PERANCANGAN SISTEM KRIPTOGRAFI PADA DOCUMENT MENGGUNAKAN ALGORITMA TRIPLE DES DAN RSA

[1]Klara Henni Kartika, [2]Alfred Yulius A.P, [3]Sandi Tendean
[1,2,3]Teknik Informatika, STMIK Widya Dharma, Pontianak
e-mail: [1]klarahennikartika@gmail.com, [2]alfredyulius703@gmail.com, [3]sanditendean@gmail.com

Abstract

The development of cryptographic technology continues to grow in the presence of a cryptographic system that is able to hide messages in order to avoid the threat of cybercrime. The importance of securing confidential documents motivates writers to lift the title "Cryptography System design On Documents Using Triple DES and RSA Algorithm". The design of the cryptographic system aims to make it easy for users to secure data documents with effective, efficient and productive. This research uses descriptive and experimental research design. Experiment is to do the design and implementation of encryption and decryption systems as a description of the problem as well as the design of descriptive research that describes the problem by collecting written as literature references such as books, journals, sources from the internet and other forms of writing that deals with the issues raised. The analysis technique used systems using system modeling tool Unified Modeling Language (UML) to describe cryptographic encryption and decryption system is built. This cryptographic applications designed using the programming language Visual Basic .Net 2010. This study was conducted to produce a draft of cryptographic software that makes it easy for users to perform secure data that is personal and public. The conclusion from the analysis is a system of cryptographic encryption and decryption can help the user to maintain the confidentiality of the document, document integrity, non-repudiation and authentication. Advice can be given that the author is still the need for modifications to the display application and development of the design of cryptographic systems in order to attract the user wishes to use this cryptographic application program.

Keywords*: system design, Cryptography, Document, Triple DES, RSA.*

Abstrak

Perkembangan teknologi kriptografi terus berkembang dengan adanya sistem kriptografi yang mampu menyembunyikan pesan agar terhindar dari ancaman cybercrime. Pentingnya mengamankan dokumen rahasia memotivasi penulis untuk mengangkat judul "Perancangan sistem Kriptografi Pada Dokumen Menggunakan Algoritma Triple DES dan RSA". Rancangan sistem kriptografi bertujuan memberikan kemudahan bagi pengguna untuk mengamankan dokumen dengan efektif, efisien dan produktif. Penelitian menggunakan desain penelitian deskriptif dan ekperimen. Eksperimen yaitu melakukan perancangan dan implementasi sistem enkripsi dan dekripsi sebagai gambaran masalah serta desain penelitian deskriptif yaitu mendeskripsikan masalah dengan mengumpulkan referensi tertulis seperti buku, jurnal, sumber-sumber dari internet maupun bentuk tulisan lain yang berkaitan dengan masalah yang diangkat. Teknik analisis sistem yang digunakan menggunakan alat pemodelan sistem Unified Modeling Language (UML) untuk menggambarkan sistem kriptografi enkripsi dan dekripsi yang dibangun. Aplikasi kriptografi dirancang menggunakan bahasa pemrograman Visual Basic .Net 2010. Penelitian dilakukan untuk menghasilkan rancangan perangkat lunak kriptografi yang memberikan kemudahan bagi pengguna dalam melakukan pengamanan data peribadi maupun umum. Kesimpulan dari hasil analisis adalah sistem kriptografi enkripsi dan dekripsi membantu pengguna untuk menjaga kerahasiaan dokumen, integritas dokumen, non-repudiation dan authentication. Saran yang dapat penulis berikan yaitu masih perlunya modifikasi pada tampilan aplikasi dan pengembangan rancangan sistem kriptografi guna menarik keinginan pengguna untuk menggunakan program aplikasi kriptografi.

Kata Kunci: Perancangan sistem, Kriptografi, Dokumen, Triple DES, RSA

1. PENDAHULUAN

Mengingat perkembangan teknologi informasi yang semakin pesat, banyak diciptakan produk-produk canggih yang mampu menangani berbagai masalah bagi pengguna. Produk canggih tersebut dirancang sedemikian rupa untuk membantu kinerja pengguna sehingga diperoleh hasil yang lebih efektif, efisien dan produktif dalam penyelesaian masalah.

Menggunakan teknologi yang serba canggih, bukan berarti tidak ada gangguan yang menghambat kinerja pengguna. Mendengar informasi yang saat ini sedang hangat diperbincangkan yaitu *cybercrime* atau yang lebih dikenal dengan kejahatan komputer. Maka dapat disimpulkan bahwa keamanan data terutama data rahasia pengguna sangat rentan untuk dicuri atau dirusak yang tidak hanya sebatas terjadi di Indonesia melainkan juga di negara lain.

Pentingnya dilakukan pengamanan terhadap data pribadi, membutuhkan suatu sistem kriptografi yang mampu menyembunyikan suatu pesan agar terhindar dari pengguna yang tidak berhak untuk mengakses pesan atau data tersebut. Pada umumnya kriptografi terdiri atas kriptografi klasik dan kriptografi modern. Sebelum komputer ada, kriptografi dilakukan dengan pensil dan kertas. Algoritma kriptografi klasik melakukan enkripsi dan dekripsi karakter per karakter. Semua algoritma kriptografi klasik termasuk ke dalam sistem kriptografi simetri. Sedangkan kriptografi modern beroperasi pada *string biner*. Kriptografi modern dipicu oleh perkembangan peralatan komputer digital. Dengan komputer digital, *cipher* yang lebih kompleks menjadi sangat mungkin untuk dihasilkan. *Cipher* yang kompleks seperti *Triple Data Encryption Standard* (DES) dan *Rivest Shamir Adleman* (RSA) adalah algoritma kriptografi modern. Di sisi lain, ada puluhan algoritma kriptografi modern yang termasuk ke dalam sistem kriptografi kunci-simetri, salah satunya adalah *Triple*-DES. *Triple*-DES mempunyai algoritma kriptografi yang termasuk ke dalam sistem kriptografi simetri sehingga berbeda dengan RSA. Algoritma RSA merupakan sistem kriptografi kunci-nirsimetri. Kedua metode ini dapat diterapkan pada proses enkripsi dan dekripsi pesan atau data yang akan diamankan.

