RANCANG BANGUN APLIKASI PENGIRIMAN DOKUMEN MENGGUNAKAN KOMBINASI ALGORITMA RSA DAN ALGORITMA XOR

Leonardo Petra Refialy

Program Studi Informatika, Fakultas Ilmu Komputer.
Jln Ot Pattimaipauw RT.003/RW.003 Talake, Kel Wainitu, Nusaniwe, Kota Ambon, Maluku

E-mail : leo.refialy@gmail.com

Abstrak

Data atau dokumen digital dalam proses pertukaran dalam suatu jaringan, haruslah benar-benar aman, utuh, dan selalu terjaga kerahasiaannya, hal ini menjadi latar belakang perlu adanya suatu sistem yang dapat menjamin keamanan dan kerahasiaan data dan dokumen tersebut dalam proses transfer data. Penelitian ini membuat suatu rancang bangun program pengamanan data dalam pengiriman dokumen dengan menggunakan algoritma kriptografi RSA dan algoritma XOR. Algoritma RSA melibatkan dua buah kunci dalam melakukan enkripsi yaitu *public key* dan *private key*. *Public key* dapat disebarluaskan ke berbagai pihak untuk melakukan enkripsi ataupun dekripsi. Pesan yang sudah terenkripsi dengan *public key* hanya dapat didekripsi dengan menggunakan *private key*. Algoritma XOR suatu teknik kriptografi (penyandian) data yang menggunakan prinsip operasi logika XOR dalam proses enkripsi dan deskripsinya. Analisis Proses Enkripsi dan Deskripsi metode XOR Algoritma enkripsi menggunakan XOR adalah dengan meng-XOR-kan plainteks (P) dengan kunci (K) menghasilkan cipherteks. Dengan menggunakan metode prototype dalam melakukan tahapan penelitian, aplikasi yang dibangun sudah mengalami penyesuaian sesuai permintaan/masukan dari pemakai aplikasi. Aplikasi dapat berjalan dengan baik karena dapat melakukan proses enkripsi dan dekripsi dokumen dalam pengiriman email dari dalam aplikasi. Kemudian dari hasil pengujian diperoleh hasil semakin besar *bit strength* kunci, semakin besar waktu proses yang diperlukan untuk membuat kunci, kedua emakin besar ukuran *file*, semakin besar waktu proses yang diperlukan untuk enkripsi/dekripsi. Format *file* tidak mempengaruhi waktu proses, dan ketiga semakin besar *bit strength*, semakin cepat waktu enkripsi.
Kata kunci : Kriptografi, RSA, XOR, email, Data

Abstract

Digital data or documents in the exchange process in a network, must be completely safe, intact, and always kept confidential, this is the background for the need for a system that can guarantee the security and confidentiality of the data and documents in the data transfer process. This research makes a design of a data security program in sending documents using the RSA cryptographic algorithm and the XOR algorithm. The RSA algorithm involves two keys in encryption, namely the public key and the private key. The public key can be distributed to various parties to perform encryption or decryption. Messages that have been encrypted with the public key can only be decrypted using the private key. XOR algorithm is a cryptographic technique (encoding) data that uses the principle of XOR logic in the encryption and description process. Analysis of the Encryption Process and Description of the XOR method The encryption algorithm using XOR is to XOR plaintext (P) with key (K) to produce ciphertext. By using the prototype method in carrying out the research stages, the applications built have undergone adjustments according to requests/input from application users. The application can run well because it can encrypt and decrypt documents in sending email from within the application. Then from the test results obtained the greater the bit strength of the key, the greater the processing time required to generate the key, secondly the larger the file size, the greater the processing time required for encryption/decryption. The file format does not affect the processing time, and thirdly, the greater the bit strength, the faster the encryption time.
Keywords : *Cryptography, RSA, XOR, email, Data*

1. PENDAHULUAN

Dalam perkembangan ilmu pengetahuan dan teknologi yang semakin maju, semakin banyak teknologi dalam menunjang proses pertukaran data/informasi secara elektronik. Hal tersebut tak terlepas dari faktor yang mengancam seperti pencurian data ataupun file/dokumen yang terkadang sangat rahasia dan sangat mahal nilainya.

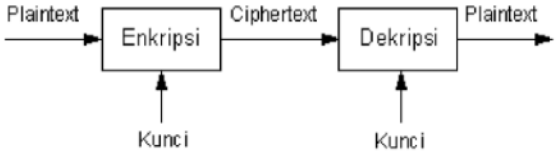
Data atau dokumen digital dalam proses pertukaran dalam suatu jaringan, haruslah benar-benar aman, utuh, dan selalu terjaga kerahasiaannya, hal ini menjadi latar belakang perlu adanya suatu sistem yang dapat menjamin keamanan dan kerahasiaan data dan dokumen tersebut dalam proses transfer data.

Tujuan dari penelitian ini adalah membuat suatu rancang bangun program pengaman data dalam aplikasi pengiriman dokumen dengan menggunakan algoritma kriptografi RSA dan algoritma XOR untuk mengamankan data.

Dengan adanya penelitian ini dapat memberikan manfaat kepada pihak-pihak terkait yang terlibat dalam proses pertukaran data/dokumen digital agar dapat menjaga keamanan dan kerahasiaan data/dokumen.

