# HALAMAN JUDUL

**ENKRIPSI FILE TEKS MENGGUNAKAN ALGORITMA RIVEST-SHAMIR-ADLEMAN (RSA)**

Diajukan sebagai persyaratan untuk memenuhi derajat Ahli Madya (Amd) pada  
Program Studi Teknik Informatika  
Jurusan Teknik Informasi  
Politeknik Negeri Samarinda

TUGAS AKHIR



Oleh:

**ELVAND ANDRIAN MANURUNG**

NIM 14615073

KEMENTERIAN RISET TEKNOLOGI DAN PENDIDIKAN TINGGI

POLITEKNIK NEGERI SAMARINDA

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI TEKNIK INFORMATIKA

# HALAMAN PERNYATAAN ORISINALITAS

Saya yang bertanda tangan di bawah ini :

Nama : Elvand Andrian Manurung

NIM : 14615073

Jurusan : Teknik Informasi

Program Studi : Teknik Informatika

Jenjang : Diploma III

Judul Tugas Akhir : Enkripsi File Teks Menggunakan Algoritma Rivest-Shamir-Adleman (RSA)

Dengan ini menyatakan bahwa Laporan Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Jika dikemudian hari terbukti ditemukan unsur plagirisme dalam Laporan Tugas Akhir ini, maka saya tersedia menerima sanksi sesuai peraturan perundangan-undangan yang berlaku.

Samarinda, 05 Agustus 2017

Elvand Andrian Manurung  
 NIM. 14615073

# HALAMAN PENGESAHAN PEMBIMBING

**ENKRIPSI FILE TEKS MENGGUNAKAN ALGORITMA**

**RIVEST-SHAMIR-ADLEMAN (RSA)**

**NAMA : ELVAND ANDRIAN MANURUNG**

**NIM : 14615073**

**JURUSAN : TEKNOLOGI INFORMASI**

**PROGRAM STUDI : TEKNIK INFORMATIKA**

**JENJANG STUDI : DIPLOMA III**

**Laporan Tugas Akhir ini telah disahkan**

**pada tanggal, 30 Agustus 2017 2015**

**Menyetujui :**

**Pembimbing I, Pembimbing II,**

**Rihartanto, ST. Agusma Wajiansyah, S.ST., MT.**

**NIP. 19711205 200312 1 001 NIP.** **19810805 200501 1 003**

**Mengesahkan :**

**Direktur Politeknik Negeri Samarinda,**

**Ir. H. Ibayasid, M.Sc.**

**NIP.19590303 198903 1 002**

**Lulus Ujian Tanggal : 05 Agustus 2017**

# HALAMAN PERSETUJUAN PENGUJI

**ENKRIPSI FILE TEKS MENGGUNAKAN ALGORITMA**

**RIVEST-SHAMIR-ADLEMAN (RSA)**

**NAMA : ELVAND ANDRIAN MANURUNG**

**NIM : 14615073**

**JURUSAN : TEKNOLOGI INFORMASI**

**PROGRAM STUDI : TEKNIK INFORMATIKA**

**JENJANG STUDI : DIPLOMA III**

**Laporan Tugas Akhir ini telah diuji dan disetujui**

**pada tanggal, 24 Agustus 2017**

**Dewan Penguji :**

**Penguji I,**

**Nama : Arief Bramanto Wicaksono Putra, S.ST., MT**

**NIP : 19830120 200801 1 006** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Penguji II,**

**Nama : Supriadi, S.ST., MT**

**NIP : 19760427 200501 1 001** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Penguji III,**

**Nama : Wahyuni Eka Sari, S.Kom., M.Eng**

**NUP :** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Mengetahui :**

**Ketua Jurusan Ketua Program Studi**

**Teknologi Informasi, Teknik Informatika,**

**Ansar Rizal, ST., M.Kom Farindika Metandi, BCompSc., MM., M.Cs**

**NIP. 19700809 199903 1 001 NIP. 19870308 200801 1 002**

# ABSTRAK

Kerahasiaan merupakan salah satu aspek penting dari informasi. Saat ini, keamanan terhadap informasi data yang tersimpan dalam *file* sudah menjadi persyaratan mutlak. *Kriptografi* merupakan solusi dalam hal menjaga kemanan dan kerahasian informasi dalam bentuk data pada *file.* Pada penelitian ini diterapkan algoritma *Rivest-Shamir-Adleman* (RSA) pada proses enkripsi *file* teks untuk melindungi data agar tetap aman. Algoritma RSA menggunakan kunci asimteris. Kunci asimetris dibangun berdasarkan pasangan bilangan prima. Implementasi algoritma RSA untuk proses enkripsi dan dekripsi berhasil dilakukan. Dalam penelitian ini pasangan bilangan prima terkecil yang berhasil diimplementasikan dalam penelitian ini adalah 11 dan 13, dan bilangan prima terbesar dibatasi maksimal 997.

**Kata Kunci :** Kriptografi, RSA, Kunci Asimetris

# *ABSTRACT*

*Confidentiality is an important aspect of information. Currently, security of an information stored in a file is required. Cryptography is a solution in terms of maintaining security and confidentiality of an information. In this research Rivest-Shamir-Adleman (RSA) algorithm is applied in text file encryption to protect data and to keep it safe. RSA algorithm uses asymteric keys. Asymmetric keys are built on the prime pair. In this research, implementation of RSA algorithm for encryption and decryption process successfully done. The pair of the smallest primes that have been successfully implemented in this study are 11 and 13, and the biggest prime is 997.*

***Keyword :*** *Cryptography, RSA, Asymmetric Key*

# KATA PENGANTAR

Puji syukur dan terima kasih kepada Tuhan Yang Maha Esa. Dan kepada kedua orang tua penulis yang selalu mendoakan serta memberikan dukungan moral dan material kepada penulis dalam mengerjakan laporan tugas akhir sehingga penulis dapat menyelesaikan Laporan tugas akhir yang berjudul “Enkripsi File Teks Menggunakan Algoritma Rivest-Shamir-Adleman (RSA)”. Terselesaikannya Laporan Tugas Akhir ini hanyalah setapak kecil permulaan dari perjalanan panjang yang masih harus ditempuh.

Laporan Tugas Akhir ini adalah salah satu persyaratan akhir Program D3 Konsentrasi Teknik Informatika Jurusan Teknologi Informasi Politeknik Negeri Samarinda.

Ucapan terima kasih disampaikan kepada semua pihak yang secara langsung maupun tidak langsung memberikan dukungan dan bantuannya hingga terselesaikannya Laporan Tugas Akhir ini. Ucapan terima kasih khusus di sampaikan kepada :

1. Bapak Ir. H. Ibayasid, M.Sc. selaku Direktur Politeknik Negeri Samarinda.
2. Bapak Ansar Rizal, ST., M.Kom. selaku Ketua jurusan Teknologi Informasi Politeknik Negeri Samarinda.
3. Bapak Farindika Metandi, BCompSc., MM., M.Cs. selaku Ketua Program Studi Teknik Informatika.
4. Bapak Rihartanto, ST. selaku Pembimbing 1 dan Bapak Agusma Wajiansyah, S.ST., MT. selaku Pembimbing 2 yang telah memberikan bimbingan dan masukan kepada penulis dalam menyelesaikan laporan tugas akhir ini.
5. Bapak Arief Bramanto W.P, S.ST., M.T. selaku Koordinator Kelompok Riset Dosen.
6. Semua rekan-rekan sesama mahasiswa Jurusan Teknologi Informasi.

Saran yang bersifat membangun sangat diharapkan untuk kesempurnaan Laporan Tugas Akhir ini. Semoga Laporan Tugas Akhir ini menjadi sumber inspirasi positif dan untuk dapat lebih bermanfaat bagi para pembaca baik yang dalam tahap pembelajaran ataupun untuk menambah luasnya pengetahuan.

Samarinda, 05 Agustus 2017

Penulis

# DAFTAR ISI

[ABSTRAK v](#_Toc491704677)

[ABSTRACT vi](#_Toc491704678)

[KATA PENGANTAR vii](#_Toc491704679)

[DAFTAR ISI ix](#_Toc491704680)

[DAFTAR GAMBAR xi](#_Toc491704681)

[DAFTAR TABEL xii](#_Toc491704682)

[BAB I PENDAHULUAN 1](#_Toc491704683)

[1.1 Latar Belakang 1](#_Toc491704684)

[1.2 Rumusan Masalah 2](#_Toc491704685)

[1.3 Batasan Masalah 2](#_Toc491704686)

[1.4 Tujuan Penelitian 2](#_Toc491704687)

[1.5 Manfaat Penelitian 3](#_Toc491704688)

[BAB II TINJAUAN PUSTAKA 4](#_Toc491704689)

[2.1 Kajian Ilmiah 4](#_Toc491704690)

[2.2 Dasar Teori 5](#_Toc491704691)

[2.2.1 Mekanisme Kriptografi 9](#_Toc491704692)

[2.2.2 Macam Macam Algoritma Kriptografi Modern 10](#_Toc491704693)

[2.2.3 Algoritma Cipher Pada Proses Asimetris Dan Simetris 11](#_Toc491704694)

[2.2.4 Algoritma Rivest-Shamir-Adleman (RSA) 12](#_Toc491704695)

[BAB III KERANGKA KONSEP DAN METODOLOGI PENELIT 17](#_Toc491704696)

[3.1 Kerangka Konsep Penelitian 17](#_Toc491704697)

[3.1.1 Kriptografi 17](#_Toc491704698)

[3.1.2 Algoritma Rivest-Shamir-Adleman (RSA) 18](#_Toc491704703)

[3.1.3 File Teks 19](#_Toc491704704)

[3.2 Metode Penelitian 20](#_Toc491704705)

[3.2.1 Riset Awal 21](#_Toc491704706)

[3.2.2 Perancangan 21](#_Toc491704707)

[3.2.3 Pengujian 22](#_Toc491704708)

[3.2.4 Analisa Hasil 22](#_Toc491704709)

[3.3 Variabel Penelitian 22](#_Toc491704710)