Triple data encryption standard merupakan pengembangan dari algoritma DES yang paling banyak dipakai di dunia, di adopsi oleh *Nasional Institue of Standards and Technology* (NIST) sebagai standar pengolahan informasi Federal AS. Sedangkan *Rivest Shamir Adleman* merupakan metode dari nama pembuat algoritma kriptografi kunci-publik yang paling populer dan dianggap aman karena algoritma ini mampu melakukan pemfaktoran bilangan yang sangat besar.

Salah satu alternatif untuk mengamankan pesan pribadi tersebut yaitu dengan merancang sebuah sistem yang dapat melakukan enkripsi dan dekripsi dengan mudah. Rancangan sistem kriptografi ini diharapkan mampu menjaga *convidentiality* (kerahasiaan), *data integrity* (perubahan data), *non-repudiation* (membuktikan suatu dokumen dikirim oleh orang yang benar), *authentication* (keaslian suatu pesan dan menguji identitas seseorang) serta mengurangi kasus *cybercrime*.

2. METODE PENELITIAN

2.1 Bentuk penelitian dan teknik pengumpulan data yang digunakan adalah:

2.1.1 Rancangan Penelitian

Pada penelitian ini, penulis menggunakan desain penelitian deskriptif dan perancangan eksperimen. Eksperimen dilakukan dengan perancangan dan implementasi sistem sebagai gambaran yang jelas dari masalah. Desain penelitian ini akan memberikan gambaran sistem kriptografi enkripsi dan dekripsi.

2.1.2 Metode Pengumpulan Data

Penulis menggunakan studi literatur dan teknik observasi. Studi literatur merupakan studi yang dapat digunakan sebagai bahan referensi tertulis untuk mengumpulkan data dengan membaca berbagai literatur seperti buku, skripsi, jurnal, sumber-sumber dari internet maupun bentuk tulisan lain yang berkaitan dengan penggunaan algoritma *Triple* DES dan RSA pada perancangan kriptografi enkripsi dan dekripsi. Teknik observasi dilakukan dengan cara menguji hasil dari permasalahan dengan mencari banyak referensi yang tepat untuk penulisan.

2.1.3 Teknik Analisis Sistem

Teknik analisis sistem yang digunakan menggunakan alat pemodelan sistem *Unified Modeling Language* (UML). *Unified Modeling Language* untuk menentukan, memvisualisasikan, membangun dan mendokumentasikan sistem kriptografi enkripsi dan dekripsi.

2.1.4 Teknik Perancangan Sistem

Perancangan sistem menggunakan bahasa pemrograman *Visual Basic .Net 2010* dan teknik berorientasi objek.

2.2 Landasan Teori

2.2.1 Perancangan Sistem

Perancangan masukan merupakan suatu aktivitas merancang komponen-komponen yang berfungsi untuk menerima semua data dari pengguna. Perancangan keluaran merupakan suatu kegiatan merancang komponen-komponen yang berfungsi untuk menyajikan hasil akhir ke pengguna sistem informasi. [1] Dengan demikian, secara lebih terperinci analisis/perancangan sistem bisa memahami bahwa tujuan dari perancangan pada dasarnya adalah untuk hal-hal yang terdaftar berikut ini.

a. Mendapatkan pemahaman yang lebih mendalam tentang sistem/perangkat lunak tentang hal-hal yang berkaitan dengan spesifikasi-spesifikasi kebutuhan non-fungsional dan batasan-batasan yang berkaitan dengan bahasa pemrograman berorientasi objek yang akan digunakan, penggunaan ulang-komponen (*component reusable*), sistem operasi yang mendasari sistem/perangkat lunak, teknologi-teknologi penebaran komponen (*deployment*) dan teknologi-teknologi konruensi, teknologi-teknologi pengelolaan transaksi, dan sebagainya.
b. Membuat asupan-asupan yang sesuai, yang merujuk pada aktivitas selanjutnya (tahap implementasi), dengan menangkap spesifikasi kebutuhan pada subsistem yang bersifat mandiri, antarmuka-antarmuka untuk subsistem-subsistem yang bersangkutan, dan kelas-kelas yang mengimplementasikan antarmuka-antarmuka. [2]

2.2.2 Kriptografi

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. [3] Untuk memastikan kerahasiaan transmisi, pemancar dapat menggunakan sebuah algoritma atau sandi untuk transmisi (misalnya enkripsi) data menjadi jelas omong kosong. Seseorang yang tidak tahu algoritma ini akan menemukan data ditransmisi tidak bermakna. [4]

2.2.3 Algoritma Kriptografi

Penggunaan kriptografi secara benar bukan sesuatu yang mudah dan sejarah penggunaan kriptografi menunjukkan bahwa kesalahan tidak hanya dilakukan oleh mereka yang "gagap teknologi." Masa depan kriptografi akan dipengaruhi oleh perkembangan matematika, terutama dalam hal algoritma, dan juga akan dipengaruhi oleh perkembangan di bidang *hardware*. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak di beberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan (parameter menjadi ekuivalen dengan kunci). [3] Seperti yang sudah dijelaskan pada bagian di atas, algoritma merupakan pola pikir terstruktur yang berisi tahap-tahap penyelesaian suatu masalah, yang nantinya akan diimplementasikan ke dalam suatu bahasa pemrograman. [5]

2.2.4 Algoritma Triple DES

Standard Triple DES menggunakan algoritma DES dengan tiga kunci seperti dalam gambar.

Gambar 1. Enkripsi Dan Dekripsi Triple Des

Dengan tiga kunci DES K1, K2 dan K3, enkripsi 3DES dilakukan sebagai berikut:

a. Enkripsi DES dengan kunci K1 dilakukan terhadap naskah asli.
b. Dekripsi DES dengan kunci K2 dilakukan terhadap hasil pertama.
c. Enkripsi DES dengan kunci K3 dilakukan terhadap hasil kedua.

Jadi enkripsi *triple* DES mempunyai rumus:

$C = E^3_{k1;k2;k3}(P) = E_{k3}(D_{k2}(E_{k1}(P)))$.

dimana

$E^3_{k1;k2;k3}(P)$ adalah enkripsi *triple* DES terhadap *P* dengan kunci *k1, k2* dan *k3*.
$E_{kn}(P)$ adalah enkripsi DES terhadap *P* dengan kunci *kn*, dan
$D_{kn}(P)$ adalah dekripsi DES terhadap *P* dengan kunci *kn*.
Dekripsi *triple* DES mempunyai rumus:

$P = D3_{k1;k2;k3}(C) = D_{k1}(E_{k2}(D_{k3}(C)))$

Dimana

$D3_{k1,k2,k3}(C)$ adalah dekripsi *triple* DES terhadap *C* dengan kunci *k1,k2* dan *k3*.