Kriptografi merupakan ilmu yang ilmu dan seni yang dipakai dalam menjaga keamanan pesan yang dikirim dari suatu tempat ke tempat lain (1). Dalam kriptografi teks terang atau cleartext atau plaintext merupakan pesan asli akan di proses sedemikian rupa menjadi ciphertext (teks buram) melalui suatu proses yang di namakan proses Enkripsi (1). Untuk pesan rahasia tersebut dapat diketahui oleh pihak yang sah maka akan dilakukan proses mengubah ciphertext (Teks Kabur) menjadi plaintext (Teks Terang) yang disebut proses dekripsi, kedua proses enkripsi dan dekripsi membutuhkan penggunaan sejumlah informasi rahasia yang disebut kunci (key)(1).

Untuk skema proses dari proses enkripsi dan dekripsi yang melibatkan kunci (key) dapat dilihat pada gambar 1 dibawah ini.



Gambar 1. Model Skema proses enkripsi dan dekripsi

Algoritma RSA merupakan Algoritma kriptografi kunci publik (asimetris) yang menggunakan 2 angka (e dan d) sebagai kunci publik dan kunci privat. Pada algoritma RSA e dan n diumumkan untuk umum sedangkan d dirahasiakan. RSA adalah algoritma public key encryption yang pertama kali dipublikasikan tahun 1977 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman di MIT (Massachusetts Institute of Technology). Algoritma RSA melibatkan dua buah kunci dalam melakukan enkripsi yaitu public key dan private key. Public key dapat disebarluaskan ke berbagai pihak untuk melakukan enkripsi ataupun dekripsi. Pesan yang sudah terenkripsi dengan public key hanya dapat didekripsi dengan menggunakan private key (2).

Algoritma XOR suatu teknik kriptografi (penyandian) data yang menggunakan prinsip operasi logika XOR dalam proses enkripsi dan deskripsinya (3). Analisis Proses Enkripsi dan Deskripsi metode XOR Algoritma enkripsi menggunakan XOR adalah dengan meng-XOR-kan plainteks (P) dengan kunci (K) menghasilkan cipherteks (4)

(C) : C= P □□K Algoritma dekripsi menggunakan XOR adalah dengan meng-XOR-kan ciphertext (C) dengan kunci (K) menghasilkan plainteks

(P) : P=C □□K.

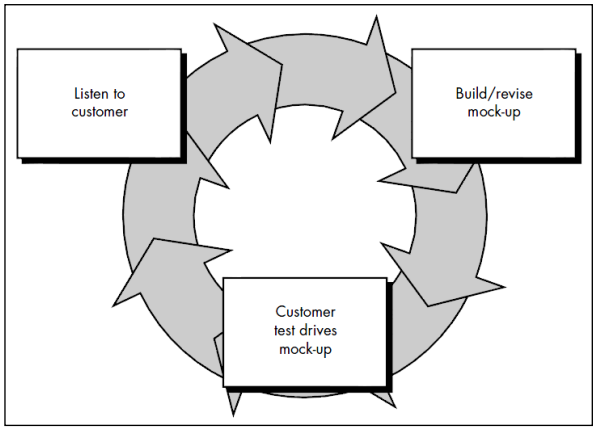
Pada penelitian ginting, dkk (5) merancang email client dengan enkripsi dan dekripsi menggunakan algoritma RSA dengan menggunakan plaintext (text) bukan mengggunakan attachment atau lampiran menghasilkan aplikasi pengamanan email yang berjalan dengan baik dan semestinya. Pada penelitian yang lain oleh Rahaijoningroem (6) melakukan penerapan pengamanan data transkrip mahasiswa dengan menggunakan algoritma RSA dalam hal ini yaitu proses enkripsi, dekripsi dan proses pembangkitan kunci dalam penelitian tersebut melihat kinerja yang diukur dengan waktu komputasi serta kompleksitas memori yang dibutuhkan dalam melakukan enkripsi dan dekripsi data. Selanjutnya pada penelitian oleh andryanto (3) mengimplementasi algoritma XOR dalam melakukan pengamanan data guru, hasil yang didapat yaitu aplikasi pengamanan data dapat melakukan fungsi enkripsi dan dekripsi dengan baik. Pada penelitian Gunawan (7) menggunakan kombinasi caesar cipher dan algortima RSA untuk pengamanan file yang mana dengan menggunakan kombinasi kedua algoritma tersebut tingkat keamanan pesan lebih baik daripada

hanya menggunakan satu algoritma saja. Berdasarkan penelitian-penelitian sebelumnya, penelitian yang dilakukan oleh penulis mencoba membangun suatu sistem pengamanan email dengan menggunakan aplikasi email client yang didalamnya melibatkan algoritma RSA serta XOR dalam enkripsi, dekripsi serta pembangkitan kunci. Aplikasi pengamanan dokumen ini bukan hanya melakukan pengaman data text saja seperti pada penelitian ginting (5) tetapi dalam aplikasi ini sudah dapat mengamankan data dalam bentuk attachment atau lampiran, serta proses pengamanan dokumen menggunakan kombinasi antara pengamanan dokumen RSA seperti pada penelitian (6) dan algoritma XOR pada penelitian Azis N (3) hal itu diperkuat oleh penelitian gunawan yang membuktikan algoritma RSA akan mempunyai tingkat keamanan yang lebih baik jika dikombinasikan dengan algoritma pengaman lain (7).

