**RANCANG BANGUN APLIKASI PENGAMANAN DATA PADA EMAIL MENGGUNAKAN ALGORITMA ELGAMAL BERBASIS WEB (STUDI KASUS : JURUSAN TEKNOLOGI INFORMASI FAKULTAS TEKNIK UNIVERSITAS TADULAKO)**



**PROPOSAL SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar

Sarjana Komputer pada Program Studi Teknik Informatika Jurusan Teknologi Informasi Fakultas Teknik Universitas Tadulako

Disusun oleh:

**RAODATUL JANNAH**

**F 551 13 036**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS TADULAKO**

**PALU**

**2018**

###### BAB I

**PENDAHULUAN**

* 1. **Latar Belakang**

Semakin pesatnya perkembangan teknologi informasi (TI) tidak akan pernah lepas dari permasalahan keamanan komputer (*Computer Security*). Keamanan komputer sebagai isu yang tidak akan pernah habis dibicarakan para pelaku bidang TI selalu menuntut adanya *update* setiap saat dan berkala. Seperti pada kasus pengiriman *file*, pengiriman *file* via internet, *email* maupun via *offline* antar sesama perusahaan merupakan cara yang paling praktis di era teknologi informasi dewasa ini.

*Email* sudah digunakan orang sejak awal terbentuknya internet dan merupakan salah satu fasilitas yang ada pada saat itu. Tidak jarang orang menyimpan berbagai data penting pada *email* tersebut, sepeti informasi akun-akun, nomor rekening relasi, dan masih banyak lainnya. Hal ini dikarenakan orang-orang takut lupa mengenai informasi penting tersebut dan dipilihlah *email* sebagai tempat penyimpanannya. Namun, akibat dari banyaknya orang yang menggunakan *email* tersebut sebagai alat penyimpan informasi, tidak sedikit orang yang berbuat nakal untuk mencari tahu mengenai informasi tersebut. Salah satu caranya adalah dengan melakukan *hack* ke *email* sang korban.

Seiring dengan perkembangan teknologi informasi, berkembang pula teknologi bidang lainnya dalam dunia teknologi informasi, seperti teknologi kriptografi (pengamanan informasi) dan algoritma pengacakan data. Pada teknologi kriptografi banyak algoritma pengacakan data yang dikembangkan, baik algoritma *asimetris* maupun *simetris*. Permasalahan tersebut dapat diatasi dengan proses enkripsi. Salah satu enkripsi yang cukup dikenal adalah dengan metode enkripsi *ElGamal*.

1

*ElGamal* ini akan memberikan *public key* serta *private key* yang digunakan dalam proses enkripsi dan dekripsi. Dalam proses pembentukan kunci publik dan rahasia, akan dibutuhkan suatu bilangan prima yang bernilai besar agar menjadi aman. Aplikasi enkripsi *email* ini akan dibangun menggunakan sistem berbasis web.

Oleh karena itu pengamanan yang kuat sangat dibutuhkan agar data-data tidak jatuh ke pihak yang tidak bertanggung jawab dan yang tidak berhak atas data tersebut. Berdasarkan hal tersebut penulis bermaksud akan mengkaji suatu penelitian skripsi yang berjudul “*Rancang Bangun Aplikasi Pengamanan Data Pada Email Menggunakan Algoritma ElGamal Berbasis Web”.*

* 1. **Rumusan Masalah**

Berdasarkan latar belakang diatas maka dapat dirumuskan permasalahan pada penelitian ini yaitu bagaimana merancang aplikasi pengaman data pada *email* dengan mengimplementasikan algoritma *ElGamal* untuk teknik enkripsi dan dekripsi berbasis web.

* 1. **Batasan Masalah**

Adapun batasan atau ruang lingkup dari penelitian ini antara lain:

1. Penulisan ini difokuskan pada segi enkripsi dan deskripsi data.
2. Data yang dienkripsi dan dideskripsi pada pesan (*email*) adalah data berupa teks dengan ekstensi (.docx, .xlsx. pdf, .txt).
3. Sistem hanya akan melakukan pengamanan *email* pada sesama pengguna akun gmail.
4. Pembatasan karakter pada format .docx, .pdf, .txt maksimal berjumlah 10.000 karakter.
   1. **Tujuan Penelitian**

Tujuan yang ingin dicapai pada penelitian ini adalah merancang dan membangun aplikasi pengamanan data *email* dengan menerapkan algoritma *ElGamal* berbasis web sehingga tingkat keamanan data pada *email* lebih terjaga.

* 1. **Manfaat Penelitian**

Adapun manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Manfaat Akademis
2. Bagi penulis dapat memperluas wawasan dalam mengaplikasikan ilmu pengetahuan di bidang informatika yang telah diperoleh di bangku perkuliahan.
3. Bagi peneliti lain dapat dijadikan sebagai bahan referensi dan acuan terhadap pengembangan ataupun pembuatan penelitian yang memiliki topik yang sama dengan penelitian ini.
4. Manfaat Praktis

Hasil penelitian ini dapat membuat aplikasi pengaman data *email* dengan menerapkan algoritma *ElGamal* untuk mempermudah pengguna mengamankan data, agar suatu data yang diamankan tidak bisa diakses oleh pihak-pihak yang tidak berwenang.

* 1. **Sistematika Penulisan**

Untuk memudahkan memahami permasalahan yang akan dibahas maka tugas akhir ini disusun dengan sistematika sebagai berikut:

BAB I: PENDAHULUAN

Bab ini terdiri dari latar belakang, rumusan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II: TINJAUAN PUSTAKA DAN LANDASAN TEORI

Bab ini menjelaskan tentang perbandingan antara penelitian ini dengan penelitian lain yang pernah ada sebelumnya serta teori-teori yang digunakan dalam menyusun proposal ini.

BAB III: METODE PENELITIAN

Bab ini menjelaskan tentang bahan dan alat yang akan digunakan dalam penelitian, tahapan penelitian, dan hipotesis yang dibuat berdasarkan teori dasar serta rumusan masalah yang ada.

BAB IV: HASIL DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dari penelitian dan perancangan yang telah dilakukan selama beberapa bulan.

BAB V: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diambil berdasarkan analisa yang telah dilakukan serta saran-saran yang bermanfaat bagi penelitian-penelitian yang akan dilakukan di kemudian hari.

DAFTAR PUSTAKA

Pada bagian ini berisi tentang referensi yang dijadikan rujukan dalam melakukan penelitian.

**BAB II**

**TINJAUAN PUSTAKA DAN LANDASAN TEORI**

1. **Tinjauan Pustaka**

Penulis senantiasa mempelajari hasil penelitian terdahulu, sebagai acuan guna membangun wawasan berpikir dalam penelitian ini. Beberapa hasil penelitian terdahulu yang dijadikan referensi adalah sebagai berikut :

1. Penelitian yang di lakukan oleh Karima et al (2017) dengan judul “Pemfaktoran Bilangan Prima pada Algoritma *ElGamal* untuk Keamanan Dokumen PDF”. Dalam penelitian ini dibahas tentang penggunaan kriptografi algoritma *ElGamal* dalam pengamanan dokumen PDF. Pemanfaatan file dokumen txt dan doc ternyata tidak menjamin dokumen tersebut aman. Terdapat pihak yang melakukan tindakan plagiat terhadap file txt dan doc, sehingga muncul inisiatif untuk mengamankan file tersebut dengan melakukan konversi file ke PDF untuk alasan keamanan. Keunggulan file PDF yang tidak mudah diplagiat serta sudah dilengkapi dengan password ternyata juga dinyatakan sudah tidak aman lagi, karena masih mudah dimodifikasi oleh pihak lain. Pada penelitian ini faktor bilangan prima dan bilangan acak pada kriptografi dengan algoritma *ElGamal* digunakan untuk mengenkripsi dokumen *plaintext* PDF menjadi dokumen *ciphertext* yang tidak mudah dimodifikas. Persamaan penelitian ini dengan penelitian yang akan penulis lakukan yaitu kedua-duanya menggunakan algoritma *ElGamal*  untuk pengamanan data nya. Sementara perbedaan penelitian ini dengan yang akan di teliti yaitu penelitian ini menggunakan kriptografi algoritma *ElGamal* untuk pengamanan data dokumen berformat PDF, sedangkan penelitian yang akan penulis lakukan yaitu menggunakan kriptografi algoritma *ElGamal* untuk pengamanan data pada *email* berbasis web.

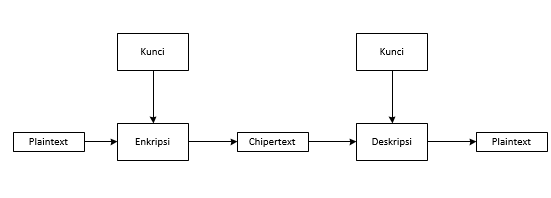
6

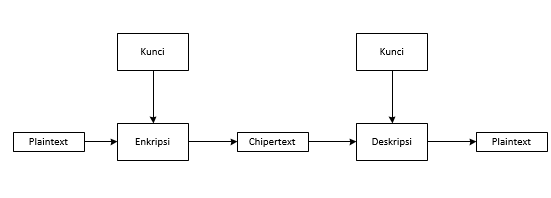
1. Penelitian yang di lakukan oleh Nugraha dan Mazharuddin (2013) dengan judul Penyembunyian pesan rahasia yang terenkripsi menggunakan algoritma RSA pada media kompresi. Jurnal Teknik POMITS Vol2, No.1, (2013), IISN: 2337-3539 (2301-9271). Saat ini banyak sekali penerimaan dan pengiriman pesan yang beredar tetapi bisa di lacak oleh orang yang tidak berkepentingan, Untuk itu diperlukan suatu cara untuk mengamankan pesan rahasia tadi agar tidak diketahui oleh orang yang tidak berkepentingan. Penyembunyian pesan rahasia yang berupa file dalam arsip ZIP dapat menjadi salah satu solusi untuk keamanan data yang bersifat rahasia jika data tersebut ingin dikirimkan. Arsip ZIP merupakan kumpulan dari beberapa file yang terkompresi dimana ukuran dari file-file tersebut beragam. Biasanya orang tidak memperhatikan ukuran file dari arsip ZIP karena ukuran dari file-file di dalam arsip ZIP tersebut terkompresi. Dari hasil uji coba yang dilakukan, file yang berisi pesan rahasia berhasil disembunyikan pada arsip ZIP serta tidak akan terbaca pada aplikasi pembaca arsip ZIP. Persamaan Penelitian ini dengan penelitian yang akan penulis lakukan yaitu kedua-duanya merupakan sistem untuk penyembunyian pesan rahasia yang terenkripsi. Sedangkan perbedaannya yaitu penelitian ini menggunakan algoritma *RSA (Rivest Shamir Adleman)* dan penulis menggunakan algoritma *Asimetris* *ElGamal.*
2. Penelitian yang di lakukan oleh Al-Anshor dan Aribowo (2014) dengan judul “Implementasi Algoritma Kriptografi Kunci Publik *ElGamal* Untuk Proses Enkripsi Dan Dekripsi Guna Pengamanan File Data”. Dalam penelitian ini akan diimplemtasikan dalam sebuah program aplikasi menggunakan bahasa pemrograman *Visual Basic* yang dapatmemberikan kemudahan bagi setiap orang yang akan mengamankan file-file penting. Salah satu algoritma yang digunakan untuk enkripsi dan dekripsi pada tugas akhir ini adalah algoritma *ElGamal*. Hasil pengujian sistem menunjukkan bahwa pengamanan file menggunakan Kriptografi dengan metode *ElGamal* ini layak dan dapat dipergunakan untuk mengamankan file data yang akan kita amankan. Persamaan dari penelitian ini dengan yang akan penulis lakukan yaitu kedua-duanya meneliti sistem pengamanan data nya menggunakan algoritma *ElGamal*. Sedangkan perbedaannya yaitu penelitian ini di aplikasikan pada perangkat dekstop dan pada penelitian yang akan dilakukan yaitu di aplikasikan berbasis web.
3. **Landasan Teori**

**2.2.1. 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, *Plaintext*, dan *Ciphertext* 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 (*sende*r) adalah entitas yang mengirim pesan kepada entitas lainnya.

Penerima (*receiver*) adalah entitas yang menerima pesan. 3. Enkripsi dan dekripsi Proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*) atau *enciphering* (standard nama menurut ISO 7498-2). Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* 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 *plaintext* dan himpunan yang berisi *ciphertext*.

Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara dua himpunan tersebut. Misalkan P menyatakan *plaintext* dan C menyatakan *ciphertext*, maka : E(P) = C merupakan fungsi enkripsi, E memetakan P ke C D(C) = P merupakan fungsi dekripsi, D memetakan C ke P Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka persamaan D(E(P)) = P harus benar. Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi. Kunci biasanya berupa string atau deretan bilangan. Dengan menggunakan kunci K, maka fungsi enkripsi dan dekripsi dapat ditulis sebagai skema diperlihatkan pada Gambar 2.1.



Gambar 2.1

(Sumber : Munir *dalam* Pabokory et al , 2015)

Kriptografi bertujuan untuk memberikan layanan pada aspek-aspek keamanan antara lain:

1. Kerahasiaan (*confidentiality*)*,* yaitu menjaga supaya pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.

2. Integritas data (*data integrity*)*,* yaitu memberikan jaminan bahwa untuk tiap bagian pesan tidak akan mengalami perubahan dari saat data dibuat/dikirim oleh pengirim sampai dengan saat data tersebut dibuka oleh penerima data.

