KETAHANAN ALGORITMA RSA TERHADAP *BRUTE FORCE ATTACK*

SKRIPSI

Oleh:
LUTFI WICAKSONO
NIM. 09610094

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013

KETAHANAN ALGORITMA RSA TERHADAP *BRUTE FORCE ATTACK*

SKRIPSI

Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Sains (S.Si)

Oleh:
LUTFI WICAKSONO
NIM. 09610094

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2013

KETAHANAN ALGORITMA RSA TERHADAP *BRUTE FORCE ATTACK*

SKRIPSI

Oleh:
LUTFI WICAKSONO
NIM. 09610094

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 2 Juli 2013

Pembimbing I,	Pembimbing II,
Abdussakir, M.Pd	Fachrur Rozi, M.Si
NIP. 19751006 200312 1 001	NIP. 19800527 200801 1 012

Mengetahui,
Ketua Jurusan Matematika

Abdussakir, M.Pd
NIP. 19751006 200312 1 001

KETAHANAN ALGORITMA RSA TERHADAP *BRUTE FORCE ATTACK*

SKRIPSI

Oleh:
LUTFI WICAKSONO
NIM. 09610094

Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Sains (S.Si)
Tanggal: 10 September 2013

Penguji Utama	: Hairur Rahmah, M.Si NIP. 19800429 200604 1 003	.....
Ketua Penguji	: Mohammad Jamhuri NIP. 19810502 200501 1 004	.....
Sekretaris Penguji	: Abdussakir, M.Pd NIP. 19751006 200312 1 001	.....
Anggota Penguji	: Fachrur Rozi, M.Si NIP. 19800527 200801 1 012	.....

Mengesahkan,
Ketua Jurusan Matematika

Abdussakir, M.Pd
NIP. 19751006 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Lutfi Wicaksono

NIM : 09610094

Jurusan : Matematika

Fakultas : Sains dan Teknologi

menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 2 Juli 2013
Yang membuat pernyataan,

Lutfi Wicaksono
NIM. 09610094

MOTTO

Tidak ada masalah yang tidak bisa diselesaikan selama ada komitmen bersama untuk menyelesaikannya.

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk:

Alm. Bapak Siswoyo dan Ibu Siti Nikmaturrohmah yang telah memberikan segalanya buat penulis

Dan untuk semua orang yang ikut berjuang dalam menyelesaikan skripsi ini
I love you ☺

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Syukur *Alhamdulillah* penulis penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik, hidayah, dan inayah-Nya, sehingga penulis dapat menyelesaikan studi di Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus menyelesaikan skripsi ini dengan baik.

Keberhasilan penulisan skripsi ini tidak lepas dari bantuan, pengarahan, dan bimbingan dari berbagai pihak, baik berupa pikiran, motivasi, tenaga, maupun doa, dan restu. Karena itu penulis mengucapkan terima kasih kepada:

1. Prof. Dr. H. Mudjia Rahardja, M.Si, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. drh. Hj. Bayyinatul Muchtaromah, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Abdussakir, M.Pd, selaku Ketua Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang sekaligus pembimbing skripsi yang telah memberikan motivasi dan bimbingan mulai semester satu hingga semester akhir.
4. Abdul Aziz, M.Si, selaku dosen wali yang telah memberikan bimbingan mulai semester satu hingga akhir.

5. Fachrur Rozi, M.Si, selaku dosen pembimbing skripsi, yang telah memberikan bimbingan dengan baik sehingga penulis dapat menyelesaikan skripsi ini.
6. Segenap sivitas akademika Jurusan Matematika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.
7. Ibu Siti Nikmaturrohmah atas perhatian dan do'anya yang tidak pernah putus.
8. Adik-adik penulis Febriani Veronika, dan Dion Ari Sasongko yang memberi motivasi penulis untuk menjadi teladan.
9. Arief Wicaksono, Adi Wardhana, Hafiz Zahiri dan teman-teman kos yang tidak bisa penulis sebut semua, terima kasih atas segala bantuannya baik berupa waktu, tenaga maupun pikiran.
10. Sahabat-sahabat senasib seperjuangan mahasiswa Jurusan Matematika 2009, terima kasih atas segala pengalaman berharga dan kenangan terindah saat menuntut ilmu bersama.
11. Semua pihak yang tidak dapat penulis sebutkan satu persatu yang turut mendukung kelancaran penyempurnaan skripsi ini.

Semoga skripsi ini dapat memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. *Amin Ya Rabbal Alamin*.

Wassalamu'alaikum Wr. Wb.

Malang, September 2013

Penulis

DAFTAR TABEL

ABSTRAK

Wicaksono, Lutfi. 2013. **Ketahanan Algoritma RSA terhadap *Brute Force Attack***. Skripsi. Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
Pembimbing: (I) Abdussakir, M.Pd. (II) Fachrur Rozi, M.Si.

Kata Kunci: Algoritma RSA, *Brute Force Attack*.

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 mutahir.

Brute force attack adalah metode mengalahkan skema kriptografi dengan mencoba semua kemungkinan *password* atau kunci. *Brute force attack* memungkinkan bisa menyerang kunci privat di hampir semua skema kriptografi, tipe serangan ini bergantung pada ukuran kunci dan mekanisme pada enkripsi yang digunakan.

Semakin besar ukuran kunci dari kunci privat akan semakin sulit dibobol oleh *brute force attack,* kriptografi kunci publik sangat ditentukan oleh kuncinya. Semakin sulit pemecahan algoritma kuncinya maka tingkat keamanannya semakin tinggi.

Pada penulisan skripsi selanjutnya dapat meneruskan kunci privat berjumlah 16 digit, tetapi harus menggunakan komputer lebih canggih lagi untuk melakukan pendekripsian pesan.

ABSTRACT

Wicaksono, Lutfi. 2013. **RSA algorithm robustness against of Brute Force Attack**. Thesis. Department of Mathematics, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang.
Supervisor: (I) Abdussakir, M. Pd. (II) Fachrur Rozi, M.Si.

Keywords: RSA algorithm, Brute Force Attack

In the cryptography, RSA is an algorithm for public key encryption. This algorithm is the first algorithm known the most suitable for signing and for encryption. It is also one of the first major discovery in public key cryptography. RSA is still widely used in electronic commerce protocols and believed very secure because it is given the keys which are quite long and its applications are extremely advanced.

Brute force attack is a method of defeating a cryptographic scheme by trying all of possible passwords or keys. Brute force attack allows can strike the private key at almost all of cryptographic scheme, this type of attack depends on the size of the key and the encryption mechanism used.

The larger the key size of the private key the more difficult to be cracked by brute force attack, public key cryptography is largely determined by the key. The more difficult the key algorithms the more increase high level of security.

At the writing of the private key can then forward the amount to 16 digits, but must use more sophisticated computers to perform description of the message.

الملخص

ويجاكسونو، لطفي. 2013. **RSA خوارزمية متانة ضد هجوم القوة الغاشمة** . أطروحة. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة ولاية الإسلامية مولانا مالك إبراهيم مالانج.
المشرف: (١) عبد الشاكر، الماجستير. (٢) فخر الرازي، الماجستير.

كلمات البحث: خوارزمية RSA ، هجوم القوة الغاشمة

في الترميز، هو خوارزمية RSA للتشفير المفتاح العام (التشفير بالمفتاح العمومي). هذه الخوارزمية هو أول خوارزمية المعروف هو الانسب لوسم (توقيع) والتشفير (التشفير) واحدة من أول اكتشاف كبير في الترميز بالمفتاح العمومي. لا يزال يستخدم على نطاق واسع في RSA بروتوكولات التجارة الإلكترونية، ويعتقد آمنة جدا لأن إعطاء المفاتيح هي طويلة جدا ومتقدمة للغاية تطبيقاتها.

هجوم القوة الغاشمة هي طريقة لهزيمة نظام التشفير من خلال محاولة كل كلمات السر أو مفاتيح ممكن. هجوم القوة الغاشمة يسمح المفتاح الخاص يمكن أن يضرب في أي مخطط التشفير تقريبا، وهذا النوع من الهجوم يعتمد على حجم المفتاح وآلية التشفير المستخدمة.

وكلما زاد حجم المفتاح من المفتاح الخاص يكون من الصعب على نحو متزايد إلى أن تصدع من قبل القوة الغاشمة الهجوم، يتم تحديد الترميز بالمفتاح العمومي إلى حد كبير بواسطة المفتاح. أكثر صعوبة خوارزميات مفتاح حل المستوى متزايد من الأمن.

في كتابة هذا المفتاح الخاص يمكن بعد ذلك إعادة توجيه المبلغ إلى 16 رقما، ولكن يجب أن تستخدم أجهزة الكمبيوتر أكثر تطورا لأداء وصف الرسالة.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Matematika sebagai ilmu pengetahuan dasar memegang peranan yang sangat penting dalam perkembangan ilmu pengetahuan lain di dunia. Pada penelitian ini akan diberikan suatu analisis matematika yang diaplikasikan dalam matematika terapan, selanjutnya akan dikembangkan suatu metode dari metode yang telah ada sebelumnya di bidang kriptografi.

Kriptografi adalah ilmu yang mempelajari bagaimana melakukan enkripsi dan dekripsi, dengan memanfaatkan model matematika tertentu. Kriptografi diilhami dengan teknik enkripsi atau teknik penyandian yang mengubah sebuah pesan yang dapat dibaca (*plaintext*) menjadi sebuah pesan yang acak dan sulit diartikan. Untuk dapat membaca pesan yang terenkripsi diperlukan proses terbalik dari enkripsi yang disebut dekripsi (Kurniawan, 2008).

Cryptosystem adalah prosedur secara matematika, bagaimana suatu *plaintext* diubah menjadi *ciphertext*. *Cryptanalysis* adalah ilmu untuk menemukan suatu metode yang dapat membongkar algoritma kriptografi, dengan cara menganalisis algoritma tersebut. Orang yang melakukan *cryptanalysis* adalah *cryptoanalist.* Dari orang-orang inilah, dapat diketahui apakah algoritma kriptografi itu lemah atau kuat (Kurniawan, 2008).

Algoritma RSA, ditemukan oleh 3 orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976, yaitu: Ron (R)ivest, Adi (S)hamir, dan

Leonard (A)dleman. RSA merupakan salah satu dari *public key cryptosystem* yang sangat sering digunakan untuk memberikan kerahasiaan terhadap keaslian suatu data digital. Keamanan enkripsi dan dekripsi data model ini terletak pada kesulitan untuk memfaktorkan modulus *n* yang sangat besar. Operasi RSA, baik enkripsi, dekripsi, penandaan, atau verifikasi intinya adalah eksponensial terhadap modul. Proses perhitungan ini ditunjukkan oleh rangkaian dari multiplikasi terhadap modul (Haro, 2006).

Dalam beberapa artikel, penulis menemukan pernyataan bahwa algoritma RSA tahan sepenuhnya dalam *brute force attack,* dalam hal ini penulis akan memecahkan masalah tersebut dengan menjelaskan kebenaran dari pernyataan tersebut.

Allah SWT berfirman tentang pembuktian kebenaran pada surat Al-Baqarah ayat 111:

وَقَالُوا۟ لَن يَدْخُلَ ٱلْجَنَّةَ إِلَّا مَن كَانَ هُودًا أَوْ نَصَٰرَىٰ ۗ تِلْكَ أَمَانِيُّهُمْ ۗ قُلْ هَاتُوا۟ بُرْهَٰنَكُمْ إِن كُنتُمْ صَٰدِقِينَ ﴿١١١﴾

Artinya: *“ dan mereka (Yahudi dan Nasrani) berkata: "Sekali-kali tidak akan masuk surga kecuali orang-orang (yang beragama) Yahudi atau Nasrani". demikian itu (hanya) angan-angan mereka yang kosong belaka. Katakanlah: "Tunjukkanlah bukti kebenaranmu jika kamu adalah orang yang benar”.*

Untuk menolak dan membatalkan anggapan bahwa yang akan masuk surga, hanyalah orang-orang Yahudi, demikian juga orang-orang Nasrani beranggapan bahwa yang akan masuk surga hanyalah orang-orang Nasrani, Allah SWT memberikan penegasan bahwa anggapan mereka itu hanyalah angan-angan

yang timbul dari khayalan mereka saja, angan-angan mereka meskipun disebutkan secara global, namun maknanya mencakup arti yang luas, yaitu angan-angan mereka agar terhindar dari siksa serta anggapan bahwa yang bukan golongan mereka akan terjerumus ke dalam siksa, dan tidak memperoleh nikmat sedikitpun. Itulah sebabnya maka dalam ayat itu angan-angan mereka dinyatakan dalam bentuk jamak. Dalam hal ini Allah SWT seakan-akan meminta bukti kebenaran yang menguatkan anggapan mereka masing-masing kalau mereka masing-masing dapat mengemukakan bukti-bukti yang benar maka dugaan mereka benar. Akan tetapi dari susunan ayat tidak demikian yang terpaham, meskipun pada arti lahir ayat terdapat tuntunan mengemukakan bukti, namun menurut maknanya menyatakan ketidakbenaran dakwaan mereka masing-masing karena mereka masing-masing memang tidak akan dapat mengemukakan bukti. Dalam ayat ini terdapat isyarat, bahwa sesuatu pendapat yang tidak didasarkan pada bukti-bukti yang benar tidaklah boleh diterima (Anonim, 2012).