2. METODOLOGI

2.1 Metode Penelitian

Metode Penelitian yang dipakai dalam penelitian ini adalah Prototype Model (8).



Gambar 2. Prototype Model (8)

Tahap-tahap dalam *Prototype Model* adalah sebagai berikut:

1. ***Listen to Costumer***
   Pada tahap ini dilakukan analisis terhadap permasalahan yang ada, yaitu mendapatkan data dan literatur yang terkait dengan proses , enkripsi, dekripsi menggunakan algoritma RSA dan XOR; melalui dokumen dan referensi yang ada.
2. ***Build***
   Selanjutnya setelah memperoleh data dan mengetahui proses enkripsi dan dekripsi dengan algoritma RSA dan XOR, langkah berikutnya adalah membuat perancangan dengan menggunakan *Unified Modeling Language* (UML) mengenai sistem yang akan dibangun nantinya. Selain itu dilakukan pula perancangan pada *user interface* berupa *prototype* sistem.
3. ***Costumer Test***
   Pada Tahap ini dilakukan pengujian sistem, yaitu menjalankan proses implementasi sistem, dengan menguji proses enkripsi dan dekripsi pengiriman *E-mail*, serta melihat hasil yang diberikan apakah sudah sesuai dengan konsep RSA dan XOR.

3. HASIL DAN PEMBAHASAN

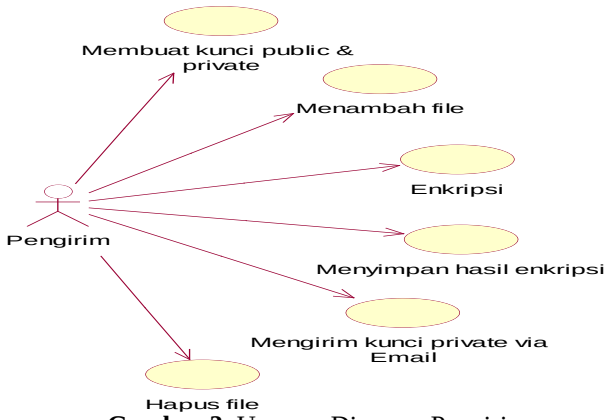
Bagian ini berisi hasil-hasil penelitian dan uraian teoritik yang dapat disajikan dalam bentuk tabel,

3.1 Hasil dan Pembahasan

3.1.1 Rancangan Sistem

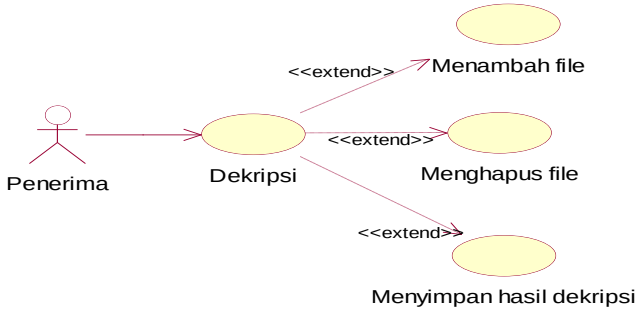
Dalam perancangan sistem aplikasi, digunakan UML (**Unified Modelling Language**) yaitu sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (9).

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem yang dibangun. Sebuah use case merepresentasikan sebuah interaksi aktor atau pengguna dengan sistem yang digunakan.



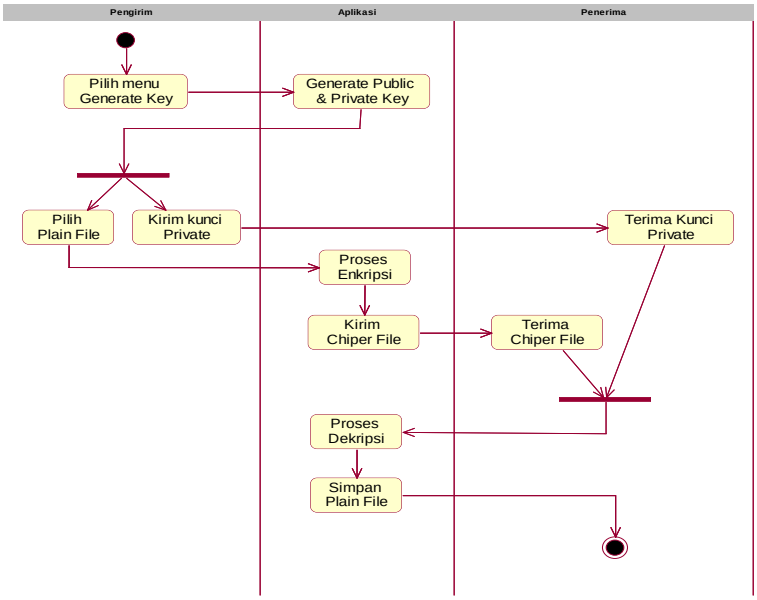
Gambar 3. Usecase Diagram Pengirim

Untuk menjalankan aplikasi RSA, pengirim dapat melakukan proses eknripsi yang diawali dengan membuat kunci public dan private. Setelah itu pengirim dapat menambah file yang akan dienkripsi, kemudian enkripsi file dimana data-data file akan diamankan. Setelah proses enkripsi selesai, pengirim dapat menyimpan file hasil enkripsi



