encapsulation), algoritma ElGamal digunakan dalam proses enkripsi dan dekripsi kunci (key encapsulation), sedangkan algoritma SHA3 digunakan untuk memverifikasi integritas data (data integrity check). Algoritma AES menggunakan AES 256-bit dengan panjang kunci 32 bytes dan menggunakan round sebanyak 14 putaran. Algoritma hash SHA3 menggunakan SHA3-512 yang mengeluarkan output berupa message digest sebesar 512-bit. Algoritma ElGamal memerlukan kunci yang berbeda untuk proses	1. Program dapat melakukan proses enkripsi dan dekripsi pada file dengan baik, sehingga dapat disimpulkan bahwa algoritma simetris AES, algoritma kunci publik ElGamal, dan fungsi hash SHA3 dapat digunakan secara bersamaan untuk menciptakan sebua h sistem enkripsi yang aman. 2. Proses enkripsi dengan sistem ini dapat dilakukan secara berulang kali (Cipher Block Chaining / CBC) dengan menggunakan

		enkripsi dan dekripsinya. 2. Rancangan Proses Enkripsi Rancangan proses enkripsi menggunakan kombinasi algoritma AES, ElGamal, dan SHA3 3. Rancangan Proses Dekripsi Rancangan proses dekripsi menggunakan kombinasi algoritma AES, ElGamal, dan SHA3	password dan kunci yang berbeda sehingga dapat memberikan proteksi berlapis pada file. 3. Program dapat melakukan enkripsi dan dekripsi terhadap jenis file yang berbeda-beda sehingga dapat disimpulkan bahwa program dapat digunakan untuk meng-enkripsi berbagai jenis file. 4. File yang terenkripsi hanya dapat didekripsi dengan menggunakan kunci publik dari pengirim dan

			kunci pribadi dari penerin sehingga program dapat memastikan bahwa pengirim dan penerima file saja yang berwenang atas file tersebut. 5. Rata-rata kecepatan proses enkripsi pada program adalah 999.43 bytes/ms, sedangkan rata-rata kecepatan proses dekripsi pada program adalah 450.38 bytes/ms
3. Keamanan Data Menggunakan Gabungan	Gabungan Algoritma	Menggabungkan antara kedua algoritma tersebut, penulis juga mencoba untuk menenkripsi file. Dan	Dari hasil perancangan dan pembuatan aplikasi Enkrispi dan

Kriptografi AES Dan RSA (Irawan & Rachmawanto, n.d.)	AES, Algoritma RSA	mendekripsi kan file tersebut. Penggunaan aplikasi ini menggunakan java dan aplikasinya menggunakan NetBeans.	deskripsi dengan Kriptografi menggunakan Algoritma AES dan algoritma RSA maka didapatkan beberapa hasil. Dari hasil percobaan yang telah dilakukan pada aplikasi ini dapat mengenkripsi dan mendekripsi file file pdf, word, ppt, excel, mp3, gambar dan video dengan sempurna. Setelah file pdf, word, ppt, excel, mp3, gambar dan video dienkripsi terdapat kenaikan besar file rata-rata mencapai 0,030463878% dari file asli. Setelah file pdf, word, ppt, excel, mp3,

			gambar dan video yang terenkripsi didekripsi maka file kembali ke ukuran semula. Semakin besar ukuran file yang dienkripsi semakin lama juga waktu yang dienkripsi.
4. Implementasi Kriptografi File Ujian Siswa Dengan Metode Rsa Berbasis Website Di SMAN 84 Jakarta (*15_135_Naskah_Publikasi_130-139 (1)*, n.d.)	Kriptografi RSA	Tahap pembangkitan pasangan kunci, tahap enkripsi, dan dekripsi. Impelementasi metode merupakan implementasi atau contoh penggunaan dari metode yang digunakan.	aplikasi kriptografi file soal ujian siswa dengan metode RSA yang dibuat mampu melakukan enkripsi dan dekripsi file soal yang diujikan, dengan karakter berupa huruf dan angka. Aplikasi ini memiliki fitur pengguna yang hanya dapat digunakan oleh admin untuk menambahkan, mengubah, atau

			menghapus akun pengguna. Proses enkripsi file yang dilakukan pada aplikasi ini memerlukan waktu yang lebih cepat daripada proses dekripsi. Aplikasi ini memerlukan kecepatan rata-rata selama 46.8 ms untuk melakukan enkripsi, sedangkan untuk mel akukan dekripsi, aplikasi ini memerlukan kecepatan proses rata-rata 56.2 ms. File hasil enkripsi pada aplikasi ini mengalami peningkatan ukuran file dengan rata-rata peningkatan sebesar 308.49% dari file semula, sedangkan ukuran