Ayat ini memotivasi penulis untuk menyelesaikan permasalahan dalam menjelaskan kebenaran apakah algortitma RSA tahan terhadap *brute force attack*. Keamanan algorima RSA akan diuji dengan *brute force attack*. *Brute force attack* adalah metode mengalahkan skema kriptografi dengan mencoba semua kemungkinan password atau kunci. *Brute force attack* memungkinkan bisa menyerang kunci privat di hampir semua skema kriptografi, tipe serangan ini bergantung pada ukuran kunci dan mekanisme pada enkripsi yang digunakan (LastBit, 2005).

Dari paparan di atas, maka penulis ingin menganalisis permasalahan tersebut dalam skripsi ini dengan judul "*Ketahanan Algoritma RSA terhadap Bruce Force Attack*".

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka dalam skripsi ini difokuskan pada permasalahan tentang bagaimana ketahanan algoritma RSA terhadap *brute force attack* dikaji dari sisi matematikanya.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penulisan skripsi ini adalah untuk mengetahui ketahanan algoritma RSA terhadap *brute force attack* dikaji dari sisi matematikanya.

1.4 Batasan Masalah

Untuk memfokuskan pembahasan tentang algoritma RSA maka pada skripsi ini terbatas pada ketahanan algoritma RSA terhadap *brute force attack*. Skripsi ini tidak membahas bilangan prima aman, kecepatan algoritma RSA, bagaimana dan cara memecahkan kode RSA.

1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan beberapa manfaat antara lain:

1. Memahami konsep algoritma RSA agar tahan terhadap *brute force attack.*

2. Mendapatkan analisis penyelesaian algoritma RSA.
3. Mendapatkan interpretasi terhadap algoritma RSA yang tahan terhadap *brute force attack*.

1.6 Metode Penelitian

Metode yang digunakan adalah berdasarkan langkah-langkah sebagai berikut:

1. Membuat kunci privat dan kunci publik.
2. Mencari jumlah kemungkinan kunci.
3. Menguji lama waktu melakukan percobaan.
4. Interpretasi hasil penyelesaian algoritma RSA.
5. Kesimpulan.

1.7 Sistematika Penulisan

Penulis membagi skripsi ini dalam empat bab agar pembaca dapat memahami isi dari skripsi ini dengan jelas dan sistematis. Rinciannya adalah sebagai berikut:

Bab I Pendahuluan

Memaparkan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian dan sistematika penulisan.

Bab II Kajian Pustaka

Menyajikan konsep-konsep (teori-teori) yang mendukung dalam skripsi ini yaitu teori bilangan, kriptografi, algoritma RSA, fungsi *hash*, dan *brute force attack.*

Bab III Pembahasan

Menganalisis dan membahas konsep matematis dalam algoritma RSA, uji algoritma RSA, kelebihan dan kekurangan algoritma RSA.

Bab IV Penutup

Memaparkan hasil dari pembahasan berupa kesimpulan dan saran.

BAB II

KAJIAN PUSTAKA

Kriptografi saat ini berkembang dengan pesat, bukan hanya sebagai suatu seni tapi juga menjadi ilmu. Memahami kriptografi dan menganalisisnya memerlukan Ilmu Matematika, karena kriptografi menggunakan matematika sebagai landasan perhitungannya. Bab 2 ini merupakan bahasan tentang konsep dasar yang berhubungan dengan kriptografi seperti definisi kriptografi, sejarah kriptografi, algoritma kriptografi, sistem kriptografi serta jenis-jenis kriptografi. Selain itu juga membahas teori-teori matematika yang berguna untuk memahami algoritma RSA terutama teori bilangan.

2.1 Teori Bilangan

Dalam pengertian yang ketat, kajian tentang sifat-sifat bilangan asli disebut dengan teori bilangan. Dalam pengertian yang lebih luas, teori bilangan mempelajari bilangan dan sifat-sifatnya. Sebagai salah satu cabang matematika, teori bilangan dapat disebut sebagai "aritmetika lanjut (*advanced aritmetics*)" karena terutama berkaitan dengan sifat-sifat bilangan asli (Muhsetyo, 1997:1).

Teori bilangan merupakan dasar perhitungan dan menjadi salah satu teori yang mendasari pemahaman kriptografi, khususnya sistem kriptografi kunci publik. Bilangan yang dimaksud hanyalah bilangan bulat *(integer)*.

2.1.1 Bilangan Bulat

Bilangan bulat adalah bilangan yang tidak mempunyai pecahan desimal. Himpunan semua bilangan bulat yang dinotasikan dengan $\mathbb{Z}$ yang diambil dari kata *Zahlen* dari bahasa Jerman atau dinotasikan dengan I yang diambil dari huruf pertama kata *Integer* dari bahasa Inggris, adalah himpunan $\{\dots, -3, -2, -1,0,1,2,3 \dots\}$. Himpunan bilangan bulat dibagi tiga, yaitu bilangan bulat positif, yaitu bilangan bulat yang lebih besar dari nol yang dituliskan $\mathbb{Z}^+$, nol, dan bilangan bulat negatif, yaitu bilangan bulat yang lebih kecil dari nol yang dituliskan $\mathbb{Z}^-$ (Abdussakir, 2009:102).

Himpunan bilangan bulat dilengkapi dengan dua buah operasi, yaitu operasi penjumlahan dan perkalian, dilambangkan $(\mathbb{Z}, +, \cdot)$ membentuk suatu sistem matematika yang disebut gelanggang atau ring (Abdussakir, 2009:102). Himpunan bilangan bulat berperan sangat penting dalam kriptografi karena banyak algoritma kriptografi yang menggunakan sifat-sifat himpunan bilangan bulat dalam melakukan proses penyandiannya.

2.1.1.1 Keterbagian

Sifat-sifat yang berkaitan dengan keterbagian *(divisibility)* merupakan dasar pengembangan teori bilangan. Jika suatu bilangan bulat dibagi oleh suatu bilangan bulat yang lain, maka hasil pembagiannya adalah bilangan bulat atau bukan bilangan bulat (Muhsetyo, 1997:43).

Definisi 2.1.1.1.1

Misalnya $a, b \in \mathbb{Z}$ dengan $a \neq 0$. a dikatakan membagi b, ditulis $a|b$, jika dan hanya jika $a = bx$, untuk suatu $x \in \mathbb{Z}$ (Abdussakir, 2009:114).

Ada beberapa hal yang dapat diambil dari definisi keterbagian di atas yaitu:

1. $1|x$, untuk setiap $x \in \mathbb{Z}$, karena ada $x \in \mathbb{Z}$, sehingga $x = 1 \cdot x$
2. $x|0$, untuk setiap $x \in \mathbb{Z}$, dengan $x \neq 0$, karena ada $0 \in \mathbb{Z}$, sehingga $0 = x \cdot 0$
3. $x|x$, untuk setiap $x \in \mathbb{Z}$, dengan $x \neq 0$, karena ada $1 \in \mathbb{Z}$, sehingga $x = x \cdot 1$
4. $x|(-x)$, untuk setiap $x \in \mathbb{Z}$, dengan $x \neq 0$, karena ada $-1 \in \mathbb{Z}$, sehingga $-x = x \cdot (-1)$

Contoh:

1. $4|12$, sebab ada $3 \in \mathbb{Z}$, sehingga $12 = 4 \cdot 3$
2. $15|60$, sebab ada $4 \in \mathbb{Z}$, sehingga $60 = 15 \cdot 4$

Teorema 2.1.1.1.1

Diberikan $a, b, c \in \mathbb{Z}$.

1. Jika $a|b$ maka $a|bx$ untuk setiap bilangan bulat x
2. Jika $a|b$ dan $b|c$ maka $a|c$
3. Jika $a|b$ dan $a|c$ maka $a|(bx + cy)$ untuk setiap $x, y \in \mathbb{Z}$
4. Jika $a|b$ dan $b|a$ maka $a = \pm b$
5. Jika $a|b$, $a > 0$, dan $b > 0$, maka $a \leq b$
6. Untuk setiap bilangan bulat $m \neq 0$, $a|b$ jika dan hanya jika $ma|mb$

(Abdussakir, 2009:115).

Bukti:

1. Jika $a|b$, maka ada $y \in \mathbb{Z}$, sehingga $b = ay$. Akibatnya, untuk setiap $x \in \mathbb{Z}$ diperoleh $bx = (ay)x = a(yx)$. Karena pada bilangan bulat berlaku sifat

tertutup pada perkalian maka terdapat $p = yx \in \mathbb{Z}$. Sehingga berlaku $bx = ap$ jadi, $a|bx$.

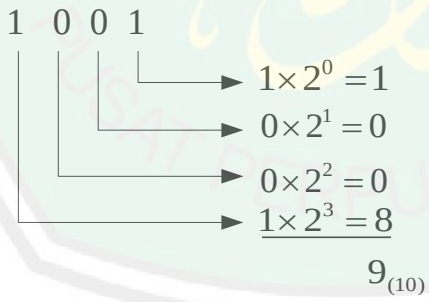
2. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Dan $b|c$, maka $c = by$ untuk $y \in \mathbb{Z}$. Diperoleh $c = by = a(xy)$, untuk suatu $xy \in \mathbb{Z}$. Jadi, $a|c$.
3. Jika $a|b$ maka $b = ap$ untuk $p \in \mathbb{Z}$. Dan $a|c$, maka $c = aq$ untuk $q \in \mathbb{Z}$. Akibatnya $bx = (ap)x$ untuk setiap $x \in \mathbb{Z}$ dan $cy = (aq)y$ untuk setiap $q \in \mathbb{Z}$. Diperoleh $bx + cy = (ap)x + (aq)y = a(px + qy)$ untuk suatu $+qy \in \mathbb{Z}$. Jadi, $a|(bx + cy)$.
4. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Dan $b|a$, maka $a = by$ untuk $y \in \mathbb{Z}$. Diperoleh $b = ax = (by)x$ maka $b - b(yx) = b(1 - yx) = 0$ karena $b \neq 0$, maka $1 - yx = 0$ atau $xy = 1$. Diperoleh $x = y = 1$ atau $x = y = -1$ sehingga didapatkan $a = \pm b$.
5. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Jika $a > 0, b > 0$ dan $b = ax$ maka $x > 0$ untuk $x = 1$ maka dipenuhi $a = b$. Sedangkan untuk $x > 1$ maka $b > a$. Jadi $a \leq b$.
6. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Akibatnya untuk $m \in \mathbb{Z}$ dan $m \neq 0$ maka berlaku $mb = m(ax) = (ma)x$. Jadi $ma|mb$. Jika $ma|mb$ dan $m \neq 0$, maka $mb = (ma)x$ untuk suatu $x \in \mathbb{Z}$. $mb = (ma)x = m(ax)$ atau $mb - m(ax) = m(b - ax) = 0$. Karena $m \neq 0$, maka $b - ax = 0$ atau $b = ax$ untuk suatu $x \in \mathbb{Z}$. Jadi $a|b$.

2.1.1.2 Bilangan Biner

Sistem bilangan biner atau yang disebut juga sistem bilangan basis dua adalah sebuah sistem bilangan yang menggunakan dua simbol yaitu 0 dan 1. Sistem bilangan biner modern ditemukan oleh Gottfried Wilhelm Leibniz pada abad ke-17. Sistem bilangan biner merupakan dasar dari sebuah sistem bilangan digital. Dari sistem biner dapat dikonversikan ke dalam bilangan oktal dan hexadesimal. Sistem biner juga dapat disebut juga dengan istilah bit atau binary digit.

Dalam istilah komputer, 1 byte = 8 bit. Kode-kode rancang bangun komputer seperti ASCII (*American Standart Code for Information Interchange*) juga menggunakan sistem pengkodean 1 byte (Buseng, 2013).

Contoh bilangan 1001 dapat diartikan:



2.1.1.3 Algoritma Pembagian

Definisi 2.1.1.3.1

Jika $a, b \in \mathbb{Z}$ dan $a > 0$, maka ada bilangan $q, r \in \mathbb{Z}$ yang masing-masing tunggal sehingga $b = qa + r$ dengan $0 \leq r < a$. Jika $a \nmid b$, maka r memenuhi ketidaksamaan $0 < r < a$ (Muhsetyo, 1997:50).

Teorema 2.1.1.3.1

Misalkan a dan b adalah bilangan bulat dengan $a > 0$. Maka terdapat bilangan bulat q dan r yang masing-masing tunggal sehingga $b = qa + r$, $0 \leq r < a$ (Abdussakir, 2009:117).

Bukti:

Diketahui a dan b adalah bilangan bulat $a > 0$. Dan $b - qa$ dengan $q \in \mathbb{Z}$ maka dapat dituliskan

$$S = \{b - qa | q \in \mathbb{Z}\}$$

Selanjutnya diambil himpunan P yang anggota himpunan S yang tidak negatif, yaitu:

$$P = \{b - qa | b - qa \geq 0, q \in \mathbb{Z}\}$$

Maka $P \neq \phi$, sebab:

1. Jika $b \geq 0$ dan $q = 0$, maka $b - qa = b - 0a = b \in P$
2. Jika $b < 0$ dan $q = b$, maka $b - qa = b - ba = b(1 - a)$

Karena $a > 0$ atau $a \geq 0$, maka $1 - a \leq 0$. Dan karena $b < 0$, maka $b(1 - a) \geq 0$. Jadi $b - ba \in P$

Karena $P \neq \phi$ dan $P \subseteq \mathbb{N}$, sesuai prinsip urutan pada $\mathbb{N}$, maka P mempunyai unsur terkecil.