Gambar 4. Usecase Diagram Penerima

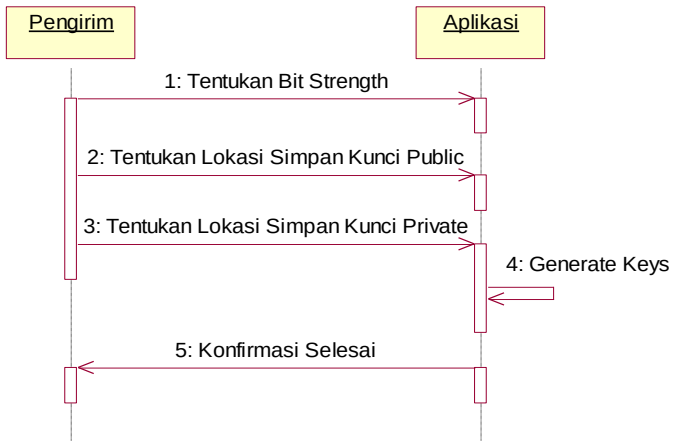
penerima dapat melakukan proses dekripsi, dimana dalam proses dekripsi penerima dapat melakukan 3 aktifitas yaitu menambah file, menghapus file dan menyimpan file hasil dekripsi. Activity diagram menggambarkan proses-proses yang terjadi mulai aktivitas dimulai sampai aktifitas berhenti.



Gambar 5. Activity Diagram Rancang Bangun Program RSA

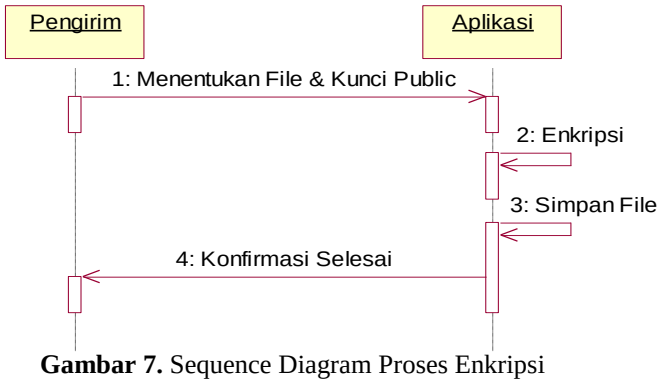
Activity diagram aplikasi RSA, pada saat aplikasi dijalankan, pengirim dapat memilih menu generate key, selanjutnya aplikasi akan melakukan proses generate public dan private key. Setelah proses tersebut selesai, maka pengirim dapat melakukan proses pengiriman private key kepada penerima ataupun memilih plain file. Apabila pengirim memilih plain file, maka aplikasi akan melakukan proses enkripsi kemudian aplikasi akan mengirimkan chipper file kepada penerima. Apabila penerima telah menerima private key dan chipper file, Selanjutnya aplikasi dapat melakukan proses pengiriman cipher text kepada penerima. Penerima menerima cipher key yang telah dikirim. Aplikasi melanjutkan dengan melakukan proses dekripsi. Selanjutnya aplikasi melakukan proses penyimpanan plain file.

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).



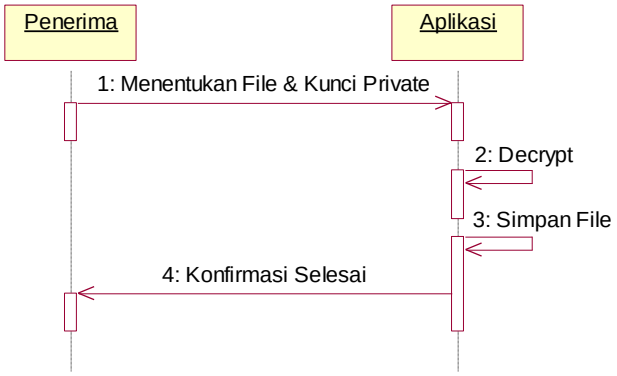
Gambar 6. Sequence Diagram Pembuatan Kunci

Sequence diagram untuk proses pembuatan kunci pada aplikasi RSA untuk mengamankan E-mail. Untuk membuat kunci, penerima diminta untuk menentukan Bit Strength kemudian menentukan lokasi penyimpanan public key. Setelah itu, penerima harus menentukan lokasi penyimpanan private key. Setelah lokasi penyimpanan kunci telah ditentukan, selanjutnya aplikasi akan melakukan proses generate keys.