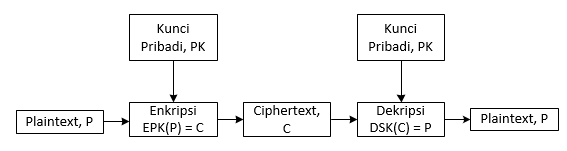
3. Otentikasi (*authentication*)*,* yaitu berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan.

4. Nirpenyangkalan (*non repudiation*)*,* yaitu memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang tertentu sehingga apabila ada seseorang yang mencoba mengakui memiliki dokumen tersebut, dapat dibuktikan kebenarannya dari pengakuan orang tersebut (Munir *dalam* Pabokory et al , 2015).

Berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan atas dua golongan, yaitu:

**2.2.1.1. Kriptografi Simetrik (*Symetric Cryptography*)**

*Symmetric cryptosystem* atau kriptografi simteris atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Algoritma kriptografi simetris dibagi menjadi 2 kategori yaitu algoritma aliran (*Stream Ciphers*) dan algoritma blok (*Block Ciphers*). Pada algoritma aliran, proses penyandiannya berorientasi pada satu bit atau satu byte data. Sedang pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau *byte* data (per blok). Ini adalah jenis kriptografi yang paling umum dipergunakan. Kunci untuk membuat pesan yang disandikan sama dengan kunci untuk membuka pesan yang disandikan itu. Jadi pembuat pesan dan penerimanya harus memiliki kunci yang sama persis. Contoh algoritma kunci simetris yang terkenal adalah DES (Data EncryptionStandard) dan RC-4. Skema algoritma ditunjukkan pada gambar 2.2.



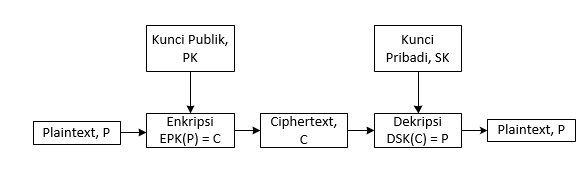
Gambar 2.2 Kriptografi Simetris

(Sumber : Kurniawan *dalam* Basri, 2014)

**2.2.1.2. Kriptografi Asimetris (*Asymetric Cryptography*)**

Pada pertengahan tahun 70-an Whitfield Diffie dan Martin Hellman menemukan teknik enkripsi asimetris yang merevolusi dunia kriptografi. Kunci asimetris adalah pasangan kunci-kunci kriptografi yang salah satunya dipergunakan untuk proses enkripsi dan yang satu lagi untuk dekripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsikan suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia tertentu dalam hal ini kunci *private* untuk melakukan pembongkaran terhadap sandi yang dikirim untuknya. Sebagai contoh jika Anto mengirim pesan untuk Badu, Anto dapat merasa yakin bahwa pesan tersebut hanya dapat dibaca oleh Badu, karena hanya Badu yang bisa melakukan dekripsi dengan

kunci privatnya. Tentunya Anto harus memiliki kunci publik Badu untuk melakukan enkripsi. Anto bisa mendapatkannya dari Badu, ataupun dari pihak ketiga seperti Tari. Skema algoritma kriptografi asimetris terlihat pada gambar 2.3.



Gambar 2.3 Kriptografi Asimetris

(Sumber : Kurniawan *dalam* Basri, 2014)

Teknik enkripsi Asimetris ini jauh lebih lambat ketimbang enkripsi dengan kunci simetris. Oleh karena itu, biasanya bukanlah pesan itu sendiri yang disandikan dengan kunci asimetris, namun hanya kunci simetrislah yang disandikan dengan kunci asimetris. Sedangkan pesannya dikirim setelah disandikan dengan kunci simetris tadi. Contoh algoritma terkenal yang menggunakan kunci Asimetris adalah RSA (merupakan singkatan penemunya yakni *Rivest, Shamir dan Adleman*).

**2.2.2. Algoritma *ElGamal***

Kriptografi *ElGamal* merupakan bagian dari kriptografi asimetris. Pertamakali dipublikasikan oleh Taher *ElGamal* pada tahun1985. Kriptografi *ElGamal* pada mulanya digunakan untuk *digital signature*,namun kemudian dimodifikasi sehingga juga biasa digunakan untuk enkripsi dan deskripsi. Algoritma *ElGamal* terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi.

Algoritma ini merupakan cipher blok, yaitu melakukan proses enkripsi pada blok-blok *plaintext* dan menghasilkan blok-blok *ciphertext* yang kemudian dilakukan proses dekripsi dan hasilnya digabungkan. Keamanan algoritma *ElGamal* terletak pada kesulitan penghitungan logaritma diskret pada bilangan modulo prima yang besar sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sukar. Algoritma ini mempunyai kerugian pada *ciphertext*nya yang mempunyai panjang dua kali lipat dari *plaintext*nya. Akan tetapi, algoritma ini mempunyai kelebihan pada enkripsi. Untuk *plaintext* yang sama, algoritma ini memberikan *ciphertext* yang berbeda (dengan kepastian yang dekat) setiap kali *plaintext* di enkripsi.

**2.2.2.1. Proses Pembentukan Kunci**

Proses pembentukan kunci ini terdiri dari pembentukan kunci publik dan juga pembentukan kunci rahasia. Kunci publik ini akan digunakan pada proses pengenkripsian sedangkan kunci rahasia digunakan pada saat proses pendeskripsian. Pada proses ini dibutuhkan sebuah bilangan prima *p*, bilangan acak *g* dan *x*, dengan syarat *g < p* dan *1 ≤ x ≤ p-2*. Kunci publik algoritma *ElGamal* terdiri atas pasangan 3 bilangan (*y*, *g*, *p*) di mana hubungan dari ketiga bilangan tersebut dapat dilihat pada persamaan (3).

*y = gx mod p* (3)

Sedangkan kunci rahasianya adalah bilangan (*x, p*)tersebut karena pada algoritma *ElGamal* menggunakan bilangan bulat dalam proses perhitungannya, maka pesan harus dikonversi ke dalam suatu bilangan bulat. Untuk mengubah pesan menjadi bilangan bulat, digunakan kode ASCII (*American Standard for Information Interchange*). Kode ASCII merupakan representasi numerik dari karakter-karakter yang digunakan pada komputer, serta mempunyai nilai minimal 0 dan maksimal 255. Oleh karena itu, berdasarkan sistem kriptografi *ElGamal* di atas maka harus digunakan bilangan prima yang lebih besar dari 255. Kode ASCII berkorespondensi 1-1 dengan karakter pesan.

**2.2.2.2. Proses Enkripsi**

Proses enkripsi menggunakan kunci publik (*y*, *g*, *p*) dan sebuah bilangan acak *k* yang dalam hal ini *1 ≤ k ≤ p-2*. Untuk setiap karakter dalam pesan dienkripsi dengan 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*). Berikut ini langkah-langkah untuk proses enkripsi:

1. Ambil sebuah karakter dalam pesan yang akan dienkripsi dan transformasi karakter tersebut ke dalam kode ASCII sehingga diperoleh bilangan bulat *m*.

1. Hitung nilai *a* dan *b* dengan persamaan persamaan (4) dan (5).

*a* = *g*k mod *p* (4)

*b* = *y*k *m* mod *p* (5)

1. Diperoleh *ciphertext* untuk karakter *m* tersebut dalam blok (*a, b*)
2. Lakukan proses di atas untuk seluruh karakter dalam pesan termasuk karakter spasi.

Salah satu kelebihan algoritma *ElGamal* adalah bahwa suatu *plaintext* yang sama akan dienkripsi menjadi *ciphertext* yang berbeda-beda. Akan tetapi, walaupun *ciphertext* yang diperoleh berbeda-beda, tetapi pada proses dekripsi akan diperoleh *plaintext* yang sama.

**2.2.2.3. Proses Dekripsi**

Dekripsi dari *ciphertext* ke *plaintext* menggunakan kunci rahasia *x* dan *p* yang disimpan kerahasiaanya oleh penerima pesan. Diberikan (*y*, *g*, *p*) sebagai kunci publik dan *x* sebagai kunci rahasia pada algoritma *ElGamal*. Jika diberikan *ciphertext* (*a, b*), maka didapatkan hubungan seperti terlihat pada persamaan (6).

*m* = *b*.*c* mod *p* (6)

dengan *m* adalah *plaintext*. Di mana nilai dari variabel *c* dicari dengan menggunakan persamaan (7).

*c* = *a p-1-x* mod *p* (7)

Berikut ini langkah-langkah untuk proses dekripsi:

1. Ambil sebuah blok *ciphertext* dari pesan yang telah dienkripsikan pengirim.
2. Dengan menggunakan *x* yang dirahasiakan oleh penerima, hitung nilai *plaintext* dengan menggunakan persamaan (6) dan persamaan (7).

Berikut ini contoh penggunaan algoritma *ElGamal* :

Misalkan A ingin berkomunikasi dengan B, A ingin mengirimkan pesan kepada B tanpa diketahui isi dari pesan tersebut. A menggunakan algoritma *ElGamal* untuk menyamarkan pesan tersebut, pesan tersebut adalah “P”. Hal pertama yang dilakukan adalah membuat kunci untuk mengenkripsi pesan tersebut. Pembuatan kunci dilakukan oleh si B, langkah-langkah yang dilakukan si B adalah:

1. Si B menentukan bilangan prima, misalkan nilai *p* adalah 503, nilai bilangan prima ini dipilih secara acak.
2. Kemudian si B menentukan bilangan acak *g* dan *x*, misalkan nilai *g* adalah 125 dan *x* adalah 50.
3. Si B kemudian mengitung nilai *y*, *y = gx mod p* = 12550 mod 503 = 173
4. Si B mendapatkan kunci publik (503, 125, 173) dan kunci rahasia (50, 503). Kemudian kunci publik tersebut dikirimkan ke A, sedangkan kunci rahasia hanya diketahui oleh si B.

Pengirim atau si A menerima kunci publik yang diberikan oleh si B. Dengan kunci tersebut si A akan mengenkripsi pesan “P” untuk dikirimkan kepada si B. Langkah-langkah yang dilakukan si A adalah :

1. Pertama yang dilakukan si A adalah mengubah pesan tersebut ke dalam bentuk ASCII. Dalam tabel ASCII “P” = 80.
2. Setelah itu si A menetukan nilai *k*, misalkan nilai *k* adalah 87.
3. Kemudian si A akan melakukan enkripsi terhadap pesan “P”, enkripsi dilakukan dengan menghitung :

*a = gk mod p* = 12587 mod 503 = 120 dan *b = yk m mod p* = 17287 80 mod 503 = 305.

Pesan yang telah terenkripsi tersebut akan dikirim oleh si A kepada si B, jadi pesan yang akan dikirim oleh si A adalah (120, 305). Untuk dapat membaca pesan tersebut si B harus melakukan dekripsi terhadap pesan tersebut, langkah-langkah yang harus dilakukan si B adalah :

1. Pertama penerima harus menghitung terlebih dahulu nilai variabel *c* sebelum mendekripsi *c* = *ap-1-x mod p* = 120503-1-50 *mod* 503 = 25.
2. Setelah didapat nilai variabel *c* barulah kemudian si B menghitung nilai dari *m*, *m = b.c mod p* = *b . c mod p* = 305 . 25 mod 503 = 80.
3. Barulah kemudian si B menkonversi dari kode ASCII ke bentuk karakter dimana 80 dalam kode ASCII adalah “P”.

Setelah si B berhasil melakukan proses dekrispsi barulah dapat terlihat atau terbaca pesan yang dikirim si A, pesan yang dikirim si A adalah “P”. Seperti itulah proses algoritma *ElGamal* dalam mengenkripsi dan mendekripsi sebuah pesan.

Contoh Kasus :

*Plaintext* : Hati

Dik :

P = 2143

g = 2

x = 1751

k = 1520

y = gx mod p a = gk md p (ax)-1 = ap-1-x mod p

= 21751 mod 2143 = 21520 mod 2143 = 203391 mod 2143

= 349 = 203 = 349

1. Ubah pesan *plaintext* ke dalam bentuk code ASCII :

H = m1 = 72

a = m2 = 97

t = m3 = 116

i = m4 = 105

1. Proses Enkripsi :

C =

C1 = = 516

C2 = = 338

C3 = = 117

C4 = = 1824

Hasil *Ciphertext* : **516, 338, 117, 1824**

1. Proses Deskripsi :

M =

M1 = 516 . 349 mod 2143 = 72

M2 = 338 . 349 mod 2143 = 97

M3 = 117 . 349 mod 2143 = 116

M4 = 1824 . 349 mod 2143 = 105

Hasil Dekripsi (ASCII) : **72, 97, 116, 105**

Hasil *Plaintext* : Hati

**2.2.2.4. *Flowchart* Cara Kerja Algoritma *ElGamal***

*Flowchart* algoritma enkripsi *ElGamal* dapat dilihat pada gambar 2.4, sedangkan *flowchart* algoritma dekripsi *ElGamal* dapat dilihat pada gambar 2.5.