[3.4 Data dan Tempat Penelitian 23](#_Toc491704711)

[BAB IV HASIL DAN PEMBAHASAN 24](#_Toc491704712)

[4.1 Proses Enkripsi Dan Dekripsi File Teks 24](#_Toc491704713)

[4.1.1 Menghitung Kunci Publik 24](#_Toc491704714)

[4.1.2 Menghitung Kunci Privat 26](#_Toc491704715)

[4.2 Proses Enkripsi menggunakan Kunci Publik 28](#_Toc491704716)

[4.3 Proses Dekripsi Menggunakan Kunci Privat 33](#_Toc491704717)

[4.5 Pengujian 38](#_Toc491704718)

[4.6 Analisa Hasil 43](#_Toc491704719)

[BAB V PENUTUP 44](#_Toc491704720)

[5.1 Kesimpulan 44](#_Toc491704721)

[5.2 Saran 44](#_Toc491704722)

**DAFTAR PUSTAKA 45**

# DAFTAR GAMBAR

[Gambar 2.2 Mekanisme Kriptografi](#_Toc487146642) 9

[Gambar 2.3 Proses Algoritma Simetri](#_Toc487146643) 10

[Gambar 2.4 Proses Algoritma Asimetris](#_Toc487146644) 11

Gambar 2.5 Tabel ASCII ............................................................................................... 13

[Gambar 3.1 Diagram Alir Kerangka Konsep Penelitian](#_Toc487146645) 17

[Gambar 3.2 Diagram Alir Metodologi Penelitian](#_Toc487146646) 20

[Gambar 4.1 Flowchart Proses Menghitung Kunci Publik RSA](#_Toc487146647) 25

[Gambar 4.2 Flowchart Proses Menghitung Kunci Privat RSA](#_Toc487146648) 27

[Gambar 4.3 Flowchart Enkripsi Pesan Dengan Menggunakan Algoritam RSA](#_Toc487146649) 29

[Gambar 4.4 Flowchart Proses Enkripsi Menggunakan Kunci Publik](#_Toc487146650) 30

Gambar 4.5 Flowchart Dekripsi Pesan Dengan Menggunakan Algoritam RSA …........ 34

Gambar 4.6 Flowchart Proses Dekripsi Menggunakan Kunci Privat …........................ 35

# DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu 5

Tabel 2.2 Hasil Enkripsi 14

Tabel 2.3 Enkripsi Code 15

Tabel 2.4 Deskripsi Code 16

Tabel 4.1 Prose Enkripsi Menggunakan Kunci Publik 31

Tabel 4.2 Proses Deskripsi Menggunakan Kunci Privat 36

Tabel 4.3 Plain Char Dan ASCII Number 38

Tabel 4.4 Hasil Enkripsi Pada Program 39

Tabel 4.5 Hasil Dekripsi Pada Program 39

Tabel 4.6 Plain Char Dan ASCII Number 40

Tabel 4.7 Hasil Enkripsi Pada Program 41

Tabel 4.8 Hasil Dekripsi Pada Program 42

# BAB I PENDAHULUAN

BAB I  
PENDAHULUAN

## Latar Belakang

Kerahasiaan merupakan salah satu aspek penting dari informasi. Penyalahgunaan data yang tidak pada tempat penyimpanan tentunya bisa menimbulkan kerugian yang besar bagi pihak yang mempunyai data dan dokumen tersebut. Salah satu alternatif yang dapat digunakan untuk menjaga kerahasiaan data dan dokumen adalah dengan menyamarkannya ke bentuk tersandi yang tidak bermakna. Hal ini dapat dilakukan dengan menggunakan kriptografi.

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Tetapi tidak semua aspek keamanan sistem informasi dapat diselesaikan dengan kriptografi. Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Pada prinsipnya, Kriptografi memiliki 4 komponen utama yaitu : *Plaintext*, *Ciphertext*, *Key*, *Algorithm*.

Algoritma yang biasa digunakan untuk memecahkan masalah dalam hal melindungi kerahasian data dalam bentuk file sendiri bermacam-macam. Contoh-nya *Digital Signature Algorithm* (DSA), *Diffie-Hellman* (DH), *ElGamal*, *Elliptic Curve Cryptography* (ECC), dan *Rivest-Shamir-Adleman* (RSA).

Pada penilitian ini akan diterapkan Algoritma Rivest-Shamir-Adleman (RSA) yang akan membantu proses pembentukan enkripsi file teks untuk melindungi data agar tetap aman dari orang-orang yang tidak berhak menyisipkan atau menghapus pesan, data, atau informasi data. Algoritma Rivest-Shamir-Adleman (RSA) termasuk dalam kelompok algoritma asimetris adalah metode yang menggunakan dua kunci berbeda yaitu kunci publik (*public key*) dan kunci privat (*private key*).

## Rumusan Masalah

Dari pembahasan di atas maka dapat dirumuskan pokok permasalahan yaitu: “Bagaimana mengimplementasikan Algoritma RSA untuk mengamankan data yang ada di dalam file teks”.

## Batasan Masalah

Ruang lingkup penelitian akan dibatasi agar penyelesaian masalah dapat lebih terarah pada tujuan tugas akhir ini. Batasan masalah dalam tugas akhir ini adalah :

1. Algoritma yang digunakan adalah RSA.
2. Menggunakan kunci Asimetris.
3. Data yang dienkripsi adalah data yang berbentuk teks di dalam file teks (\*.txt).
4. Pasangan bilangan prima berurutan yang bisa digunakan pada algoritma RSA adalah 11 dan 13.

Beberapa asumsi yang digunakan adalah :

1. Menggunakan tools pemrograman yang sesuai.
2. Kunci Asimetris dimasukkan melalui keyboard terdiri dari kombinasi huruf dan angka.
3. File teks dibuat menggunakan notepad.

## Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Mengimplempentasikan algoritma RSA untuk mengamankan isi file teks.
2. Membangun kunci asimetris yang akan digunakan pada algoritma RSA.

## Manfaat Penelitian

Setelah melakukan penelitian maka diharapkan penelitian akan memiliki manfaat sebagai berikut :

1. Menambah literatur tentang implementasi algoritma RSA untuk keamanan data.
2. Untuk memenuhi syarat mendapatkan gelar derajat Ahli Madya (Amd).

# BAB II TINJAUAN PUSTAKA

BAB II  
TINJAUAN PUSTAKA

## Kajian Ilmiah

RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika *modulo*. Baik kunci enkripsi maupun deskripsi keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum (sehingga disebut dengan kunci publik), namun kunci untuk dekripsi bersifat rahasia (kunci privat). Kenyataannya, memfaktorkan bilangan non prima menjadi faktor primanya bukanlah pekerjaan yang mudah. Belum ada algoritma yang mangkus (efisien) yang ditemukan untuk pemfaktoran itu. Semakin besar bilangan non primanya, tentu semakin sulit pula pemfaktorannya. Semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA (Arifin, 2009).

Penelitian yang terkait dengan algoritma kriptografi RSA adalah aplikasi enkripsi sms dengan metode RSA pada smartphone berbasis android. Aplikasi SMS diterapkan pada *platform* android, karena *platform* android paling banyak digunakan oleh berbagai perangkat mobile. Aplikasi yang dibuat mampu melakukan enkripsi dan deskripsi terhadap alfabet, angka dan simbol-simbol. Aplikasi juga diharapkan mampu melindungi pesan rahasia yang dikirimkan saat dilakukan penyadapan terhadap perangkat *mobile* (Sasmita dkk, 2014).

RSA mempunyai proses enkripsi dan dekripsi yang menggunakan kunci berbeda. Proses dekripsi algoritma RSA sering mempunyai kendala dalam hal ukuran kunci dekripsi yang relatif besar sehingga memperlambat proses (Arief, 2016).

Penelitian dan beberapa referensi di bidang sistem pendukung keputusan yang telah dilakukan sebelumnya dipakai untuk mendasari penelitian ini, terutama berkenaan dengan metode yang dipergunakan.

Secara umum penelitian-penelitian yang telah dilakukan terdahulu dapat dirangkum pada (Tabel 2.1).

Tabel 2.1 Penelitian Terdahulu

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Peneliti** | **Tahun** | **Judul Penelitian** |
| 1 | Zainal Arifin | 2009 | Studi Kasus Penggunaan Algoritma RSA Sebagai Algoritma Kriptografi yang Aman |
| 2 | I Gusti Made Arya Sasmita, dkk | 2014 | Aplikasi Enkripsi SMS Dengan Metode RSA Pada Smartphone Berbasis Android |
| 3 | Ashari Arief, dkk | 2016 | Implementasi Kriptografi Kunci Publik dengan Algoritma RSA-CRT pada Aplikasi Instant Messaging |
| 4 | Elvand Andrian Manurung | 2017 | Enkripsi File Teks Menggunakan Algoritma Rivest-Shamir-Adleman (RSA) |

## Dasar Teori

Kriptografi adalah bidang ilmu untuk menjaga keamanan pesan (*message*). Kriptografi telah banyak diimplementasikan di banyak hal. *Smart card*, Anjungan Tunai Mandiri (ATM), *Pay TV*, *Mobile Phone*, dan Komputer adalah beberapa contoh produk teknologi yang menggunakan kriptografi untuk keamanan-nya. Cara kerjanya adalah dengan mengubah pesan asli yang dapat dimengerti / dibaca manusia (*plaintext*) ke bentuk lain yang tidak dapat dimengerti / dibaca oleh manusia (*ciphertext*). Proses transformasi *plaintext* menjadi chipertext diistilahkan dengan enkripsi. Sedang proses pengembalian pesan *chipertext* menjadi *plaintext* diistilahkan dengan dekripsi (Primartha, 2011).