Gambar 7. Sequence Diagram Proses Enkripsi

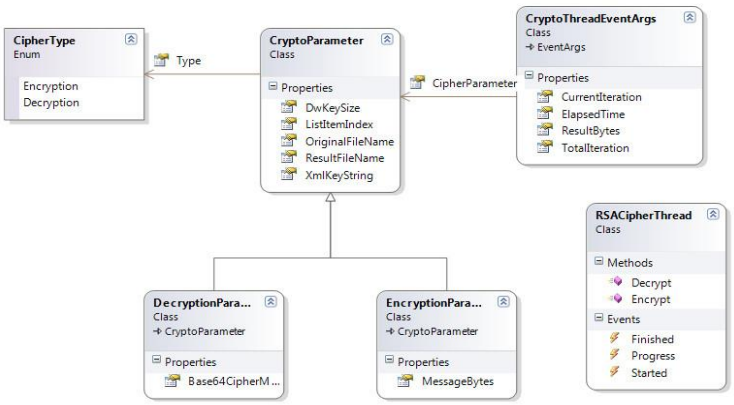
Sequence diagram untuk proses enkripsi. Tahap pertama, pengirim diminta untuk menambahkan file yang akan dienkripsi kemudian pengirim diminta untuk memilih file dan kunci public. Setelah itu, aplikasi akan melakukan proses enkripsi untuk mengenksripsi file yang telah ditambahkan tadi. Kemudian pengirim diminta untuk menentukan lokasi penyimpanan hasil enkripsi. Apabila telah selesai, aplikasi akan memberikan konfirmasi kepada pengirim.



Gambar 8. Sequence Diagram Proses Deskripsi

Sequence diagram untuk proses dekripsi. Tahap pertama, penerima diminta untuk menambahkan file yang akan didekripsi kemudian penerima diminta untuk menentukan file dan kunci private untuk mendekripsi file yang telah ditambahkan. Setelah itu, penerima diminta untuk menentukan lokasi penyimpanan hasil dekripsi. Apabila proses tersebut telah selesai, maka aplikasi akan memberikan konfirmasi kepada penerima.

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain beriontasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.



Gambar 9. Class Diagram Enkripsi Dekripsi

3.2 Implementasi dan Pembahasan

3.2.1 Antarmuka untuk Proses Pembuatan Kunci

User1 (pengirim) akan membuat public key dan private key yang diperlukan dalam proses enkripsi dan dekripsi. User1 (pengirim) membuat pasangan kunci dengan memilih tombol keys pada program.

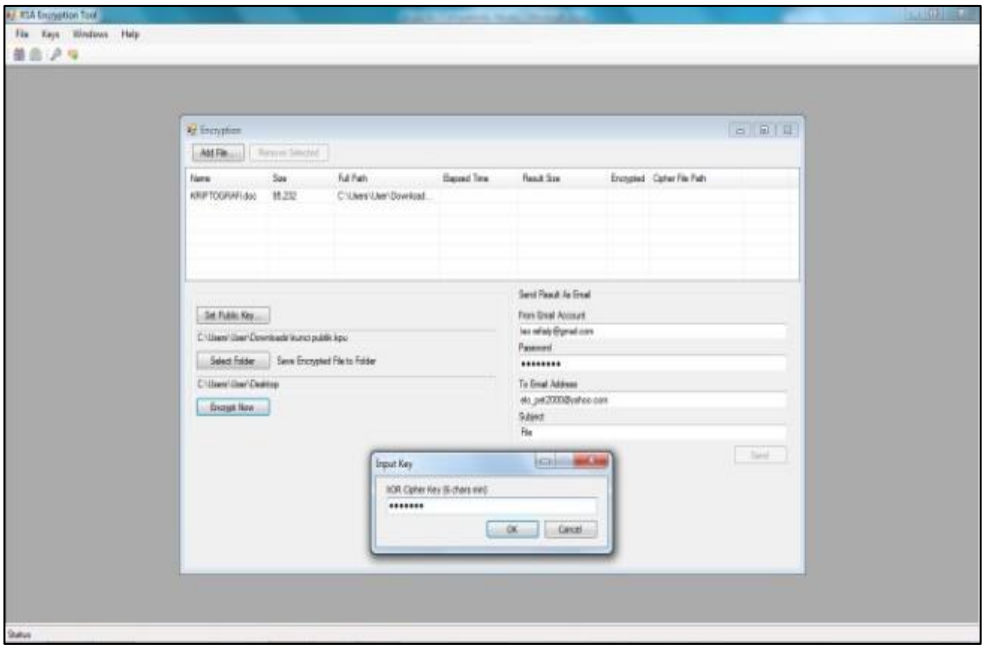
Dalam membuat pasangan kunci, akan memerlukan inputan user1 (pengirim) untuk membuat public key dan private key, dengan memilih tombol yang disediakan pada program. Setelah proses input untuk public key dan private key, user lalu memilih tombol generate untuk proses generate pasangan public key dan private key yang sudah dibuat.

Kode Program 1 Perintah untuk Proses Pembuatan Kunci

```
        for (int i = 0; i <= iterations; i++)
            {
                byte[] tempBytes = new byte[(dataLength - maxLength * i > maxLength) ? maxLength : dataLength - maxLength *
i];
                Buffer.BlockCopy(messageBytes, maxLength * i, tempBytes, 0, tempBytes.Length);
                byte[] encryptedBytes = rsaCryptoServiceProvider.Encrypt(tempBytes, true);

                Array.Reverse(encryptedBytes);
                string base64String = Convert.ToBase64String(encryptedBytes);
                resultBytes.AddRange(Encoding.Default.GetBytes(base64String));
            }
}
```

3.2.2 Antarmuka untuk Proses Enkripsi

Untuk dapat menjalankan proses enkripsi, user1 (pengirim) harus membuka file yang akan dienkripsi terlebih dahulu dengan memilih tombol add file. Setelah melakukan proses add file (plaintext) yang akan dienkripsi maka user1 (pengirim) sudah dapat melakukan enkripsi pada plaintext yang telah di add.