Misalkan r adalah unsur terkecil dari P.

Karena $r \in P$, maka $r \geq 0$ dan $r = b - qa$ atau $b = qa + r$, untuk suatu $q \in \mathbb{Z}$.

Selanjutnya akan dibuktikan bahwa $r \geq a$. Maka $0 \leq r - a$ dan $r - a$

$= (b - qa) - a = b - (q + 1)a$.

Jadi, $r - a \in P$.

Karena $a > 0$, maka $r - a < r$.

Jadi, ada elemen $(r - a)$ di P yang kurang dari r. Hal ini bertentangan dengan pernyataan bahwa r adalah unsur terkecil di P. Dengan demikian maka harus $r < a$. Dari $r \geq 0$ dan $r < a$, maka $0 \leq r < a$ sehingga $b = qa + r$, untuk $0 \leq r < a$.

Berikutnya akan ditunjukkan bahwa q dan r masing-masing tunggal. Andaikan ada q_1 dan q_2 dengan $q_1 \neq q_2$ dan $r_1$dan r_2 dengan $r_1 \neq r_2$ sehingga $b = q_1a + r_1, 0 \leq r_1 < a$

dan $b = q_2a + r_2, 0 \leq r_2 < a$.

Maka $q_1a + r_1 = q_2a + r_2$ atau $r_2 - r_1 = a(q_1 - q_2)$.

Berarti $a|(r_2 - r_1)$ atau $(r_2 - r_1)$ adalah kelipatan dari a.

Di sisi lain karena $0 \leq r_1 < a$ dan $0 \leq r_2 < a$.

Ambil $0 \leq r_1 < a \times (-1) = -a \leq -r_1 < 0$ dan $0 \leq r_2 < a$.

$$\begin{array}{l} 0 \leq r_2 < a \\ \underline{-a < -r_1 < 0 \quad +} \\ -a < (r_2 - r_1) < a \end{array}$$

Maka $-a < (r_2 - r_1) < a$.

Satu-satunya kelipatan a yang terdapat di antara $-a$ dan a adalah 0. Sehingga diperoleh $r_2 - r_1 = 0$ atau $r_2 = r_1$

Karena $r_2 - r_1 = a(q_1 - q_2)$ maka $a(q_1 - q_2) = 0$

Karena $a > 0$ maka $q_1 - q_2 = 0$ atau $q_1 = q_2$

Jadi q dan r masing-masing adalah tunggal.

Jadi, $b = qa + r, 0 \leq r < a$

Dalam teorema di atas, yaitu $b = qa + r, 0 \neq r < a.b$ disebut bilangan yang dibagi *(dividend)*, a disebut pembagi *(divisor)*, q disebut hasil bagi *(quotient)*, dan r disebut sisa pembagi *(remainder)* jika $a|b$ maka diperoleh bahwa sisa pembaginya adalah 0. Sehingga dapat disimpulkan untuk $a > 0$ bahwa:

a) $a|b$ jika dan hanya jika $b = qa + r$ dan $r = 0$

b) $a\nmid b$ jika dan hanya jika $b = qa + r$ dengan $0 \leq r < a$

Teorema 2.1.1.3.2

Jika $b \in \mathbb{Z}$ dan $b > 1$, maka setiap $n \in \mathbb{Z}^+$ dapat ditulis secara tunggal dalam bentuk $n = a_k b^k + a_{k-1} b^{k-1} + \cdots a_2 b^2 + a_1 b^1 + a_0 b^0$

Yang mana $k \in \mathbb{Z}$ dan $k \geq 0, a_i \in \mathbb{Z}$ dan $0 \leq a_i \leq b - 1$ untuk $i = 0,1,2, \ldots, k$, dan $a_k \neq 0$ (Muhsetyo, 1997:54).

Bukti:

Karena $b \in \mathbb{Z}$ dan $b > 1$, maka $b > 0$, sehingga menurut algoritma pembagian, hubungan antara n dan b adalah:

$n = bq_0 + a_0, 0 \leq a_0 \leq b - 1$

Jika $q_0 \neq 0$, maka hubungan antara q_0 dan b menurut algoritma pembagian:

$q_0 = bq_1 + a_1, 0 \leq a_1 \leq b - 1$

Jika langkah yang sama dikerjakan, maka diperoleh:

$q_1 = bq_2 + a_2, 0 \leq a_2 \leq b - 1$

$q_2 = bq_3 + a_3, 0 \leq a_3 \leq b - 1$

... ...

$q_{k-2} = bq_{k-1} + a_{k-1}, 0 \leq a_{k-1} \leq b - 1$

$q_{k-1} = bq_k + a_k, 0 \leq a_k \leq b-1$

Langkah terakhir ditandai dengan munculnya $q_k = 0$. Karena barisan $q_0, q_1, \dots, q_k$ adalah barisan bilangan bulat tidak negatif yang menurun, maka paling banyak ada q_0 suku yang positif, dan 1 suku q_k yang bernilai nol. Dari persamaan-persamaan di atas dapat ditentukan bahwa:

$n = bq_0 + a_0$

$n = b(bq_1 + a_1) + a_0 = b^2q_1 + ba_1 + a_0$

$n = b^2(bq_2 + a_2) + ba_1 + a_0 = b^3q_2 + b^2q_1 + ba_1 + a_0$

… …

$n = b^{k-1}q_{k-2} + b^{k-2}a_{k-2} + b^{k-3}a_{k-3} + \cdots + ba_1 + a_0$

$n = b^kq + b^{k-1}a_{k-1} + b^{k-2}a_{k-2} + \cdots + ba_1 + a_0$

$n = b^{k+1}q_k + b^ka_k + b^{k-1}a_{k-1} + \cdots + ba_1 + a_0$

Karena $q_k = 0$:

$n = b^ka_k + b^{k-1}a_{k-1} + \cdots + ba_1 + a_0$

$n = a_kb^k + a_{k-1}b^{k-1} + \cdots + a_1b^1 + a_0b^0$

Contoh:

Perhatikan langkah berturut-turut dalam pembagian algoritma untuk menuliskan 567 dalam basis 2 dan 567dalam basis 3.

Jawab:

Untuk basis 2:

567 =2·283+1 17 =2·8+1

283 =2·141+1 8 =2·4+0

141 =2·70+1 4 =2·2+0

70 =2·35+1 2 =2·1+0

35 =2·17+1 1 =2·0+1

$(567)_{10}=(1000110111)_2$

Untuk basis 3:

567 =3·189+0

189 =3·63+0

63 =3·21+0

21 =3·7+0

7 =3·2+1

2 =3·0+2 $(567)_{10} = (210000)_3$

2.1.2 Fungsi Euler

Fungsi Euler digunakan untuk menyatakan banyaknya bilangan bulat $< n$ yang relatif prima terhadap n.

Definisi 2.1.2.1

Suatu himpunan bilangan bulat $\{r_1, r_2, \dots, r_3\}$ disebut dengan sistem residu tereduksi modulo m jika:

a) $(r_1, m) = 1 (i = 1, 2, \dots, k)$.

b) $r_i \not\equiv r_j$ (mod m) untuk semua $i \neq j$.

c) Jika $(x, m) = 1$, maka $x \equiv r_i$ mod (m) (Muhsetyo, 1997:279).

Contoh:

Himpunan {1,5} adalah sistem tereduksi modulo 6 karena:

a. $r_1 = 1, r_2 = 5, (r_1, 6) = (1,6) = 1$ dan $(r_2, 6) = (5,6) = 1$

b. $1 \not\equiv 5 \pmod 6$

c. $(7,6) = 1 \rightarrow 7 \equiv 1 \pmod 6$

$(11,6) = 1 \rightarrow 11 \equiv 5 \pmod 6$, dan seterusnya

Definisi 2.1.2.2

Jika m adalah suatu bilangan bulat positif, maka banyaknya residu di dalam sistem residu tereduksi modulo m adalah $\phi(m)$. $\phi(m)$ disebut fungsi ϕ-Euler dari m. Dari definisi dapat diketahui bahwa $\phi(m)$ adalah sama dengan banyaknya bilangan bulat positif kurang dari m yang relatif prima dengan m (Muhsetyo, 1997:279).

Contoh:

1. Himpunan {1} adalah sistem residu tereduksi modulo 2 sehingga $\phi(2) = 1$
2. Himpunan $\{1, 2\}$ adalah sistem residu tereduksi modulo 3 sehingga $\phi(3) = 2$
3. Himpunan $\{1, 3, 5, 7, 9, 11, 13, 15\}$ adalah sistem residu tereduksi modulo 16 sehingga $\phi(16) = 8$

2.1.3 Metode *Fast Exponentiation*

Metode *fast exponetiation* ini digunakan untuk menghitung operasi pemangkatan besar bilangan bulat modulo dengan cepat. Metode ini memanfaatkan ekspansi biner dari bilangan Z (Hamidah, 2009) yaitu:

$$Z = \sum_{i=0}^{k} a_i \cdot 2^i$$

Karena z ditulis dengan ekspansi biner maka $a \in \{0,1\}$, sehingga:

$$g^z = g^{\sum_{i=0}^{k} a_i \cdot 2^i} = \prod_{i=0}^{k} (g^{2^i})^{a_i} = \prod_{0 \le i \le k, a_i = 1} g2^i$$

$$g^z = g^{\left((a_0 \cdot 2^0) + (a_1 \cdot 2^1) + (a_2 \cdot 2^2) + \cdots + (a_k \cdot 2^k)\right)}$$

Metode *fast exponetiation* didasarkan pada pernyataan berikut ini:

$$g2^{i+1} = (g^{2^i})^a$$

Contoh:

Akan dihitung $6^{73} \pmod{100}$

Jawab:

Pertama tentukan expansi biner dari 73

$73 = 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^0$ atau $73 = (1001001)_2$

Kemudian dihitung

$6^{2^0} = 6$

$6^{2^1} = 36$

$6^{2^2} = 36^2 = 96 \pmod{100}$

$6^{2^3} = 16 \pmod{100}$

$6^{2^4} = 16^2 = 56 \pmod{100}$

$6^{2^5} = 56^2 = 36 \pmod{100}$

$6^{2^6} = 56^2 = 96 \pmod{100}$

Sehingga diperoleh:

6^{73} $= 6 \cdot 6^{2^3} \cdot 6^{2^4} \cdot 6 \pmod{100}$

$= 6 \cdot 16 \cdot 96 \pmod{100}$

$= 16 \pmod{100}$

Jadi, $6^{73} \pmod{100} = 16$

2.1.4 Aritmetika Modulo dan Kekongruenan

Definisi 2.1.4.1

Diketahui $a, b, m \in \mathbb{Z}$. a disebut kongruen dengan b modulo m, ditulis $a \equiv b \pmod{m}$, jika $(a - b)$ habis dibagi m, yaitu $m|(a - b)$ tidak habis dibagi m, yaitu $m \nmid (a - b)$, maka ditulis $a \not\equiv b \pmod{m}$, dibaca a tidak kongruen dengan b modulo m. Karena $(a - b)$ habis dibagi oleh m jika dan hanya jika $(a - b)$ habis dibagi oleh $-m$, maka: $a \equiv b \pmod{m}$ jika dan hanya jika $b \equiv a \pmod{m}$ (Muhsetyo, 1997:138).

Contoh:

1. $17 \equiv 2 \pmod{3}$ (3 habis dibagi $17 - 2 = 15 \rightarrow 15 \div 3 = 5$)
2. $-7 \not\equiv 15 \pmod{3}$ (3 tidak habis dibagi $-7 - 15 = -22$)

2.1.5 Bilangan Prima

Sifat pembagian pada bilangan bulat melahirkan konsep-konsep bilangan prima dan aritmetika modulo, dan salah satu konsep bilangan bulat yang digunakan dalam penghitungan komputer adalah bilangan prima. Dengan ditemukannya bilangan prima, teori bilangan berkembang semakin jauh dan lebih

mendalam. Banyak dalil dan sifat dikembangkan berdasarkan bilangan prima. Bilangan prima juga memainkan peranan yang penting pada beberapa algoritma kunci publik, seperti algoritma RSA.

Definisi 2.1.5.1

Jika p suatu bilangan bulat positif lebih dari 1 yang hanya mempunyai pembagi positif 1 dan p, maka p disebut bilangan prima. Jika suatu bilangan bulat $q > 1$ bukan suatu bilangan prima, maka q disebut bilangan komposit.

Untuk menguji apakah p merupakan bilangan prima atau bilangan komposit, dapat menggunakan cara yang paling sederhana, yaitu cukup membagi p dengan sejumlah bilangan prima, yaitu $2, 3, \ldots,$ bilangan prima $\leq \sqrt{p}$. Jika p habis dibagi salah satu dari bilangan prima tersebut, maka p adalah bilangan komposit tetapi jika p tidak habis di bagi oleh semua bilangan prima tersebut, maka p adalah bilangan prima.