Gambar 2.4. *Flowchart* Enkripsi *ElGamal*



Gambar 2.5. *Flowchart* Dekripsi *ElGamal*

**2.2.2.5. Masa Perhitungan Algoritma *ElGamal***

Algoritma *ElGamal* merupakan algoritma kriptografi asimetris yang menggunakan dua jenis kunci, yaitu kunci publik dan kunci privat. Algoritma *ElGamal* mempunyai kunci publik berupa tiga pasang bilangan dan kunci rahasia berupa satu bilangan. Properti yang digunakan pada algoritma *ElGamal* antara lain :

1. Bilangan prima, *p* (tidak rahasia)

2. Bilangan acak, *g* ( *g* < *p*) (tidak rahasia)

3. Bilangan acak, *x* (*x* < *p*)(rahasia, kunci *private*)

4. *y* = *gx* mod *p* (tidak rahasia, kunci publik)

5. *m* (*plaintext*) (rahasia)

6. *a* dan *b* (*ciphertext*) (tidak rahasia)

Proses pertama dalam algoritma *ElGamal* adalah pembangkitan kunci. Pada proses ini dibutuhkan sebuah bilangan prima *p*, elemen *g* dan sembarang *x*0,1,..., *p* 2.

Langkah-langkah :

1. Pilih sembarang bilangan prima *p>255* ( *p* dapat dipublikasikan)

2. Pilih dua buah bilangan acak, *g* dan *x*, dengan syarat *g* < *p* dan 0  *x*  *p*–2

3. Hitung *y* = *gx* mod *p.*

Kunci publik algoritma *ElGamal* berupa pasangan 3 bilangan, yaitu (*y,g,p*). Sedangkan kunci privatnya adalah bilangan *x* tersebut.

Setelah proses pembangkitan kunci selesai kemudian pesan dienkripsi menggunakan kunci publik (*y,g,p*) dan sembarang bilangan acak rahasia *k*

0,1,..., *p* 2. Misalkan *m* adalah pesan yang akan dikirim.

Langkah-langkah Enkripsi Algoritma *ElGamal* :

1. Susun *plaintext* menjadi blok-blok *m*1, *m*2, …, *mn,* dengan setiap blok adalah satu karakter pesan.

2. Konversikan masing-masing karakter ke dalam kode ASCII, maka diperoleh *plaintext* sebanyak *n* bilangan, yaitu *m*1, *m*2,…, *mn,*

3. Untuk *i* dari 1 sampai *n* kerjakan :

a. Pilih sembarang bilangan acak rahasia

*k i* 0,1,..., *p* 2.

b. Hitung *ai* = *gki* mod *p*

c. Hitung *bi* = *ykimi* mod *p*

4. Diperoleh *ciphertext* yaitu (*ai,bi*) , *i*1, 2,..., *n.* jadi ukuran *Ciphertext* dua kali ukuran *plaintext* nya.

Dari algoritma diatas, kemudian dibuat *pseudocode* untuk memudahkan proses analisis. Dalam *pseudocode* algoritma *ElGamal* ini terdapat tiga buah perulangan,

perulangan yang pertama adalah operasi pembangkitan kunci, dan dua yang lainnya adalah proses enkripsi. Pada *pseudocode* ini *plaintext* dan *ciphertext* diterjemahkan ke dalam bentuk bilangan integer.

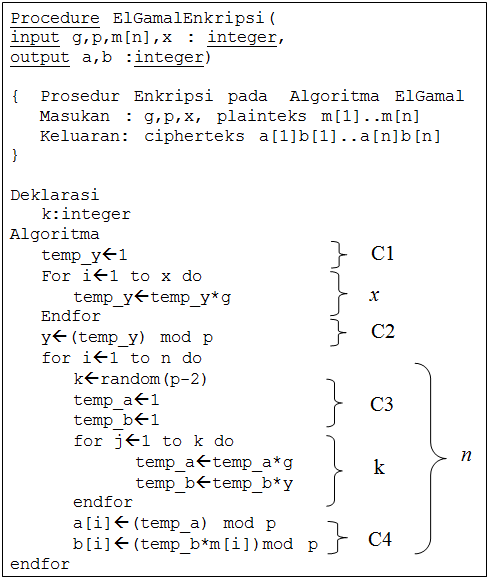
Berdasarkan *pseudocode* di gambar 2.6, kemudian dihitung banyaknya langkah-langkah pada operasi- operasi yang khas. Seperti yang terlihat pada persamaan 8.

Banyak langkah = C1 + *x* + C2 + *n*. ( C3 + k + C4 )

= *x* + *n*k + C (8)

Dari banyak langkah yang telah dihitung, maka diperoleh kompleksitas waktu untuk algoritma enkripsi *ElGamal*, Seperti yang terlihat pada persamaan 9.

*T*(*n*) = *x* + *n*k + C (9)



Gambar 2.6*. Pseudocode* enkripsi *ElGamal*

Sehingga kompleksitas waktu untuk kasus terburuk algoritma enkripsi *ElGamal* dapat dinyatakan pada persamaan 10.

Tmax(n) = x + nk + C (10)

Pada penggunaan yang sebenarnya, pemakaian x menggunakan seratus digit bilangan sebagai kunci privatnya, yang mengakibatkan lamanya proses enkripsi banyak tersita pada proses pembangkitan kunci, atau dapat dikatakan bahwa lama proses enkripsi sebanding dengan nilai x. Sedangkan n yang merupakan proses perulangan untuk setiap karakter *plaintext* dapat akan berpengaruh jika jumlah teks yang dienkripsi sangat panjang. Jika diasumsikan pemilihan nilai random k menggunakan bilangan yang sangat besar, bahkan mendekati nilai x, dan n menggunakan *plaintext* yang sangat panjang maka nilai k dan n akan sebanding dengan x. Oleh karena itu, nilai k dan n tidak bisa diabaikan karena turut berpengaruh terhadap lama proses enkripsi. Sehingga kompleksitas waktu untuk kasus terburuk, Seperti yang terlihat pada persamaan 11.

*Tmax*(*n*) = *x2* + *x* + C (11)

Dari kompleksitas waktu tersebut, dapat diketahui kompleksitas waktu asimtotik dari proses enkripsi algoritma *ElGamal*, Seperti yang terlihat pada persamaan 12.

T(n) = O(x2) (12)

Dari kompleksitas waktu tersebut, dapat diketahui kompleksitas waktu asimtotik dari proses enkripsi algoritma *ElGamal* adalah T(n) = O(x2).

Proses dekripsi pada algoritma *ElGamal* menggunakan kunci publik *p* dan kunci rahasia *x*. Dapat ditunjukkan bahwa *plaintext* m dapat diperoleh dari *ciphertext* menggunakan kunci rahasia *x*.

Langkah-langkah :

1. *Ciphertext* (*ai,bi*) , *i* 1, 2,..., *n*, kunci publik *p* dan kunci rahasia *x.*

2. Untuk *i* dari 1 sampai *n* kerjakan :

2.1. Hitung *ai p* – 1 – *x* mod *p*

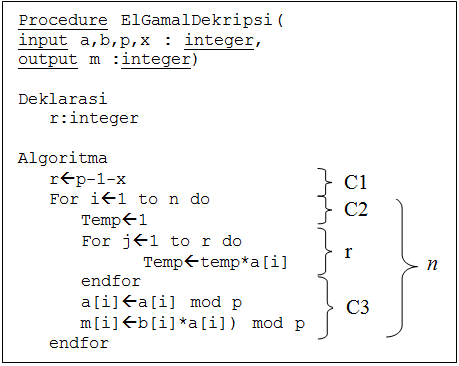
2.2. Hitung *mi* = *bi*/*a x* mod *p* = *bi*(*a x*)– 1mod *p*

*i*

*i*

3. Diperoleh *plaintext* *m*1, *m*2, …, *mn.*

4. Konversikan masing-masing bilangan *m*1, *m*2, …, *mn* ke dalam karakter sesuai dengan kode ASCII-nya, kemudian hasilnya digabungkan kembali.

Berbeda dari proses Enkripsinya, pada proses dekripsi, algoritma hanya menggunakan dua buah perulangan dikarenakan pada proses ini tidak ada langkah pembangkitan kunci.

Gambar 2.7. *Pseudocode* dekripsi *ElGamal*

Berdasarkan pseudocode pada gambar 2.7, kemudian dihitung banyaknya langkah proses dekripsi.

Banyak langkah = C1 + n . ( C2 + r + C3 )

= nr + C

Dari banyak langkah yang telah dihitung, maka diperoleh kompleksitas waktu untuk algoritma dekripsi *ElGamal* adalah : T(n) = nr + C

Pada penggunaan yang sebenarnya, pemilihan kunci privat *x* menggunakan bilangan yang sangat besar, sehingga mempengaruhi nilai r. Hal ini menjadikan variabel yang berpengaruh terhadap kompleksitas waktu algoritma dekripsi *ElGamal* adalah *x,* dan proses dekripsi menggunakan *ciphertext* yang sangat panjang hingga mendekati *x*, sehingga kompleksitas waktu untuk algoritma dekripsi *ElGamal* adalah:

Persamaan diatas adalah kompleksitas waktu untuk kasus terbaik dan terburuk, sehingga kompleksitas waktu asimtotik algoritma dekripsi *ElGamal* dapat dinyatakan seperti yang terlihat pada persamaan 13.

T(n) = O(x2) (13)

Dengan demikian, kompleksitas waktu dari algoritma enkripsi dan dekripsi *ElGamal* adalah kuadratik

* + 1. ***Elektronik Mail* (*Email*)**

Surat elektronik sudah mulai dipakai di tahun 1960 an. Pada saat itu internet belum terbentuk, yang ada hanyalah kumpulan *mainframe* yang terbentuk sebagai jaringan. Mulai tahun 1980 an, surat elektronik sudah bisa dinikmati oleh khalayak umum. Sekarang ini banyak perusahaan pos di berbagai negara menurun penghasilannya disebabkan masyarakat sudah tidak memakai jasa pos lagi. Untuk mengirim surat elektronik kita memerlukan suatu program *email client*. Dengan program inilah kita dapat mengirimkan *email* kita pada alamat tujuan melalui jaringan *public*, dengan menggunakan protokol SMTP. Sedangkan untuk dapat menerima pesan, hasil kiriman pihak lain untuk kita, maka *email* *client* menggunakan protokol POP3 atau IMAP. Keamanan data di surat elektronik tidaklah terjamin dan selalu ada resiko terbuka untuk umum, dalam artian semua isinya dapat dibaca oleh orang lain. Hal ini disebabkan oleh karena surat elektronik itu akan melewati banyak *server* sebelum sampai di tujuan. Tidak menutup kemungkinan ada orang yang menyadap surat elektronik yang dikirimkan tersebut. Surat elektronik dapat diamankan dengan melakukan teknik pengacakan (enkripsi). Salah satu program enkripsi yang populer adalah PGP (*Pretty Good Privacy*).

Dengan memakai PGP maka isi akan dienkrip, dan hanya orang yang tertuju dapat mendekripsi dan membaca surat elektronik tersebut. Kerugiannya adalah membuat repot pihak pengirim dan penerima (karena keduanya harus memiliki program PGP, dan pengirim juga harus memiliki kunci umum penerima, dan melakukan enkripsi pesan dengan kunci tersebut).

* + 1. ***Hypertext Preprocessor* (PHP)**

Menurut Bernadhed (2013), berpendapat bahwa “PHP adalah sebuah bahasa pemograman yang berjalan dalam sebuah *web-server* (*server side*).

PHP disebut bahasa pemrograman ***server side*** karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada *web browser* (*client*). Terdapat beberapa versi PHP salah satunya yaitu PHP 5 versi 5.6.32.

PHP 5 merupakan kelanjutan dari evolusi yang berjalan pada PHP. Walaupun pada PHP 4 sudah banyak *library* yang ditambahkan, PHP 5 menawarkan peningkatan dari fungsionalitas dan penambahan beberapa fitur, antara lain:

1. Peningkatan dari kemampuan pemrograman berorientasi objek.
2. *Exception handling*, yang menstandarisasi logika atas pemberitahuan kesalahan pemrograman Peningkatan terhadap penanganan string.
3. Peningkatan dukungan terhadap XML dan *Web Service*, yang menggunakan libxml2.
4. Dukungan terhadap SQLite *database server.*
5. PHP versi 5 dapat membuat file swf dan applet java.

Salah satu kelebihan PHP 5 adalah PHP 5 dapat menangani OOP (*Object Oriented Programming*) lebih baik dari pendahulunya. Ini juga yang menjadi alasan mengapa nomor versi php lompat ke versi 5. Berikut beberapa perbedaannya.

Pada php4, jika kita meng *copy* sebuah *object*, maka *object* tersebut akan terduplikat menjadi 2 dengan property yang sama. Sedangkan pada php5, *object* yang baru hanya berupa referensi ke *object* aslinya.

**BAB III**

**METODE PENELITIAN**

**3.1. Bahan dan Alat Penelitian**

**3.1.1. Bahan Penelitian**

Dalam penelitian ini, bahan penelitian yang akan digunakan adalah hasil jurnal, *textbook*, dan dokumentasi lainnya dari hasil studi literatur dan wawancara serta data-data yang bekaitan dengan pengamanan data pada *email* yaitu data dokumen dengan format file .docx, .xlxs, .pdf dan .txt yang didapatkan dari hasil survei pada Jurusan Teknologi Informasi Fakultas Teknik Universitas Tadulako.

**3.1.2. Alat Penelitian**

Pada penelitian ini digunakan alat penelitian berupa perangkat lunak dan perangkat keras.