			file hasil dekripsi tidak mengalami perubahan dari ukuran file semula
5. Algoritma Optimasi Chaos pada Ridge Polynomial Neural Network Untuk Kriptanalisis Kunci Publik Elgamal (Pramitasari, n.d.)	Kunci public ElGamal dengan menggunakan Algoritma Optimasi Chaos pada Ridge Polynominal Neural Network	Penelitian menggunakan data pelatihan adalah 40 baris terdiri dari kunci public dan cyphertext. Data pengujian adalah 9 baris kunci public dan cyphertext. Data target pelatihan dan pengujian adalah kunci privat	Nilai awal Penerimaot Ridge Polynomial Neural Network menggunakan hasil dari Algoritma Optimasi Chaos karena untuk mengatasi iterasi yang terjebak ke dalam minimum local. Variabel chaos di dalam range tertentu memiliki beberapa fitur yaitu Randomness, property ergodis dan regularity. Fungsi chaos sederhana adalah Logistic Map. Hasil menunjukkan bahwa penelitian

			menghasilkan kunci privat dengan akurat dan baik dimana menghasilkan iterasi yang konvergen.
6. Manajemen Kunci Menggunakan Kombinasi dari Pertukaran Kunci Diffie–Hellman dengan Enkripsi AES (Pendahuluan et al., n.d.)	Diffie Hellman; algoritma AES	Aplikasi kriptografi untuk keamanan data dengan menggunakan pertukaran kunci Diffie-Hellman dan algoritma AES untuk mengamankan data pada saat pengiriman, pengirim pertama dari penerima menukar kunci atau menghasilkan kunci untuk file yang dienkripsi dengan menggunakan Diffie-Hellman pertukaran kunci, setelah mendapatkan file kunci akan dienkripsi dengan menggunakan algoritma AES untuk enkripsi dan dekripsi file, pengirim akan mengirim file ke pihak penerima	Hasil menunjukkan bahwa dengan menggunakan Diffie-Hellman dengan kunci 1024 bit keamanannya setara dengan algoritma kunci 3DES 2, jika Diffie-Hellman dengan 2048 kunci bit maka keamanannya sama dengan kunci 3DES 3, jika Diffie-Hellman dengan kunci 3072 bit maka keamanan yang sama dengan AES-128 bit, jika Diffie-Hellman

		melalui server terlebih dahulu untuk menyimpan data, kemudian penerima mendekripsi file untuk membaca file tersebut sehingga dapat dibaca kembali oleh penerima. Dalam tugas ini, aplikasi yang menggabungkan Pertukaran Kunci Diffie-Hellman dengan algoritma enkripsi AES. Aplikasi ini yang pertama kali menggunakan Pertukaran Kunci Diffie-Hellman untuk mencapai tujuan pertukaran kunci rahasia bersama.	dengan kunci 7680 bit maka keamanannya setaraAES-182 bit, dan jika Diffie-Hellman dengan kunci 15360 bit maka keamanan setara dengan AES-256 bit.
7. Aplikasi Enkripsi Citra Dan Teks Menggunakan Algoritma Diffie-Hellman Dan ElGamal (Nisa et al., 2020)		Aplikasi yang dibangun adalah aplikasi enkripsi Click or tap here to enter text.yang dapat mengirim pesan terenkripsi dan mendekripsi pesan yang diterima. Faktor yang mendasari dibentuknya aplikasi ini adalah keamanan data. Keamanan data telah menjadi aspek	Hasil pengujian diperoleh rata-rata waktu enkripsi file teks sebesar 119,9 ms, dekripsi 248,3 ms, dan throughput 600,96 Kbps. Ratarata waktu enkripsi file citra sebesar

		yang sangat penting dari suatu sistem informasi. Untuk keperluan tersebut, maka diperlukan sebuah teknik kriptografi dengan menggunakan algoritma enkripsi dan dekripsi data. Algoritma enkripsi yang digunakan dalam penelitian ini adalah algoritma enkripsi Diffie-Hellman dan ElGamal.	2623,4 ms dan waktu dekripsi 4349,5 ms. Dari hasil pengujian pada beberapa dimensi citra yang berbeda, rata-rata MSE pada citra sebesar 213,9 dan PSNR sebesar 173,27 dB. Nilai ini menunjukkan kualitas citra cukup baik (diatas 40 dB).
8. Algoritma Elgamal dengan Pertukaran Kunci Diffie Hellman pada Aplikasi Keamanan Citra Sidik Jari Berbasis Android" (Elgamal	metode Diffie Hellman	Citra yang akan dienkripsi pada aplikasi ini adalah citra sidik jari berukuran 180 x 230 piksel. Proses enkripsi akan menghasilkan chiper citra berukuran 460 x 360 piksel atau berukuran 2 kali dari ukuran citra aslinya, selanjutnya di dekripsi untuk menghasilkan plain citra (citra asli). Dari 30 sampel yang di uji dengan	Hasil enkripsi dan dekripsi berhasil dilakukan dengan waktu rata-rata waktu enkripsi 102,482 detik dan rata-rata waktu dekripsi 34,151 detik.