Triple DES dapat digunakan dengan satu kunci k sebagai berikut:

$E^3_{k,k,k}(P) = E_k(D_k(E_k(P)))$
$= Ek(P)$ dan
$D^3_{k,k,k}(C) = D_k(E_k(D_k(C)))$
$= D_k(C)$.

Jadi *triple* DES dengan satu kunci ekuivalen dengan DES.

Dengan dua kunci k1 dan k2, penggunaan *triple* DES adalah sebagai berikut:

$E^3_{k1,k2,k1}(P) = E_{k1}(D_{k2}(E_{k1}(P)))$ *dan*
$D^3_{k1,\,k2,\,k1}(C) = D_{k1}(E_{k2}(D_{k1}(C)))$.

Enkripsi *triple* DES dengan dua atau tiga kunci masih cukup tangguh untuk penggunaan saat ini. Walaupun enkripsi *triple* DES agak lamban komputasinya dibandingkan dengan enkripsi block cipher lainnya yang masih cukup tangguh, *triple* DES tergolong populer. [3]

2.2.5 Algoritma RSA

RSA dapat digunakan, baik untuk *key distribution* (termasuk *key exchange*), maupun untuk *digital signature*. Karena merupakan sistem pertama yang dapat digunakan untuk *key distribution* dan *digital signature*, RSA menjadi sistem kriptografi *public key* yang terpopuler. menggunakan *random number generator*, dua bilangan prima yang sangat besar p dan q (masing-masing lebih dari 200 digit). Untuk produk $n = p.q$, jika p dan q diketahui, fungsi Euler dapat dikomputasi yaitu $\phi(n) = (p-1)(q-1)$.

Menggunakan *random number generator*, suatu bilangan e antara 1 dan $\phi(n)$ yang koprima dengan $\phi(n)$. Berikutnya komputasi *inverse* dari e modulo $\phi(n)$: $d \equiv e^{-1} (\text{mod } \phi(n))$.

Kemudian mempublikasi kunci publiknya $K_E=(n,e)$ dan merahasiakan kunci privatnya $K_D=(n,d)$.

Rumus untuk mengenkripsi atau mendekripsi menggunakan kunci publik adalah

$M^e \text{ mod } n$

dimana M adalah representasi naskah asli (menggunakan bilangan bulat) jika mengenkripsi, atau representasi naskah acak jika mendekripsi.

Rumus untuk mengenkripsi atau mendekripsi menggunakan kunci privat adalah

$M^d \text{ mod } n$

dimana M adalah representasi naskah asli jika mengenkripsi, atau representasi naskah acak jika mendekripsi.

Naskah yang dienkripsi menggunakan kunci publik dapat didekripsi menggunakan kunci privat:

$(M^e)^d \equiv M^{ed} \equiv M \text{ (mod } n)$.

Jika $\text{ged}(M,n) = 1$, maka menggunakan teorema 32 kita dapatkan $(M^e)^d \equiv M^{ed}$

$\equiv M^{\phi(n)+1}$

$\equiv M^{\phi(n)}M$

$\equiv M \text{ (mod } n)$.

Untuk $\text{ged}(M,n) > 1$ dimana M bukan kelipatan n, ini hanya bisa terjadi jika M merupakan kelipatan p atau kelipatan q, tetapi bukan kelipatan keduanya.

Jika M merupakan kelipatan p, maka

$M^{ed} \equiv 0 \text{ (mod } p)$

dan

$M^{ed} \equiv M^{\phi(p)\phi(q)} M \equiv M \text{ (mod } q)$.

Menggunakan *Chinese Remainder Theorema*, dengan $n = pq$, kita dapatkan $M^{ed} \equiv M \text{ (mod } n)$.

Untuk M kelipatan q, hal serupa dapat ditunjukan. Untuk M kelipatan n, kita dapatkan

$(M^e)^d \equiv 0 \equiv M \text{ (mod } n)$

jadi

$(M^e)^d \equiv M \text{ (mod } n)$

untuk sembarang M.

Naskah yang dienkripsi menggunakan kunci privat dapat didekripsi menggunakan kunci publik:

$(M^d)^e \equiv M^{de} \equiv M \text{ (mod } n)$.

Secara umum, jika $f \neq d \text{ (mod } \phi(n))$ maka

$(M^e)^f \neq M \text{ (mod } n)$,

yang berarti sesuatu yang dienkripsi menggunakan kunci publik tidak dapat didekripsi selain menggunakan kunci privat. Juga, jika $f \neq e \text{ (mod } \phi(n))$ maka secara umum

$(M^d)^f \neq M \text{ (mod } n)$,

yang berarti sesuatu yang dienkripsi menggunakan kunci privat tidak dapat didekripsi selain menggunakan kunci publik.

Keamanan dari RSA bersandar pada fakta bahwa mengetahui n dan d secara umum tidak membantu untuk mencari e yaitu *inverse modulo* $\phi(n)$ dari d. Hal ini karena mengetahui n tidak membantu mencari $\phi(n)$ jika n tidak bisa diuraikan menjadi

$n = pq$.