1. Perangkat lunak (*Software*)

1. *Sublime Text 3*  sebagai *text editor*
2. *Php Versi 5.6.32* sebagai bahasa pemrograman

2. Perangkat Keras (*Hardware*)

1. *Processor intel core i5* pada laptop

*b. RAM 4 GB DDR 3* pada laptop

*c. Hard disk drive (HDD)* 500 *GB* pada laptop

*d. GPU Intel HD graphics 4000* pada laptop

29

**3.2. Desain Penelitian**

**3.2.1. Jenis Penelitian**

Jenis penelitian pada penelitian ini menggunakan studi kasus, dimana peneliti mengeksplorasi suatu masalah dengan batasan terperinci, memiliki pengambilan data yang mendalam, dan menyertakan berbagai sumber informasi.

**3.2.2. Tipe Penelitian**

Peneliti dalam penelitian ini menggunakan tipe penelitian kualitatif. Menurut Gumilang (2016) penelitian kualitatif dilakukan pada kondisi alamiah dan bersifat penemuan dan peneliti sebagai instrument kunci. Penelitian kualitatif tidak berbentuk angka dan penelitian kualitatif tidak memiliki rumus atau aturan *absolute* untuk mengolah dan menganalisa data.

**3.3. Objek, Waktu, dan Lokasi Penelitian**

Objek dalam penelitian ini adalah pengamanan data yang akan dirancang oleh peneliti. Penelitian ini dilakukan di Jurusan Teknologi Informasi fakultas Teknik Universitas Tadulako. Adapun waktu yang ditargetkan oleh peneliti adalah 4 (empat) bulan dari pengajuan proposal penelitian ini dimulai pada bulan Oktober 2017 sampai Januari 2018.

**3.4. Jenis, dan Sumber Data**

Adapun jenis dan sumber data yang digunakan adalah data primer dan data sekunder.

1. Data Primer

Data primer, merupakan data yang di peroleh langsung secara dari pihak yang diperlukan datanya.

1. Data Sekunder

Data sekunder, merupakan data yang mengacu pada informasi yang dikumpulkan dari sumber yang telah ada. Data Sekunder adalah data yang telah dikumpulkan oleh para peneliti, data yang diterbitkan dalam jurnal statistik dan lainnya, serta informasi yang tersedia dari sumber publikasi atau non publikasi entah di dalam atau luar organisasi, semua yang dapat berguna bagi peneliti. Adapun yang data sekunder pada penelitian ini didapatkan melalui jurnal-jurnal dan referensi yang berkaitan dengan pengamanan data *email* berbasis web.

**3.5. Teknik Pengumpulan Data**

Adapun teknik pengumpulan data yang digunakan adalah teknik studi literatur.

1. Teknik Studi Literatur

Teknik ini dilakukan dengan pengumpulan data-data yang berhubungan dengan topik penelitian. Data-data bisa didapatkan dari berbagai sumber seperti jurnal, makalah, buku dokumentasi, internet, dan lain-lain.

1. Teknik Wawancara

Teknik ini dilakukan dengan wawancara secara langsung melalui tatap muka dan tanya jawab dengan pihak yang terkait. Wawancara dilakukan terhadap bagian kepala Jurusan Teknologi Informasi fakultas Teknik Universitas Tadulako.

**3.6. Metode Analisa Data**

Dalam penggalian data dan informasi, peneliti menggunakan metode *Unified Modelling Language* (*UML)* yaitu suatu metode yang dapat digunakan untuk memahami serta mendokumentasikan sistem perangkat lunak secara visual. Adapun jenis *UML* yang digunakan yaitu *Use Case* dan *Sequence Diagram* dan analisis perhitungan algoritma Elgamal, Selain jenis UML tersebut, penulis juga menggunakan *Context Diagram* dan *Data Flow Diagram* (DFD). Pada *diagram konteks* ini terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Ia akan memberi gambaran tentang keseluruan sistem.

**3.7. Metode Pengembangan Sistem**

*Waterfall model* merupakan salah satu model proses perangkat lunak yang mengambil kegiatan proses dasar seperti spesifikasi, pengembangan, validasi, evolusi, dan merepresentasikanya sebagai fase-fase proses yang berbeda seperti analisis dan definisi persyaratan, perancangan perangkat lunak, implementasi, pengujian unit, integrasi sistem, pengujian sistem, operasi, dan pemeliharaan. Model *waterfall* dapat dilihat pada gambar 3.1. Adapun penjelasanya sebagai berikut:

1. Definisi Persyaratan(*Requirements Definition*)

Pada tahapan ini, penulis mengumpulkan informasi kebutuhan perangkat lunak secara lengkap untuk selanjutnya dianalisis kemudian didefinisikan sebagai kebutuhan sistem yang harus dipenuhi. Melalui proses ini peneliti akan mendefinisikan secara rinci mengenai fungsi-fungsi, batasan, dan tujuan dari perangkat lunak sebagai spesifikasi sistem. Informasi dan data yang dikumpulkan berdasarkan buku, jurnal-jurnal, dan referensi yang berkaitan dengan pengamanan data pada *email* berbasis web.

1. Perancangan Sistem dan Perangkat Lunak (*System and Software Design*)

Pada tahap ini, penulis merealisasikan kebutuhan perangkat lunak kedalam perancangan perangkat lunak. Perancangan dimulai dengan merepresentasikan hubungan data-data yang didapatkan, secara logis menjadi struktur data sebagai keperluan sistem. Selanjutnya dilakukan perancangan tahapan-tahapan sistem secara lebih detail mencakup fungsi dan algoritma yang digunakan dan kemudian perancangan antar muka (*interface*) perangkat lunak dilakukan sesuai dengan kebutuhan sistem sebelum masuk dalam tahap implementasi.

1. Implementasi dan Pengujian Unit(*Implementation and Unit Testing*)

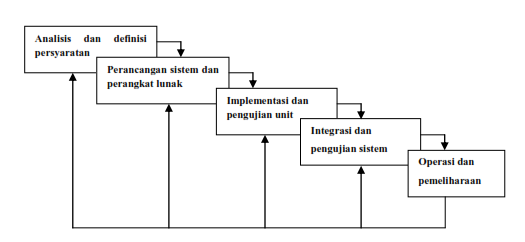
Pada tahap ini, perancangan sistem dan perangkat lunak direalisasikan ke dalam serangkaian kode-kode bahasa pemrograman menjadi suatu unit program. Kemudian dilakukan pengujian unit melalui proses verifikasi untuk memastikan spesifikasinya terpenuhi.

1. Integrasi dan Pengujian Sistem(*Integration and System Testing*)

Pada tahap ini, unit-unit program diintegrasikan menjadi sebuah kesatuan sistem yang selanjutnya dilakukan pengujian secara keseluruhan untuk menguji keterhubungan dari setiap fungsi perangkat lunak untuk memastikan keseluruhan spesifikasinya terpenuhi.

1. Operasi dan Pemeliharaan(*Operation and Maintenance*)

Pada tahap ini, perangkat lunak yang telah sukses dalam pengujian sistem dilakukan pemeliharaan yang meliputi pengembangan, perbaikan, dan penambahan fungsi ataupun persyaratan yang perlu dilakukan.



\

Gambar 3.1. *Waterfall Model*

(Sumber: Sommerville dalam Angga dan bunyamin, 2015)

**3.8. Tahapan dan Diagram Alir Penelitian**

Tahapan yang dilakukan pada penelitian ini adalah:

1. Survei Literatur

Melakukan pengumpulan bahan, informasi, dan referensi yang relevan berkaitan dengan topik penelitian

1. Identifikasi Masalah

Melakukan identifikasi masalah untuk menentukan solusi dan pembahasan apa yang diperlukan, berkaitan dengan topik penelitian.

1. Studi Pustaka

Mempelajari literatur yang akan digunakan dengan masalah yang dihadapi untuk digunakan sebagai kajian pendukung teori dalam penelitian ini.

1. Pengumpulan Data

Melakukan wawancara secara langsung kepada pihak yang terkait untuk mendapatkan informasi dan keperluan sistem, seperti mahasiswa dan dosen Jurusan Teknologi Informasi.

1. Perancangan Perangkat Lunak

Melakukan perancangan kebutuhan perangkat lunak dan tahapan-tahapan pada proses operasi aplikasi mencakup tentang Pengamanan data pada *email* mengguunakan algoritma Asimetris *ElGamal* berbasis web.

Diagram blok sistem aplikasi pengamanan data pada *email* dapat dilihat pada diagram alir digambar 3.2.



Gambar 3.2. Diagram blok sistem

1. Pembuatan Perangkat Lunak

Melakukan pembuatan perangkat lunak antara lain melakukan persiapan *software* yang akan digunakan, membuat desain *interface,* dan melakukan *coding.*

1. Penginputan Data

Melakukan penginputan data ke sistem untuk digunakan sebagai bahan pengolahan sistem.

1. Menjalankan Perangkat Lunak

Setelah pembuatan perangkat lunak dan penginputan data selesai, selanjutnya menjalankan perangkat lunak untuk mengetahui apakah program yang dibuat telah memenuhi spesifikasi dan dapat melakukan proses pengamanan data pada *email* dengan baik.

1. Pengujian Perangkat Lunak

Pada tahap ini dilakukan pengujian perangkat lunak secara menyeluruh untuk memastikan fungsi-fungsi dari perangkat lunak telah berjalan dengan baik sesuai yang diharapkan dengan memastikan tidak ada *bug* ataupun *logic error* pada perangkat lunak.

1. Kesimpulan dan Saran

Menuliskan kesimpulan melalui analisa akhir sistem yang diperoleh serta memberikan saran-saran yang diperlukan untuk pengembangan sistem.

Tahapan-tahapan penelitian dapat dilihat pada diagram alir digambar 3.3.



Gambar 3.3. *Flowchart* Tahapan Penelitian.

**BAB IV**

**HASIL DAN PEMBAHASAN**

* 1. **Bahan dan Alat Penelitian**
     1. **Analisis Sistem**

Pada analisis sistem, aplikasi pengamanan data pada *email* dapat melakukan beberapa hal yaitu :

1. Aplikasi dapat terhubung dengan gmail.
2. Aplikasi yang dibuat dapat mengamankan dan menjamin keamanan data pada email.
3. Aplikasi pengaman data dilakukan menggunakan teknik enkripsi pada saat proses pengiriman pesan.
4. Aplikasi dapat mengirimkan pesan ke email tujuan.
5. Aplikasi dapat mendekripsi pesan yang telah masuk ke email.
6. Aplikasi dapat mengunduh file ketika proses dekripsi pesan telah selesai.
7. Diagram Konteks

Skema diagramkonteks aplikasi pengamanan data pada *email* dapat dilihat pada gambar 4.1.



Gambar 4.1. Diagram konteks aplikasi pengamanan data pada *email*

39

Keterangan gambar 4.1. dapat dilihat sebagai berikut:

1. Pengirim

Pengirim melakukan penginputan data-data dan file yang akan di kirim ke penerima, dalam hal ini data-data dan file yang dikirim adalah *email* tujuan, subjek, isi pesan dan lampiran file. Pada proses kirim pesan terdapat proses enkripsi data isi pesan dan lampiran file yang akan merubah data *plaintext* menjadi *ciphertext*

1. Penerima

Penerima menerima data dan file yang dikirim oleh pengirim, kemudian melakukan dekripsi pesan dengan menginputkan kunci *private* *x* dan *p*. dimana kunci *private* *x* dan *p* didapat dari *email* yang dikirim oleh pengirim bersamaan dengan data dan file. Dekripsi pesan berfungsi untuk merubah dari data *ciphertext* menjadi *plaintext* (pesan asli).

1. *Data Flow Diagram* (DFD)

Skema DFD level 1 dan DFD level 2 aplikasi pengamanan data pada *email* dapat dilihat masing-masing pada gambar 4.2 dan gambar 4.3.



Gambar 4.2. *Data Flow Diagram* Level 1 aplikasi pengamanan data pada *email*



Gambar 4.3. *Data Flow Diagram* Level 2 aplikasi pengamanan data

pada *email*

1. *Use Case Diagram*

Skema *Use case diagram* aplikasi pengamanan data pada *email* dapat dilihat pada gambar 4.4.



Gambar 4.4. *Use Case Diagram* aplikasi pengamanan data pada *email*

Keterangan gambar 4.4. dapat dilihat sebagai berikut:

1. *Login* pada aplikasi

Pengguna (pengirim dan penerima) melakukan *login* sistem dengan memasukan *username* dan *password*.

1. *Login* pada *email*

Pada tahap ini penerima melakukan *login* untuk membuka akun *email*nya.

c. Buat pesan pada aplikasi

Pada tahap ini pengirim menginputkan pesan yang akan dikirim pada penerima.

1. Mengirim dan mengenkripsi pesan pada aplikasi

Pada tahap ini pengirim mengenkripsi pesan *plaintext* ke dalam bentuk *ciphertext* dan proses enkripsi pesan tersebut terjadi bersamaan saat pesan dikirim

1. Menerima pesan pada kotak masuk *email*

Pada tahap ini penerima pesan baru pada kotak masuk yang dimana pesan tersebut hanya berupa *ciphertext*.

1. Membuka pesan pada kotak masuk aplikasi

Pada tahap ini penerima membuka pesan baru pada kotak aplikasi kemudian memilih pesan yang akan didekripsi.