dengan Pertukaran Kunci Diffie Hellman pada Aplikasi Keamanan Citra Sidik Jari Berbasis Android Nurul Yalisa & Arhami, 2018)		menggunakan kunci 16 bit.	
9. Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing (Fatima et al., 2022)	Algoritma Aes dan Rsa	Analisis Perbandingan Algoritma Aes dan Rsa untuk Keamanan Data di Cloud Computing	Enkripsi algoritma yang paling aman dan paling sedikit memakan waktu merupakan kebutuhan utama saat ini. Algoritme AES adalah metode dengan kompleksitas waktu yang lebih rendah dan karena perilakunya yang fleksibel dan terukur, algoritma ini mudah diimplementasikan,

			meninggalkan algoritma RSA dalam hal kebutuhan memori. Algoritma AES melindungi data dengan tingkat keamanannya yang tinggi dan dapat melakukan serangan balik terhadap berbagai serangan. Tidak seperti RSA, algoritma AES memerlukan lebih sedikit ruang penyimpanan sekaligus memberikan kinerja lebih tinggi tanpa batasan besar. Namun, dengan teknologi yang bergerak cepat, model hybrid mengambil alih algoritma keamanan biasa. Dengan demikian, model hybrid AES dan

			RSA akan meningkatkan keamanan cloud secara keseluruhan.
10. Secure Sensitive Data Sharing Using RSA and ElGamal Cryptographic Algorithms with Hash Functions (Adeniyi et al., 2022)	RSA dan ElGamal	Amankan Berbagi Data Sensitif Menggunakan RSA dan ElGamal Algoritma Kriptografi dengan Fungsi Hash	Kebutuhan akan keamanan informasi saat ini sudah tidak dapat diabaikan lagidi masyarakat kita karena meningkatnya kemunculan kejahatan dunia maya, pembajakan, penipuan, dan lainnya setiap hari kasus penipuan. Seperti yang telah diketahui bahwa masalah keamanan dan keselamatan adalah yang paling utama masalah mendesak yang dihadapi

			potensi data terdistribusi, pengiriman dan penerimaan data dianggap rentan terhadap serangan eksternal. Oleh karena itu, perlindungan data melalui enkripsi/dekripsi sangat penting. Penelitian ini mengkaji dua algoritma asimetris (RSA dan ElGamal) dikembangkan dalam meningkatkan layanan keamanan informasi. Selain itu, penerapan fungsi hash SHA-256 pada tanda tangan digital RSA dan Kriptosistem ElGamal diterapkan untuk membangun integritas informasi.

			Tekniknya memastikan perlindungan keamanan data sensitif pengguna dan pada saat yang sama menyediakan pengguna dengan kontrol penuh atas data mereka. Berbagai manfaat yang terkait dengan penelitian ini dankebenaran sistem yang diterapkan membuatnya cocok untuk data sensitif apa pun yang amansistem berbagi. Oleh karena itu, disarankan agar implementasi lebih lanjut seperti aman operasi pengiriman, penyimpanan, dan ekstraksi dari

			sistem berbagi data sensitif seharusnya diterapkan untuk perlindungan penuh dan maksimal terhadap data sensitif.

II.2. Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Pesan yang akan dienkripsi disebut sebagai plaintext (teks biasa). Disebut demikian karena informasi ini dengan mudah dapat dibaca dan dipahami oleh siapa saja. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah plaintext melibatkan penggunaan suatu bentuk kunci. Pesan plaintext yang telah dienkripsi (atau dikodekan) dikenal sebagai ciphertext (teks sandi). Di dalam kriptografi kita akan sering menemukan berbagai istilah atau terminology. Beberapa istilah yang harus diketahui yaitu :

1. Pesan, Plainteks, dan Cipherteks Pesan (message) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah (plaintext) atau teks jelas (cleartext).
2. Pengirim dan Penerima Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (sender) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (receiver) adalah entitas yang menerima pesan.
3. Enkripsi dan dekripsi Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (encryption) atau enciphering (standard nama menurut ISO 7498-2). Sedangkan proses mengembalikan cipherteks menjadi plainteks semula disebut dekripsi (decryption) atau deciphering (standard nama menurut ISO 7498-2).