Sistem kriptografi kunci publik adalah bahwa kunci kriptografi dibuat sepasang, satu kunci untuk enkripsi dan satu kunci untuk dekripsi. Kunci untuk enkripsi bersifat publik (tidak rahasia) sehingga dinamakan kunci publik, sedangkan kunci dekripsi bersifat rahasia sehingga dinamakan kunci privat (kunci rahasia). Hasil kajian teknik enkripsi data dengan metode algoritma simetri menunjukan bahwa kelemahan-kelemahan pada metode simetri dengan kunci tunggal bisa diselesaikan dengan menggunakan metode lain yang disebut dengan metode kunci publik yang menggunakan 2 (dua) kunci yaitu *public key* dan *private key* (Listiyono, 2009).

Dalam kriptografi, RSA adalah algoritma untuk enkripsi kunci publik (*public-key encryption*). Algoritma ini adalah algoritma pertama yang diketahui paling cocok untuk menandai (*signing*) dan untuk enkripsi (*encryption*) dan salah satu penemuan besar pertama dalam kriptografi kunci publik. RSA masih digunakan secara luas dalam protokol-protokol perdagangan elektronik, dan dipercayai sangat aman karena diberikan kunci-kunci yang cukup panjang dan penerapan-penerapannya yang sangat *up-to-date* (mutakhir) (Pahrizal dan Pratama, 2016).

Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (*Massachussets Instittute of Technology*) pada tahun 1976. Ketiga peneliti tersebut adalah Ron Riverst, Adi Shamir, dan Leonard Adleman. Algoritma RSA digunakan untuk membangkitkan 2 kunci yaitu kunci publik dan kunci privat. Kunci publik adalah dua buah variabel bilangan (e,n) yang digunakan untuk mengenkripsi data. Sedangkan kunci privat adalah dua buah variabel bilangan (d,n) yang digunakan untuk melakukan dekripsi data. Nilai e, d dan n adalah nilai bilangan bulat positif. Keamanan algoritma RSA bergantung pada kesulitan memfaktorkan bilangan besar. “*We have proposed a method for implementing a publik-key cryptosystem whose security rests in part on the difficulty of factoring large numbers*” (R.L. Rivest, A.Shamir, and L. Adleman, 1977).

Algoritma kriptografi dapat dibagi ke dalam kelompok algoritma simetris dan algoritma asimetris. Algoritma simetris merupakan algoritma kriptografi yang menggunakan kunci yang sama baik untuk proses enkripsi maupun dekripsi. Algoritma simetris dapat dikelompokkan menjadi dua kategori, yaitu cipher aliran (*stream chiper*) dan cipher blok (*block chiper*). Cipher aliran merupakan algoritma kriptografi yang beroperasi dalam bentuk bit tunggal. Sedangkan algoritma kriptografi kategori cipher blok beroperasi dalam bentuk blok bit pada penelitian ini RSA menggunakan algoritma *chiper block* yang dimana setiap blok terdiri dari beberapa bit (1 blok terdiri dari 64 bit atau 128 bit) (Andriyanto dan Pardede, 2008).

Pada umumnya RSA menggunakan algoritma asimetris dimana pada proses enkripsi dan dekripsi mempunyai dua kunci yang berbeda, yaitu kunci umum (*public key*) yang digunakan untuk proses enkripsi yaitu perubahan data teks asli (*plaintext*) menjadi teks rahasia (*ciphertext*) yang sifatnya tidak rahasia, dan kunci pribadi (*private key*) yang digunakan untuk proses dekripsi yaitu pengembalian data teks rahasia (*ciphertext*) menjadi teks asli (*plaintext*) yang sifatnya rahasia dan masing-masing pihak memiliki kunci pribadi yang berbeda (Ginting dkk, 2015). Keamanan kunci privat dan kunci publik pada algoritma RSA sangat bergantung pada dua variabel p dan q dimana variabel ini di gunakan untuk menciptakan kedua kunci tersebut (Zainnudin dan Mulyana, 2016).

RSA merupakan suatu algoritma yang populer digunakan karena kesederhanaan, serta memiliki kecepatan proses yang tidak kalah cepat dibandingkan dengan algoritma kriptografi lainnya. Secara umum algoritma RSA banyak digunakan pada proses penyandian yang bersifat software aplikasi Sistem Pembangkitan Kunci pada Algoritma RSA sedangkan salah satu kelemahan sistem penyandian yang bersifat online adalah pengguna diharuskan terkoneksi dengan jaringan internet setiap ingin melakukan proses enkripsi dan dekripsi informasi rahasia (Busran, 2014).

Salah satu cara untuk mengamankan data atau informasi dengan Kriptologi. Kriptografi adalah studi keamanan (kerahasiaan). Algoritma kriptografi teknis terdiri dari substitusi dan transposisi teknis teknik. Teknik kriptografi dapat dipercaya untuk menangani masalah keamanan data atau informasi, karena selain menggunakan bahasa pemrograman komputer, kriptografi juga menggunakan rumus matematika, mulai dari rumus sederhana untuk formula kompleks. Dalam kriptografi terdapat dua konsep utama, yaitu enkripsi dan dekripsi. Enkripsi adalah proses dimana data atau informasi diubah menjadi bentuk yang tidak dikenali sebagai informasi awalnya dengan menggunakan metode tertentu sedangkan dekripsi adalah mengubah kembali bentuk tersamar tersebut menjadi informasi awal (Pradipta, 2016).

Banyak cara yang dilakukan oleh pengembang TI dalam urusan keamanan, salah satunya menggunakan metode kriptografi. Kriptografi adalah sebuah ilmu yang mempelajari tentang penyembunyian huruf atau tulisan sehingga membuat tulisan tersebut tidak dapat dimengerti atau dibaca, proses ini disebut dengan enkripsi. Untuk membacanya kembali maka lakukanlah proses dekripsi atau pengembalian ke tulisan aslinya. Hal ini banyak dimanfaatkan untuk menyamarkan dokumen-dokumen penting sehingga hanya orang-orang tertentu yang dapat membuka dan membacanya. Selain itu membuat dokumen tersebut aman apabila jatuh ke pihak lain. Seiring dengan perkembangannya, kriptografi mulai dimanfaatkan untuk menyamarkan *file-file* non dokumen seperti gambar, *video* maupun suara (Arief, 2015).

Syarat-syarat yang ada pada pembangkitan Kunci RSA adalah :

1. Pilih nilai p dan q yang dimana kedua nilai bilangan tersebut merupakan bilangan prima acak besar.
2. Hitung nilai n dari hasil p\*q (dimana nilai n digunakan pada proses pembangkitan kunci publik dan kunci privat).
3. Hitung nilai Q(n) = (p-1) \* (q-1) yang dimana nilai Q(n) berfungsi sebagai proses menentukan nilai kunci publik dan privat.
4. Pilih nilai e dimana 1 < e < Q(n) dan e adalah relatif prima dengan Q(n), relatif prima artinya nilai PBB (kunci publik, Q(n)) = 1.
5. Buat sebuah kunci privat d, dimana d menggunakan persamaan d = 1+kQ(n)/e dan k adalah salah satu bilangan bulat 1,2,3, ,..Q(n).

### 2.2.1 Mekanisme Kriptografi

Enkripsi adalah proses untuk mengubah pesan yang tidak disandikan (*plaintext*) ke dalam bentuk yang tidak dapat dibaca (*ciphertext*), dekripsi adalah proses untuk mengubah pesan yang tidak dapat dibaca kedalam bentuk yang dapat dibaca. Proses enkripsi dan dekripsi diatur oleh kunci kriptografi tertentu. Urutan proses kerja secara umum dapat digambarkan pada Gambar 2.2 sebagai berikut:

Plaintext

Enkripsi

Ciphertext

Dekripsi

Plaintext

Kunci Enkripsi

Kunci Dekripsi

Gambar 2.2 Mekanisme Kriptografi

### 2.2.2 Macam Macam Algoritma Kriptografi Modern

Algoritma kriptografi dibagi menjadi 2 bagian berdasarkan kunci yang dipakainya, yaitu Algoritma Simetris (menggunakan satu kunci untuk enkripsi dan deskripsinya) dan Algoritma Asimetris (menggunakan dua kunci yang berbeda untuk enkripsi dan dekripsinya).

1. Algoritma Simetris

Algoritma ini disebut simetris karena memiliki *key* atau kunci yang sama dalam proses enkripsi dan dekripsi sehingga algoritma ini juga sering disebut algoritma kunci tunggal atau algoritma satu kunci. *Key* dalam algoritma ini bersifat rahasia atau *private key* sehingga algoritma ini juga disebut dengan algoritma kunci rahasia (Prayudi, 2005).

Algoritma yang memakai kunci simetris di antaranya adalah :

* + 1. Data encryption Standard (DES)
    2. RC2,RC4,RC5,RC6
    3. International Data Encryption Algorithm (IDEA)
    4. One Time Pad (OTP)
    5. A5
    6. Dan lain sebagainya.

Proses algoritma simetri dapat dilihat pada Gambar 2.3.

 Gambar 2.3 Proses Algoritma Simetri

1. Algoritma Asimetris

Algoritma ini disebut asimetris karena kunci yang digunakan untuk enkripsi berbeda dengan kunci yang digunakan untuk dekripsi. Kunci yang digunakan untuk enkripsi adalah kunci publik atau *public key* sehingga algoritma ini juga disebut dengan algoritma kunci publik. Sedangkan kunci untuk dekripsi menggunakan kunci rahasia atau *private key* (Prayudi, 2005).

Algoritma yang memakai kunci publik di antaranya adalah :

* 1. Digital Signature Algorithm (DSA)
  2. Diffie-Hellman (DH)
  3. RSA
  4. ElGamal
  5. Elliptic Curve Cryptography (ECC)
  6. Dan lain sebagainya.

Proses algoritma asimetris dapat dilihat pada Gambar 2.4.

Gambar 2.4 Proses Algoritma Asimetris

### 2.2.3 Algoritma Cipher Pada Proses Asimetris Dan Simetris

1. Stream Cipher

*Stream cipher* (aliran cipher) Merupakan suatu cipher yang berasal dari hasil XOR dimana digit *plaintext* digabungkan dengan *keystream*. Dalam *stream cipher*, masing-masing digit *plaintext*-nya di enkripsi satu per satu dengan digit yang sesusai dari *keystream*.