Teorema 2.1.5.1

Jika p adalah suatu bilangan prima dan $p|ab$, maka $p|a$ atau $p|b$

(Muhsetyo, 1997:100).

Bukti:

Anggaplah $p \nmid a$, karena p adalah suatu bilangan prima dan $p|a$, maka p hanya mempunyai pembagi 1 dan p, sehingga $(a, p) = 1$. Menurut teorema, jika $a|ab$ dan $(a, p) = 1$, maka $p|b$. Dengan cara serupa, dan dianggap $p|b$, maka dapat dibuktikan bahwa $p|a$.

Teorema 2.1.5.2

Jika p adalah suatu bilangan prima dan $p|a_1a_2, \dots, a_n$, maka paling sedikit membagi satu faktor $a_k(1 \leq k \leq n)$ (Muhsetyo, 1997:100).

Bukti:

$p|a_1a_2, \dots, a_n$ atau $p|a_1(a_2, a_3 \dots, a_n)| \rightarrow p|a_1$ atau $p|a_2, a_3 \dots, a_n$, jika $p \nmid a_1$ maka terbukti p paling sedikit membagi satu faktor a_k, jika $p \nmid a_1$ maka $p|a_2, a_3 \dots, a_n$ atau $p|a_2(a_3, a_4 \dots, a_n)$, $p|a_2(a_3, a_4 \dots, a_n) \rightarrow p|a_2$ atau $p|a_3a_4, \dots, a_n$. Demikian seterusnya diperoleh $p|a_{n-1}, a_n$, $p|a_{n-1}, a_n \rightarrow p|a_{n-1}$ atau $p|a_n$. Ini berarti bahwa p paling sedikit membagi faktor a_k.

Teorema 2.1.5.3 (Teorema Fermat)

Jika p adalah suatu bilangan prima dan $(a, p) = 1$, maka $a^{\phi(p)} \equiv 1(\text{mod } p)$ (Muhsetyo, 1997:152).

Bukti:

Karena p adalah suatu bilangan prima dengan $p \nmid a$, maka $(p, a) = 1$ (jika $(p, a)|1$) yaitu p dan a tidak relatif prima, maka p dan a mempunyai faktor selain 1 dan p, bertentangan dengan sifat p sebagai bilangan prima), selanjutnya, karena $(p, a) = 1$ maka untuk $a^{\phi(p)} \equiv 1(\text{mod } p)$.

P adalah bilangan prima, berarti dari bilangan-bilangan bulat:

$\{0, 1, 2, 3, \dots, p - 1\}$

Yang tidak relatif prima dengan p hanyalah 0, sehingga:

$\{1, 2, 3, \dots, p - 1\}$

Merupakan sistem residu tereduksi modulo p, dengan demikian

$\phi(p) = p - 1$

Karena

$\phi(p) = p - 1$ dan $a^{\phi(p)} \equiv 1$, maka $a^{p-1} \equiv 1 (\text{mod } p)$

Contoh:

Carilah nilai-nilai x yang memenuhi $2^{250} \equiv x (\text{mod } 7)$ dan $0 \leq x < 7$

Jawab:

Karena 7 adalah bilangan prima, maka $\phi(7) = 7 - 1 = 6$

Karena 7∤2 dan 7 adalah bilangan prima, maka:

$2^{\phi(7)} \equiv 1 (\text{mod } 7)$

$2^6 \equiv 1 (\text{mod } 7)$

$2^{250} \equiv (2^6)^{41} \cdot 2^4 \equiv 1 \cdot 2^4 (\text{mod } 7) \equiv 1 \cdot 16 (\text{mod } 7) \equiv 16 (\text{mod } 7) \equiv 2 (\text{mod } 7)$

Jadi $x = 2$.

2.1.6.1 Relatif Prima

Dua buah bilangan bulat a dan b dikatakan relatif prima jika PBB $(a, b) = 1$ (Tomiexz, 2011).

Contoh:

20 dan 3 relatif prima sebab FPB $(20, 3) = 1$. Begitu juga 7 dan 11 relatif prima karena FPB $(7, 11) = 1$. Tetapi 20 dan 5 tidak relatif prima sebab FBB $(20, 5) = 5$ dan 1.

Jika a dan b relatif prima, maka terdapat bilangan bulat m dan n sedemikian sehingga: $ma + nb = 1$.

Contoh: Bilangan 20 dan 3 adalah relatif prima karena FPB $(20, 3) = 1$, atau dapat ditulis: $2 \cdot 20 + (-13) \cdot 3 = 1$, dengan $m = 2$ dan $n = -13$. Tetapi 20 dan 5 tidak relatif prima karena FPB $(20, 5) = 5$ dan 1 sehingga 20 dan 5 tidak dapat karena FBB $(20, 5) = 5$ dan 1 sehingga 20 dan 5 tidak dapat dinyatakan dalam $m. 20 + n \cdot 5 = 1$.

2.2 Kriptografi

2.2.1 Pengertian Kriptografi

Kriptografi atau yang dalam bahasa inggrisnya *cryptography* berasal dari bahasa Yunani, *cryptography* terdiri dari kata *kryptos* yang berarti tersembunyi dan *graphein* yang berarti menulis atau tulisan. Menurut terminologi, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain (Ariyus, 2006:9). Kriptografi juga dapat disebut dengan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data. Sebuah pesan rahasia harus terjaga keamanannya, salah satu cara dengan penyandian pesan yang bertujuan meyakinkan rahasia dengan menyembunyikan informasi dari orang-orang yang tidak ditujukan informasi tersebut kepadanya (Munir, 2006:3).

Kriptografi berkembang sedemikian rupa sehingga melahirkan bidang yang berlawanan yaitu *cryptoanalysis*, yaitu suatu ilmu dan seni yang dipelajari untuk memecahkan *chiperteks* menjadi *plainteks* tanpa mengetahui kunci yang digunakan atau aksi untuk memecahkan mekanisme kriptografi dengan cara

mendapatkan *plainteks* atau kunci dari *cipherteks* yang digunakan untuk mendapatkan informasi berharga kemudian mengubah atau memalsukan pesan dengan tujuan untuk menipu penerima yang sesungguhnya, memecahkan *cipherteks* (Flourensia, 2005:4). Secara sederhana adalah seseorang yang ingin menembus kerahasiaan dari sebuah kode dengan cara membangun algoritma baru yang dapat memecahkan algoritma yang sudah ada, pelakunya disebut kriptoanalis. Menurut Munir (2006:8), jika seorang kriptografer mentransformasi *plainteks* dan *chiperteks* dengan suatu algoritma dan kunci maka sebaliknya seorang kriptoanalis berusaha memecahkan *chiperteks* untuk menemukan *plainteks* atau kunci. Setiap detiknya dalam dunia internet terjadi banyak sekali pertukaran informasi, dan banyak pula pencurian informasi oleh pihak-pihak yang tidak bertanggung jawab. Ada beberapa ancaman keamanan yang terjadi terhadap informasi di antaranya:

1. *Interuption*, adalah ancaman terhadap *avaibability* informasi, yaitu data yang ada dalam komputer dirusak atau dihapus sehingga saat informasi tersebut dibutuhkan tidak ada lagi.
2. *Interception*, adalah ancaman terhadap kerahasiaan. Informasi yang ada disadap oleh orang yang tidak berhak mendapat akses ke komputer di mana informasi tersebut disimpan.
3. *Modification*, adalah ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalu lintas informasi yang sedang dikirim dan dirubah sesuai keinginan orang tersebut.

4. *Fabrication*, adalah ancaman terhadap integritas. Orang yang tidak berhak berhasil menirukan atau memalsukan suatu informasi yang ada sehingga si penerima informasi mengira telah mendapatkan informasi dari pengirim yang sebenarnya. Jadi, dari sini dapat diketahui kriptografi diciptakan dengan tujuan, kerahasiaan, yaitu menjamin bahwa pesan dalam keadaan aman dari pihak yang tidak berhak, integritas data, yaitu menjamin bahwa pesan masih asli atau tidak dimanipulasi, keaslian, yaitu mengidentifikasi pesan dan pengirim pesan, dan *nonrepudiation*, yaitu mencegah penyangkalan pihak yang berkomunikasi (menolak penyangkalan) (Ariyus, 2006:7).

2.2.2 Sejarah Kriptografi

Kriptografi memiliki sejarah yang panjang dan mengagumkan. Penulisan rahasia ini dapat dilacak kembali ke 3000 tahun sebelum masehi saat digunakan oleh bangsa Mesir. Mereka menggunakan *hieroglyphcs* untuk menyembunyikan tulisan dari orang yang tidak diharapkan. *Hieroglyphcs* diturunkan dari bahasa Yunani, *hieroglyphica* yang berarti ukiran rahasia. Pada tahun 400 SM tentara Sparta di Yunani, menggunakan alat dari daun papyrus yang dililitkan pada sebatang kayu atau silinder berdiameter tertentu yang disebut *scytale* untuk mengirimkan pesan rahasia di medan perang. Sekitar tahun 50 SM, Julius Caesar, kaisar Romawi, menggunakan *cipher substitusi*, yaitu huruf-huruf alfabet disubstitusi dengan huruf-huruf yang lain pada alfabet yang sama. *Cipher substitusi* digunakan untuk mengirim pesan pada jendral di medan perang agar tidak terbaca oleh musuh dan hanya dapat dibaca oleh jendralnya saja, yang mana sang jendral telah diberi tahu bagaimana cara membacanya.

Tahun 1460, Leon Battista Alberti, di Italia mengembangkan *disk cipher* untuk enkripsi. Sistemnya terdiri dari dua disk konsentris. Setiap disk memiliki alfabet di sekelilingnya, dan dengan memutar satu disk berhubungan dengan yang lainnya, huruf pada satu alfabet dapat ditransformasi ke huruf pada alfabet yang lain. Bangsa Arab yang mahir dalam Ilmu Matematika, Statistik, dan Linguistik juga mengembangkan kriptografi terbukti dengan ditemukannya buku karangan al-Kindi yang ditulis pada abad 9 H yang berjudul *"A Manuscript on Deciphering Cryptographic Messages"*. Pada 1790, Thomas Jefferson mengembangkan alat enkripsi dengan menggunakan tumpukan yang terdiri dari 26 disk yang dapat diputar secara individual. Pesan dirakit dengan memutar setiap disk ke huruf yang tepat di bawah batang berjajar yang menjalankan panjang tumpukan disk. Kemudian, batang berjajar diputar dengan sudut tertentu, kemudian dilihat bahwa huruf-huruf yang berada di bawah batang adalah pesan yang terenkripsi. Penerima akan menjajarkan karakter-karakter *cipher* di bawah batang berjajar, memutar batang kembali dengan sudut A dan membaca pesan *plainteks*. Sejak saat itu sistem disk digunakan secara luas terutama pihak militer (Flourensia, 2005:5).

Pada tahun 1920, Boris Hagelin di Scockholm, Swedia. membuat mesin *Hagelin* dikenal sebagai M-209, dilanjutkan Herbert O. Yardley, yang membuat alat bernama *Black Chamber* digunakan untuk menyadap informasi Jepang. Pada tahun 1919, Hugo Koch dari Belanda mengembangkan enigma atau otor mekanis untuk pengkodean dan pendekodean selama perang dunia II. Pengembangan paling mengejutkan dalam sejarah kriptografi terjadi pada tahun 1976 saat Diffie dan Hellman mempublikasikan *New Directions in Cryptography*. Tulisan ini

memperkenalkan konsep revolusioner kriptografi kunci publik dan juga memberikan metode baru dan jenius untuk pertukaran kunci, keamanan yang berdasar pada kekuatan masalah logaritma diskrit. Ide dasar dari sistem kriptografi kunci publik adalah bahwa kunci kriptografi dibuat sepasang, satu kunci untuk enkripsi dan satu kunci untuk dekripsi.

Pada tahun 1978 Rivest, Shamir dan Adleman menemukan rancangan enkripsi kunci publik dan tanda tangan, yang sekarang disebut RSA. Tahun delapan puluhan menunjukkan peningkatan luas di area ini, sistem RSA masih aman. Kelas lain yang merupakan rancangan kunci publik praktis ditemukan oleh Elgamal pada 1985. Rancangan ini juga berdasar pada masalah logaritma diskret. Pada tahun 1994 pemerintah US mengadopsi *Digital Signature Standard*, sebuah mekanisme yang berdasar pada rancangan kunci publik Elgamal (Flourensia, 2005:6).

Selama bertahun-tahun kriptografi hanya digunakan oleh pihak militer. Agen keamanan nasional semua Negara bekerja keras untuk mempelajari kriptografi. Maka dari itu kriptografi terus berkembang karena semakin banyaknya informasi yang harus diamankan kerahasiaannya. Selama tiga puluh tahun terakhir ini bukan hanya agen militer yang berniat menggunakan kriptografi namun pribadi-pribadi yang lain yang tidak ingin diketahui kehidupan pribadinya juga menggunakan kriptografi (Ariyus, 2006:10).