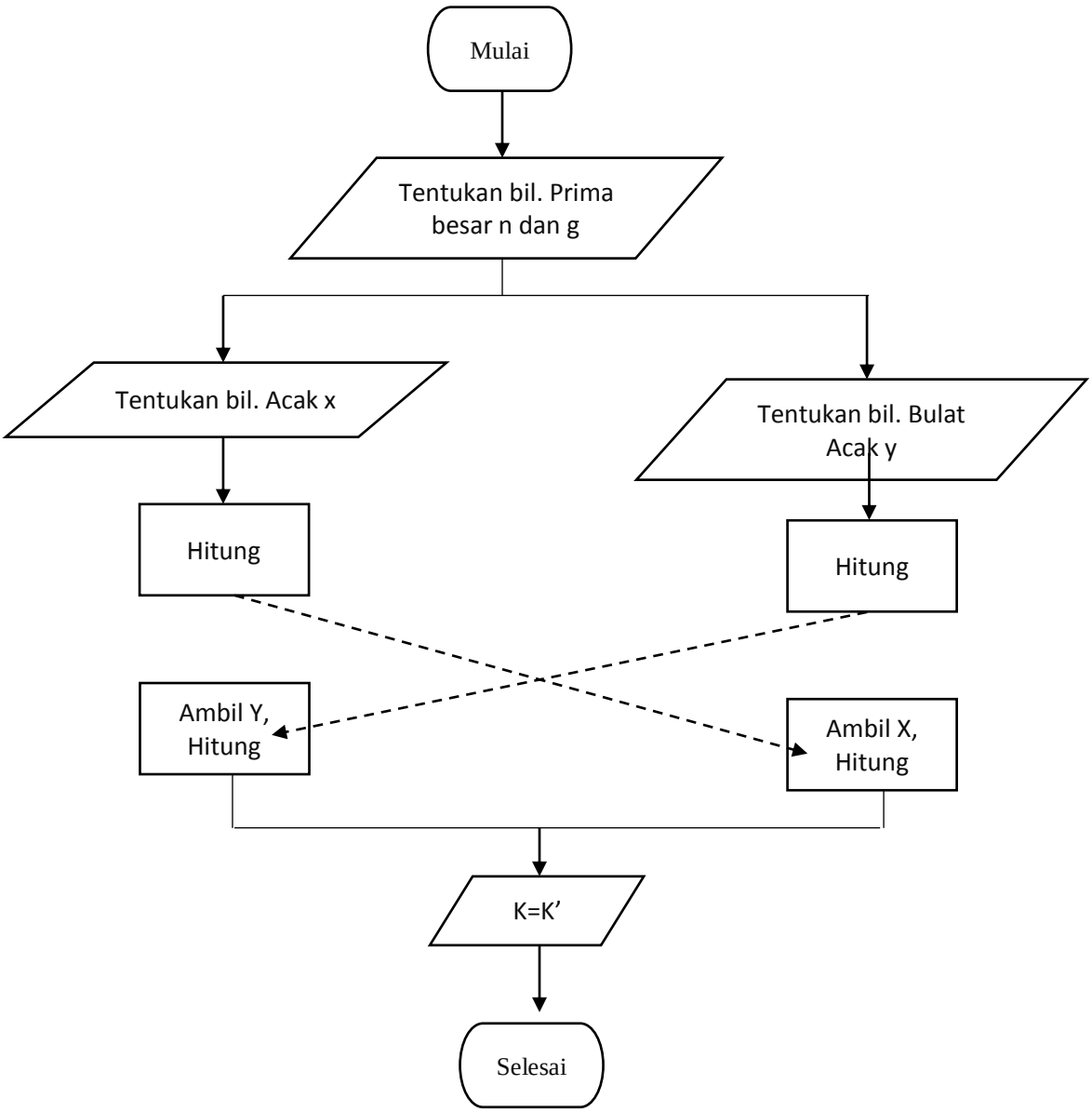
4. Cipher dan kunci Algoritma kriptogarfi disebut juga cipher, yaitu aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa cipher memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi cipherteks. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. (Nandar Pabokory et al., 2015)