1. Block Cipher

Merupakan suatu algoritma yang mana input dan output nya berupa satu blok, dan setiap blok terdiri dari beberapa bit (1 blok terdiri dari 64 bit atau 128 bit). *Block cipher* mempunyai banyak aplikasi, aplikasi tersebut di gunakan untuk memberikan layanan *confidentiality* (kerahasiaan), integritas data atau authentication (pengesahan pemakai), dan juga bisa memberikan layanan keystream generator untuk *stream cipher*.

### 2.2.4 Algoritma Rivest-Shamir-Adleman (RSA)

RSA merupakan salah satu kriptografi asimetris, algoritma asimetris di rancang sedemikian rupa sehingga kunci yang di pakai untuk enkripsi berbeda dari kunci yang digunakan untuk dekripsi. Lebih jauh lagi, kunci dapat dihitung dari kunci enkripsi. algoritma ini disebut algoritma kunci publik karena kunci enkripsi dapat dibuat publik sehingga orang lain dapat mengetahuinya, namun hanya orang tertentu dan sekaligus pemilik kunci dekripsi yang dapat melakukan dekripsi pesan tersebut. Dalam hal ini kunci enkripsi sering disebut dengan kunci publik dan kunci dekripsi sering disebut dengan kunci privat. Kunci privat kadang-kadang sering disebut kunci rahasia (Susanto dan Tritanto, 2011).

Rumus enkripsi :

C = Me mod n................................................................................................. 2.1

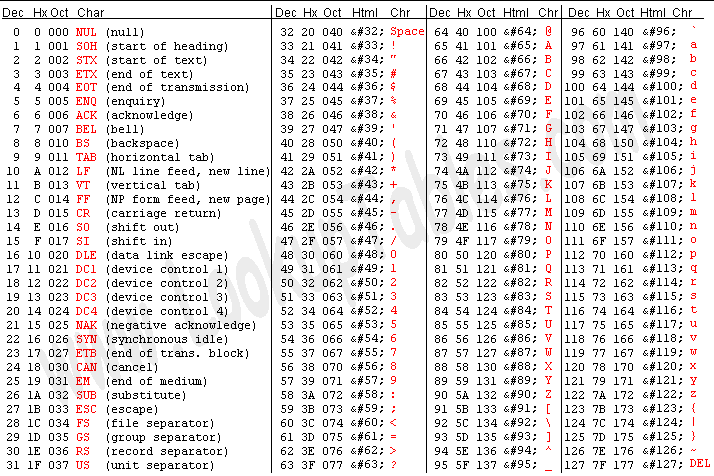
Rumus dekripsi :

M = Cd mod n = (Me)d mod n = Med mod n.................................................... 2.2

Besaran-besaran yang digunakan pada algoritma RSA :

1. p dan q bilangan acak prima (rahasia)
2. n = p\*q (tidak rahasia)
3. Qn = (p-1)\*(q-1) (rahasia)
4. e = 1 < e < Q(n) (kunci enkripsi) (tidak rahasia)
5. d = 1+kQ(n)/e (kunci dekripsi) (rahasia)
6. M (*plaintext*) (rahasia)
7. C (*ciphertext*) (tidak rahasia)

Jika didapat p = 11 dan q = 13 yang akan digunakan dalam proses enkripsi dan dekripsi dengan plaintext “TEKNIKINFORMASI”, yaitu dengan cara :

1. p = 11, q = 13, nilai p dan q merupakan bilangan prima acak.
2. n = 143, nilai n didapat dengan rumus n = p\*q.
3. Q(n) = 120, nilai Qn didapat dengan rumus Qn = (p-1)\*(q-1).
4. e = 7, nilai e didapat dengan rumus e = 1 < e < Q(n).
5. d = 103, nilai d didapat dengan rumus d = 1+kQ(n)/e.

Sumber : [www.Lookup](http://www.Lookup) Tables.com

Gambar 2.5 Tabel *ASCII*

*Plaintext* : “TEKNIKINFORMASI”

Kode *ASCII* dari *Plaintext*

Tabel 2.2 Hasil Enkripsi

| Char | ASCII |
| --- | --- |
| T | 84 |
| E | 69 |
| K | 75 |
| N | 78 |
| I | 73 |
| K | 75 |
| I | 73 |
| N | 78 |
| F | 70 |
| O | 79 |
| R | 82 |
| M | 77 |
| A | 65 |
| S | 83 |
| I | 73 |

Tabel 2.2 menunjukan nilai *ASCII* dari setiap karakter yang akan di enkripsi yaitu “84,69,75,78,73,75,73,78,70,79,82,77,65,83,73”. Selanjutnya proses enkripsi dilakukan dengan cara sebagai berikut :

Enkripsi (*cipher code*) = (*Character*)e mod n.

Sehingga didapatkan data seperti di tunjukkan dalam Tabel 2.3

Tabel 2.3 Enkripsi *Code*

| Plain Text | Ascii | e | n | Pangkat | chiper char | Chiper Text |
| --- | --- | --- | --- | --- | --- | --- |
| T | 84 | 7 | 143 | 29,509,034,655,744 | 72 | H |
| E | 69 | 7 | 143 | 7,446,353,252,589 | 108 | L |
| K | 75 | 7 | 143 | 13,348,388,671,875 | 114 | R |
| N | 78 | 7 | 143 | 17,565,568,854,912 | 78 | N |
| I | 73 | 7 | 143 | 11,047,398,519,097 | 83 | S |
| K | 75 | 7 | 143 | 13,348,388,671,875 | 114 | R |
| I | 73 | 7 | 143 | 11,047,398,519,097 | 83 | S |
| N | 78 | 7 | 143 | 17,565,568,854,912 | 78 | N |
| F | 70 | 7 | 143 | 8,235,430,000,000 | 60 | < |
| O | 79 | 7 | 143 | 19,203,908,986,159 | 40 | ( |
| R | 82 | 7 | 143 | 24,928,547,056,768 | 69 | E |
| M | 77 | 7 | 143 | 16,048,523,266,853 | 77 | M |
| A | 65 | 7 | 143 | 4,902,227,890,625 | 65 | A |
| S | 83 | 7 | 143 | 27,136,050,989,627 | 8 | BackSpace |
| I | 73 | 7 | 143 | 11,047,398,519,097 | 83 | S |

*CipherText* yang dihasilkan adalah :

“H,L,R,N,S,R,S,N,<,(,E,M,A,Backspace,S”

Tabel 2.4 menunjukan nilai *ASCII* setiap karakter yang akan di dekripsi yaitu “72,108,114,78,83,114,83,78,60,40,69,77,65,8,83”. Selanjutnya proses dari dekripsi dilakukan dengan cara sebagai berikut :

Deskripsi (*Plaintext*) = (*Cipher code*)d mod n

Sehingga didapatkan data seperti di tunjukkan data dalam tabel 2.4

Tabel 2.4 Dekripsi *Code*

| Chiper Char | d | n | Hasil Pangkat | Hasil Mod | Plain Text |
| --- | --- | --- | --- | --- | --- |
| 72 | 103 | 143 | 5.4605E+224 | 84 | T |
| 108 | 103 | 143 | 1.1073E+246 | 69 | E |
| 114 | 103 | 143 | 7.6822E+248 | 75 | K |
| 78 | 103 | 143 | 8.779E+228 | 78 | N |
| 83 | 103 | 143 | 1.616E+232 | 73 | I |
| 114 | 103 | 143 | 7.6822E+248 | 75 | K |
| 83 | 103 | 143 | 1.616E+232 | 73 | I |
| 78 | 103 | 143 | 8.779E+228 | 78 | N |
| 60 | 103 | 143 | 1.4332E+215 | 70 | F |
| 40 | 103 | 143 | 7.0674E+193 | 79 | O |
| 69 | 103 | 143 | 3.1677E+222 | 82 | R |
| 77 | 103 | 143 | 1.8424E+228 | 77 | M |
| 65 | 103 | 143 | 2.3042E+219 | 65 | A |
| 8 | 103 | 143 | 1.8788E+109 | 83 | S |
| 83 | 103 | 143 | 1.616E+232 | 73 | I |

*PlainText* yang dihasilkan adalah :  
“T,E,K,N,I,K,I,N,F,O,R,M,A,S,I”

# BAB III KERANGKA KONSEP DAN METODOLOGI PENELIT

BAB III

KERANGKA KONSEP PENELITIAN DAN METODOLOGI PENELITIAN

## Kerangka Konsep Penelitian

Gambar 3.1 Diagram Alir Kerangka Konsep Penelitian

### Kriptografi

Didalam kriptografi terdapat dua jenis kunci yaitu:

1. Kunci Simetris
2. Kunci Asimetris

Algoritma asimetris dapat dikelompokkan menjadi dua kategori yaitu :

1. Block Cipher

*Block cipher* merupakan suatu algoritma yang mana proses input dan outputnya berupa 1 blok, dan setiap blok terdiri dari beberapa bit (1 blok terdiri dari 64 bit atau 128 bit).

1. Stream Cipher

*Stream cipher* merupakan suatu algoritma yang mana setiap bit *plaintext* dengan bit kunci adalah kunci utama (kunci induk) yang digunakan untuk membangkitkan kunci acak semu. Pada kerangka konsep penilitian RSA menggunakan algoritma *block cipher*.



### Algoritma Rivest-Shamir-Adleman (RSA)

Algoritma RSA adalah algoritma kunci publik yang populer, Algoritma RSA diciptakan oleh tiga orang ilmuwan dari MIT (*Massachusset Institute of Technology*) pada tahun 1976, ilmuwan tersebut adalah Ron Rivest, Adi Shamir, dan Leonard Adleman, nama RSA sendiri diambil dari gabungan nama ketiga penemunya yaitu, (R)ivest (S)hamir (A)dleman. Algoritma ini dinilai aman karena melibatkan proses pemfaktoran bilangan prima yang besar.