Gambar 10. Antarmuka proses Enkripsi

Gambar 9 menunjukkan proses enkripsi. File yang akan dienkripsi dalam kriptosistem yang dibangun adalah file dokumen word (.doc, .docx), excel (.xls, .xlsx), compressed file (.zip, .rar), PDF file (.pdf), image (.jpeg, .gif, .png), text file (.txt). User1 harus memilih tombol set public key untuk memanggil kunci public yang telah dibuat sebelumnya pada proses generate keys. Setelah itu, user1 (pengirim) harus memilih tombol select folder sebagai tempat atau lokasi penyimpanan file yang telah dienkripsi menjadi ciphertext. User lalu melakukan memilih tombol encrypt now untuk melakukan menjalankan enkripsi file (plaintext). Dalam prosesnya, enkripsi file melalui 2 tahapan enkripsi, yaitu dengan algoritma RSA yang terlebih dahulu dengan enkripsi dengan algoritma XOR.

Kode Program 2 Perintah untuk Proses Enkrpsi File

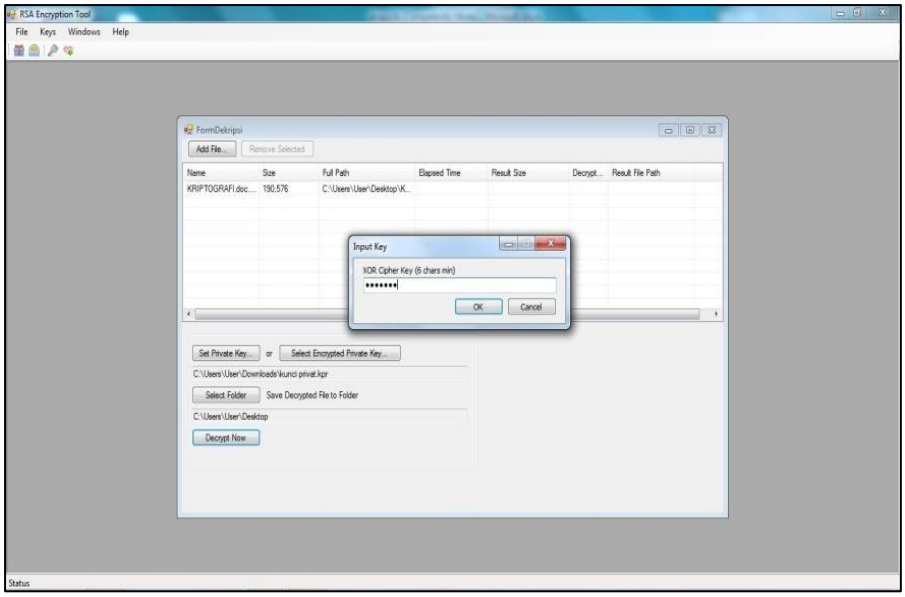
```
public void Encrypt(Object o)
{
    EncryptionParameter
       param = (EncryptionParameter)o;
    param.Type = CipherType.Encryption;

    int dwKeySize = param.DwKeySize;
    string xmlString = param.XmlKeyString;

    RSACryptoServiceProvider rsaCryptoServiceProvider
    = new RSACryptoServiceProvider(dwKeySize);
    rsaCryptoServiceProvider.FromXmlString(xmlString);
    int keySize = dwKeySize / 8;
    byte[] messageBytes = param.MessageBytes;

 #region . XOR .                    [17]
    messageBytes = XORCipher.XOR(messageBytes, Encoding.Default.GetBytes(param.XORKey));
    File.WriteAllBytes(param.ResultFileName + ".xor", messageBytes);
    #endregion                      [22]

    List<byte> resultBytes = new List<byte>();
    int maxLength = keySize - 42;
    int dataLength = messageBytes.Length;
    int iterations = dataLength / maxLength;
```

file ciphertext yang telah melalui tahapan enkripsi akan dikirim oleh User1 (pengirim) kepada User2 (penerima) melalui aplikasi RSA.

User1 (Pengirim) akan mengirim kunci private kepada user2 (Penerima) yang nantinya akan digunakan untuk proses dekripsi file. Kunci private yang dikirim akan terlebih dahulu dienkripsi dengan algoritma XOR. User memilih tombol keys lalu pilih send private keys atau langsung memilih tombol shorcut send private keys. User1 (pengirim) akan diminta untuk memasukkan kunci private yang akan dikirim, E-mail, password E-mail. Alamat E-mail tujuan beserta subject E-mail. Kemudian user1 (pengirim) lalu memilih tombol send, akan muncul tampilan untuk input key untuk algoritma XOR. Setelah proses input XOR key, user lalu memilih tombol ok untuk melanjutkan proses sending private key.