II.3. Pembangkit Kunci Algoritma Diffie-Hellman

Algoritma pertukaran kunci Diffie Hellman (protokol Diffie-Hellman) berguna untuk mempertukarkan kunci rahasia pada komunikasi menggunakan kriptografi simetris. Kekuatan algoritma ini adalah pada sulitnya melakukan perhitungan logaritma diskrit. Langkah-langkahnya adalah sebagai berikut, 1. Misalkan Pengirim dan Penerima adalah pihak-pihak yang berkomunikasi. Mula-mula Pengirim dan Penerima menyepakati 2 buah bilangan yang besar (sebaiknya prima) P dan Q, sedemikian sehingga P < Q. Nilai P dan Q tidak perlu rahasia, bahakan Pengirim dan Penerima dapat membicarakannya melalui saluran yang tidak aman sekalipun. 2. Pengirim membangkitkan bilangan bulat acak x yang besar dan mengirim hasil perhitungan berikut kepada Penerima : X = Px mod Q. 3. Penerima membangkitkan bilangan bulat acak y yang besar dan mengirim hasil perhitungan, berikut kepada Pengirim: Y = P y mod Q. 4. Pengirim menghitung K = Y x mod Q. 5. Penerima menghitung K’ = X y mod Q. Jika perhitungan

dilakukan dengan benar maka K = K'. Dengan demikian Pengirim dan Penerima telah memiliki sebuah kunci yang sama tanpa diketahui pihak lain.(Pembangkit et al., 2022)

Adapun flowchart pembangkitan kunci dari algoritam Deffie-Hellman dapat dilhat seperti pada Gambar II.1.



Gambar II.1 *Flowchart* Algoritma Diffie Hellman

Kekuatan pertukaran kunci Diffie-Hellman adalah sulitnya melakukan perhitungan logaritma diskrit, juga ditambah dengan lamanya memfaktorkan bilangan besar menjadi faktor-faktor primanya.

Keamanan pada algoritma RSA terdapat pada sulitnya memfaktorkan bilangan besar menjadi faktor-faktor primanya. Dimana pada RSA pemfaktoran N merupakan hasil dari dua atau lebih bilangan prima sedemikian hingga $N=P\cdot Q$ Jika N berhasil difaktorkan menjadi P dan Q maka selanjutnya dapat dihitung D karena nilai E diumukan dan tidak bersifat rahasia. Penemu algoritma RSA ini menyarankan nilai P dan Q memiliki panjang lebih dari 100 digit karena dengan demikian hasil kali P dan Q akan memiliki panjang lebih dari 200 dikit. Sedikit fakta yang ada bahwa untuk mencari faktor prima dari bilangan yang kurang lebih 200 digit membutuhkan waktu komputasi selama 4 milyar tahun sedangkan untuk bilangan 500 digit membutuhkan waktu 1025 tahun dimana diasumsikan bahwa algoritma pemfaktoran yang digunakan adalah algoritma tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 milidetik.

Tingkat keamanan algoritma *Elgamal* ini terletak pada kesulitan dalam menghitung logaritma diskrit. Kelebihan dari algoritma ini adalah pembangkitan kunci yang menggunakan logaritma diskrit dan metode enkripsi dekripsi yang menggunakan proses komputasi yang besar sehingga hasil enkripsinya berukuran dua kali dari ukuran semula. Proses enkripsi pada *plaintext* yang sama diperoleh ciphertext yang berbeda-beda, sedangkan untuk proses dekripsi diperoleh *plaintext* yang sama *ElGamal* dapat digunakan untuk mengamankan data karena algoritma *ElGamal* dalam pembentukan salah satu kuncinya menggunakan

bilangan prima dan menitikberatkan kekuatan kuncinya pada pemecahan masalah logaritma diskrit sehingga keamanan kuncinya lebih terjamin. Meskipun memiliki kelemahan tersebut, namun algoritma *ElGamal* memiliki kelebihan yang jauh lebih banyak, sehingga dalam paper ini menggunakan algoritma *ElGamal* dalam meningkatkan keamanan data. (Karima & Saputro, 2018). Kelemahan algoritma ElGamal yaitu membutuhkan resource yang besar dan processor yang mampu melakukan perhitungan besar.

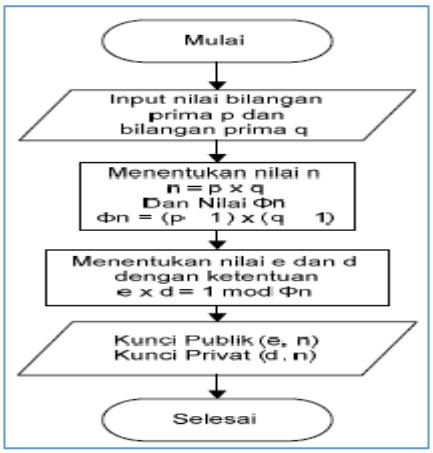
II.4. Algoritma Rivest-Shamir-Adleman (RSA)