2.2.3 Algoritma Kriptografi

Algoritma adalah urutan langkah-langkah logis umtuk penyelesaian masalah yang disusun secara sistematis, jadi algoritma kriptografi atau sering

disebut dengan *chiper* merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut. Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu:

1. Enkripsi, merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirim agar terjaga kerahasiaannya. Pesan asli disebut *plainteks* yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi dapat diartikan sebagai *chiper* atau kode.
2. Dekripsi, merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk aslinya. Algoritma yang digunakan berbeda dengan yang digunakan untuk enkripsi.
3. Kunci, merupakan kunci yang digunakan untuk proses enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian yaitu kunci rahasia *(private key)* dan kunci umum *(public key)* (Ariyus, 2008:43).

Pada proses pengiriman pesan dalam kriptografi, pesan yang dikirim oleh pengirim pesan kepada penerima pesan terjaga keamanannya. Allah SWT menjamin kesucian dan kemurnian Al-Qur'an dijelaskan pada firman-Nya surat Al-Hijr ayat 9:

إِنَّا نَحْنُ نَزَّلْنَا ٱلذِّكْرَ وَإِنَّا لَهُۥ لَحَٰفِظُونَ ۝٩

Artinya: "*Sesungguhnya Kami-lah yang menurunkan Al-Quran, dan Sesungguhnya Kami benar-benar memeliharanya"(QS. Al-Hijr: 9).*

Ayat ini memberikan jaminan tentang kesucian dan kemurnian Al-Qur'an selama-lamanya. Al-Qur'an tidak perlu diragukan lagi keasliannya karena yang menjamin adalah Allah SWT. Melalui perantara yang benar-benar terjamin

kesuciannya dan dipercayakan kepada orang yang benar-benar terjamin kejujurannya.

Dalam ayat di atas Allah menyebut dirinya sendiri dengan kata "Kami". Menurut kaidah bahasa Indonesia, kata "Kami" biasanya digunakan untuk merujuk pada orang pertama jamak, padahal Kami dalam ayat di atas merujuk hanya pada Allah semata (tunggal). Penggunaan kata "Kami" dalam ayat tersebut setidaknya memiliki dua makna, yaitu yang pertama Allah sedang mengagungkan diri-Nya. Apakah dikatakan sombong jika Allah mengagungkan dirinya sendiri? Tentu saja tidak, karena memang Allah maha segala-galanya, tidak ada satupun yang dapat menandingi kuasa-Nya. Yang tidak boleh adalah jika manusia mengagung-agungkan dirinya sendiri, karena hal tersebut dapat menyebabkan tumbuhnya sifat sombong.

Makna yang kedua yaitu Allah ingin melibatkan makhluk-Nya terhadap apa yang Dia kehendaki. Pada ayat diatas dikatakan *"Sesungguhnya Kami lah yang menurunkan Adz-Dzikir (Al-Qur'an) ..."*. Maksud kata "Kami" dalam kalimat tersebut adalah terdapat makhluk Allah yang dilibatkan dalam proses turunnya Al-Qur'an, yaitu malaikat Jibril yang bertugas untuk menyampaikan wahyu kepada Nabi Muhammad S.A.W. Lanjutan ayatnya: ".... *dan sesungguhnya Kami pula yang memeliharanya (menjaganya)"*. Salah satu cara Allah untuk menjaga Al-Qur'an yaitu dengan melibatkan para penghafal Al-Qur'an. Dengan adanya para penghafal Al-Quran maka apabila terdapat kesalahan sedikit saja dalam penulisan atau pembacaan Al-Qur'an maka akan segera diketahui. Dengan demikian kemurnian Al-Quran akan tetap terjaga hingga akhir zaman.

Jadi secara garis besar inti dari ayat diatas adalah: Allah yang menurunkan Al-Qur'an dan Allah pula yang akan menjaganya hingga akhir zaman. Jika Allah menjaga Al-Qur'an maka Allah akan menjaga ahlul qur'an (para penghafal Al-Qur'an) pula. Sebagai contoh di Palestina, secara hitung-hitungan matematika seharusnya Palestina sudah hancur sejak lama karena menghadapi gempuran yang hebat dari berbagai pihak. Tapi pada kenyataannya tidak demikian, Palestina tetap ada hingga saat ini. Mungkin inilah salah satu wujud kebesaran Allah karena di sana banyak terdapat para penghafal Al-Qur'an sehingga Allah tetap melindungi mereka. Di negara tersebut terdapat lebih dari 53000 anak di bawah 15 tahun yang telah mampu menghafal Al-Qur'an sebanyak 30 juz (Adharani, 2010).

Dalam hal ayat di atas dapat diambil bahwa dalam penyampaian pesan menggunakan algoritma RSA dapat terjaga rahasianya sehingga sampai kepada si penerima pesan.

2.2.4 Algoritma Simetri

Algoritma ini juga disebut sebagai algoritma klasik karena memakai kunci yang sama untuk proses enkripsi dan dekripsinya. Keamanan algoritma ini terletak pada kuncinya. Jika kunci telah diketahui oleh orang lain maka informasi akan terbongkar. Sifat kunci yang seperti ini membuat pengirim harus selalu memastikan bahwa jalur yang digunakan dalam pendistribusian kunci adalah jalur yang aman atau memastikan bahwa seseorang yang ditunjuk membawa kunci untuk dipertukarkan adalah orang yang dapat dipercaya. Masalahnya akan rumit jika sebanyak *n* pengguna dan setiap dua orang harus bertukar kunci yang berbeda. Maka akan terjadi sebanyak $c_2^n = \frac{n!}{(n-2)!2!} = \frac{n(n-1)}{2}$ jumlah kunci agar

semuanya aman. Contoh algoritma simetri adalah: substitusi, transposisi atau permutasi, *data encryption standard* (DES), *advanced encryption standard* (AES), *one time pad* (OTP), dan sebagainya (Ariyus, 2006:14).

2.2.5 Algoritma Asimetri

Algoritma RSA termasuk di dalam algoritma Asimetri, Asimetri sering disebut algoritma kunci, kerana kunci yang digunakan untuk enkripsi dan dekripsinya berbeda. Pada algoritma kriptografi kunci terbagi menjadi dua bagian, yaitu kunci publik dan kunci pribadi. Kunci A enkripsi dekripsi B publik adalah kunci yang semua orang boleh mengetahui sedangkan kunci pribadi adalah kunci yang dirahasiakan, hanya boleh diketahui oleh satu orang. Kunci-kunci tersebut saling berhubungan satu dengan yang lainnya. Dengan kunci publik orang dapat mengenkripsi pesan sedangkan untuk mendekripsi pesan hanya orang yang mempunyai kunci pribadi yang dapat melakukannya. Contoh dari algoritma Asimetri adalah *digital signature algorithm* (DSA), *elliptic curve cryptografi* (ECC), Diffie-Hellman (DH), Elgamal, dan lain sebagainya (Ariyus, 2006:15).

2.2.6 Sistem Kriptografi

Definisi 2.2.6.1

Sistem kriptografi adalah suatu 5-*tuple* $(\mathcal{PCKED})$ yang memenuhi kondisi sebagai berikut:

1. $\mathcal{P}$ adalah himpunan *plainteks*
2. $\mathcal{C}$ adalah himpunan *chiperteks*
3. $\mathcal{K}$ adalah himpunan kunci atau ruang kunci *(Keyspace)*
4. $\mathcal{E}$ adalah himpunan fungsi enkripsi $ek: \mathcal{P} \longrightarrow \mathcal{C}$

5. $\mathcal{D}$ adalah himpunan fungsi dekripsi $dk: \mathcal{C} \longrightarrow \mathcal{P}$

6. Untuk $k \in \mathcal{K}$ terdapat $\mathcal{E}k \in \mathcal{E}$ dan $dk \in \mathcal{D}$ setiap $ek: \mathcal{P} \longrightarrow \mathcal{C}$ dan $dk: \mathcal{C} \longrightarrow \mathcal{P}$ merupakan fungsi sehingga $dk\big(ek(x)\big) = x$ untuk setiap *plainteks* $x \in \mathcal{P}$ (Stinson,1995:48).

Suatu sistem kriptografi terdiri dari suatu algoritma, seluruh kemungkinan *plainteks*, *cipherteks*, dan kunci-kuncinya. Sistem kriptografi merupakan suatu fasilitas untuk mengkonversikan *plainteks* menjadi *cipherteks*, dan sebaliknya.

2.3 Algoritma RSA

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 mutahir (Arifin, 2009).

2.3.1 Sejarah RSA

Algortima RSA dijabarkan pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Len Adleman dari MIT. Huruf RSA itu sendiri juga berasal dari inisial nama mereka (Rivest-Shamir-Adleman). Clifford Cocks, seorang matematikawan Inggris yang bekerja untuk GCHQ, menjabarkan tentang sistem ekivalen pada dokumen internal di tahun 1973. Penemuan Clifford Cocks tidak terungkap hingga tahun 1997 dikarenakan alasan *top secret classification*. Algoritma

tersebut dipatenkan oleh Massachusetts Institute of Technology pada tahun 1983 di Amerika Serikat sebagai U. S. Patent 4405829. Paten tersebut berlaku hingga 21 September 2000. Semenjak algoritma RSA dipublikasikan sebagai aplikasi paten, regulasi di sebagian besar negara-negara lain tidak memungkinkan penggunaan paten. Hal ini menyebabkan hasil temuan Clifford Cocks di kenal secara umum, paten di Amerika Serikat tidak dapat mematenkannya (Arifin, 2009).

2.3.2 Kecepatan Algoritma RSA

Sebuah operasi RSA, baik enkripsi, dekripsi, penandaan, atau verifikasi intinya adalah sebuah eksponensial terhadap modul. Proses perhitungan ini ditunjukkan oleh sebuah rangkaian dari multiplikasi terhadap modul. Dalam aplikasi praktikal, adalah umum untuk menentukan sebuah eksponen kecil yang umum sebagai kunci publik. Faktanya, keseluruhan kelompok dari *user* (pemakai) dapat memakai eksponen yang sama, dengan berbeda modulus. (Terdapat beberapa pembatasan pada faktor-faktor prima dari eksponen publik ketika diputuskan). Hal ini menyebabkan proses enkripsi lebih cepat daripada proses dekripsi dan verifikasi lebih cepat dari pada penandaan. Dengan algoritma eksponensial modular (*modular exponentiation*) yang khas yang digunakan untuk mengimplementasikan algoritma RSA, operasi kunci publik membutuhkan *O(k2)* langkah, operasi kunci privat membutuhkan *O(k3)* langkah, pembangitan kunci membutuhkan *O(k4)* langkah, di mana *k* adalah jumlah bit dari modulus.

Teknik multiplikasi cepat, seperti metode pada *fast fourier transform* (FFT), membutuhkan langkah-langkah yang lebih sedikit secara asimtotik. Dalam

praktiknya, hal di atas tidaklah umum melihat pada kompleksitas perangkat lunak yang lebih besar dan kenyataan bahwa mungkin akan lebih lambat untuk beberapa ukuran kunci yang khas. Sebagai perbandingan, algoritma DES dan beberapa *chiper* blok yang lain jauh lebih cepat daripada algoritma RSA. DES secara umum 100 kali lebih cepat pada perangkat lunak dan antara 1.000 dan 10.000 kali lebih cepat pada perangkat keras, tergantung dari implementasinya. Implementasi dari algoritma RSA mungkin akan mempersempit celah beberapa bit dalam tahun-tahun mendatang, melihat tingginya permintaan, tapi bagaimana pun juga, *chiper* blok akan bertambah lebih cepat (Arifin, 2009).

2.3.3 Membuat Kunci Privat (*Private Key*) dan Kunci Publik (*Public Key*)

Berikut ini langkah-langkahnya adalah:

1. Pilih dua buah bilangan prima sembarang, sebut *a* dan *b*. Jaga kerahasiaan *a* dan *b* ini.
2. Hitung $n = a \times b$. Besaran *n* tidak dirahasiakan.
3. Hitung $m = (a - 1) \times (b - 1)$. Sekali *m* dapat dihitung, *a* dan *b* dapat dihapus untuk mencegah diketahuinya oleh orang lain.
4. Pilih sebuah bilangan bulat untuk kunci publik, sebut namanya *e*, yang relatif prima terhadap *m*.
5. Bangkitkan kunci dekripsi, *d*, dengan kekongruenan $ed \equiv 1 (mod\ m)$. Lakukan enkripsi terhadap isi pesan dengan persamaan $ci = pi\ e\ mod\ n$, yang dalam hal ini *pi* adalah blok *plainteks*, *ci* adalah *chiperteks* yang diperoleh, dan *e* adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa nilai *pi* harus

terletak dalam himpunan nilai *0, 1, 2, . . . , n – 1* untuk menjamin hasil perhitungan tidak berada di luar himpunan.

6. Proses dekripsi dilakukan dengan menggunakan persamaan *pi = ci d mod n*, yang dalam hal ini *d* adalah kunci dekripsi (kunci privat). Perhatikan pada langkah 4, kekongruenan *ed ≡ 1(mod m)* sama dengan *ed mod m = 1*. Menurut persamaan *chinese remainder theorem (CRT)* yang menyatakan bahwa *a ≡ b (mod m)* ekivalen dengan *a = b + km*, maka *ed ≡ 1(mod m)* ekivalen dengan *ed = 1 + km*, sehingga *d* dapat dihitung dengan: $d = \frac{1+km}{e}$.