Antarmuka untuk Proses Dekripsi File

Pada tahapan proses dekripsi file, user2 (Penerima) yang telah menerima ciphertext atau file enkripsi yang dikirim oleh user1 (pengirim) akan melakukan proses dekripsi untuk melihat isi file enkripsi tersebut. Ciphertext akan didekripsi dengan menggunakan kunci private (private key) yang juga telah dikirim oleh user1 (penerima). Kunci private yang dikirim oleh user1 (pengirim) akan didekripsi terlebih dahulu dengan algoritma xor. user2 (penerima) memilih tombol add file. Setelah melakukan proses add file yang akan didekripsi, berikut user2 (penerima) melakukan dekripsi pada ciphertext yang telah di add terlebih dahulu, dalam proses dekripsinya, user2 akan memilih tombol set encrypt private key untuk memanggil kunci private yang terenkripsi yang telah dikirim oleh user1. User2 memilih tombol select folder sebagai tempat atau lokasi penyimpanan file yang telah dienkripsi menjadi ciphertext. User2 lalu melakukan memilih tombol decrypt now untuk melakukan menjalankan dekripsi ciphertext. Dalam prosesnya, enkripsi ciphertext melalui 2 tahapan enkripsi, yaitu dengan algoritma RSA yang terlebih dahulu dengan enkripsi dengan algoritma XOR. Algoritma XOR.



Gambar 11. Antarmuka Proses Dekripsi

3.3Evaluasi Prototype System

Evaluasi prototype sistem merupakan tahap terakhir pada metode pengembangan sistem yang digunakan, yaitu mengevaluasi apakah prototype sistem yang telah dibangun sesuai dengan kebutuhan. Jika masih kurang atau belum sesuai, maka proses prototyping modelakan berulang lagi yang dimulai dengan pengumpulan kebutuhan, dilanjutkan perancangan, kemudian evaluasi prototype. Namun jika telah sesuai maka proses prototyping selesai.

Pada prototype pertama, dirancang aplikasi hanya menggunakan algoritma kriptografi RSA dalam kriptosistem yang dibangun. Kemudian prototype tersebut dievaluasi oleh customer, dalam hal ini adalah pembimbing. Customer memberikan masukan atau saran untuk menambahkan suatu algoritma kriptografi dalam proses enkripsi dekripsi pada aplikasi tersebut. Berdasarkan hasil evaluasi pada prototype pertama, maka dirancang prototype kedua.

Pada prototype kedua, rancangan aplikasi kriptosistem yang dibangun ditambahkan penambahan algoritma kriptografi yaitu algoritma XOR. Proses enkripsi dekripsi dikerjakan terlebih dahulu dengan algoritma XOR, selanjutnya dilanjutkan dengan proses enkripsi, costumer memberi masukan tersebut dengan tujuan membuat tingkat keamanan proses enkripsi lebih rumit, dengan adanya double encryption atau proses enkripsi ganda yaitu dengan mengkombinasikan algoritma kriptografi XOR dan RSA. Kemudian prototype kedua dievaluasi kembali oleh customer. Customer memberikan masukan atau saran untuk pengiriman kunci privat oleh aplikasi dengan aman. Berdasarkan hasil evaluasi pada prototype kedua, maka dirancang prototype ketiga.

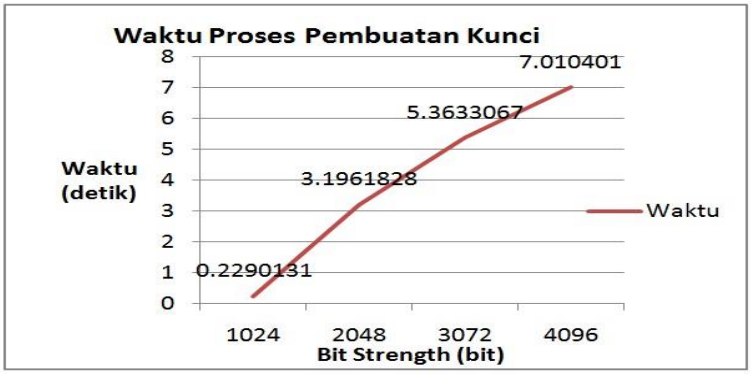
Pada prototype ketiga, rancangan aplikasi kriptosistem yang dibangun diubah pada proses pengiriman kunci privat, dibuat enkripsi kunci dengan menggunakan algoritma XOR, dan proses pengiriman kunci privat dibuat terpisah dengan pengiriman dokumen atau data hasil enkripsi. Kemudian prototype ketiga dievaluasi kembali oleh customer, dan dinyatakan bahwa prototype ketiga telah sesuai dengan kebutuhan customer. Kemudian dilanjutkan dengan melengkapi dan membenahi aplikasi yang dibuat

3.4 Pengujian Aplikasi

Setelah aplikasi selesai dibuat, dilakukan pengujian untuk menguji kerja program, proses pengujian dimulai dengan menguji perbandingan panjang kunci (bitstrength) dengan waktu generate key.

Perbandingan panjang kunci (bitstrength) dengan waktu generate key.

Berdasarkan hasil pengujian perbandingan panjang kunci (bitstrength) dengan waktu generate key pada tabel 1, menunjukkan bahwa semakin besar bit strength kunci, semakin besar juga waktu untuk proses generate key.