Algoritma RSA merupakan salah satu algoritma kunci publik yang sampai saat ini paling banyak dikenal. Salah satu sifat dari sistem kripto dengan kunci publik adalah : D(E(m)) = E(D(m)) dengan E adalah proses enkripsi, D adalah proses dekripsi dan m adalah pesan/data. Sifat ini yang nantinya sebagai dasar untuk merancang sistem pengamanan pesan/data yang mampu menjamin keaslian dan kerahasiaan data.

Algoritma RSA merupakan algorithm *block cipher* (algoritma yang bekerja per blok data), dimana plaintext dan *ciphertext*-nya adalah bilangan bulat (*integer*) antara 0 dan n-1 untuk sebuah n tertentu. Algoritma ini bekerja dengan menghitung eksponen dari pesan (atau *plaintext*) dalam operasi *modulo* n (*modulo* = sisa pembagian). Jadi, sebelum sebuah *plaintext* dikirimkan, *plaintext* tersebut harus dipangkatkan dengan sebuah bilangan, yang biasa disebut sebagai e yang adalah kunci publik. Kemudian di sisi penerima, pesan tersandi (atau *ciphertext*) tersebut harus dipangkatkan dengan kunci miliknya sendiri, yang biasa dikenal d sebagai kunci privat, untuk memperoleh kembali pesan yang dikirimkan. Semua perhitungan dikerjakan pada operasi *modulo* n.

Pada kerangka konsep penelitian proses yang dilakukan menggunakan metode Rivest-Shamir-Adleman (RSA) adalah proses enkripsi dan dekripsi. Enkripsi merupakan proses penyandian *plaintext* menjadi *ciphertext*. Plaintext disebut dengan pesan asli, sedangkan *ciphertext* adalah pesan ter-enkrip (tersandi) yang merupakan hasil enkripsi. Sedangkan dekripsi adalah kebalikan dari enkripsi yakni mengubah *ciphertext* menjadi *plaintext*, sehingga kembali menjadi data awal/asli. Proses enkripsi metode Rivest-Shamir-Adleman (RSA) dapat dilakukan dengan cara mengikuti alur seperti yang di tunjukkan pada (Gambar 2.2). Sedangkan proses dekripsi dapat mengikuti alur seperti yang di tunjukkan pada (Gambar 2.4).

### File Teks

Pada penelitian ini akan digunakan *file* teks. *File* teks merupakan *file* yang berisi informasi-informasi dalam bentuk teks. Data yang berasal dari dokumen pengolah kata, angka yang digunakan dalam perhitungan, nama dan alamat dalam basis data merupakan contoh masukan data teks yang terdiri dari karakter, angka dan tanda baca. Masukan dan keluaran data teks direpresentasikan sebagai set karakter atau sistem kode yang dikenal oleh sistem komputer. Karakter yang umum digunakan untuk masukan dan keluaran pada komputer, salah satunya yaitu *ASCII*.

## Metode Penelitian



Gambar 3.2 Diagram Alir Metode Penelitian

### Riset Awal

Sebelum melakukan penelitian terlebih dahulu mempelajari segala hal yang terkait dengan topik penelitian. Bagian utama yang perlu dipelajari dan didalami adalah :

1. Konsep Kriptografi
2. Konsep Dasar File Teks
3. Algoritma Rivest-Shamir-Adleman

Setelah melakukan riset awal, kemudian dilakukan tahapan-tahapan seperti pada Gambar 3.2.

### Perancangan

Berdasarkan bahan dan materi riset awal yang sudah di kumpulkan, kemudian dilakukan perancangan implementasi algoritma RSA yang diperlukan untuk melakukan enkripsi dan dekripsi.

##### Proses Enkripsi

Pada tahap ini data yang berupa teks dienkripsi atau disandikan ke sebuah *file* teks yang tidak dapat dimengerti maknanya (*ciphertext*). Alur dari proses enkripsi sebagai berikut:

1. Input data teks pada *file*.
2. Teks yang ada di bagi menjadi beberapa blok kemudian diubah ke dalam bentuk bilangan *ASCII*.
3. Teks yang sudah menjadi bilangan *ASCII* akan diproses dengan perhitungan kunci publik dengan simbol e dan n.
4. Kemudian proses dari perhitungan e sebagai kunci publik dihitung dengan n sebagai nilai hasil pembagian atau *mod*/*modulo* seperti pada (pers. 2.1).
5. *Output*-nya adalah sebuah pesan acak yang tidak bisa di baca (*ciphertext*).
6. Proses Dekripsi

Pada tahap ini data teks yang ter-enkripsi dikembalikan ke bentuk data awal yang dapat dimengerti maknanya (*plaintext*). Alur dari proses dekripsi sebagai berikut :

Input data teks hasil enkripsi.

Teks yang sudah ter-enkripsi di bagi menjadi beberapa blok kemudian di ubah kedalam bentuk bilangan *ASCII*.

Teks yang sudah menjadi bilangan *ASCII* akan di proses dengan perhitungan kunci privat dengan simbol d dan n.

Kemudian proses dari perhitungan d sebagai kunci privat di hitung dengan n sebagai nilai hasil pembagian atau *mod*/*modulo* seperti pada (pers 2.2).

Output-nya adalah pesan yang bisa dibaca (*plaintext*).

### Pengujian

Melakukan enkripsi dan dekripsi pada file teks sesuai dengan kriteria data input yang sudah dispesifikasikan pada bab satu.

### Analisa Hasil

Hasil yang diperoleh dari pengujian kemudian dianalisa terutama pada hasil

enkripsi dan dekripsi file teks.

## Variabel Penelitian

Sebagai variabel penelitian dalam tugas akhir ini yang akan dikaji secara mendalam adalah metode RSA yang digunakan untuk pengamanan data teks dan menjaga kerahasiaan data.

## Data dan Tempat Penelitian

Data-data yang diperlukan :

1. Referensi-referensi dari penilitian Tugas Akhir sebelumnya
2. Tempat Penelitian : Politeknik Negeri Samarinda

# BAB IV HASIL DAN PEMBAHASAN

BAB IV  
HASIL DAN PEMBAHASAN

Bab ini membahas tentang algoritma RSA dalam melakukan enkripsi dan dekripsi pada File Teks.

## Proses Enkripsi Dan Dekripsi File Teks

RSA melibatkan kunci publik dan kunci privat. Kunci publik boleh diketahui oleh semua orang sehingga kunci publik tersebut mempunyai sifat yang tidak rahasia. Maksudnya adalah pada proses enkripsi kunci publik e dan n yang merupakan kunci publik perlu dibuat untuk mengubah pesan (*plaintext*) menjadi tidak bisa dibaca (*ciphertext*) oleh semua orang. Sedangkan kunci privat hanya diketahui oleh si pembuat kunci dan orang tertentu saja sehingga kunci privat tersebut mempunyai sifat yang rahasia. Maksudnya adalah bahwa saat akan melakukan proses dekripsi kunci privat d dan n yang merupakan kunci privat digunakan agar pesan yang tidak bisa dibaca (*ciphertext*) dapat dibaca (*plaintext*) untuk si penerima pesan.

### Menghitung Kunci Publik

Pada algoritma RSA menghitung kunci publik adalah hal yang pertama kali dilakukan saat akan mengenkripsi *file* teks. Teks yang sudah ada kemudian di enkripsi dengan menggunakan kunci publik. Alur dari proses perhitungan kunci publik adalah sebagai berikut:

1. Pilih nilai p dan q yang merupakan bilangan prima acak.
2. Hitung n yang merupakan hasil dari p\*q.
3. Hitung Q(n) yang merupakan hasil dari Q(n) = (p-1)\*(q-1).
4. Pilih nilai pada e yang merupakan kunci publik . syarat dari pembangkit kunci e ini adalah 1 < e < Q(n).

Adapun Flowchart menghitung kunci publik sebelum melakukan proses enkripsi dapat dilihat pada Gambar 4.1.



Gambar 4.1 Flowchart Proses Menghitung Kunci Publik RSA

Sebelum melakukan enkripsi, menghitung kunci publik adalah hal yang harus dilakukan seperti pada alur Gambar 4.1, tahap awal dilakukan dengan meng-input pasangan bilangan prima dimana nilai input p dan q merupakan bilangan prima acak dan n merupakan hasil dari input **p\*q**. Setelah melakukan proses tersebut nilai Q(n) di hitung dengan cara **Q(n) = (p-1)\*(q-1)**, serta Q(n) di gunakan juga untuk mencari kunci publik, kunci publik didapat dengan menggunakan cara **1 < e < Q(n)**, e adalah relatif prima dengan Q(n), relatif prima artinya nilai PBB (kunci publik, Q(n)) = 1. Setelah kunci publik didapat tidak perlu lagi meng-input nilai p dan q. Kemudian didapatlah hasil e sebagai proses sebelum melakukan enkripsi seperti pada pers 2.1 dan n digunakan dalam hal merubah (*plaintext*) menjadi (*ciphertext*) seperti pada tabel 2.3.

### Menghitung Kunci Privat

Pada algoritma RSA menghitung kunci privat adalah salah satu proses yang akan dilakukan setelah perhitungan kunci publik pada *file* teks selesai. teks yang sudah di enkripsi kemudian akan di dekripsi kan kembali dengan menggunakan kunci privat. Alur dari proses perhitungan kunci privat adalah sebagai berikut:

1. Pilih p, q, e. p dan q merupakan bilangan prima acak dan e adalah hasil dari proses perhitungan pada kunci publik.
2. Hitung Q(n) dengan cara Q(n) = (p-1)\*(q-1).
3. Input kunci publik e, e didapat dengan cara 1 < e < Q(n), e adalah relatif prima dengan Q(n), relatif prima artinya nilai PBB (kunci publik, Q(n)) = 1.
4. Pilih k, nilai k adalah salah satu bilangan bulat 1, 2, 3, ,.Qn.
5. Kunci privat dihitung dan didapat dengan cara d = 1+kQ(n)/e.

Adapun *Flowchart* menghitung kunci privat sebelum melakukan proses dekripsi dapat dilihat pada Gambar 4.2.