Untuk menjaga keamanan tersebut, ada beberapa hal yang perlu diperhatikan dalam memilih p dan q:

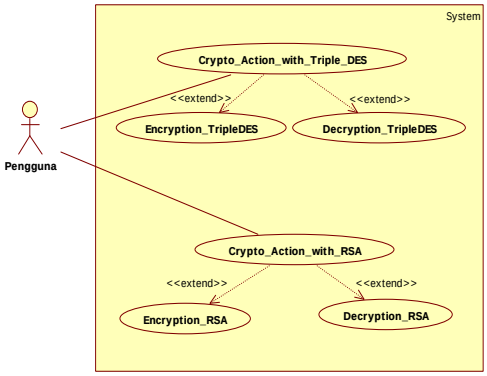
a. Nilai p harus cukup jauh dari nilai q. Sebaiknya panjang dari p harus berbeda beberapa digit dari q. Jika nilai p terlalu dekat dengan nilai q, maka *Fermat factorication* dapat digunakan untuk menguraikan $n = pq$.
b. Sebaiknya $\text{ged}(p - 1, q - 1)$ tidak terlalu besar.
c. Sebaiknya p -1 dan q -1 mempunyai faktor prima yang besar. [3]

3. HASIL DAN PEMBAHASAN

3.1 Use Case Diagram

Use case diagram menggambarkan fungsionalitas dari sistem kriptografi menggunakan algoritma *triple* DES dan RSA. *Use case* mempresentasikan sebuah interaksi antara aktor dengan sistem kriptografi. Sebuah *use case* dapat membantu dalam menggambarkan aktivitas-aktivitas pada sistem kriptografi yang dibuat. Gambaran mengenai aktivitas sistem ini akan mempermudah penulis untuk mengetahui alur kerja pada sistem kriptografi.

Sebab itulah *use case* digunakan pada penelitian ini guna menggambarkan seluruh aktivitas yang akan dikerjakan pada sistem kriptografi.



Gambar 2. Use Case Diagram

Diagram *use case* di atas menggambarkan interaksi antara pengguna dengan aplikasi kriptosistem. Interaksi hanya akan berlangsung jika pengguna sebagai aktor berperan aktif dalam mengendalikan aplikasi kriptosistem ini.

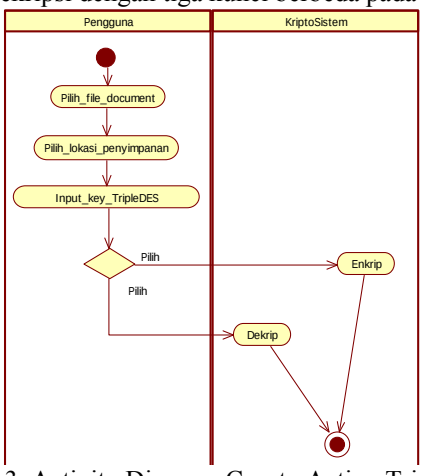
a. Pengguna atau aktor dapat secara langsung masuk pada *form* kriptosistem yang menampilkan *cryptosystem triple* DES dan *cryptosystem* RSA.
b. Terdapat dua kriptosistem yang bisa dipilih yaitu *cryptosystem triple* DES atau *cryptosystem* RSA.
c. Pengguna dapat memilih *file text* untuk dienkripsi dengan mengklik *button browse*.
d. Menentukan lokasi untuk melakukan penyimpanan hasil dekripsi dengan mengklik *button choose*.
e. Menginputkan kunci pengaman untuk *algorithm triple* DES atau dengan mengklik *button keygen* pada *algorithm* RSA.
f. Pengguna dapat memilih *button encryption* atau memilih *button decryption* untuk menghasilkan keluaran berupa *file text* yang tersimpan pada lokasi yang sudah ditentukan pengguna sebelumnya.
g. Untuk *algorithm* RSA, pengguna dapat menyimpan *private key* jika ingin melakukan dekripsi terhadap dokumen pada masa yang akan datang. Pengguna juga dapat menginput *private key* dari kunci pengaman yang sebelumnya sudah disimpan ketika akan melakukan proses dekripsi pada dokumen *ciphertext*.

3.2 Activity Diagram

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang pada penelitian ini, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Pada penelitian ini, *activity diagram* berfungsi untuk memberikan gambaran berupa *flowchart* mengenai proses enkripsi dan dekripsi yang dilakukan pada sistem kriptografi.

3.2.1 Activity Diagram Crypto Action Triple DES

Crypto action menyajikan sebuah kripto sistem dengan *algorithm triple* DES. *Algorithm triple* DES menyediakan dua fungsi enkripsi dan dekripsi dengan tiga kunci berbeda pada setiap proses perhitungannya.



Gambar 3. Activity Diagram Crypto Action Triple DES

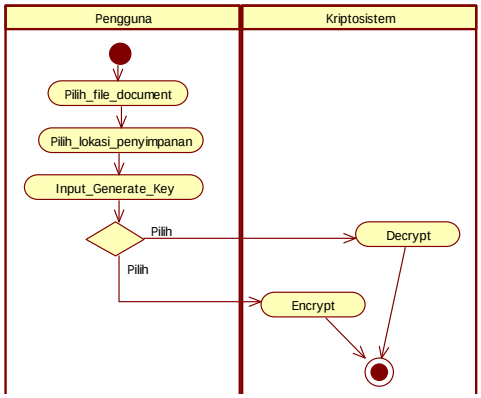
Pada gambar di atas menunjukkan *activity diagram* pada *crypto action triple* DES. *Crypto action* menggambarkan prosedur kerja program aplikasi kriptografi *triple* DES dan RSA pada umumnya.

a. Aktivitas yang dapat dilakukan pada *form* utama yaitu memilih *algorithm cryptosystem triple* DES.
b. Kemudian pengguna dapat menginputkan *file* dengan memilih *button browse.*
c. Menentukan lokasi dimana *file* hasil dekripsi akan disimpan pada aktivitas *decryption* selanjutnya.
d. Menginputkan *password* pada *textbox private key* sebagai kunci pengaman.
e. Memilih *button encryption* yang berfungsi untuk mengenkrip *file* dokumen. Keluaran dari hasil enkripsi yaitu berupa *file text* berisi *ciphertext*.

f. Pengguna juga bisa memilih *button decryption* yang berfungsi untuk mendekrip *file ciphertext*. Keluaran dari hasil dekripsi yaitu berupa *file* hasil dekripsi.

3.2.2 Activity Diagram Crypto Action RSA

Algorithm kriptografi yang terpopuler lainnya selain *triple* DES yaitu *algorithm* kriptografi RSA. RSA menyediakan dua fungsi yang sama dengan *triple* DES yaitu enkripsi dan dekripsi. Bedanya, RSA melakukan enkripsi terhadap *file* dokumen dengan menggunakan kunci publik dan mendekripsi *file ciphertext* dengan menggunakan kunci privat.