1. Dekripsi pesan

Padan tahap ini penerima melakukan dekripsi pesan dengan menginputkan *private key* yang dimana digunakan untuk membuka pesan yang dalam bentuk *ciphertext* ke dalam bentuk *plaintext* (file asli).

1. *Sequence Diagram*

Skema *Sequence diagram* aplikasi pengamanan data pada *email* dapat dilihat pada gambar 4.5.



Gambar 4.5. *Sequence Diagram* proses enkripsi aplikasi pengamanan data

pada *email*



Gambar 4.6. *Sequence Diagram* proses dekripsi aplikasi pengamanan data

pada *email*

* + 1. **Implementasi Sistem**
       1. **Implementasi Perangkat Keras dan Perangkat Lunak**

Implementasi perangkat keras yang digunakan untuk membangun sistem ini yaitu rancang bangun pengamanan data pada *email* dengan spesifikasi *Processor Core i5, RAM 2 Gb,* monitor dengan resolusi 1366x768 dan sistem operasi 64 bit.

Untuk implementasi perangkat lunak bahasa pemrograman yang digunakan yaitu *Hypertext Preprocessor* (PHP), media yang digunakan sebagai editor untuk membangun sistem yaitu *Sublime Text 3* dan *xammp* sebagai penghubung ke *database server MySQL* dengan bahasa pemrograman.

* + - 1. **Implementasi *Input***

Pada aplikasi pengamanan data pada *email* terdapat beberapa *form* untuk memasukan data, antara lain sebagai berikut:

1. *Input* data *login*

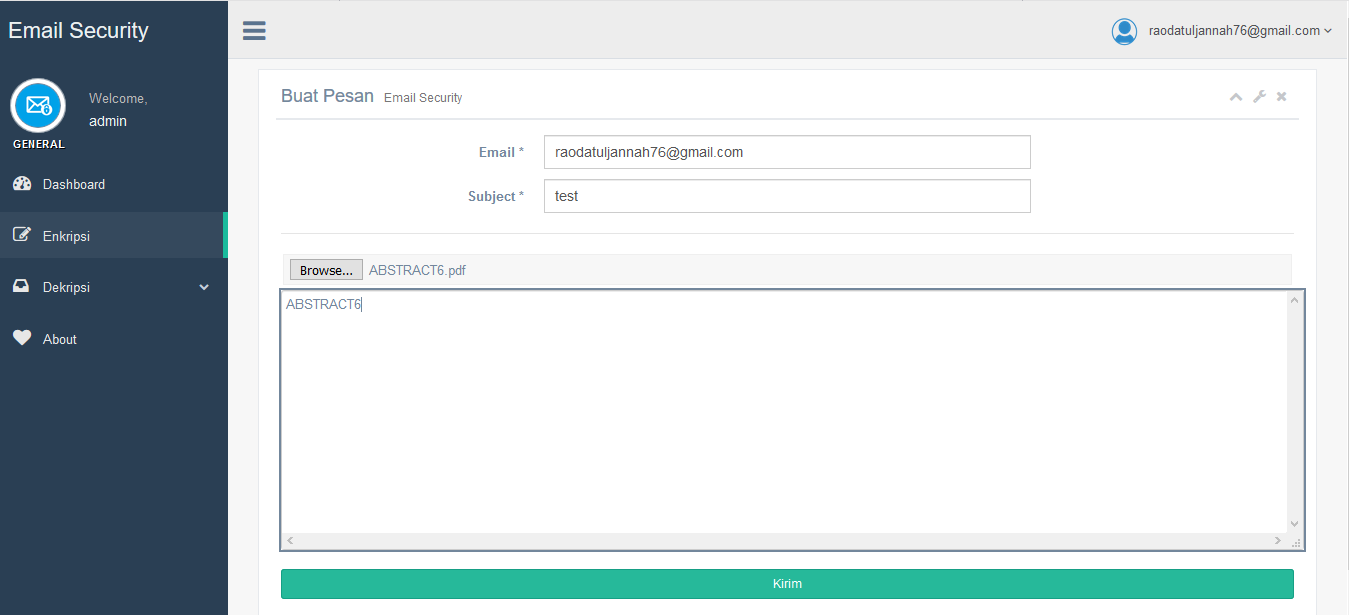
*Form login* ini digunakan untuk memasukkan akun *email* pengguna agar bisa mengakses aplikasi. *Form login* dapat dilihat pada gambar 4.7.



Gambar 4.7. *Form login*

1. *Input* data pesan (buat pesan)

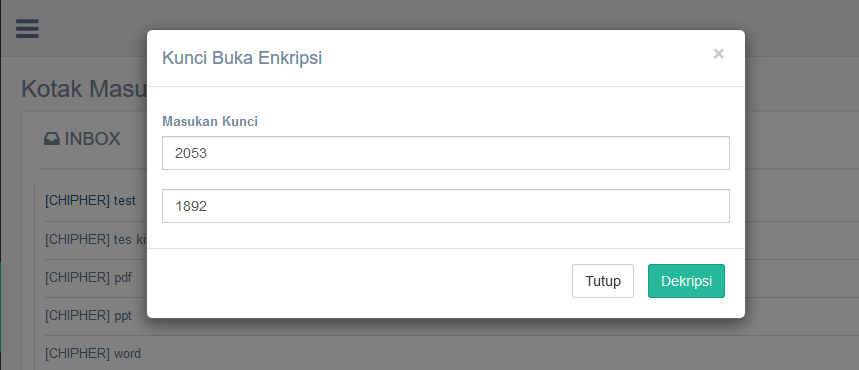
*Form* ini digunakan untuk memasukkan data pesan yang akan dikirim ke *email* tujuan. Pada *form* ini terdapat proses enkripsi pada file atau data yang terlampir ketika *user* mengirim pesan ke *email* tujuan, data yang dimasukkan yaitu alamat *email*, subjek, isi pesan, danlampiran file. Keterangan seperti yang terlihat pada gambar 4.8.



Gambar 4.8. *Form input* data pesan

1. *Input* dekripsi pesan *email* (kunci *private x* dan *p*)

*Form* ini digunakan untuk mendekripsi pesan yang terenkripsi dengan menginputkan kunci *private x* dan *p* menjadi file *plaintext*/data asli seperti yang terlihat pada gambar 4.9.



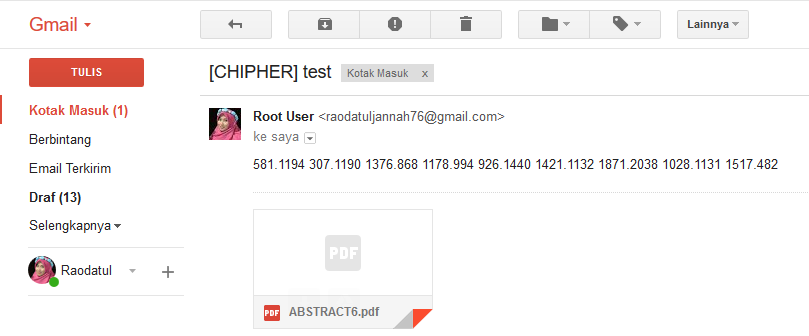
Gambar 4.9. *Form input* dekripsi pesan

* + - 1. **Implementasi *Output***

Pada aplikasi pengamanan data pada *email* terdapat beberapa *form* sebagai *output* antara lain sebagai berikut:

1. *Output* pesan pada *email* yang telah terenkripsi

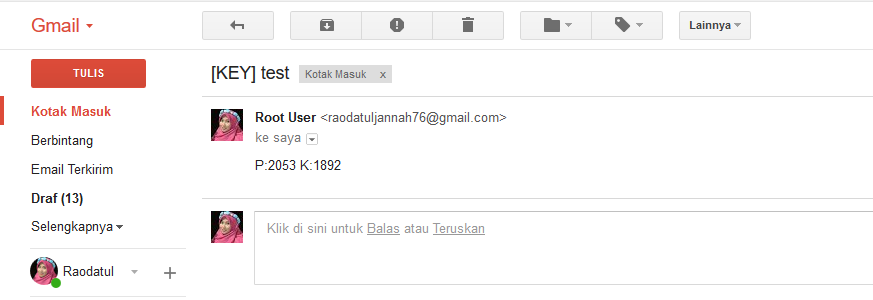
*Output* ini merupakan tampilan pesan pada *email* yang telah terenkripsi yang dikirim oleh *user,* seperti yang terlihat pada gambar 4.10.



Gambar 4.10. Pesan pada *email* yang telah terenkripsi

1. *Output* kunci *private x* dan *p* pada *email*

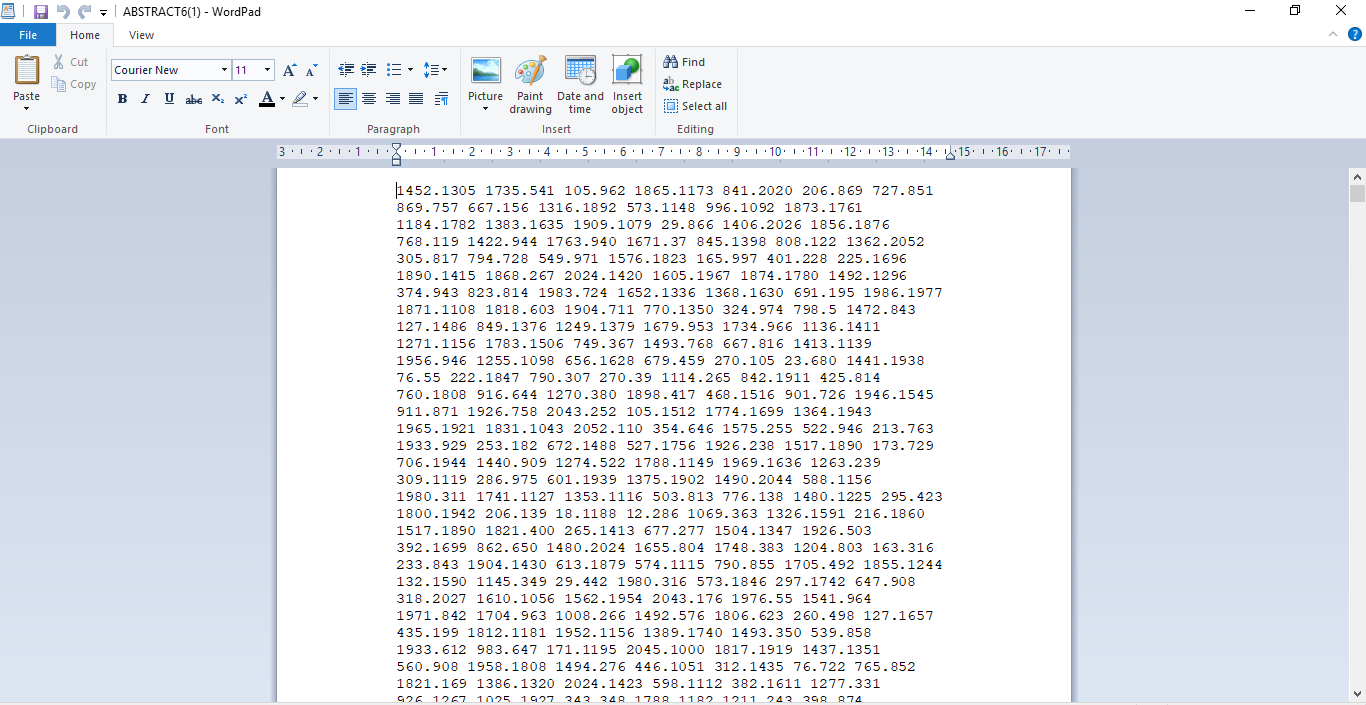
*Output* ini merupakan tampilan kunci *private x* dan *p* pada *email* yang dikirim oleh *user* bersamaan dengan pesan yang terlampir, seperti yang terlihat pada gambar 4.11.



Gambar 4.11. Kunci *private x* dan *p* pada *email*

1. *Output* pesan yang terenkripsi ketika dibuka dengan aplikasi WordPad

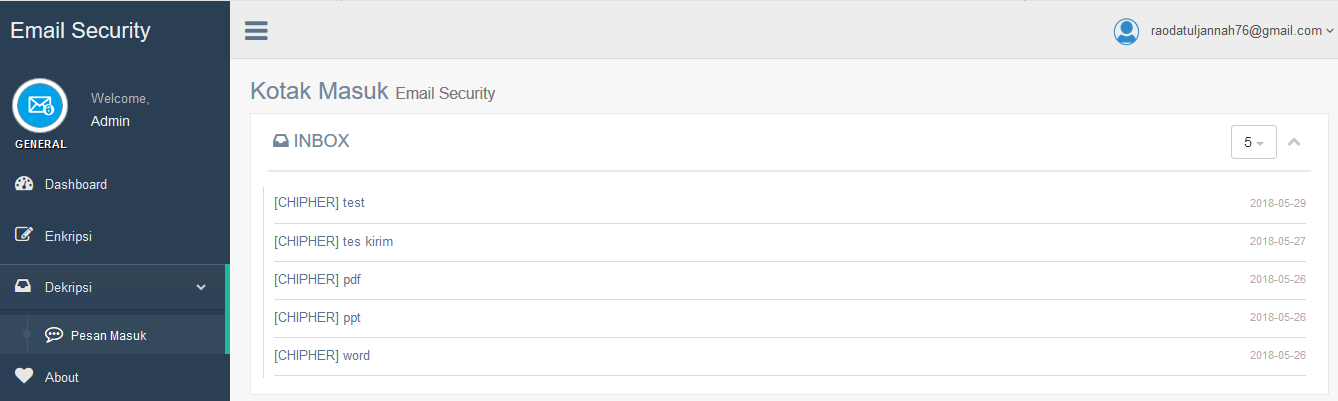
*Output* ini merupakan tampilan pesan yang telah terenkripsi dalam menjadi *ciphertext* ketika dibuka dengan aplikasi WordPad, seperti yang terlihat pada gambar 4.12.