Gambar 4.2 *Flowchart* Proses Menghitung Kunci Privat RSA

Sebelum melakukan dekripsi, menghitung kunci privat pada alur Gambar 4.2 adalah hal yang harus dilakukan setelah melakukan proses perhitungan kunci publik seperti yang ada pada alur Gambar 4.1. Tahap awal dilakukan dengan meng-*input* p dan q merupakan bilangan prima acak dan e adalah kunci publik sehingga e berguna untuk untuk membantu dalam hal proses menghitung kunci privat, dimana e didapat dengan menggunakan cara **1 < e < Q(n)** dan e adalah relatif prima dengan Q(n). Setelah itu input k dimana nilai k merupakan bilangan bulat 1,2,3..,.Qn. Setelah itu kunci privat dihitung dan didapat dengan cara **d = 1+kQ(n)/e**. Kemudian didapat d sebagai kunci privat, dimana d akan digunakan dalam hal proses sebelum melakukan dekripsi seperti pada pers 2.2 dan n digunakan dalam hal merubah (*ciphertext*) menjadi (*plaintext*) seperti pada tabel 2.4.

## Proses Enkripsi Menggunakan Kunci Publik

Proses enkripsi menggunakan kunci pubik adalah suatu proses enkripsi menggunakan hasil dari perhitungan kunci publik yang ada pada alur (Gambar 4.1). Adapun alur dari proses enkripsi yang menggunakan perhitungan kunci publik adalah sebagai berikut :

* + - 1. *Input file* teks yang akan dienkripsi (*plaintext*).
      2. Pilih dua pasangan bilangan prima acak p dan q untuk proses enkripsi.
      3. Hitung n, nilai n digunakan untuk melakukan proses enkripsi.
      4. Hitung Qn, nilai Qn digunakan untuk mendapatkan kunci publik.
      5. Pilih nilai pada e yang merupakan kunci publik dan kunci publik ini didapatkan dengan menggunakan cara 1 < e < Q(n), e adalah relatif prima dengan Q(n).
      6. Proses enkripsi dilakukan berulang-ulang untuk semua huruf sesuai jumlah huruf pada *plaintext*.
      7. *Output*-nya adalah sebuah pesan acak yang tidak bisa di baca (*ciphertext*).

Setelah proses enkripsi maka di dapatkan teks yang tidak bisa di baca tanpa proses dekripsi.

*Flowchart* utama dalam hal enkripsi pesan dengan menggunakan algoritma RSA dapat dilihat pada Gambar 4.3.



Gambar 4.3 *Flowchart* Enkripsi Pesan Dengan Menggunakan Algoritam RSA

Pada saat melakukan enkripsi pesan hal yang pertama kali harus dilakukan adalah meng-*input plaintext* dimana nanti nya setiap karakter pada *plaintext* akan dibaca. Setelah itu input kunci publik dimana kunci publik didapat dengan menggunakan rumus 1 < e < Q(n), e adalah relatif prima dengan Q(n), relatif prima artinya nilai PBB (kunci publik, Q(n)) = 1. Kemudian hitung n dengan rumus n = p\*q. Setelah itu *plaintext* yang sudah menjadi karakter dipangkatkan dengan e, dimana e adalah kunci publik dan diproses dengan n seperti yang ada pada Tabel 2.3. Sehingga didapatlah hasil pesan berupa *ciphertext*.

*Flowchart* proses enkripsi menggunakan kunci publik pada Algoritma RSA dapat dilihat pada Gambar 4.4.



Gambar 4.4 *Flowchart* Proses Enkripsi Menggunakan Kunci Publik

Pada saat melakukan enkripsi adapun langkah-langkah yang harus dilakukan dalam hal proses merubah (*plaintext*) menjadi (*ciphertext*). Proses enkripsi yang ada pada alur Gambar 4.4 menunjukan *plaintext* di-*input*, kemudian nilai n didapat dengan rumus **n = p\*q**. Setelah itu *input* Q(n), nilai Q(n) didapat dengan rumus **Q(n) = (p-1)\*(q-1)**. kemudian *input* e, nilai e didapat dengan menggunakan rumus **1 < e < Q(n)**, e adalah relatif prima dengan Q(n). Nilai e tidak melalui proses perhitungan nilai n tetapi nilai e digunakan pada saat melakukan proses enkripsi. Selanjutnya *text* yang ada pada *file* dibaca kemudian urutan data yang ada pada *text* diurutkan dengan jumlah panjangnya huruf yang digunakan. Kemudian nilai **ab (m[i]\*\*e)** didapat dengan cara huruf-huruf yang sudah ada diubah kedalam huruf *ASCII* dipangkatkan dengan kunci publik. Setelah nilai ab didapat nilai ba di proses dengan mengikuti pers 2.1. Sehingga didapat ba yang merupakan *ciphertext* didapat dari proses enkripsi menggunakan kunci publik.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plain Text | ASCII(M) | e | n | Me | Me Mod n | cipher text |
| K | 75 | 7 | 221 | 13,348,388,671,875 | 114 | r |
| R | 82 | 7 | 221 | 24,928,547,056,768 | 108 | l |
| I | 73 | 7 | 221 | 11,047,398,519,097 | 44 | , |
| P | 80 | 7 | 221 | 20,971,520,000,000 | 24 | Cancel |
| T | 84 | 7 | 221 | 29,509,034,655,744 | 33 | ! |
| O | 79 | 7 | 221 | 19,203,908,986,159 | 105 | i |
| G | 71 | 7 | 221 | 9,095,120,158,391 | 215 | × |
| R | 82 | 7 | 221 | 24,928,547,056,768 | 108 | L |
| A | 65 | 7 | 221 | 4,902,227,890,625 | 91 | [ |
| F | 70 | 7 | 221 | 8,235,430,000,000 | 60 | < |
| I | 73 | 7 | 221 | 11,047,398,519,097 | 44 | , |

Tabel 4.1 Proses Enkripsi Menggunakan Kunci Publik

*Plaintext* : “KRIPTOGRAFI”  
*ASCII* (M) : “75,82,73,80,84,79,71,82,65,70,73”  
Me : Merupakan proses nilai dari *ASCII* pada *plaintext* di pangkatkan dengan e sebagai kunci publik  
n : Merupakan nilai dari proses hasil sisa pembagian dari suatu bilangan terhadap bilangan lainnya atau bisa disebut juga dengan *mod*/*modulo*

Misal Nilai p=13 dan q=17 kemudian hitung n, yang dimana n merupakan *mod*/*modulo* dengan rumus **n = p\*q**, setelah itu hitung nilai Q(n) yang dimana **Q(n)=(p-1)\*(q-1)**. Setelah menghitung nilai Q(n), kunci publik didapat dengan rumus **1 < e < Q(n)**, e adalah relatif prima dengan Q(n). dimana nilai dari e merupakan kunci publik dan n digunakan dalam hal merubah (*plaintext*) menjadi (*ciphertext*) seperti pada Tabel 2.3.

p = 13 , q = 17  
n = 13 \* 17 = 221   
Q(n) = (13-1)\*(17-1)  
 = (12)\*(16)  
 = 192

Faktor dari 60 : 1,2,3,4,5,6,10,12,15,20,30,60  
Faktor dari 7 : 1,7

60 dan 7 relatif prima karena hanya memiliki faktor yang sama yaitu 1.

Sehingga kunci publik e,n adalah (7,221). Adapun perhitungan proses enkripsi seperti pada Tabel 4.1 adalah sebagai berikut :

*Plaintext* = “KRIPTOGRAFI”

*ASCII* (M) = “75,82,73,80,84,79,71,82,65,70,73”

n = “221”

C = Me Mod n

C = “757 Mod n : 757 Mod 221 = 114 “

“827 Mod n : 827 Mod 221 = 108 “

“737 Mod n : 737 Mod 221 = 44 “

“807 Mod n : 807 Mod 221 = 24 “

“847 Mod n : 847 Mod 221 = 33 “

“797 Mod n : 797 Mod 221 = 105 “

“717 Mod n : 717 Mod 221 = 215 “

“827 Mod n : 827 Mod 221 = 108 “

“657 Mod n : 657 Mod 221 = 91 “

“707 Mod n : 707 Mod 221 = 60 “

“737 Mod n : 737 Mod 221 = 44 “

*CipherText* = “rl,Cancel!i×L[<,”.

## Proses Dekripsi Menggunakan Kunci Privat

Proses dekripsi menggunakan kunci privat adalah suatu proses dekripsi yang mana setelah proses enkripsi dengan menggunakan perhitungan publik selesai. Maka hasil dari proses enkripsinya akan didekripsikan kembali dengan menggunakan proses perhitungan kunci privat yang ada pada RSA. Adapun alur dari proses dekripsi yang menggunakan kunci privat adalah sebagai berikut :

* + - 1. *Input ciphertext*.
      2. Input p,q,k. dimana p,q didapat dengan cara n = p\*q dan k di dapat dengan memilih nilai k, dimana k adalah salah satu bilangan bulat 1, 2, 3, ,.Qn.
      3. Proses dekripsi dilakukan berulang-ulang untuk semua huruf sesuai jumlah huruf pada *ciphertext*.
      4. *Output*-nya adalah pesan yang bisa dibaca (*plaintext*)

Setelah melalui proses enkripsi pesan yang tidak dapat dibaca kemudian bisa dibaca dengan menggunakan proses perhitungan kunci privat.

*Flowchart* utama dalam hal dekripsi pesan dengan menggunakan Algoritma RSA dapat dilihat pada Gambar 4.5.



Gambar 4.5 *Flowchart* Dekripsi Pesan Dengan Menggunakan Algoritma RSA

Setelah melakukan enkripsi pesan seperti pada alur Gambar 4.3, Proses dekripsi dilakukan seperti pada alur Gambar 4.5. Hal yang pertama kali dilakukan pada saat akan mengdekripsi pesan adalah meng-*input ciphertext*, dimana nanti nya setiap karakter pada *ciphertext* akan dibaca. Setelah itu *input* kunci privat dimana kunci privat didapat dengan rumus d = 1+kQ(n)/e, kemudian input n dimana nilai n dihitung dengan rumus n = p\*q. setelah itu *chipertext* yang sudah dibaca menjadi karakter dipangkatkan dengan kunci privat dan di proses dengan n seperti yang ada pada Tabel 2.4. Sehingga didapatlah hasil pesan berupa *plaintext*.