Algoritma RSA adalah salah satu teknik kriptografi dimana kunci untuk melakukan enkripsi berbeda dengan kunci untuk melakukan dekripsi. (*Volume-81-Artikel-9*, n.d.)Algoritma RSA adalah sebuah algoritma yang memiliki kunci yang cukup panjang. Kunci RSA pada umumnya sepanjang 1024—2048 bit. Beberapa pakar meyakini bahwa kunci 1024-bit ada kemungkinan dipecahkan pada waktu dekat (hal ini masih dalam perdebatan), tetapi tidak ada seorangpun yang berpendapat kunci 2048- bit akan pecah pada masa depan yang terprediksi. Jika (sepanjang 256-bit atau lebih pendek, maka kunci RSA akan dapat ditemukan dalam beberapa jam hanya dengan menggunakan PC, dengan menggunakan perangkat lunak yang tersedia. Jika (sepanjang 512-bit atau lebih pendek, (akan dapat difaktorisasi dalam hitungan ratusan jam seperti pada tahun 1999 dengan menggunakan ratusan komputer. Secara teori, perangkat keras bernama TWIRL dan penjelasan dari Shamir dan Tromer pada tahun 2003 mengundang berbagai pertanyaan akan keamanan dari kunci 1024-bit. Saat ini disarankan bahwa

setidaknya sepanjang 2048-bit.(*BAB I PENDAHULUAN*, n.d.)

II.4.1 Proses Pembangkitan Kunci Algoritma RSA

Pembangkitan kunci dari algoritma RSA dapat dilihat seperti gambar II.2 berikut.



Gambar II.2 *Flowchart* Proses Pembangkitan Kunci Algoritma RSA

Adapun penjelasan dari Gambar II.4.1 adalah sebagai berikut :

1. Pilih dua bilangan prima p dan q dimana $p \neq q$
2. Hitung $n = p \times q$ dan $\phi(n) = (p-1)(q-1)$
3. Menentukan nilai e secara acak, $1 < e < \phi$ dan prima relatif terhadap ϕ.

   Prima relatif maksudnya jika faktor persekutuan terbesar (*FPB*) dari *e* dan ϕ sama dengan satu (*FPB(e,* ϕ) = 1). Dan hitung bilangan bulat positif *d* yang unik, $1 < d < \phi$ dimana d = e-1 mod ϕ sehingga e.d = 1 (mod ϕ) atau e.d = k(ϕ) + 1 untuk suatu bilangan bulat k.
4. Didapat kunci publik berupa pasangan (n, e) dan kunci privat (n, d) dimana e adalah *public exponent,* d adalah *private exponent* dan n adalah modulus.
5. Nilai e dan n dipublikasikan tapi nilai d, p, q harus tetap dirahasiakan.

II.4.2 Proses Enkripsi Algoritma RSA

Proses enkripsi dengan algoritma RSA dilakukan dengan menghitung *exponen plaintext* dalam operasi modulo n (modulo = sisa pembagian) untuk setiap blok pesan atau data sehingga dapat menghasilkan *ciphertext*. Eksponen yang digunakan adalah public exponent e. Operasi ini bisa dituliskan dengan persamaan berikut :

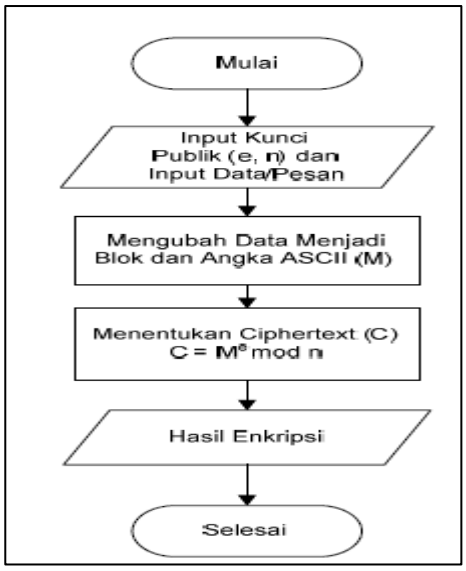
$C = M^e \bmod n$

C = ciphertext

M = plaintext (pesan)

e = public exponent

n = modulus



Gambar II. 3 *Flowchart* Proses Enkripsi Algoritma RSA

Penjelasan dari Gambar II.3 mengenai langkah-langkah dalam proses enkripsi algoritma RSA adalah sebagai berikut:

1. Menginputkan kunci publik yang berupa pasangan *(*e, n) beserta pesan atau data yang akan dienkripsi.
2. Representation pesan atau data menjadi blok-blok dan diubah menjadi bilangan bulat positif M.
3. Hitung C = Me mod n.
4. Dan dihasilkan *ciphertext* (C) yang merupakan hasil dari enkripsi.