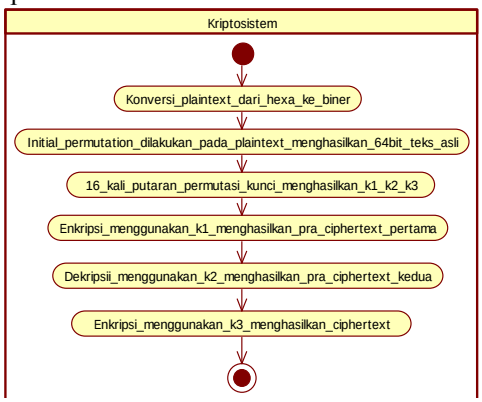
Gambar 4. Activity Diagram Crypto Action RSA

Pada gambar di atas menunjukkan aktivitas *encryption* pada *algorithm system* RSA yaitu dengan memilih *cryptosystem* RSA, memberikan beberapa *input* dan melakukan eksekusi enkrip dan dekrip.

a. Aktivitas *encryption* pada *algorithm system* RSA yaitu dengan memilih *cryptosystem* RSA.
b. *File* sumber yang akan dienkrip dapat di-*input* dengan mengklik *button browse*.
c. Untuk menyimpan hasil dekrip pada aktivitas *decryption* setelah melakukan *encryption* dapat dilakukan dengan mengklik *button choose*.
d. Pengguna dapat membangkitkan kunci *private key* dan *public key* dengan mengklik *button generate key*.
e. Kunci pengaman untuk melakukan dekripsi selanjutnya dapat juga di simpan agar pengguna bisa melakukan *decryption* pada waktu yang di inginkan. Untuk menyimpan private key, pengguna bisa mengklik *button save private key*.
f. Terakhir, pengguna bisa melakukan *encryption* terhadap file dokumen menggunakan *algorithm* RSA dengan mengklik *encryption*.
g. Pada waktu yang berbeda, *file ciphertext* dapat juga didekripsi kembali dengan menggunakan kunci privat RSA dan mengklik *button decryption*.

3.2.3 Activity Diagram Encryption

a. Activity Diagram Encryption Triple DES



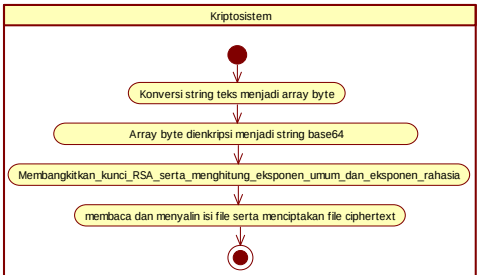
Gambar 5. Activity Diagram Encryption Triple DES

Activity diagram pada Gambar 5 di atas menggambarkan aktivitas pada kriptosistem saat melakukan enkripsi terhadap *file* dokumen. *Form* utama kriptosistem ini dapat di akses setelah melalui *form progressbar*.

1) Aktivitas yang dilakukan pada proses enkripsi *triple* DES yaitu mengkonversi bilangan hexa menjadi bilangan biner.
2) Melakukan permutasi terhadap bilangan biner dari teks asli menjadi 64 bit.
3) Melakukan 16 kali putaran permutasi terhadap kunci eksternal dan menghasilkan tiga kunci yang berbeda.
4) Kemudian proses enkripsi bisa dilakukan dengan menggunakan kunci pertama dan menghasilkan pra-*ciphertext* pertama.

5) Selanjutnya mendekripsi *plaintext* dari pra-*ciphertext* pertama menggunakan kunci kedua sehingga diperoleh pra-*ciphertex*t kedua.
6) Tahap terakhir yaitu melakukan enkripsi terhadap *plaintext* pra-*ciphertext* kedua dan dihasilkan *text ciphertext*.

b. Activity Diagram Encryption RSA



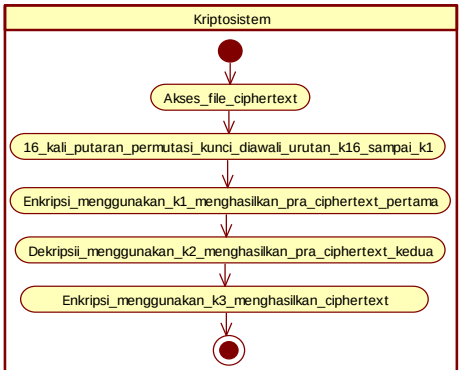
Gambar 6. Activity Diagram Encryption RSA

Activity diagram pada Gambar 6 di atas menggambarkan aktivitas pada kriptosistem saat melakukan enkripsi terhadap *file* dokumen. *Form* utama kriptosistem ini dapat di akses setelah melalui *form progressbar*.

1) Aktivitas *encryption* pada *algorithm system* RSA yaitu dengan melakukan konversi terhadap string teks menjadi array *byte*.
2) Melakukan enkripsi terhadap array *byte* sehingga menjadi string base64.
3) Proses pembangkitan kunci yang berfungsi untuk menciptakan kunci publik dan kunci privat.
4) Hasil enkripsi akan dibaca dan disalin untuk disimpan dalam *file ciphertext* yang telah dibuat.

3.2.4 Activity Diagram Decryption

a. Activity Diagram Decryption Triple DES

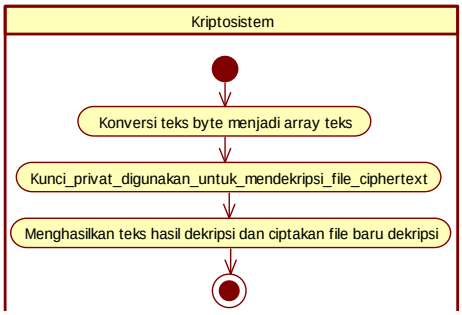


Gambar 7 Activity Diagram Decryption Triple DES

Activity diagram pada Gambar 7 di atas menggambarkan aktivitas pada kriptosistem saat melakukan dekripsi terhadap *file* dokumen. *Form* utama kriptosistem ini dapat di akses setelah melalui *form progressbar*.