Gambar 12. Pengujian panjang kunci (bitstrength) dengan waktu generate key

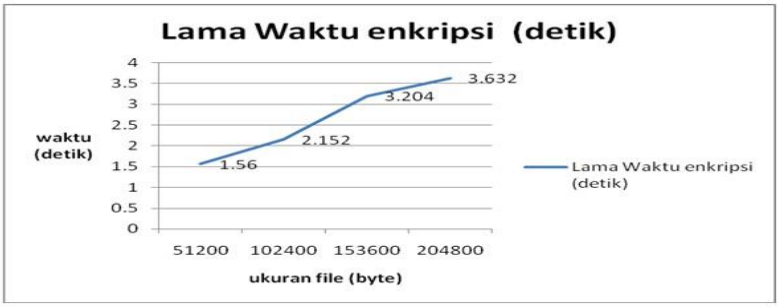
Tabel 1. Pengujian panjang kunci (bitstrength) dengan waktu generate key.

No.	Bitstrength (bit)	Waktu (Second)
1.	1024	0.229013
2.	2048	3.196183
3.	3072	5.363307
4.	4096	7.010401

Tabel 2. Pengujian enkrpsi ukuran file format.txt terhadap waktu

Nama File	Format File	Panjang Kunci	Ukuran Plaintext (byte)	Ukuran Ciphertext (byte)	Lama Waktu dekripsi (detik)	Rasio Ukuran File
kripto1	Txt	1024	102512	51200	1.444	49.945%
kripto2	Txt	1024	204852	102400	2.597	49.987%
kripto3	Txt	1024	307364	153600	3.653	49.973%
kripto4	Txt	1024	409704	204800	4.572	49.987%

Berdasarkan hasil pengujian kriptosistem pada data dengan format .txt dengan ukuran file berbeda dan panjang kunci sama pada tabel 2, menunjukkan bahwa ukuran file, dapat mempengaruhi waktu proses kriptosistem. Semakin besar ukuran plaintext maka semakin lama waktu proses enkripsi.



Gambar 13. Pengujian enkripsi ukuran file format.txt terhadap waktu.

4. PENUTUP

Dari pegujian yang dilakukan didapat hasil yaitu 3 catatan pangujian sistem. Pada pengujian 1 tentang Waktu Proses Pembuatan Kunci didapat kesimpulan semakin besar bit strength kunci, semakin besar waktu proses yang diperlukan untuk membuat kunci. Hal itu dikarenakan semakin besar bit strength maka semakin

besar pula bilangan prima yang harus dibangkitkan. Pengujian 2 tentang pengaruh ukuran file terhadap waktu proses enkripsi/dekripsi didapat kesimpulan semakin besar ukuran file, semakin besar waktu proses yang diperlukan untuk enkripsi/dekripsi. Format file tidak mempengaruhi waktu proses. Pengujian 3 tentang pengaruh ukuran bit strength kunci terhadap waktu proses enkripsi, didapat kesimpulan semakin besar bit strength, semakin cepat waktu enkripsi. Hal itu dikarenakan nilai bit strength merupakan faktor pembagi dalam proses enkripsi sehingga semakin besar nilai bit strength, semakin sedikit perulangan yang dilakukan dalam proses enkripsi. Adapun saran yang dapat diberikan untuk pengembangan aplikasi ini lebih lanjut adalah Penelitian yang dilakukan masih bersifat umum sehingga untuk ke depannya, penelitian ini bisa dilanjutkan dan difokuskan ke masalah-masalah kriptografi yang lebih khusus dengan suatu studi kasus yang lebih spesifik.

DAFTAR PUSTAKA

[1] Mukhtar H. Kriptografi Untuk Keamanan Data. Deepublish; 2018.
[2] Ashioba NC, Yoro RE. RSA Cryptosystem using Object-Oriented Modeling Technique. Int J Inf Commun Technol Res. 2014;4(2):57–61.
[3] N. Aziz, "Perancangan Aplikasi Enkripsi Deskripsi Menggunakan Metode Caesar Shiper dan Operasi XOR," *Ikraith-Informatika*, vol. 2(2), no. 1, hal. 72–80, 2018.
[4] Suhardi S. Aplikasi Kriptografi Data Sederhana dengan Metode Exlusive-or (Xor). J Teknovasi J Tek dan Inov. 2018;3(2):23–31.
[5] Ginting A, Isnanto RR, Windasari IP. Implementasi Algoritma Kriptografi RSA untuk Enkripsi dan Dekripsi Email. J Teknol dan Sist Komput. 2015;3(2):253–8.
[6] Rahajoeningroem T, Aria M. Studi Dan Implementasi Algoritma Rsa Untuk Pengamanan Data Transkrip Akademik Mahasiswa. Maj Ilm Unikom. 2011;
[7] Gunawan I. Kombinasi algoritma Caesar cipher dan algoritma RSA untuk pengamanan file dokumen dan pesan teks. InfoTekJar J Nas Inform dan Teknol Jar. 2018;2(2):124–9.
[8] Khosrow-Pour DBA. Encyclopedia of information science and technology. IGI Global; 2005.
[9] Fauzan R, Siahaan D, Rochimah S, Triandini E. Use case diagram similarity measurement: A new approach. In: 2019 12th International Conference on Information & Communication Technology and System (ICTS). IEEE; 2019. hal. 3–7.