II.4.3 Proses Dekripsi Algoritma RSA

Sedangkan pada proses deskripsi, yang dilakukan hampir sama dengan enkripsi tapi eksponen yang digunakan adalah *private exponent d* untuk mengembalikan pesan seperti semula. Operasi ini bisa dituliskan dengan persamaan berikut :

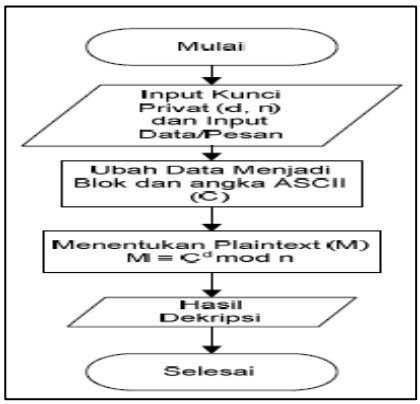
$M = C^d \bmod n$

C = ciphertext

d = private exponent

n = modulus

M = plaintext (pesan)



Gambar II.4 *Flowchart* Proses Dekripsi Algoritma RSA

Penjelasan dari gambar II.4 mengenai langkah-langkah dalam proses dekripsi algoritma RSA adalah sebagai berikut:

1. Menginputkan kunci privat yang berupa pasangan *(*d, n) beserta pesan atau data yang akan didekripsi.
2. Representasikan data menjadi blok-blok dan diubah menjadi bilangan bulat positif C.
3. Hitung $M = C^e \bmod n$.
4. Dan dihasilkan *plaintext* (M) yang merupakan hasil dari dekripsi dan merupakan pesan atau data yang sebenarnya.

II.5. Algoritma ElGamal

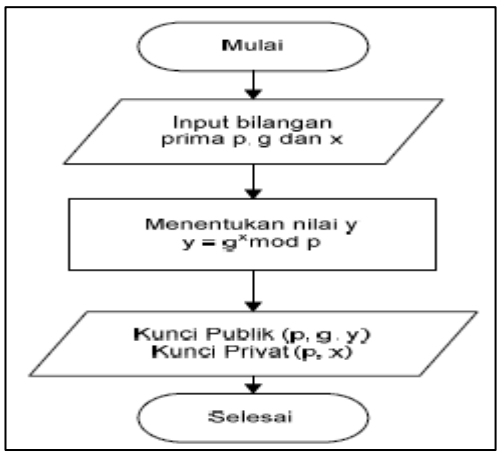
Algoritma Elgamal diciptakan oleh ilmuan mesir, yaitu Taher El Gamal pada tahun 1984. Algoritma ElGamal merupakan algoritma dalam kriptografi yang termasuk dalam kategori algoritma asimetris. Keamanan algoritma ElGamal terletak pada kesulitan perhitungan logaritma diskret pada bilangan modulo prima yang besar sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sukar.

Algoritma ElGamal mempunyai kunci publik yang berupa tiga pasang bilangan dan kunci rahasia berupa satu bilangan. Algoritma ini mempunyai kerugian pada cipherteksnya yang mempunyai panjang dua kali lipat dari plainteksnya. Akan tetapi, algoritma ini mempunyai kelebihan pada enkripsi. Untuk plainteks yang sama, algoritma ini memberikan cipherteks yang berbeda (dengan kepastian yang dekat) setiap kali plainteks di enkripsi. Algoritma

ElGamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Sama halnya dengan algoritma RSA, algoritma ElGamal juga merupakan cipher blok, yaitu melakukan proses enkripsi pada blok-blok plainteks dan menghasilkan blok-blok cipherteks yang kemudian dilakukan proses dekripsi dan hasilnya digabungkan.

II.5.1 *Flowchart* Proses Pembangkitan Kunci ElGamal

Adapun *flowchart* proses pembangkitan kunci Algoritma ElGamal adalah seperti pada Gambar II.5