1) Aktivitas yang dilakukan pada proses dekripsi *triple* DES yaitu menginput *file ciphertex*t.
2) Melakukan 16 kali putaran permutasi terhadap kunci dengan urutan terbalik dari proses permutasi kunci untuk enkripsi.
3) Kemudian proses dekripsi bisa dilakukan dengan menggunakan kunci ketiga dan menghasilkan pra-*plaintext* pertama.
4) Selanjutnya mengenkripsi *ciphertext* dari pra-*plaintext* pertama menggunakan kunci kedua sehingga diperoleh pra-*plaintext* kedua.
5) Tahap terakhir yaitu melakukan dekripsi terhadap *ciphertext* pra-*plaintext* kedua dan dihasilkan *text* hasil dekripsi.

b. Activity Diagram Decryption RSA



Gambar 8. Activity Diagram Decryption RSA

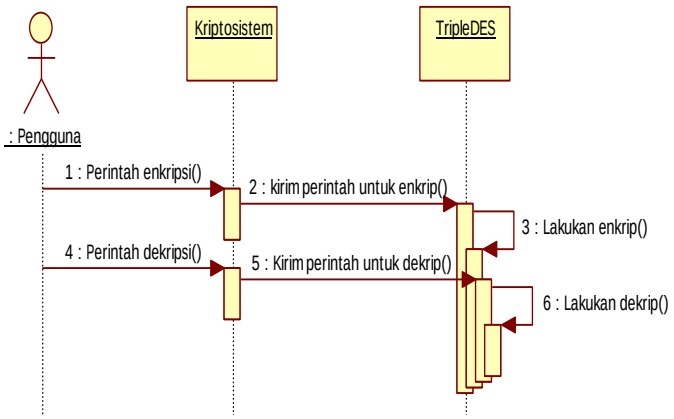
Activity diagram pada Gambar 8 di atas menggambarkan aktivitas pada kriptosistem saat melakukan dekripsi terhadap *file* dokumen. *Form* utama kriptosistem ini dapat di akses setelah melalui *form progressbar*.

1) Aktivitas *decryption* pada *algorithm system* RSA yaitu dengan melakukan konversi teks *byte* menjadi array teks.
2) Kunci private berperan penting untuk melakukan dekripsi terhadap *file ciphertext*.
3) Hasil dekripsi akan disimpan dengan menciptakan *file* dekripsi baru.

3.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem kriptografi (termasuk pengguna, *display* dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* yang digunakan pada penelitian ini yaitu *sequence diagram* vertikal guna mempermudah pemahaman terhadap alur kerja *cryptosystem triple* DES dan RSA.

3.3.1 Sequence Diagram Crypto Action Triple DES

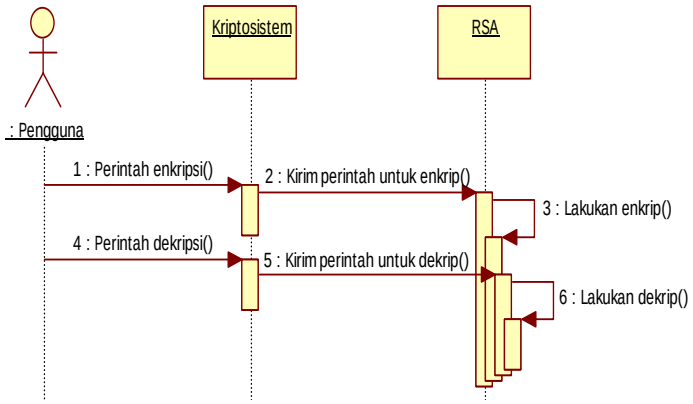


Gambar 9. Sequence Diagram Crypto Action Triple DES

Sequence diagram *crypto action triple* DES merupakan aktivitas yang menggambarkan interaksi antara pengguna dengan sistem kriptografi. Aktivitas yang berurutan berfungsi memberikan kemudahan berinteraksi yang bersifat dinamis pada sistem kriptografi. Sequence pertama yang dilakukan yaitu menentukan tindakan antara enkripsi atau dekripsi.

a. Setelah *input file*, penentuan lokasi dan *key* di-*input*, maka kriptosistem akan mengirim instruksi yang diberikan pengguna pada sistem *crypto action triple* DES dan dokumen segera dienkrip.
b. Prosedur untuk meng-*input* pada proses enkripsi juga sama dengan prosedur untuk proses dekripsi. Pengguna dapat memberikan perintah dekripsi pada kriptosistem dan kriptosistem akan mengirim perintah tersebut pada sistem *algorithm triple* DES.

3.3.2 Sequence Diagram Crypto Action RSA



Gambar 10. Sequence Diagram Crypto Action RSA

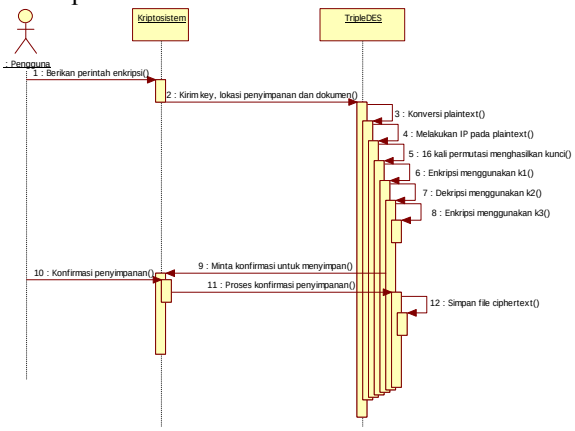
Sequence diagram *crypto action* RSA merupakan aktivitas yang menggambarkan interaksi antara pengguna dengan lingkungan sistem. Aktivitas yang berurutan berfungsi memberikan kemudahan berinteraksi yang bersifat dinamis pada sistem kriptografi. Berikut diuraikan urutan proses interaksi *crypto action* pada *algorithm* RSA.

a. Terdapat dua proses yang bisa dilakukan oleh pengguna kriptosistem yaitu enkripsi dan dekripsi. Pada enkripsi, pengguna perlu memberikan perintah pada kriptosistem untuk melakukan enkrip.
b. Perintah yang diberikan oleh pengguna akan dikirim pada *algorithm* RSA untuk dilakukan proses enkripsi.
c. Untuk proses dekripsi, hal sama seperti proses enkripsi juga dilakukan yaitu pengguna memberikan perintah terhadap kriptosistem kemudian perintah akan dikirim oleh kriptosistem pada sistem *algorithm* RSA.

d. Instruksi dekripsi diproses oleh *algorithm* RSA setelah menerima kiriman perintah dari kriptosistem yang diberikan oleh pengguna.