2.4 Fungsi *Hash*

Data yang tersimpan di dalam memori komputer perlu ditempatkan dalam suatu cara sedemikian sehingga pencariannya dapat dilakukan dengan cepat. Setiap data yang berupa *record* mempunyai *field* kunci yang unik yang membedakan suatu *record* dengan *record* lainnya. Fungsi *hash* (*hash function*) digunakan untuk menempatkan suatu *record* yang mempunyai nilai kunci *k*. Fungsi *hash* yang paling umum berbentuk $h(k) = \text{mod } m$ yang dalam hal ini m adalah jumlah lokasi memori yang tersedia (misalkan memori berbentuk sel-sel yang diberi indeks 0 sampai $m - 1$). Fungsi h di atas menempatkan *record* dengan kunci k pada suatu lokasi memori yang beralamat $h(k)$. Andaikan $m = 11$, sehingga mempunyai sel-sel memori yang diberi indeks 0 sampai 10, akan disimpan data *record* yang masing-masing mempunyai kunci $15, 558, 32, 132, 102,$ dan 5. Pada mulanya sel-sel memori dalam keadaan kosong (Munir, 2010:214).

Keenam data *record* tersebut masing-masing disimpan pada lokasi yang dihitung sebagai berikut:

$h(15) = 15 \bmod 11 = 4$

$h(558) = 558 \bmod 11 = 8$

$h(32) = 32 \bmod 11 = 10$

$h(132) = 132 \bmod 11 = 0$

$h(102) = 102 \bmod 11 = 3$

$h(5) = 5 \bmod 11 = 5$

Keadaaan sel-sel memori setelah penyimpanan keenam data *record* tersebut digambarkan seperti berikut ini:

132			102	15	5			558		32
0	1	2	3	4	5	6	7	8	9	10

Karena fungsi *hash* bukanlah fungsi satu ke satu (beberapa nilai k yang berbeda dapat menghasilkan nilai $h(k)$ yang sama), maka dapat terjadi bentrokan ($collision$) dalam penempatan suatu data *record*. Misalnya jika menempatkan data *record* dengan kunci 257. Perhitungan *hash* menghasilkan $h(257) = 257 \bmod 11 = 4$.

Untuk mengatasi bentrokan perlu diterapkan kebijakan resolusi bentrokan (*collision resolution policy*). Suatu kebijakan resolusi bentrokan adalah mencari sel tak terisi tertinggi berikutnya (dengan 0 diasumsikan mengikuti 10). Jika di terapkan kebijakan ini, maka data *record* dengan kunci 257 ditempatkan pada lokasi 6.

Jika ingin mencari data *record* tertentu, maka digunakan fungsi *hash* kembali. Misalkan mencari data *record* dengan kunci *p*, maka dihitung $h(p) = q$. Jika *record p* sama dengan isi sel tidak sama dengan isi sel pada lokasi q, maka melihat pada posisi tertinggi berikutnya (sekali lagi, 0 diasumsikan mengikuti 10), jika record *p* tidak berada pada posisi ini, dilihat lagi pada posisi berikutnya, demikian seterusnya. Jika mencapai sel kosong atau kembali ke posisi semula, disimpulkan bahwa *record p* tidak ada (Munir, 2010:215).

2.5 *Brute Force Attack*

Brute force attack adalah metode mengalahkan skema kriptografi dengan mencoba semua kemungkinan *password* atau kunci. *Brute force attack* adalah metode mengalahkan skema kriptografi dengan mencoba jumlah besar kemungkinan *password* atau kunci. *Brute force attack* memungkinkan dapat menyerang kunci privat di hampir semua skema kriptografi, tipe serangan ini bergantung pada ukuran kunci dan mekanisme pada enkripsi yang digunakan (Lastbit, 2005).

Dalam bidang kriptografi, *exhaustive search* merupakan teknik yang digunakan penyerang untuk menemukan kunci enkripsi dengan cara mencoba semua kemungkinan kunci, serangan semacam ini dikenal dengan nama *exhaustive key search attack* atau *brute force attack*. Contoh: panjang kunci enkripsi pada algoritma DES (*Data Encryption Standard*) = 64 bit. Dari 64 bit tersebut, hanya 56 bit yang digunakan (8 bit paritas lainnya tidak dipakai). Jumlah

kombinasi kunci yang harus dievaluasi oleh pihak lawan adalah sebanyak $(2)(2)(2)(2)(2)\ \ldots(2)(2) = 256 = 7.205.759.403.792.794.$ Jika untuk percobaan dengan satu kunci memerlukan waktu 1 detik, maka untuk jumlah kunci sebanyak itu diperlukan waktu komputasi kurang lebih selama 228.493.132 tahun. Meskipun algoritma *exhaustive search* tidak cocok, namun sebagaimana ciri algoritma *brute force* pada umumnya nilai plusnya terletak pada keberhasilannya yang selalu menemukan solusi (jika diberikan waktu yang cukup) (Jadid, 2011).

BAB III

PEMBAHASAN

Kriptografi kunci publik sangat ditentukan oleh kuncinya. Semakin sulit pemecahan algoritma kuncinya maka tingkat keamanannya semakin tinggi. Bab 3 ini adalah bahasan mengenai konsep-konsep matematis yang melandasi pembentukan algoritma RSA, sehingga dapat memperkuat kuncinya. Juga proses penyandian dalam algoritma RSA dan kelebihan serta kelemahan algoritma RSA.

3.1 Konsep Matematis dalam Algoritma RSA

Teori bilangan yang merupakan bagian Ilmu Matematika banyak mendasari disiplin ilmu mengenai komputer dan salah satunya adalah dalam bidang kriptografi terutama algoritma RSA. Algoritma RSA menggunakan bilangan prima sebagai salah satu kuncinya dan mendasarkan kekuatan keamanannya pada masalah logaritma diskrit. Jadi, bilangan prima dan logaritma diskrit adalah bagian dari konsep matematika yang melandasi algoritma RSA.

3.1.1 Bilangan Prima

Bilangan prima memiliki peranan yang sangat penting pada algoritma RSA. Bilangan prima digunakan sebagai salah satu kunci dalam algoritma RSA. Jadi, sangat penting untuk mencari bilangan prima yang besar agar keamanan kunci lebih besar pula. Untuk mencari bilangan prima yang besar dapat menguji keprimaan sebuah bilangan bulat menggunakan tes-tes keprimaan berikut:

3.1.1.1 Tes Keprimaan

3.1.1.1.1 Tes Lehmann

Tes yang paling sederhana adalah menggunakan algoritma Lehmann sebagai berikut:

Algoritma 3.1 (Algoritma Lehmann):

Input: p (yang akan diuji keprimaannya)

Output: p adalah bilangan prima atau bilangan komposit

Langkah:

1) Bangkitkan bilangan acak a yang lebih kecil dari p

2) Hitung $a^{\frac{(p-1)}{2}} \pmod p$

3) Jika $a^{\frac{(p-1)}{2}} \not\equiv 1$ atau $(-1)(mod\ p)$, maka p tidak prima

4) Jika $a^{\frac{(p-1)}{2}} \equiv 1$ atau $(-1)(\text{mod } p)$, maka peluang p bukan prima adalah lima puluh persen.

Pengujian menggunakan algoritma Lehmann dianjurkan diulangi sebanyak lima kali dengan nilai a yang berbeda. Jika hasil perhitungan langkah kedua sama dengan 1 atau (-1), maka peluang p adalah prima mempunyai kesalahan tidak lebih dari lima puluh persen. Bilangan acak yang digunakan pada algoritma Lehmann dapat dipilih nilai yang kecil agar perhitungan lebih cepat. Algoritma Lehmann menentukan keprimaan suatu bilangan dengan cara yang sangat sederhana dan masih sangat diragukan kevalidannya.

3.1.1.1.2 Tes Fermat

Tes fermat adalah tes yang umum dilakukan untuk mencari keprimaan sebuah bilangan, kemudian sedikit diubah menjadi:

Algoritma 3.2 (Algoritma Fermat):

Input: p (yang akan diuji keprimaannya)

Output: p adalah bilangan prima atau bilangan komposit

Langkah:

1. Ambil sebarang bilangan bulat positif a, $2 \leq a \leq p-1$
2. Hitung y
3. Jika $y \neq 1$ maka *output* "komposit"
4. *Output* "prima"

Namun, teorema fermat memiliki kelemahan. Tidak selamanya nilai p yang diperoleh dari $a^{(p-1)} \equiv 1 \pmod{p}$ menghasilkan p sebuah bilangan prima. Contoh:

Diberikan $p = 341$ dan $a = 2$. Maka menurut teorema fermat:

$a^{(p-1)} \equiv 1 \pmod{p}$

$a^{340} \equiv 1 \pmod{341}$

Padahal, $341 = 11 \cdot 31$, habis di bagi oleh bilangan prima, maka 341 adalah sebuah bilangan komposit bukan bilangan prima.

Bilangan bulat seperti 341 ini disebut dengan bilangan prima semu (*pseudo primes*). Bilangan prima semu relatif jarang muncul, maka tes keprimaan suatu bilangan dengan teorema fermat masih dapat digunakan. Tes fermat memiliki kelemahan yang lain yaitu tidak dapat mendeteksi kekompositan bilangan tertentu yang disebut dengan bilangan *Carmichael*.

3.1.1.1.3 Tes Rabin-Miller

Tes Rabin-Miller melengkapi kekurangan dari tes fermat. Segala kekurangan tes fermat telah dapat disempurnakan oleh tes Rabin-Miller. Dapat dibuat algoritma Rabin-Miller sebagai berikut:

Algoritma 3.3 (Algoritma Rabin-Miller):

Input : p, m, dan *b*

Output : p adalah bilangan prima atau bilangan komposit

Langkah:

1) Bangkitkan bilangan acak *a* yang lebih kecil dari *p.*
2) Nyatakan $j = 0$ dan hitung $z = a^m \pmod{p}$.
3) Jika $z = 1$ atau $z = p - 1$, maka *p* lolos dari pengujian dan mungkin prima.
4) Jika $z > 0$ dan $z = 1$, maka *p* bukan prima.
5) Nyatakan $j = j + 1$. Jika $j < 1$ dan $z \neq p - 1$, nyatakan $z = z^2 \pmod{p}$ dan kembali kelangkah (4) jika $z = p - 1$, maka p lolos pengujian dan mungkin prima.
6) Jika $j = b$ dan $z \neq p - 1$, maka *p* tidak prima.

3.2 Uji Algoritma RSA

3.2.1 Kasus Pertama

Berikut ini langkah-langkah algoritma RSA:

1. Pilih dua bilangan prima sembarang, sebut $a = 7$ dan $b = 11$, jaga kerahasiaan a dan b ini.
2. Hitung $n = 7 \times 11$, besaran n = 77 tidak dirahasiakan.

3. Hitung $m = (a - 1)\, x\, (b - 1)$. Sekali m dapat dihitung, 7 dan 11 dapat dihapus untuk mencegah diketahui oleh orang lain, didapat $m = 60$.
4. Pilih satu bilangan bulat untuk kunci publik, sebut namanya $e = 17$, yang relatif prima terhadap 60.
5. Bangkitkan kunci dekripsi d, dengan kekongruenan $ed \equiv 1(mod\ m)$. Lakukan enkripsi terhadap isi pesan dengan persamaan $ci = pi\ e\ mod\ n$, yang dalam hal ini pi adalah blok *plainteks*, ci adalah *chiperteks* yang diperoleh, dan e adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa nilai pi harus terletak dalam himpunan nilai $0, 1, 2, \ldots, n - 1$ untuk menjamin hasil perhitungan tidak berada di luar himpunan.
6. Proses dekripsi dilakukan dengan menggunakan persamaan $pi = ci\ d\ mod\ n$, yang dalam hal ini d adalah kunci dekripsi (kunci privat). Perhatikan pada langkah 4, kekongruenan $ed \equiv 1(\text{mod } m)$ sama dengan $ed \bmod m = 1$. Menurut persamaan *chinese remainder theorem* (CRT) yang menyatakan bahwa $a \equiv b\ (mod\ m)$ ekuivalen dengan $a = b + km$, maka $ed \equiv 1(mod\ m)$ ekuivalen dengan $ed = 1 + km$, sehingga d dapat dihitung dengan: $d = \frac{(1+km)}{e}$, $d = \frac{(1+k\times 60)}{17}$, dengan mencoba nilai-nilai $k = 1, 2, 3, \ldots, 15$ diperoleh nilai d yang bulat adalah 53, kunci ini sebanyak 2 digit, dalam 1 digit adalah 8 bit berarti 16 bit. Ini adalah kunci privat (untuk dekripsi).