Gambar 4.12. Pesan yang terenkripsi ketika dibuka dengan aplikasi WordPad

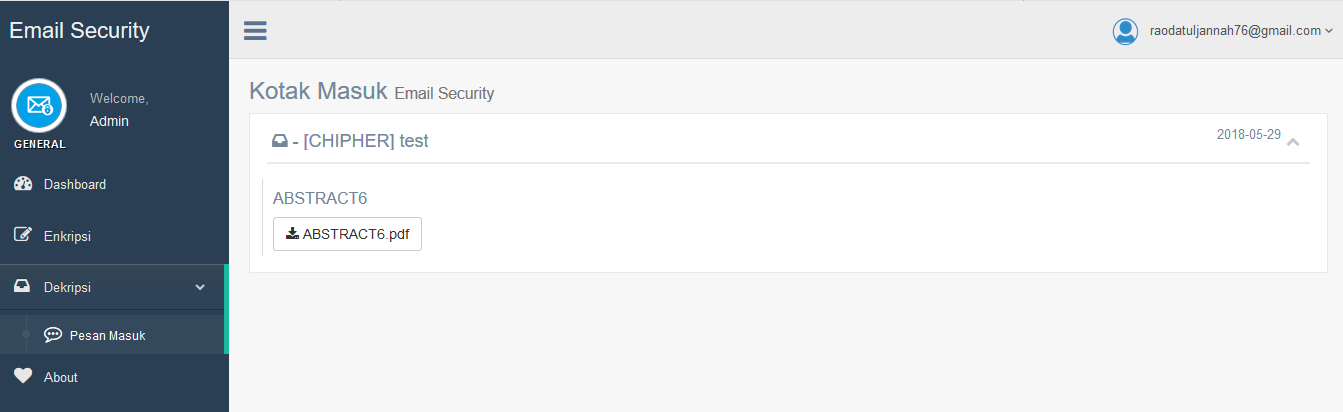
1. *Output form* kotak masuk (*inbox*)

*Form* ini digunakan untuk *user* dapat melihat pesan yang masuk dan dapat memilih pesan mana yang akan di dekripsi. Pada aplikasi ini *form* kotak masuk hanya akan menampilkan pesan dengan subjek awalan [CIPHER], untuk subjek dengan awalan bukan [CIPHER] tidak akan ditampilkan di aplikasi, seperti yang terlihat pada gambar 4.13.

Gambar 4.13. *Form* kotak masuk (*inbox*)

1. *Output Form* unduh file

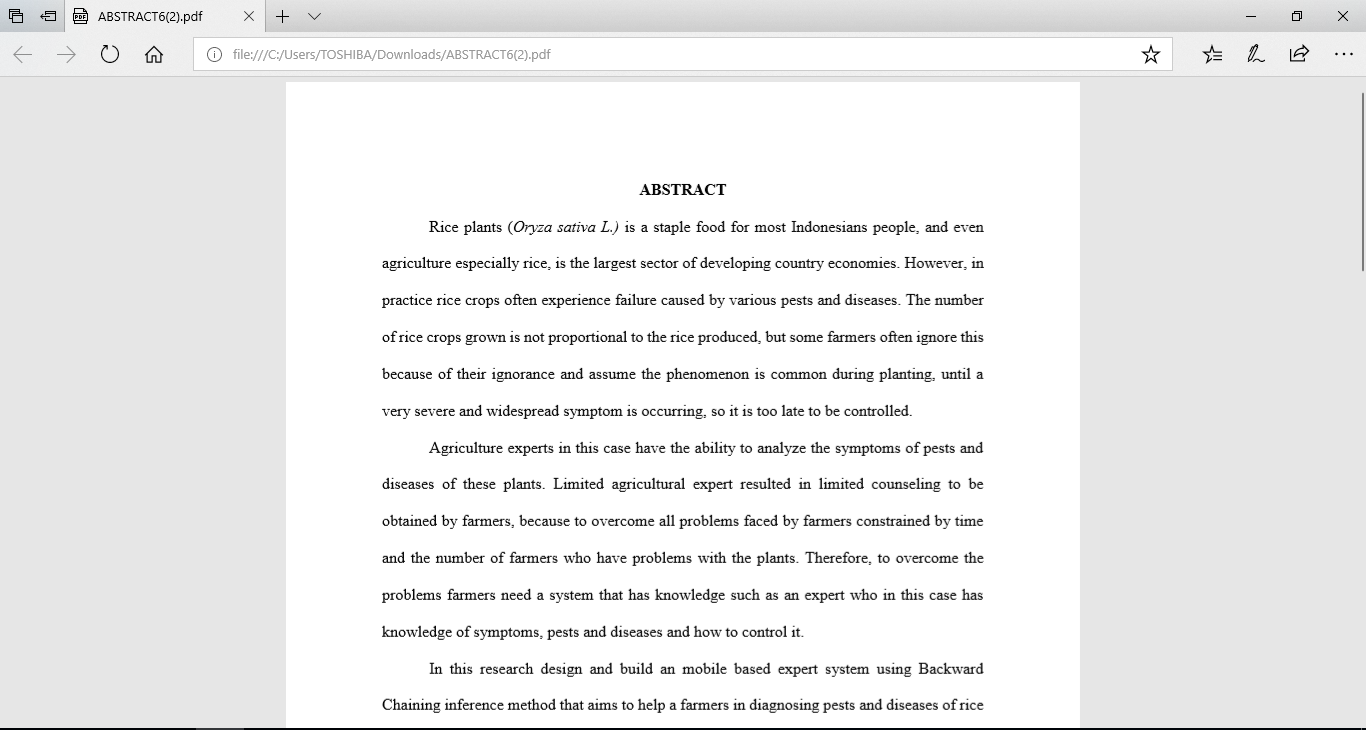
*Form* ini digunakan untuk mengunduh file yang telah terdekripsi untuk disimpan di direktori komputer, sehingga dapat dibaca oleh *user*, seperti yang terlihat pada gambar 4.14.



Gambar 4.14. *Form* unduh file

1. *Output* pesan yang telah terdekripsi

*Output* ini merupakan tampilan pesan yang telah diunduh dan sudah terdekripsi ke pesan asli (*plaintext*), seperti yang terlihat pada gambar 4.15.



Gambar 4.15. *Output* Pesan yang telah terdekripsi (*plaintext*)

* + 1. **Pengujian Sistem**

Setelah pembuatan aplikasi telah selesai, maka dilakukan pengujian terhadap aplikasi. Pengujian sistem yang dilakukan merupakan pengujian terhadap aplikasi pengamanan data pada *email*. Pengujian fungsi sistem dilakukan dengan metode *blackbox* untuk mengetahui apakah fungsi-fungsi yang terdapat dalam sistem berjalan dengan baik atau tidak.

Hasil pengujian fungsi sistem dan pengujian lapangan dapat dilihat pada tabel berikut:

1. Pengujian *Login*

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada *form login*. Pengujian *login* dapat dilihat pada tabel 4.1.

Tabel 4.1. Tabel pengujian fungsi *login*

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Masukan** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| *Username* | Data yang dimasukan benar  lalu masuk ke halaman *inbox* | Menampilkan halaman *inbox* | Baik |
| *Password* | Data yang dimasukan benar  lalu masuk ke halaman *inbox* | Menampilkan halaman *inbox* | Baik |

1. Pengujian buat pesan

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada *form* buat pesan. Pengujian buat pesandapat dilihat pada tabel 4.2.

Tabel 4.2. Tabel pengujian buat dan kirim pesan

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** **Masukan** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Input data | Dapat menginput *email* tujuan, subjek, isi pesan serta melampirkan file | Sistem berjalan dengan baik dengan berhasil menginput data *email*, subjek, isi pesan, dan melampirkan file. | Baik |
| Kirim pesan | File dan kunci *private* ( *x* dan *p*) terkirim ke *email* dengan data inputan isi pesan (*body*) dan file yang terlampir ter enkripsi. | Sistem berjalan dengan baik dengan berhasil mengenkripsi dan mengirim pesan serta kunci dekripsi ke *email*. | Baik |

1. Pengujian kotak masuk (*inbox*) pesan

Pengujian ini digunakan untuk pengecekan dan pengujian fungsi-fungsi yang ada pada *form* kotak masuk. Pengujian inidapat dilihat pada tabel 4.3.

Tabel 4.3. Tabel kotak masuk pesan

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** **Masukan** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Pilih pesan yang akan di dekripsi | Muncul jendela pop up masukan kunci *private* *x* dan *p* | Sistem menampilkan dengan baik jendela pop up | Baik |
| Dekripsi | File yang ter enkripsi dapat ter dekripsi menjadi *plaintext* (data asli) | Sistem berhasil mendekripsi pesan enkripsi (*ciphertext*) menjadi *plaintext* (data asli) | Baik |

1. Pengujian download file

Tabel 4.4. Tabel pengujian fungsi download file

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** **Masukan** | **Ekspektasi** | **Pengamatan** | **Kesimpulan** |
| Download file | File terdownload dan tersimpan ke direktori komputer | Sistem berjalan dengan baik dengan berhasil mendownload file yang telah ter dekripsi | Baik |

* 1. **Pembahasan**

**4.2.1. Tahapan Alir Sistem**

Pada aplikasi pengamanan data pada *email* terdapat 2 proses yaitu proses enkripsi yang dilakukan pengirim dan proses dekripsi yang dilakukan penerima, pada masing-masing proses terdapat langkah-langkah sebagai berikut :

1. Proses Enkripsi
2. Input Pesan

Pada *form* input pesan pengguna harus menginput pesan atau melampirkan file yang akan di enkripsi dan di kirim pada *email* tujuan.

1. Pembangkitan kunci publik dan *private*

Pada tahap ini pembangkitan kunci publik dan *private* di proses pada saat pengiriman file. Kunci publik digunakan pada saat proses enkripsi sedangkan kunci *private* digunakan pada saat proses dekripsi.

1. Enkripsi file

Proses enkripsi file atau data asli merupakan proses dimana data asli pada isi pesan dan isi file yang dilampirkan berubah menjadi *ciphertext*.

1. Mengirim file dan kunci *private*

Pada proses ini file dan isi pesan yang telah terenkripsi akan dikirim ke *email* tujuan.

Tahapan-tahapan diatas dapat digambarkan pada diagram alir yang terdapat pada gambar 4.16. dibawah ini:



Gambar 4.16. *Flowchart* sistem enkripsi aplikasi pengamanan data pada *email*

1. Proses Dekripsi
2. Buka kotak masuk pesan

Pada *form* ini, akan menampilkan pesan yang telah masuk ke *email* pengguna.

1. Pilih pesan dan *input* kunci *private x* dan *p*

Pilih pesan masuk yang akan di dekripsi dengan menginputkan kunci *private x* dan *p*.

1. Dekripsi file

Pada proses dekripsi file, pesan yang telah di pilih dan di inputkan kunci *private* *x* dan *p* akan secara langsung terdekripsi atau mengembalikan pesan seperti semula (data asli).

1. Unduh file yang telah di dekripsi

Pada form ini, pesan telah terdekripsi dan untuk membaca pesan pengguna harus mengunduh file tersebut ke direktori komputernya dan pesan siap untuk di baca.