3.3.3 Sequence Diagram Encryption

a. Sequence Diagram Encryption Triple DES

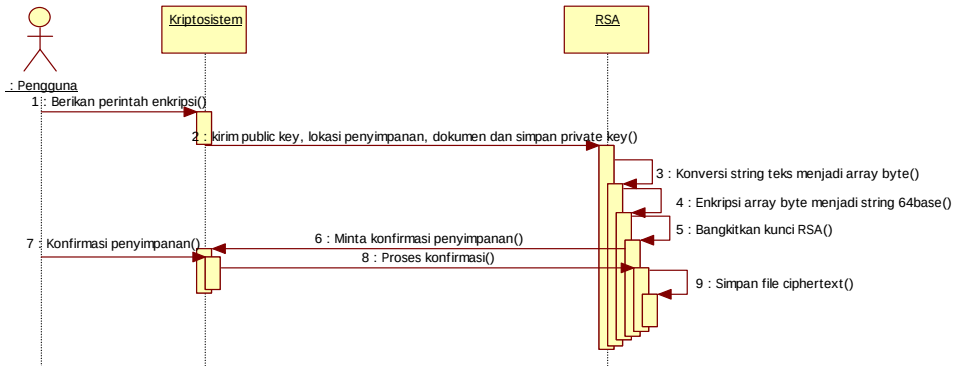


Gambar 11. Sequence Diagram Encryption Triple DES

Sequence diagram pada Gambar 11 di atas menjelaskan bahwa pengguna dapat masuk melalui *progressbar* terlebih dahulu kemudian akan ditampilkan *form* kriptosistem yang berisi pilihan *triple* DES dan RSA. Pennguna dapat menentukan *triple* DES untuk melakukan enkripsi terhadap *file* dokumen.

1) Kriptosistem akan mengirimkan *file* untuk dienkrip, lokasi penyimpanan yang telah ditentukan dan *key* sebagai kunci pengaman pada sistem *triple* DES.
2) Pada bagian *algorithm triple* DES, diawali dengan melakukan konversi bilangan dari hexa ke bilangan biner.
3) Dilakukan proses permutasi terhadap bilangan biner dari teks asli menjadi bilangan biner 64 bit.
4) Melakukan 16 kali putaran permutasi terhadap kunci eksternal sehingga menghasilkan kunci k1, k2 dan k3.
5) Lakukan proses enkripsi menggunakan k1 dan menghasilkan pra-*ciphertext* pertama.
6) Kemudian hasil pra-*ciphertext* pertama didekripsi menggunakan k2 sehingga menghasilkan pra-*ciphertext* kedua.
7) Tahap akhir yaitu mengenkripsi pra-*ciphertext* kedua menggunakan k3 sehingga dihasilkan *file ciphertext*.
8) Dialog untuk *save as* akan tampil sebagai tanda permintaan konfirmasi dari algoritma pada kriptosistem. Kemudian pengguna dapat mengkonfirmasi permintaan dengan memilih *save*. Konfirmasi akan diproses oleh sistem dan *file ciphertext* akan disimpan.

b. Sequence Diagram Encryption RSA



Gambar 12. Sequence Diagram Encryption RSA

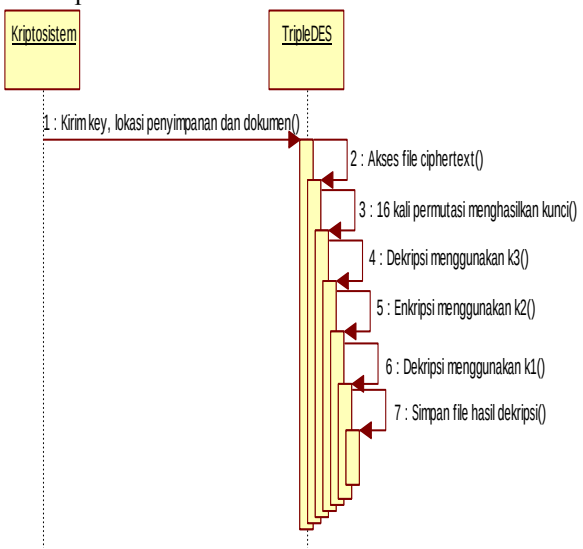
Sequence diagram pada Gambar 12 menjelaskan prosedur untuk melakukan enkripsi yaitu melalui *progressbar* sebagai jalan menuju *form* kriptosistem. Pada *form* kriptosistem terdapat dua *algorithm cryptosystem* yang bisa pengguna pilih. Pada *sequence* ini, pengguna dapat memilih *cryptosystem* RSA untuk melakukan proses enkripsi terhadap *file*.

1) Pada *cryptosystem* RSA, diawali dengan meng-*input file* dokumen, lokasi penyimpanan dan *generate key* yang selanjutnya akan dikirim pada algorithm RSA untuk diproses.
2) Mengirim perintah untuk penyimpanan *private key* yang berfungsi untuk mendekrpsi *file ciphertext*.
3) Pada *algorithm* RSA dilakukan aktivitas konversi terhadap string teks menjadi array *byte*.
4) Hasil konversi array *byte* akan dienkripsi menjadi string base64.
5) Melakukan proses pembangkitan kunci untuk *public key* dan *private key*.

6) Setelah enkripsi dilakukan, permintaan konfirmasi penyimpanan akan ditampilkan pada kriptosistem dan akan diterima oleh pengguna sistem. Ketika pengguna mengkonfirmasi permintaan maka konfirmasi penyimpanan segera diproses dan menyimpan *file ciphertext*.