Dengan menggunakan program *Microsoft excel* akan dicari nilai d

Nilai d akan dicari dengan menggunakan rumus $d = \frac{(1+k\times60)}{17}$

Tabel 3.1 Mencari Kunci Dekripsi pada $d = \frac{(1+k\times60)}{17}$

No.	Nilai k	Hasil nilai d
1	1	3.588235
2	2	7.117647
3	3	10.64706
4	4	14.17647
5	5	17.70588
6	6	21.23529
7	7	24.76471
8	8	28.29412
9	9	31.82353
10	10	35.35294
11	11	38.88235
12	12	42.41176
13	13	45.94118
14	14	49.47059
15	15	53

3.2.1.1 Proses Enkripsi dan Dekripsi algoritma RSA

Dalam proses enkripsi dan dekripsi algoritma RSA akan diambil contoh plainteks berupa pesan yang bertuliskan "LUTFI". Dalam Kode ASCII *(American Standard for Information Interchange)*, pesan tersebut dipotong berdasarkan blok-blok sebagai berikut:

L: 76

U: 85

T: 84

F: 70

I: 73

Kemudian dalam melakukan proses enkripsi digunakan $c = p^e \bmod n$, sehingga didapat:

$c_1 = 76^{17} \bmod 77 = 76$

$c_2 = 85^{17} \bmod 77 = 57$

$c_3 = 84^{17} \bmod 77 = 28$

$c_4 = 70^{17} \bmod 77 = 49$

$c_5 = 73^{17} \bmod 77 = 61$

Proses inilah yang kemudian dikirim kepada penerima pesan.

Kemudian untuk mengubah pesan sandi menjadi pesan sesungguhnya dilakukan proses dekripsi pesan dengan menggunakan $p = c^d \bmod n$ sehingga didapat:

$p_1 = 76^{53} \bmod 77 = 76$

$p_2 = 57^{53} \bmod 77 = 85$

$p_3 = 28^{53} \bmod 77 = 84$

$p_4 = 49^{53} \bmod 77 = 70$

$p_5 = 61^{53} \bmod 77 = 73$

Dalam proses ini diperoleh pesan asli yang semula yaitu “LUTFI”.

3.2.2 Kasus Kedua

Berikut ini langkah-langkah algoritma RSA:

1. Pilih dua bilangan prima sembarang, sebut $a = 47$ dan $b = 71$, jaga kerahasiaan a dan b ini.
2. Hitung $n = 47 \times 71$, besaran n =3337 tidak dirahasiakan.
3. Hitung $m = (a - 1)\, x\, (b - 1)$. Sekali m dapat dihitung, 47 dan 71 dapat dihapus untuk mencegah diketahui oleh orang lain, didapat m = 3220.

4. Pilih satu bilangan bulat untuk kunci publik, sebut namanya $e = 79$, yang relatif prima terhadap 3220.
5. Bangkitkan kunci dekripsi d, dengan kekongruenan $ed \equiv 1(mod\, m)$. Lakukan enkripsi terhadap isi pesan dengan persamaan $ci = pi\, e\, mod\, n$, yang dalam hal ini pi adalah blok *plainteks*, ci adalah *chiperteks* yang diperoleh, dan e adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa nilai pi harus terletak dalam himpunan nilai $0, 1, 2, \ldots, n - 1$ untuk menjamin hasil perhitungan tidak berada di luar himpunan.
6. Proses dekripsi dilakukan dengan menggunakan persamaan $pi = ci\, d\, mod\, n$, yang dalam hal ini d adalah kunci dekripsi (kunci privat). Perhatikan pada langkah 4, kekongruenan $ed \equiv 1(\mathrm{mod}\, m)$ sama dengan $ed \bmod m = 1$. Menurut persamaan *chinese remainder theorem* (CRT) yang menyatakan bahwa $a \equiv b\,(mod\, m)$ ekuivalen dengan $a = b + km$, maka $ed \equiv 1(mod\, m)$ ekuivalen dengan $ed = 1 + km$, sehingga d dapat dihitung dengan: $d = \frac{(1+km)}{e}$, $d = \frac{(1+k\times 3220)}{79}$, dengan mencoba nilai-nilai $k = 1, 2, 3, \ldots, 25$ diperoleh nilai d yang bulat adalah 1029, kunci ini sebanyak 4 digit, dalam 1 digit adalah 8 bit berarti 32 bit. Ini adalah kunci privat (untuk dekripsi).

Nilai d akan dicari dengan menggunakan rumus $d = \frac{(1+k\times 3220)}{79}$

Tabel 3.2 Mencari Kunci Dekripsi pada $d = \frac{(1+k\times3220)}{79}$

No.	Nilai k	Hasil nilai d
1	1	40.77215
2	2	81.53165
3	3	122.2911
4	4	163.0506
5	5	203.8101
6	6	244.5696
7	7	285.3291
8	8	326.0886
9	9	366.8481
10	10	407.6076
11	11	448.3671
12	12	489.1266
13	13	529.8861
14	14	570.6456
15	15	611.4051
16	16	652.1646
17	17	692.9241
18	18	733.6835
19	19	774.443
20	20	815.2025
21	21	855.962
22	22	896.7215
23	23	937.481
24	24	978.2405
25	25	1019

3.2.2.1 Proses Enkripsi dan Dekripsi algoritma RSA

Dalam proses enkripsi dan dekripsi algoritma RSA akan diambil contoh plainteks berupa pesan yang bertuliskan "LUTFI". Dalam Kode ASCII *(American Standard for Information Interchange)*, pesan tersebut dipotong berdasarkan blok-blok sebagai berikut:

L: 76

U: 85

T: 84

F: 70

I: 73

Kemudian dalam melakukan proses enkripsi digunakan $c = p^e \bmod n$, sehingga didapat:

$c_1 = 76^{79} \bmod 3337 = 1903$

$c_2 = 85^{79} \bmod 3337 = 3048$

$c_3 = 84^{79} \bmod 3337 = 1995$

$c_4 = 70^{79} \bmod 3337 = 1490$

$c_5 = 73^{79} \bmod 3337 = 725$

Proses inilah yang kemudian dikirim kepada penerima pesan.

Kemudian untuk merubah pesan sandi menjadi pesan sesungguhnya dilakukan proses dekripsi pesan dengan menggunakan $p = c^d \bmod n$ sehingga didapat:

$p_1 = 1903^{1019} \bmod 3337 = 76$

$p_2 = 3048^{1019} \bmod 3337 = 85$

$p_3 = 1995^{1019} \bmod 3337 = 84$

$p_4 = 1490^{1019} \bmod 3337 = 70$

$p_5 = 725^{1019} \bmod 3337 = 73$

Dalam proses ini diperoleh pesan asli yang semula yaitu “LUTFI”.

3.2.3 Kasus Ketiga

Berikut ini langkah-langkah algoritma RSA:

1. Pilih dua bilangan prima sembarang, sebut $a = 25799$ dan $b = 12899$, jaga kerahasiaan a dan b ini.

2. Hitung n = 25799 $\times$ 12899, besaran n = 332781301 tidak dirahasiakan.
3. Hitung m = $(a - 1)\, x\, (b - 1)$. Sekali m dapat dihitung, 25799 dan 12899 dapat dihapus untuk mencegah diketahui oleh orang lain, didapat m = 332742604.
4. Pilih satu bilangan bulat untuk kunci publik, sebut namanya $e = 1019$, yang relatif prima terhadap 332742604.
5. Bangkitkan kunci dekripsi d, dengan kekongruenan $ed \equiv 1(\mathrm{mod}\, m)$. Lakukan enkripsi terhadap isi pesan dengan persamaan $ci = pi\ e \bmod n$, yang dalam hal ini pi adalah blok *plainteks*, ci adalah *chiperteks* yang diperoleh, dan e adalah kunci enkripsi (kunci publik). Harus dipenuhi persyaratan bahwa nilai pi harus terletak dalam himpunan nilai $0, 1, 2, \ldots, n - 1$ untuk menjamin hasil perhitungan tidak berada di luar himpunan.
6. Proses dekripsi dilakukan dengan menggunakan persamaan $pi = ci\ d \bmod n$, yang dalam hal ini d adalah kunci dekripsi (kunci privat). Perhatikan pada langkah 4, kekongruenan $ed \equiv 1(\mathrm{mod}\ m)$ sama dengan $ed \bmod m = 1$. Menurut persamaan *chinese remainder theorem* (CRT) yang menyatakan bahwa $a \equiv b\ (mod\ m)$ ekuivalen dengan $a = b + km$, maka $ed \equiv 1(mod\ m)$ ekuivalen dengan $ed = 1 + km$, sehingga d dapat dihitung dengan: $d = \frac{(1+km)}{e}$, $d = \frac{(1+k\times 332742604)}{1019}$, dengan mencoba nilai-nilai k = $1, 2, 3, \ldots, 8$ diperoleh nilai d yang bulat adalah 2612307, kunci ini sebanyak 7 digit, dalam 1 digit adalah 8 bit berarti 56 bit. Ini adalah kunci privat (untuk dekripsi).

Nilai d akan dicari dengan menggunakan rumus $d = \frac{(1+k\times 332742604)}{1019}$

Tabel 3.3 Mencari Kunci Dekripsi pada $d = \frac{(1+k\times 332742604)}{1019}$

No.	Nilai k	Hasil nilai d
1	1	326538.3759
2	2	653076.7507
3	3	979615.1256
4	4	1306153.5
5	5	1632691.875
6	6	1959230.25
7	7	2285768.625
8	8	2612307

3.2.3.1 Proses Enkripsi dan Dekripsi algoritma RSA

Dalam proses enkripsi dan dekripsi algoritma RSA akan diambil contoh plainteks berupa pesan yang bertuliskan "LUTFI". Dalam Kode ASCII *(American Standard for Information Interchange)*, pesan tersebut dipotong berdasarkan blok-blok sebagai berikut:

L: 76

U: 85

T: 84

F: 70

I: 73

Kemudian dalam melakukan proses enkripsi digunakan $c = p^e \bmod n$, sehingga didapat:

$c_1 = 76^{1019} \bmod 332781301 = 166509775$

$c_2 = 85^{1019} \bmod 332781301 = 293761044$

$c_3 = 84^{1019} \bmod 332781301 = 20730870$

$c_4 = 70^{1019} \bmod 332781301 = 297371060$

$c_5 = 73^{1019} \bmod 332781301 = 271722564$

Proses inilah yang kemudian dikirim kepada penerima pesan.

Kemudian untuk merubah pesan sandi menjadi pesan sesungguhnya dilakukan proses dekripsi pesan dengan menggunakan $p = c^d \bmod n$ sehingga didapat:

$p_1 = 166509775^{2612307} \bmod 332781301 = 76$

$p_2 = 293761044^{2612307} \bmod 332781301 = 85$

$p_3 = 20730870^{2612307} \bmod 332781301 = 84$

$p_4 = 297371060^{2612307} \bmod 332781301 = 70$

$p_5 = 271722564^{2612307} \bmod 332781301 = 73$

Dalam proses ini diperoleh pesan asli yang semula yaitu "LUTFI".

Dari hasil beberapa kunci privat di atas kemudian lihat tabel *brute force attack* berikut:

Tabel 3.4 Tabel *Brute Force Attack*

Ukuran bit (Jumlah digit × 8)	**Jumlah kemungkinan kunci ($2^{\text{Jumlah bit}}$)**	**Banyaknya percobaan (0,5 × Jumlah kemungkinan kunci)**	**Lama waktu Banyaknya percobaan $\times 10^6$ percobaan perdetik**
16 bit	$2^{16} = 65536$	32768	3,7 milidetik
32 bit	$2^{32} = 4{,}3 \times 10^9$	2147483648	35,8 menit
56 bit	$2^{56} = 7{,}2 \times 10^{16}$	36028797018963968	1142 tahun
128 bit	$2^{128} = 3{,}4 \times 10^{38}$	170141183460469231731687303 7158841057280	$5{,}4 \times 10^{24}$ tahun

(Sumber: Stalling, 1994).

Penjelasan dari Tabel *brute force attack:*

Setiap karakter terdiri dari 1 byte, setiap 1 byte = 8 bit. Untuk mengetahui jumlah kemungkinan kunci digunakan rumus jumlah kemungkinan kunci = $2^{\text{bit kunci}}$, kemudian untuk mencari banyaknya percobaan dilakukan dengan cara 0,5 ×

jumlah kemungkinan kunci, dan untuk mencari lama waktu percobaan dengan cara banyaknya percobaan $\times$ 10^6/detik, untuk keterangan 1 tahun = 31536000. Dari tabel 3.4 dapat diketahui bahwa semakin besar ukuran kunci dari kunci privat akan semakin sulit dibobol oleh *brute force attack*. Kriptografi kunci publik sangat ditentukan oleh kuncinya. Semakin sulit pemecahan algoritma kuncinya maka tingkat keamanannya semakin tinggi dan semakin aman kuncinya maka algoritma ini semakin tahan terhadap *brute force attack*.

Dari permasalahan ketiga, didapatkan bahwa dalam pendekripsian pesan tidak didapatkan nilai untuk mendekripsikan pesan, dari tabel *brute force attack* di atas dapat diketahui nilai *d* berjumlah 7 digit adalah 56 bit, maka untuk mencari kunci privatnya dalam pendekripsikan pesan dibutuhkan 1142 tahun, algoritma tetap dapat dibobol jika ada waktu cukup untuk melakukannya.