*Flowchart* proses dekripsi menggunakan kunci privat pada Algoritma RSA dapat dilihat pada Gambar 4.6.



Gambar 4.6 *Flowchart* Proses Dekripsi Menggunakan Kunci Privat

Setelah melakukan proses enkripsi menggunakan kunci publik seperti pada alur Gambar 4.4. Kemudian proses dekripsi dilakukan dengan menggunakan kunci privat seperti pada alur Gambar 4.6. Pada saat akan melakukan dekripsi adapun langkah-langkah yang harus dilakukan dalam hal proses merubah (*chipertext*) menjadi (*plaintext*) proses dekripsi yang ada pada alur Gambar 4.6 menunjukan *chipertext* di-*input* kemudian nilai n didapat dan dihitung dengan menggunakan rumus **n = p\*q**, dan k di dapat dengan memilih nilai k, dimana nilai k adalah salah satu bilangan bulat 1, 2, 3, ,.Qn. kemudian hasil dari proses enkripsi yang ada pada alur Gambar 4.4 dipangkatkan dengan kunci privat (**c=i\*\*k/n**) berdasarkan pada pemrograman berbasis python kemudian nilai c di proses dengan mengikuti pers 2.2. Sehingga didapat c yang merupakan *plaintext* dari proses dekripsi menggunakan kunci privat.

Tabel 4.2 Proses Deskripsi Menggunakan Kunci Privat

| Chiper Text | ASCII(C) | d | n | Cd | Cd Mod n | Plain Text |
| --- | --- | --- | --- | --- | --- | --- |
| r | 114 | 55 | 221 | 1,35E+116 | 75 | K |
| l | 108 | 55 | 221 | 6,89E+114 | 82 | R |
| , | 44 | 55 | 221 | 2,45E+94 | 73 | I |
| Cancel | 24 | 55 | 221 | 8,16E+79 | 80 | P |
| ! | 33 | 55 | 221 | 3,30E+87 | 84 | T |
| i | 105 | 55 | 221 | 1,46E+114 | 79 | O |
| × | 215 | 55 | 221 | 1,92E+131 | 71 | G |
| l | 108 | 55 | 221 | 6,89E+114 | 82 | R |
| [ | 91 | 55 | 221 | 5,59E+110 | 65 | A |
| < | 60 | 55 | 221 | 6,29E+101 | 70 | F |
| , | 44 | 55 | 221 | 2,45E+94 | 73 | I |

*ASCII* (C) : “114,108,44,24,33,105,215,108,91,60,44”  
Cd : Merupakan proses nilai dari *ASCII* (C) pada *chipertext* di pangkatkan dengan d sebagai kunci privat  
n : Merupakan nilai dari proses hasil sisa pembagian dari suatu bilangan terhadap bilangan lainnya atau bisa disebut juga dengan *mod*/*modulo*

Jika nilai dari e dan n (7,221) adalah perhitungan proses kunci publik seperti pada Tabel 4.1. Maka lakukanlah proses perhitungan nilai dari d sebagai kunci privat dengan menggunakan rumus **d=1+kQ(n)/e**.

d = 1+(2\*(192))/7

d = 1+(384/7)

d = 1+54

d = 55

Sehingga diperoleh proses perhitungan kunci privat d,n adalah (55,221). Adapun perhitungan proses dekripsi seperti pada Tabel 4.2 adalah sebagai berikut :

*ChiperText* = “rl,Cancel!i×L[<,”

*ASCII* (C) = “114,108,44,24,33,105,215,108,91,60,44”

n = “221”

M = Cd Mod n

M = “11455 Mod n : 11455 Mod 221 = 75 “

“10855 Mod n : 10855 Mod 221 = 82 “

“4455 Mod n : 4455 Mod 221 = 73 “

“2455 Mod n : 2455 Mod 221 = 80 “

“3355 Mod n : 3355 Mod 221 = 84 “

“10555 Mod n : 10555 Mod 221 = 79 “

“21555 Mod n : 21555 Mod 221 = 71 “

“10855 Mod n : 10855 Mod 221 = 82 “

“9155 Mod n : 9155 Mod 221 = 65 “

“6055 Mod n : 6055 Mod 221 = 70 “

“4455 Mod n : 4455 Mod 221 = 73 “

*PlainText* = “KRIPTOGRAFI”

1. **Pengujian**

Pengujian dilakukan untuk mengimplementasikan Algoritma kriptografi RSA untuk mengenkripsi data teks yang ada pada file menggunakan pemrograman Python. Editor pemrograman yang digunakan adalah eclipse. Dengan data teks yang tersimpan di dalam file (.txt). Contoh data yang digunakan untuk proses enkripsi adalah teks “KRIPTOGRAFI”. Selanjutnya data teks yang ada pada noteped diproses lagi untuk mencari *ASCII Number* dengan *Plain Char* “KRIPTOGRAFI” menggunakan pemrograman yang telah dibuat. Sehingga diperoleh *ASCII Number* seperti yang ada pada Tabel 4.3.

Tabel 4.3 *Plain Char* Dan *ASCII Number*

|  |  |
| --- | --- |
| Plain Char | ASCII Number |
| K | 75 |
| R | 82 |
| I | 73 |
| P | 80 |
| T | 84 |
| O | 79 |
| G | 71 |
| R | 82 |
| A | 65 |
| F | 70 |
| I | 73 |

Setelah mendapatkan *ASCII Number* seperti pada Tabel 4.3 untuk melakukan proses enkripsi selanjutnya diperlukan pasangan bilangan prima (p dan q) adalah 13 dan 17. Dari pasangan bilangan prima yang digunakan diperoleh n = 221, qn = 192, dan kunci publik = 7. Kemudian proses enkripsi dilakukan dengan mengikuti alur pada Gambar 4.4. Sehingga diperoleh hasil enkripsi berupa numerik seperti yang ditunjukan pada Tabel 4.4.

Tabel 4.4 Hasil Enkripsi Pada Program

|  |  |  |  |
| --- | --- | --- | --- |
| PlainText | | CipherText | |
| Char | ASCII | ASCII | Char |
| S | 83 | 941 | έ |
| a | 97 | 791 | ̗' |
| m | 109 | 1093 | х |
| a | 97 | 791 | ̗' |
| r | 114 | 1003 | ϫ |
| i | 105 | 117 | u |
| n | 110 | 1035 | Ћ |
| d | 100 | 803 | ̣' |
| a | 97 | 791 | ̗' |
|  | 32 | 1055 | П |
| A | 65 | 761 | ˹ |

Setelah melakukan proses enkripsi seperti pada Tabel 4.4. Dihasilkan teks proses enkripsi adalah “rl,Cancel!i×l[<,”.

Selanjutnya proses dekripsi dilakukan dengan mengikuti alur pada Gambar 4.6. untuk melakukan proses dekripsi selanjutnya diperlukan pasangan bilangan prima (p dan q) adalah 13 dan 17 dengan kunci privat = 55. Sehingga diperoleh data *plaintext* “KRIPTOGRAFI” dalam bentuk karakter seperti yang ada pada Tabel 4.5.

Tabel 4.5 Hasil Dekripsi Pada Program

| CipherText | | PlainText | |
| --- | --- | --- | --- |
| ASCII | Char | Char | ASCII |
| 114 | r | K | 75 |
| 108 | l | R | 82 |
| 44 | , | I | 73 |
| 24 |  | P | 80 |
| 33 | ! | T | 84 |
| 105 | i | O | 79 |
| 215 | × | G | 71 |
| 108 | l | R | 82 |
| 91 | [ | A | 65 |
| 60 | < | F | 70 |
| 44 | , | I | 73 |

Setelah melakukan proses dekripsi seperti pada Tabel 4.5. Dihasilkan teks proses dekripsi adalah “KRIPTOGRAFI”.

Pengujian lain dilakukan dengan membuat data *text* berisi huruf dan angka pada notepad dengan data yang tersimpan kedalam notepad dengan jumlah data yang berbeda dengan data pada Tabel 4.3. Contoh data yang akan digunakan untuk proses enkripsi adalah teks “Samarinda Angkatan 2014”. Selanjutnya data teks yang ada pada noteped diproses lagi untuk mencari ASCII Number dengan *Plain Char* “Samarinda Angkatan 2014” menggunakan pemrograman yang telah dibuat. Sehingga diperoleh *ASCII Number* seperti yang ada pada Tabel 4.6.

Tabel 4.6 *Plain Char* Dan *ASCII Number*

| Plain Char | ASCII Number |
| --- | --- |
| S | 83 |
| a | 97 |
| m | 109 |
| a | 97 |
| r | 114 |
| i | 105 |
| n | 110 |
| d | 100 |
| a | 97 |
|  | 32 |
| A | 65 |
| n | 110 |
| g | 103 |
| k | 107 |
| a | 97 |
| t | 116 |
| a | 97 |
| n | 110 |
|  | 32 |
| 2 | 50 |
| 0 | 48 |
| 1 | 49 |
| 4 | 52 |

Setelah mendapatkan *ASCII Number* seperti pada Tabel 4.6 untuk melakukan proses enkripsi selanjutnya diperlukan pasangan bilangan prima (p dan q) adalah 31 dan 37. Dari pasangan bilangan prima yang digunakan diperoleh n = 1147, qn = 1080, dan kunci publik = 7. Kemudian proses enkripsi dilakukan dengan mengikuti alur pada Gambar 4.4. Sehingga diperoleh hasil enkripsi berupa numerik seperti yang ditunjukan pada Tabel 4.7.