Tahapan-tahapan diatas dapat digambarkan pada diagram alir yang terdapat pada gambar 4.17. dibawah ini:



Gambar 4.17. *Flowchart* sistem dekripsi aplikasi pengamanan data pada *email*

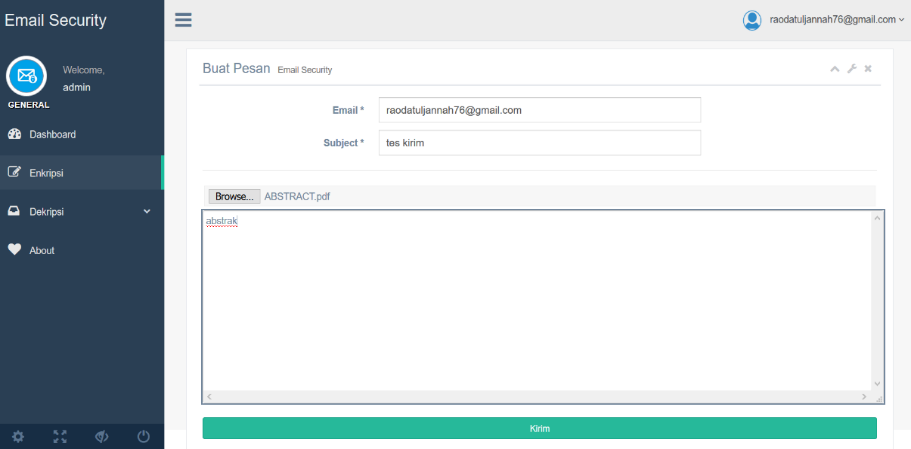
* + 1. **Pengamanan Data Pada *Email***

1. Penginputan data dan file (Enkripsi)

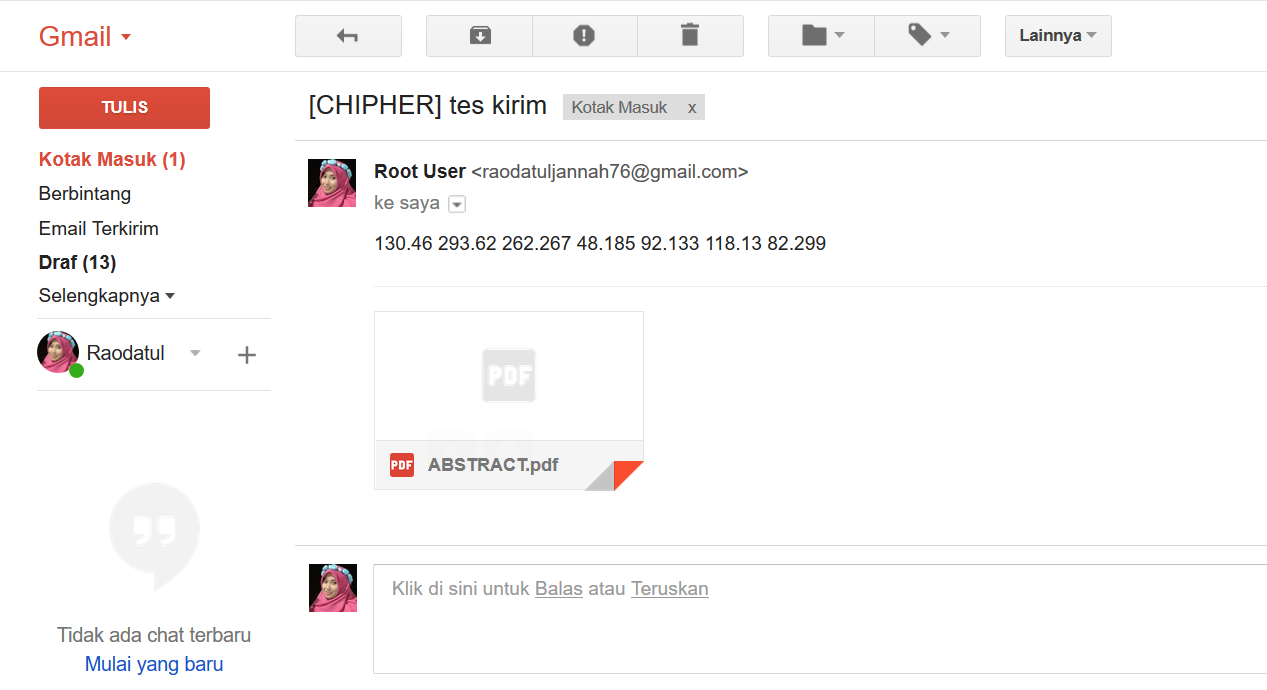
Pengamanan data pada *email* dimulai dari tahap penginputan data dan file yaitu berupa alamat *email* tujuan, subjek *email*, lampiran file berupa dokumen dengan ekstensi .docx, .xlsx, .txt, dan .PDF. Kemudian data dan file yang telah di*inputkan* siap dikirim ke *email* tujuan. Pada proses pengiriman data dan file terdapat proses enkripsi yang dimulai dari proses pembangkitan kunci publik dan *privat*e. Kunci publik digunakan pada proses enkripsi sedangkan kunci *private* digunakan pada proses dekripsi. Pada saat program mengirim pesan ke *email* tujuan proses enkripsi terjadi. Proses awal yaitu pembangkitan kunci, dengan kunci yang dibangkitkan yaitu secara *random*. Kunci *p* yang dibangkitkan dengan nilai batas awal yaitu 256 dan batas akhir dengan nilai 256\*10 = 2560. Proses pembangkitan kunci akan terus berulang sampai didapatkan bilangan prima.

Pada penjelasan diatas proses pembentukan kunci dibutuhkan sebuah bilangan prima p, bilangan acak g dan x, dengan syarat g < p dan 1 ≤ x ≤ p-2. Kunci publik algoritma *ElGamal* terdiri atas pasangan 3 bilangan (y, g, p) di mana hubungan dari ketiga bilangan tersebut yaitu *y = gx mod p.*

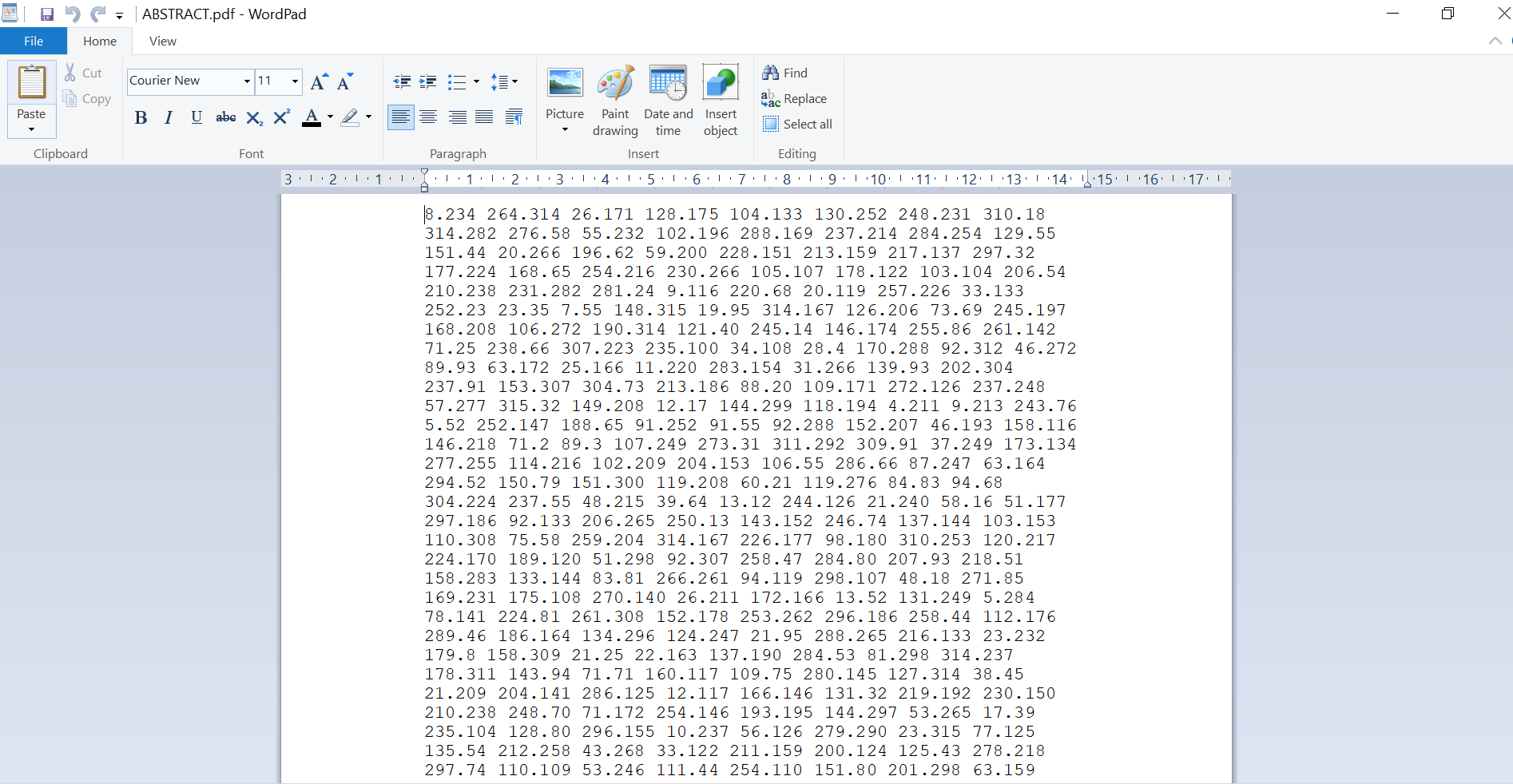
Kemudian selanjutnya proses enkripsi menggunakan kunci publik (*y*, *g*, *p*) dan sebuah bilangan acak *k* yang dalam hal ini *1 ≤ k ≤ p-2*. Untuk setiap karakter dalam pesan dienkripsi dengan menggunakan bilangan k yang berbeda-beda. Satu karakter yang direpresentasikan dengan menggunakan bilangan bulat ASCII akan menghasilkan kode dalam bentuk blok yang terdiri atas dua nilai (*a*, *b*), dengan nilai a yaitu *a* = *g*k mod *p* dan nilai b yaitu *b* = *y*k *m* mod *p.* Proses input data dan file dapat dilihat pada gambar 4.18. dan hasil pesan yang telah terkirim dan terenkripsi (*ciphertext*) dapat dilihat pada gambar 4.19, serta hasil enkripsi (*ciphertext*) yang dibuka dengan menggunakan aplikasi WordPad dapat dilihat pada gambar 4.20.



Gambar 4.18. Proses *input* data dan file.



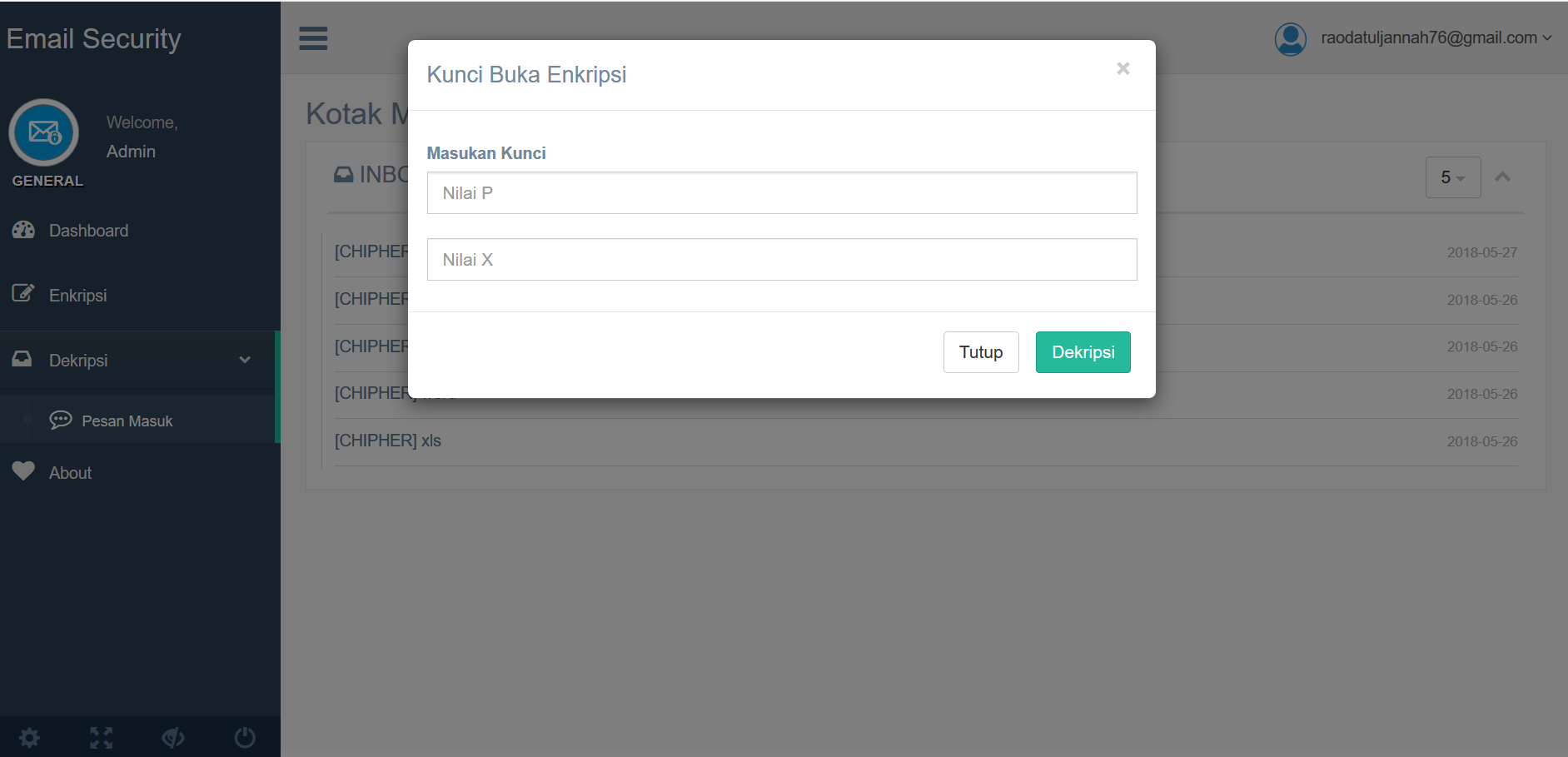
Gambar 4.19. Pesan yang telah terkirim ke *email* dalam bentuk *ciphertext*.