3.3 Kelebihan dan Kekurangan Algoritma RSA

3.3.1 Kelebihan Algoritma RSA

Kelebihan utama dari RSA yang merupakan kriptografi kunci publik adalah menambah keamanan dan kenyamanan. Kunci privat tidak pernah diperlukan untuk dikirim atau diberi tahu ke orang lain. Pada sebuah sistem kunci rahasia, secara terang-terangan kunci rahasia ini harus dikirim (dapat secara manual atau melalui sebuah saluran komunikasi), dan akan terjadi suatu kemungkinan di mana penyerang dapat mencari tahu kunci rahasia tersebut saat proses pengiriman. Kelebihan utama lainnya adalah sistem RSA yang merupakan sistem kunci publik ini dapat menyediakan sebuah metode untuk tanda tangan

digital atau tanda tangan elektronik. Autentikasi melalui kunci rahasia memerlukan pembagian dari beberapa rahasia dan terkadang juga memerlukan rasa kepercayaan terhadap pihak ketiga. Sebagai hasilnya, pengirim dapat menolak pesan autentikasi sebelumnya dengan cara membuktikan bahwa rahasia yang dibagikan bagaimanapun caranya disetujui oleh pihak lain yang berbagi rahasia tersebut.

3.3.2 Kekurangan Algoritma RSA

Kekurangan dari pemakaian kriptografi kunci publik, dalam hal ini RSA, adalah dalam masalah kecepatan. Banyak metode enkripsi kunci rahasia yang populer yang memiliki kecepatan enkripsi-dekripsi yang lebih cepat dibandingkan dengan metode enkripsi kunci publik yang ada sekarang. Namun kriptografi kunci publik dapat digunakan dengan kriptografi kunci rahasia untuk mendapatkan metode enkripsi yang terbaik di dunia. Untuk enkripsi, solusi terbaik adalah dengan cara mengkombinasi sistem kunci publik dan sistem kunci rahasia untuk mendapatkan kedua keuntungan yang dimiliki oleh kedua metode enkripsi ini, keuntungan keamanan dari segi sistem kunci publik, dan keuntungan kecepatan dari segi sistem kunci rahasia. Sistem kunci publik dapat digunakan untuk mengenkripsi sebuat kunci rahasia, yang dapat digunakan untuk mengenkripsi file atau pesan yang berukuran besar sekalipun. Kriptografi kunci publik dapat menjadi lemah terhadap pemalsuan identitas *user*, bagaimana pun juga, walaupun jika kunci privat dari pemakai tidak tersedia. Serangan yang sukses pada otoritas sertifikasi akan memperbolehkan lawan untuk menyelinap siapapun yang lawan

pilih dengan cara memilih sertifikat kunci publik dari sebuah otoritas yang memilikinya untuk menggabungkan kunci tersebut ke nama *user* yang lain.

3.4 Cara Allah Menjaga Kesucian Al-Qur'an

Dalam surat sebelumnya yaitu surat Al-Hijr: 9 bahwa Allah menjaga kesucian Al-Qur'an, dalam ayat ini adalah cara Allah menjaga kesucian Al-Qur'an pada surat Al-Qiyamah ayat 17-18:

إِنَّ عَلَيْنَا جَمْعَهُۥ وَقُرْءَانَهُۥ ﴿١٧﴾ فَإِذَا قَرَأْنَٰهُ فَٱتَّبِعْ قُرْءَانَهُۥ ﴿١٨﴾

Artinya: *"Sesungguhnya atas tanggungan Kami-lah mengumpulkannya (di dadamu) dan (membuatmu pandai) membacanya, apabila Kami telah selesai membacakannya maka ikutilah bacaannya itu".*

Dalam ayat ini Allah menjelaskan sebab larangan mengikuti bacaan Jibril ketika dia sedang membacakannya itu, adalah karena: sesungguhnya atas tanggungan Allah-lah mengumpulkannya di dalam dada Muhammad dan membuatnya pandai membacanya. Allah-lah yang bertanggung jawab bagaimana supaya Al-Qur'an itu tersimpan dengan baik dalam dada atau ingatan Muhammad, dan memantapkannya dalam kalbunya. Allah pula yang memberikan bimbingan kepadanya bagaimana cara membaca ayat itu dengan sempurna dan teratur, sehingga Muhammad hafal dan tidak lupa selama-lamanya. Oleh sebab itu bila Jibril selesai membacakan ayat-ayat yang harus diturunkan, hendaklah Muhammad menuruti membacanya. Nanti Muhammad mendapatkan dirinya selalu ingat dan hafal akan ayat-ayat itu. Tegasnya pada waktu Jibril membaca, hendaklah Muhammad diam dan mendengarkan bacaannya. Dari sudut lain ayat ini juga berarti: Bila telah selesai dibacakan kepada Muhammad ayat-ayat Allah

hendaklah umat Muhammad segera mengamalkan hukum-hukum dan syariat-syariatnya. Semenjak turunnya perintah ini Rasulullah senantiasa mengikuti dan mendengarkan dengan penuh perhatian wahyu yang dibacakan Jibril. Setelah Jibril pergi, barulah beliau membacanya dan bacaannya itu tetap tinggal dalam ingatan beliau. Demikian diterangkan dalam hadis Bukhari dari Siti 'Aisyah (Ihsan, 2013).

Begitu juga pada algoritma RSA, di dalam penelitian-penelitian sebelumnya sudah teruji bahwa untuk meningkatkan keamanan pada algoritma RSA hendaknya mempersulit pemecahan algoritma kuncinya, dengan memperpanjang bilangan primanya, jika sudah teruji kebenarannya hendaknya mengikuti cara tersebut agar mendapatkan keamanan terhadap serangan-serangan kriptografi.

BAB IV

PENUTUP

4.1 Kesimpulan

Dari kasus ketiga diketahui nila $a = 25799$, $b = 12899$, sehingga didapat nilai $n = 332781301$, $m = 332742604$. Untuk melakukan proses enkripsi digunakan $e = 1019$, yang relatif prima terhadap 332742604. Kemudian setelah melakukan proses enkripsi dibutuhkan proses dekripsi untuk mendapatkan pesan asli dengan $d = \frac{(1+km)}{e}$, dengan nilai $k = 8$, sehingga didapat $d = \frac{(1+k\times 332742604)}{1019}$ dan diperoleh nilai $d = 2612307$. Dari Tabel *brute force attack* diketahui nilai d berjumlah 7 digit adalah 56 bit, maka untuk mencari kunci privatnya dalam pendekripsikan pesan dibutuhkan 1142 tahun. Semakin besar ukuran kunci dari kunci privat akan semakin sulit dibobol oleh *brute force attack.*

Algoritma ini tetap dapat dibobol jika ada waktu untuk melakukannya. Kriptografi kunci publik sangat ditentukan oleh kuncinya. Semakin sulit pemecahan algoritma kuncinya maka tingkat keamanannya semakin tinggi sehingga algoritma RSA tahan terhadap *brute force attack.*

4.2 Saran

Pada penulisan skripsi selanjutnya dapat meneruskan kunci privat berjumlah 16 digit, tetapi harus menggunakan komputer lebih canggih lagi untuk melakukan pendekripsian pesan.

DAFTAR PUSTAKA

Ana, D.. 2012. Jaminan Keamanan Mau?. *http://Jaminan Keamanan.htm* (diakses pada tanggal 25 Mei 2013).

Anonim. 2012. Tafsir Al-Baqarah 111-120. *http://TafsirKitab Al-Qur'an Al-Baqarah 11-120.htm*(diakses pada tanggal 20 Juni 2013).

Abdussakir. 2009. *Matematika 1 Kajian Integratif Matematika dan Al-Qur'an.* Malang: UIN Malang Press.

Adharani, Y..2010. Tafsir Q.S Al-Hijrayat 9.*http:// Tafsir Q.S Al-Hijrayat 9 htm* (diakses pad atanggal 23 Juni 2013).

Arifin, Z.. 2009. *Studi Kasus Penggunaan Algoritma RSA Sebagai Algoritma Kriptografi yang Aman*, Jurnal Informatika Mulawarman. Pdf (diakses tanggal 3 September 2009).

Ariyus, D.. 2006. *Kriptografi*. Yogyakarta: CV. Andioffset.

Buseng, V.. 2013. Sistem Bilangan Biner. *http://catatan-goblog.blogspot.com/2013/04/sistem-bilangan-biner.html* (diakses pada tanggal 23 September 2013).

Flourensia, S.R.. 2005. Cryptografi. *http://cryptografi/124p/04/final0.1.(diakses* pada tanggal 06 Agustus 2009).

Hamidah, S.N..2009. Konsep Matematis dan Proses Penyandian Kriptografi ElGamal. *Skripsi* tidak diterbitkan. Malang: Universitas Islam Negeri Malang.

Haro, G.A.. 2006. *Studi dan Penggunaan Algoritma RSA Sebagai Algoritma Kriptografi yang Aman*, 2006-2007/Makalah/Makalah 0607-101. pdf, diakses tanggal 4 November 2009.

Ihsan, M.. 2013. Tafsir surat: Al-Qiyaamah. *http://Alquran_Tafsir.asp?suratke=75* (diakses pada tanggal22Agustus 2013).

Jadid, A.. Ilmu Yunta fa'u Bihi. *http://ilmunyuntafa'ubihi Algoritma Brute Force.htm* (diakses tanggal 19 Maret 2013).

Kurniawan, A.. 2008. *Konsep dan Implementasi Cryptography Dengan.NET*. Jakarta: PC Media.

LastBit. 2005. Brute Force Attack. *http://www.lastbit.com/rm_bruteforce.asp* (diakses pada tanggal 12 Mei 2009).

Muhsetyo, G.. 1997. *Dasar-Dasar Teori Bilangan*. Jakarta: PGSM.

Munir, R.. 2006. *Kriptografi*. Bandung: Informatika Bandung.

Munir, R.. 2010. *Matematika Diskrit*. Bandung: Informatika Bandung.

Stallings, W.. 1994. *Data and Computer Communication*. New York: Macmillan Publishing Company.

Stinson, D.R.. 1995. *Cryptography Theory and Practice*. Florida: CRC Press, Inc.

Tomiexz. 2011. Teori Bilangan. *http://tomiexz.wordpress.com/2011/11/23/teori-bilangan/* (diakses pada tanggal 23 September 2013).

Lampiran 1. Tabel Kode ASCII

No.	Kode
0	NULL (null)
1	SOH (start of heading)
2	STX (start of text)
3	ETX (end of text)
4	EOT (end of transmission)
5	ENQ (enquiry)
6	ACK (acknowledge)
7	BEL (bell)
8	BS (backspace)
9	TAB (horizontal tab)
10	LF (new line)
11	VT (vertical tab)
12	FF (new page)
13	CR (carriage return)
14	SO (shift out)
15	SI (shift in)
16	DLE (data link espace)
17	DC1 (device control 1)
18	DC2 (device control 1)
19	DC3 (device control 1)
20	DC4 (device control 1)
21	NAK (negative acknowledge)
22	SYN (synchronus idle)
23	ETB (end of trans. Blok)
24	CAN (cancel)
25	EM (end of medium)
26	SUB (substitute)
27	ESC (escape)
28	FS (file separator)
29	GS (group separator)
30	RS (record separator)
31	US (unit separator)
32	space
33	!
34	"
35	#
36	$
37	%

No.	Kode
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[
92	\
93	]
94	^
95	_
96	‘
97	a
98	b
99	c
100	d
101	e
102	f

38	&
39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@

103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124	|
125	}
126	~
127	DEL

Lampiran 2

Mencari nilai d akan dicari dengan rumus $d = \frac{(1+k\times 60)}{17}$ menggunakan Microsoft Excel:

No.	Nilai k	Hasil nilai d
1	1	3.588235
2	2	7.117647
3	3	10.64706
4	4	14.17647
5	5	17.70588
6	6	21.23529
7	7	24.76471
8	8	28.29412
9	9	31.82353
10	10	35.35294
11	11	38.88235
12	12	42.41176
13	13	45.94118
14	14	49.47059
15	15	53
16	16	56.52941176
17	17	60.05882353
18	18	63.58823529
19	19	67.11764706
20	20	70.64705882

Lampiran 3

Mencari nilai d akan dicari dengan rumus $d = \frac{(1+k\times3220)}{79}$ menggunakan Microsoft Excel:

No.	Nilai k	Hasil nilai d
1	1	40.77215
2	2	81.53165
3	3	122.2911
4	4	163.0506
5	5	203.8101
6	6	244.5696
7	7	285.3291
8	8	326.0886
9	9	366.8481
10	10	407.6076
11	11	448.3671
12	12	489.1266
13	13	529.8861
14	14	570.6456
15	15	611.4051
16	16	652.1646
17	17	692.9241
18	18	733.6835
19	19	774.443
20	20	815.2025
21	21	855.962
22	22	896.7215
23	23	937.481
24	24	978.2405
25	25	1019
26	26	1059.759494
27	27	1100.518987
28	28	1141.278481
29	29	1182.037975
30	30	1222.797468

Lampiran 4

Mencari nilai d akan dicari dengan rumus $d = \frac{(1+k\times 332742604)}{1019}$ menggunakan Microsoft Excel:

No.	Nilai k	Hasil nilai d
1	1	326538.3759
2	2	653076.7507
3	3	979615.1256
4	4	1306153.5
5	5	1632691.875
6	6	1959230.25
7	7	2285768.625
8	8	2612307
9	9	2938845.375
10	10	3265383.75
11	11	3591922.125
12	12	3918460.5
13	13	4244998.874
14	14	4571537.249
15	15	4898075.624