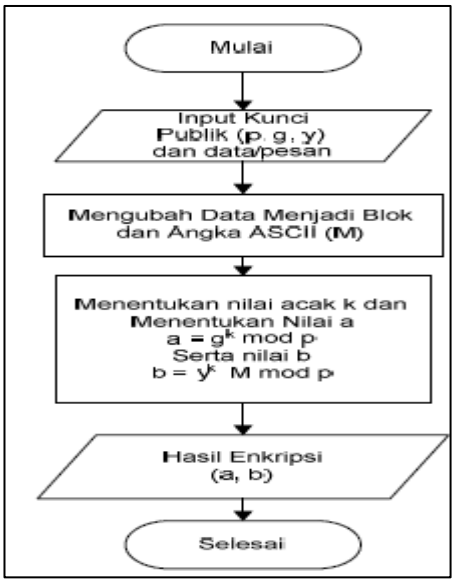


Gambar II.5 *Flowchart* Proses Pembangkitan Kunci El Gamal

Pembentukan kunci terdiri atas pembangkitan kunci publik dan kunci privat. Dalam gambar II.5 pada proses ini dibutuhkan sebuah bilangan prima *p* yang digunakan untuk membentuk dua bilangan acak *g* dan *x* dimana $g < p$ dan $1 \leq x \leq p - 2$. Kunci publik algoritma ElGamal terdiri atas pasangan 3 bilangan *{p, g,y}* dimana $y = g^x \bmod p$ sedangkan untuk kunci privatnya terdiri atas dua bilangan {p, x}.

II.5.2 Proses Enkripsi Algoritma ElGamal

Adapun flowchart proses Enkripsi Algoritma ElGamal adalah seperti pada Gambar II.6



Gambar II.6 *Flowchart* Enkripsi Algoritma El Gamal

Pada gambar II.6 diatas Proses enkripsi membutuhkan kunci publik (*p, g, y*) dan sebuah bilangan integer acak *k* dengan ketentuan $1 \leq k \leq p - 1$ yang dijaga kerahasiaannya oleh yang mengenkripsi pesan atau teks. Untuk setiap karakter dalam pesan dienkripsi menggunakan bilangan *k* yang berbeda-beda. Satu karakter yang direpresentasikan dengan menggunakan bilangan bulat ASCII akan menghasilkan kode dalam bentuk blok yang terdiri atas dua nilai *(a, b)*.

Langkah proses enkripsi:

a. Ambil sebuah karakter dalam pesan yang akan dienkripsi dan transformasi karakter tersebut kedalam kode ASCII sehingga diperoleh bilangan bulat *M*.

Adapun daftar kode ASCII masing-masing karakter dapat dilihat pada gambar II.7

dan gambar II.8 berikut:

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	⌂

Gambar II.7 Kode-kode ASCII Bagian 1

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	⌂

Gambar II.8 Kode-kode ASCII Bagian 2

b. Hitung nilai *a* dengan rumus

$$a = g^k \bmod p$$

dan hitung nilai *b* dengan rumus $b = y^k M \bmod p$

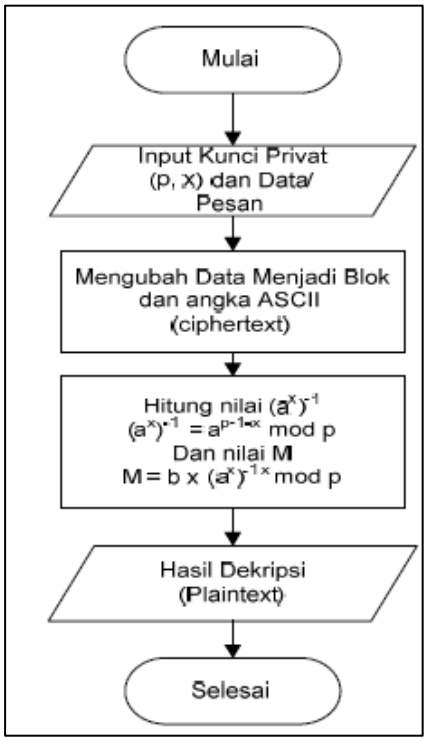
$$b = y^k M \bmod p$$

c. Diperoleh cipherteks untuk karakter *M* tersebut dalam blok *(a, b).*

d. Lakukan proses diatas untuk seluruh karakter dalam pesan termasuk karakter spasi.

II.5.3 Proses Dekripsi Algoritma ElGamal

Adapun *flowchart* proses dekripsi Algoritma ElGamal adalah seperti pada Gambar II.9



Gambar II.9 *Flowchart* Dekripsi Algoritma *El Gamal*

Pada Gambar II.9 diatas, dekripsi dari cipherteks menggunakan kunci privat *(p, x)*

yang disimpan kerahasiaannya.

Langkah proses dekripsi :

a. Ambil sebuah blok cipherteks dari pesan yang telah dienkripsikan.
b. Dengan menggunakan *x* dan hitung nilai plainteks dengan menggunakan rumus

$$(a^x)^{-1} = a^{p-1-x} \, mod \, p$$

Kemudian hitung dengan rumus

M = $b \, x \, (a^x) \, mod \, p$