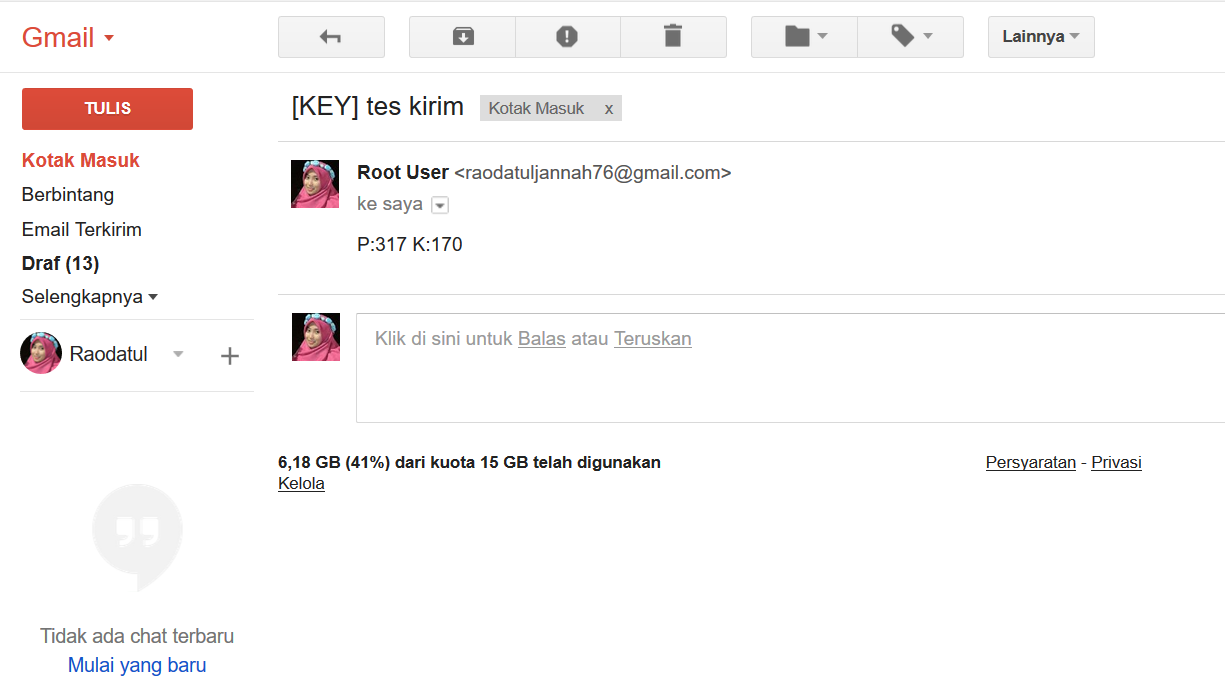
Gambar 4.20. Hasil enkripsi (*ciphertext*) yang dibuka dengan menggunakan aplikasi WordPad.

1. Melakukan dekripsi data dan file yang terkirim ke *email* pada kotak masuk

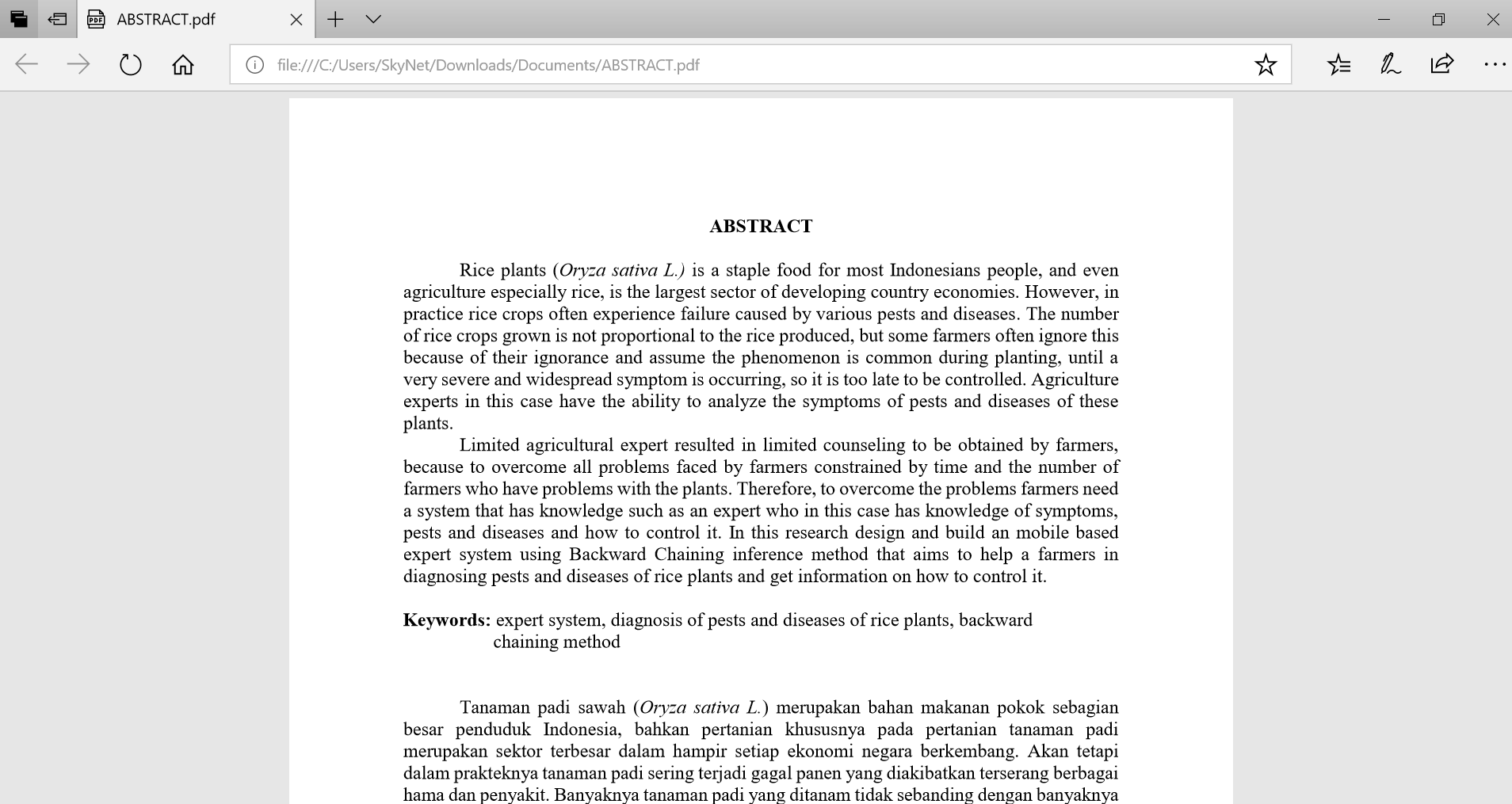
Tahap kedua yaitu melakukan dekripsi data dan file dengan memilih pesan yang akan didekripsi pada kotak masuk (*inbox*). Pesan yang ditampilkan pada kotak masuk hanya akan menampilkan pesan dengan subjek awalan [CIPHER], untuk subjek dengan awalan bukan [CIPHER] tidak akan ditampilkan di kotak masuk aplikasi, kemudian setelah memilih pesan yang akan didekripsi pengguna menginputkan kunci *private x* dan *p*. Kunci *private* *x* dan *p* secara otomatis terdapat pada akun *email* yang terkirim bersamaan dengan data dan file. Pada tahap dekripsi. dari *ciphertext* ke *plaintext* menggunakan kunci *private* *x* dan *p* yang disimpan kerahasiaanya oleh penerima pesan. Diberikan (*y*, *g*, *p*) sebagai kunci publik dan *x* sebagai kunci rahasia pada algoritma *ElGamal*. Jika diberikan *ciphertext* (*a, b*), maka didapatkan hubungan persamaan yaitu *m* = *b*.*c* mod *p* dengan *m* adalah *plaintext*. Di mana nilai dari variabel *c* dicari dengan menggunakan persamaan *c* = *a p-1-x* mod *p.* Proses dekripsi data dan file, dengan penginputan kunci *private x* dan *p* dapat dilihat pada gambar 4.21, kunci *private x* dan *p* pada *email* yang digunakan untuk mendekripsi data dan file dapat dilihat pada gambar 4.22 dan hasil dekripsi dari pesan enkripsi *ciphertext* ke data asli (*plaintext*) dapat dilihat pada gambar 4.23.



Gambar 4.21. Proses dekripsi data dan file pada kotak masuk dengan menginputkan kunci *private x* dan *p*.



Gambar 4.22. Kunci *private x* dan *p* pada *email* yang digunakan untuk mendekripsi data dan file.



Gambar 4.23. Hasil dekripsi dari pesan enkripsi *ciphertext* ke data asli (*plaintext*)

**BAB V**

**PENUTUP**

* 1. **Kesimpulan**

Berdasarkan hasil pengujian dan analisis aplikasi pengamanan data pada *email* menggunakan algortima *ElGamal* berbasis web studi kasus Jurusan Teknologi Informasi Fakultas Teknik Universitas Tadulako, dapat disimpulkan bahwa :

1. Penelitian ini menghasilkan aplikasi yang dapat mengamankan data pada *email* dengan mengimplementasikan algoritma *ElGamal* yang berjalan dengan baik secara efektif dan efisien.

2. Pembangkitan kunci algoritma *ElGamal* pada nilai p, g, dan k ditentukan secara *random* agar memudahkan pengguna untuk mendapatkan kunci publik dan *private* dan pengguna tidak harus menentukan nilai pada masing-masing kunci. Kunci *private* yang digunakan untuk melakukan dekripsi terpadat pada akun *email* tujuan yang terkirim bersamaan dengan file yang telah terenkripsi (*ciphertext*).

3. Hasil enkripsi data dan file menggunakan algoritma *ElGamal* membuat ukuran data dan file bertambah menjadi lebih besar.

4. Aplikasi yang dibuat dapat digunakan untuk dokumen *office* dengan ekstensi .docx, .txt, .xlsx, dan PDF.

59

* 1. **Saran**

Saran untuk penelitian dan pengembangan aplikasi selanjutnya mengingat masih banyaknya hal-hal yang belum dapat diimplementasikan pada penelitian ini, maka penulis berharap pembaca dapat mengkaji lebih dalam tentang kriptografi *ElGamal* sehingga dapat mengembangkan aplikasi pengamanan data pada *email* ini ke dalam algoritma kriptografi yang lain diantaranya algoritma *RSA, AES, DES* dan lain-lain.

Pada aplikasi ini keamanan dari gmail sendiri masih harus di non-aktifkan secara manual agar akun gmail dapat diakses. Diharapkan pada penelitian selanjutnya aplikasi yang dibuat secara otomatis dapat mengakses akun gmail tanpa harus menonaktifkan keamanan akun gmail tersebut.

**DAFTAR PUSTAKA**

Al-Anshori dan Aribowo. 2014. *Implementasi Algoritma Kriptografi Kunci Publik Elgamal Untuk Proses Enkripsi Dan Dekripsi Guna Pengamanan File Data.* Yogyakarta : Universitas Ahmad Dahlan.

Angga dan Bunyamin. 2015. *Pengembangan Aplikasi Penjualan Dan Pembelian Bahan Bangunan Di Toko Bagja Jaya Menggunakan Metodologi Waterfall.* Jurnal Algoritma. Vol. 2012, No.1.

Basri. 2016. *Kriptografi Simetris Dan Asimetris Dalam Perspektif Keamanan Data Dan Kompleksitas Komputasi*. Jurnal Ilmiah Ilmu Komputer. Vol. 2, No. 2.

Bernadhed. 2013. *Sistem Informasi Pelayanan Produk Berbasis Vendor Berkart*. Yogyakarta: STMIK AMIKOM

Date. 2015. Pengertian Sistem Basis Data Menurut Para Ahli, <https://dosenit.com/kuliah-it/database/pengertian-sistem-basis-data-menurut-para-ahli>*,* diakses : 16 april 2017.

Enterprise. 2014*. MySQL Untuk Pemula*. Jakarta: Elex Media Komputindo.

Everest. 2015. Pengertian Sistem Basis Data Menurut Para Ahli. <https://dosenit.com/kuliah-it/database/pengertian-sistem-basis-data-menurut-para-ahli>, diakses : 16 april 2017.

Gumilang, S.G., 2016. Metode Penelitian Kualitatif Dalam Bidang Bimbingandan Konseling, *Jurnal Fokus Konseling,* Vol. 2, No.2, Kediri.

Iswandy & Eka, 2015. *Sistem Penunjang Keputusan Untuk Menentukan Penerimaan Dana Santunan Sosial Anak Nagari dan Penyaluranya Bagi Mahasiswa dan Pelajar Kurang Mampu di Kenagarian Barung-Barung Balantai Timur.* Jurnal TEKNOIF, Vol. 3, No. 2, Padang.

Kabetta.2017. *Analisis Kompleksitas Waktu Algoritma Kriptografi Elgamal dan Data Encryption Standard.* Jurnal Teknikom. Vol. 1, No. 1.

Karima et al. 2017. *Pemfaktoran Bilangan Prima pada Algoritme ElGamal untuk Keamanan Dokumen PDF.* Jurnal JNTETI. Vol. 6, No. 3.

Nugraha dan Mazharuddin. 2013. *Penyembunyian pesan rahasia yang terenkripsi menggunakan algoritma RSA pada media kompresi*. Jurnal Teknik Pomits. Vol. 2, No. 1.

61

Pabokory et al. 2015. *Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard.* Jurnal Informatika Mulawarman. Vol. 10, No.1.