Tabel 4.7 Hasil Enkripsi Pada Program

| PlainText | | CipherText | |
| --- | --- | --- | --- |
| Char | ASCII | ASCII | Char |
| S | 83 | 941 | έ |
| a | 97 | 791 | ̗' |
| m | 109 | 1093 | х |
| a | 97 | 791 | ̗' |
| r | 114 | 1003 | ϫ |
| i | 105 | 117 | u |
| n | 110 | 1035 | Ћ |
| d | 100 | 803 | ̣' |
| a | 97 | 791 | ̗' |
|  | 32 | 1055 | П |
| A | 65 | 761 | ˹ |
| n | 110 | 1035 | Ћ |
| g | 103 | 82 | R |
| k | 107 | 81 | Q |
| a | 97 | 791 | ̗' |
| t | 116 | 277 | ĕ |
| a | 97 | 791 | ̗' |
| n | 110 | 1035 | Ћ |
|  | 32 | 1055 | П |
| 2 | 50 | 69 | E |
| 0 | 48 | 973 | ύ |
| 1 | 49 | 9 | \t |
| 4 | 52 | 1034 | Њ |

Setelah melakukan proses enkripsi seperti pada Tabel 4.7. Dihasilkan teks proses enkripsi adalah “έ̗'х̗'ϫuЋ ̣'̗'П˹ЋRQ̗'ĕ̗'ЋПEύ\tЊ”.

Setelah melakukan proses enkripsi seperti pada Tabel 4.7 Selanjutnya proses dekripsi dilakukan dengan mengikuti alur pada Gambar 4.6. untuk melakukan proses dekripsi selanjutnya diperlukan pasangan bilangan prima (p dan q) adalah 31 dan 37 dengan kunci privat = 463. Sehingga diperoleh data *plaintext* “Samarinda Angkatan 2014” dalam bentuk karakter seperti yang ada pada Tabel 4.8.

Tabel 4.8 Hasil Dekripsi Pada Program

| CipherText | | PlainText | |
| --- | --- | --- | --- |
| ASCII | Char | Char | ASCII |
| 941 | έ | S | 83 |
| 791 | ̗' | a | 97 |
| 1093 | х | m | 109 |
| 791 | ̗' | a | 97 |
| 1003 | ϫ | r | 114 |
| 117 | u | i | 105 |
| 1035 | Ћ | n | 110 |
| 803 | ̣' | d | 100 |
| 791 | ̗' | a | 97 |
| 1055 | П |  | 32 |
| 761 | ˹ | A | 65 |
| 1035 | Ћ | n | 110 |
| 82 | R | g | 103 |
| 81 | Q | k | 107 |
| 791 | ̗' | a | 97 |
| 277 | ĕ | t | 116 |
| 791 | ̗' | a | 97 |
| 1035 | Ћ | n | 110 |
| 1055 | П |  | 32 |
| 69 | E | 2 | 50 |
| 973 | ύ | 0 | 48 |
| 9 | \t | 1 | 49 |
| 1034 | Њ | 4 | 52 |

Setelah melakukan proses dekripsi seperti pada Tabel 4.8. Dihasilkan teks proses dekripsi adalah “Samarinda Angkatan 2014”.

1. **Analisa Hasil**

Berdasarkan hasil pada contoh pengujian Algoritma RSA menggunakan kunci asimteris. Kunci asimetris dibangun menggunakan pasangan bilangan prima. Pada penelitian ini diterapkan algoritma RSA dalam melakukan proses enkripsi pesan teks (*plaintext*) menjadi pesan acak (*ciphertext*) dan proses dekripsi pesan acak menjadi pesan teks aslinya berhasil dilakukan. Adapun kekurangan implementasi algoritma RSA dalam penelitian ini adalah pasangan bilangan prima berurutan terkecil yang dapat digunakan untuk menentukan kunci publik dan kunci privat adalah 11 dan 13. Kemudian untuk pasangan bilangan prima terbesar pada penelitian ini dibatasi sampai maksimal 997.

# BAB V PENUTUP

BAB V

**PENUTUP**

## Kesimpulan

Berdasarkan hasil penelitian, maka diperoleh kesimpulan sebagai berikut :

1. RSA dapat digunakan untuk enkripsi suatu data dalam bentuk file teks.
2. Kunci privat di peroleh dari hasil perhitungan kunci publik.
3. RSA menggunakan kunci asimetris dalam melakukan proses enkripsi dan dekripsi.
4. Kunci Asimetris dibangun berdasarkan pasangan bilangan prima.
5. Pasangan bilangan prima berurutan terkecil yang bisa digunakan pada aplikasi yaitu 11 dan 13. Pasangan bilangan prima terbesar yang digunakan dibatasi sampai 997.

## Saran

Berikut adalah beberapa saran untuk pengembangan lebih lanjut tentang tugas akhir ini :

1. Penggunaan RSA digunakan untuk melakukan enkripsi dan dekripsi pada jenis data selain teks.
2. Membangun aplikasi untuk mengimplementasikan RSA.

DAFTAR PUSTAKA

Andriyamto. T, Pardede C.L.D. 2008. *Studi Dan Perbandingan Algoritma IDEA Dan Algoritma Blowfish*. Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008). ISSN 1411-6286.

Arief A, Saputra R. 2016*. Implementasi Kriptografi Kunci Publik Dengan Algoritma RSA-CRT Pada Aplikasi Instant Messaging*. Scientific Journal Of Information. Volume 3. Nomor 1. Hal 46-54.

Arief M, Fitriyani, Ikhsan N. 2015. *Kriptografi RSA Pada Aplikasi File Transfer Client-Server Based*. Jurnal Ilmiah Teknolog informasi Terapan. Volume 1, No 3, ISSN 2407-3911.

Arifin. Z. 2009. *Studi Kasus Penggunaan Algoritma RSA Sebagai Algoritma Kriptografi Yang Aman*. Jurnal Informatika Mulawarman. Vol 4, No. 3, Universitas Mulawarman.

Busran, Putra A.N. 2014. *Rekayasa Perangkat Lunak Kriptografi Menggunakan Algoritma RSA Pada Sistem Kemanan File Berbasis Java*. Jurnal TEKNOIF. Vol. 2 No. 1, ISSN 2338-2724.

Ginting A, Isnanto R.R, Windasari P.I. 2015. *Implementasi Algoritma Kriptografi RSA Untuk Enkripsi Dan Dekripsi Email*. Jurnal Teknologi Dan Sistem komputer. Vol. 3, No. 2, ISSN 2338-0403.

Listiyono. H. 2009. *Implementasi Algoritma Kunci Public Pada Algoritma RSA. Dinamika Informatika*. Vol 1 No. 2, ISSN 2085-3343.

Pradipta. A. 2016. *Implementasi Metode Caesar Chiper Alphabet Majemuk Dalam Kriptografi Untuk Pengamanan Informasi*. Indonesian Journal on Networking and Security. Vol. 5 No. 2, ISSN 2345-6654.

Pahrizal dan Pratama D, 2016. *Implementasi Algoritma RSA untuk Pengamanan Data Berbentuk Teks*. Jurnal Pseudocode, Volume III Nomor 1, Februari 2016, ISSN 2355 – 5920.

Prayudi Y, Halik I. 2005. *Studi Dan Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/Dekripsi Data*. Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005). Hal 149-158.

Primartha. R. 2011. *Penerapan Enkripsi Dan Dekripsi File Menggunakan Algoritma Data Encryption Standard (DES)*. Jurnal Sistem Informasi (JSI). Vol 3, No. 2, ISSN 2355-4641.

Rivest R.L, Shamir A, dan Adleman L. 1977. *A Method For Obtaining Digital Signatures and Public-Key Cryptosystems*. Laboratory for Computer Science, Massachusetts Institute of Technology.Cambridge.

Satriawan I.W.D, Sasmita I.G.M.A, Bayupati I.P.A. 2014. *Aplikasi Enkripsi SMS Dengan Metode RSA Pada Smartphone Berbasis Android. Merpati*. Volume 2. Nomor 2. Hal 127-134.

Susanto A, Tritanto R. 2011. *Kombinasi Algoritma RSA Dan Algoritma Cipher Transposisi Untuk Keamanan Database*. Jurnal Dian. Volume 11. Nomor 3.

LAMPIRAN

Kode Program :

|  |
| --- |
| Import math  # Pembangkit Kunci Algoritma RSA  # pilih dua bilangan prima p dan q  p = int(input("Masukkan nilai p: "))  q = int(input("Masukkan nilai q: "))  print("p:", p, " q:", q)  # hitung qn = (p-1)(q-1)  qn = (p - 1) \* (q - 1)  print("qn: ", qn)  # pilih e coprime dari qn  e = 7  while (math.gcd(e, qn) != 1):  e += 1;    print("e: ", e)  # hitung d sehingga d\*e congruent dengan 1 mod qn  d = 2  while d \* e % qn != 1:  d += 1;    print("d:", d)  # kunci publik dan private  print("\nJadi \nK publik: ", e)  print("K private: ", d)  # enkripsi |

|  |
| --- |
| import math  # Enkripsi Algoritma RSA  p = int(input("Masukkan nilai p: "))  q = int(input("Masukkan nilai q: "))  n = p \* q  print("n: ", n)  qn = (p - 1) \* (q - 1)  print("qn: ", qn)  e = 7  while (math.gcd(e, qn) != 1):  e += 1;    print("e: ", e)  m = []  cryp = []  f = open('algoritmarsa.txt', 'r')  for i in f:  data = i  f.close()  jmlhuruf = len(data)  for i in range(jmlhuruf):  m.append(data[i])  print(m)  for i in range(jmlhuruf):    ab = ord(m[i]) \*\* e  ba = ab % n  print(m[i], ord(m[i]), ba)    cryp.append(ba)    print(cryp)  f = open('enkriprsa.txt', 'w')  for i in cryp:  f.write(str(i) + '\n') |

|  |
| --- |
| # Dekripsi Algoritma RSA  p = int(input("Masukkan nilai p: "))  q = int(input("Masukkan nilai q: "))  k = int(input("Masukkan kunci privat: "))  m = []  cryp = []  f = open('enkriprsa.txt', 'r')  for i in f:  cryp.append(eval(i))  f.close()  print(cryp)  n = p \* q  f = open('dekriprsa.txt', 'w')  for i in cryp:  # print(i[0])  c = i \*\* k % n  m.append(chr(c))  f.write(chr(c))  print(m)  f.close() |