3.3.4 Sequence Diagram Decryption

a. Sequence Diagram Decryption Triple DES

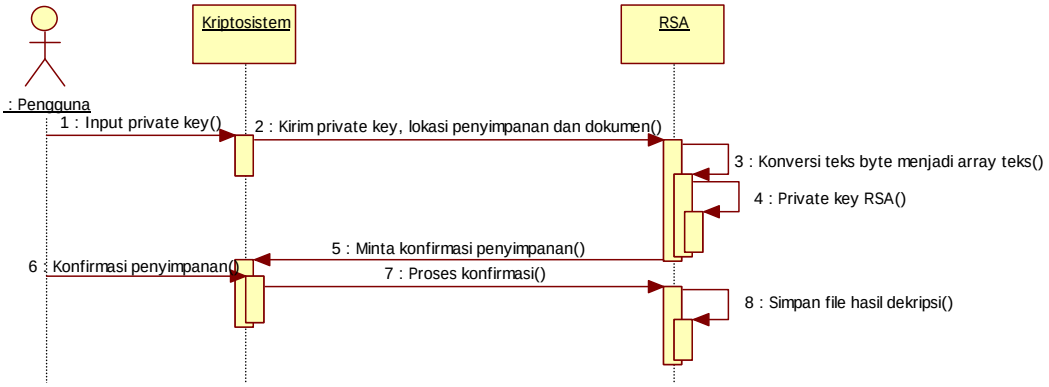


Gambar 13. Sequence Diagram Decryption Triple DES

Sequence diagram pada Gambar 13 di atas menjelaskan bahwa pengguna dapat masuk melalui *progressbar* terlebih dahulu kemudian akan ditampilkan *form* kriptosistem yang berisi pilihan *triple* DES dan RSA.

1) Mengirim kunci pengaman untuk melakukan dekrip, lokasi penyimpanan dan file dokumen pada sistem *triple* DES untuk diproses selajutnya.
2) Pilih kriptografi *triple* DES sebagai *algorithm* dekripsi kemudian akses *file ciphertext*.
3) Melakukan 16 kali putaran permutasi terhadap kunci namun dengan urutan yang terbalik dari k16 sampai k1.
4) Kemudian lakukan proses dekripsi dengan menggunakan kunci k3 dan menghasilkan pra-*plaintext* pertama.
5) Untuk proses selanjutnya dilakukan enkripsi terhadap file pra-*plaintext* yaitu dengan menggunakan kunci k2.
6) Tahap terakhir yaitu mendekripsi *file* pra-*plaintext* kedua menggunakan kunci k1 sehingga dihasilkan *file plaintext* hasil dekripsi.
7) Hasil dekripsi terhadap *file ciphertext* akan disimpan pada lokasi yang telah ditentukan sebelumnya.

b. Sequence Diagram Decryption RSA



Gambar 14. Sequence Diagram Decryption RSA

Sequence diagram pada Gambar 14 menjelaskan prosedur untuk melakukan dekripsi yaitu melalui *progressbar* sebagai jalan menuju *form* kriptosistem. Pilih kriptografi RSA sebagai *algorithm* untuk melakukan enkripsi dan dekripsi.

1) Pengguna dapat meng-*input private key* dengan mengetik manual kunci pengaman pada *textbox private key*.
2) Melakukan pengiriman *private key*, lokasi penyimpanan dan *file ciphertext* yang akan didekrip.
3) Konversi teks *byte* menjadi array *byte* menggunakan generator *algorithm* RSA.
4) Akses kunci privat untuk melakukan dekripsi terhadap *file ciphertext*.

5) Setelah *private key* di-*input,* proses dekripsi dilakukan untuk mentransformasi *file ciphertext* kembali menjadi *file* teks asli dekripsi.
6) Terakhir *file* hasil dekripsi akan tersimpan pada lokasi yang telah ditentukan pengguna.
7) Setelah dekripsi dilakukan, permintaan konfirmasi penyimpanan akan ditampilkan pada kriptosistem dan akan diterima oleh pengguna sistem. Ketika pengguna mengkonfirmasi permintaan maka konfirmasi penyimpanan segera diproses dan menyimpan *file* hasil dekripsi.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan mengenai perancangan sistem kriptografi pada dokumen menggunakan algoritma *triple* DES dan RSA yang telah diuraikan pada bab-bab sebelumnya, maka dapat disimpulkan bahwa:

a. Aplikasi sistem kriptografi dibangun untuk memberi kemudahan bagi pengguna dalam mengamankan dokumen penting sehingga terhindar dari gangguan privasi pengguna oleh pihak tak berwenang.
b. Tampilan sederhana aplikasi sistem kriptografi *triple* DES dan RSA ini bertujuan untuk mempermudah pengguna dalam proses penggunaan aplikasi.
c. Algoritma *triple* DES dan RSA mampu mengamankan dokumen penting sehingga pengguna tidak perlu khwatir privasinya diganggu.

5. SARAN

Adapun saran yang dapat disampaikan oleh penulis guna pengembangan program aplikasi kedepannya yaitu:

a. Perlu tutorial bagi pengguna sebelum menggunakan program aplikasi kriptografi *triple* DES dan RSA ini guna melancarkan proses eksekusi enkripsi dan dekripsi serta terhindar dari kesalahan penggunaan program.
b. Tampilan aplikasi kriptografi ini masih perlu dimodifikasi untuk menambah daya tarik pengguna agar tertarik menggunakan program aplikasi kriptografi *triple* DES dan RSA.
c. Program aplikasi kriptografi *triple* DES dan RSA ini masih jauh dari sempurna sehingga penulis mengharapkan penelitian dan pengujian lebih lanjut sesuai kebutuhan kriptografer.

UCAPAN TERIMA KASIH

Pada penulisan skripsi ini, penulis telah banyak memperoleh bimbingan, pengarahan, saran, petunjuk dan dukungan moril terutama dari kedua dosen pembimbing serta pihak-pihak yang tidak bisa disebutkan satu per satu. Pada kesempatan yang berbahagia ini, penulis mengucapkan rasa syukur yang mendalam serta terimakasih kepada Sekolah Tinggi Manajemen Informatika dan Komputer Widya Dharma Pontianak.

DAFTAR PUSTAKA

[1] Pratama, I Putu Agus Eka. (2014). *Sistem Informasi dan Implementasinya*. Informatika. Bandung.
[2] Nugroho, Adi. (2010). *Rekayasa Perangkat Lunak Menggunakan Uml Dan Java*. Edisi I. Andi. Yogyakarta.
[3] Kromodimoeljo, Sentot. (2010). *Teori & Aplikasi Kriptografi*. SPK IT Consulting. Jakarta.
[4] Muis, Saludin. (2013). *Pengantar Kriptografik Kuantum Teknik Enkripsi Masa Depan*. Edisi Pertama. Graha Ilmu. Yogyakarta.
[5] Kristanto, Andri. (2013). *Algoritma dan Pemrograman Dengan C++*. Edisi Ketiga. Graha Ilmu. Yogyakarta.