# Aplikasi Teori Bilangan Dalam Kriptografi Untuk Keamanan Teknologi Informasi Dalam Bentuk Algoritma RSA

Rizki Ihza Parama - 13515104[1]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*13515104@itb.ac.id*

**Abstrak— Di zaman modern ini dimana segala aktivitas manusia dibantu oleh komputer keamanan data dan sistem sangatlah krusial bagi keberlangsungan sistem yang telah ada. Apabila peretas dapat mencuri dan membobol sistem dengan mudah maka kepercayaan terhadap sistem-sistem digital akan memudar. Oleh karena itu untuk menjaga sistem yang ada dari serangan pihak-pihak yang tidak bertanggung jawab dibutuhkan sebuah sistem enkripsi yang ampuh. Salah satu sistem ini adalah algoritma RSA yang memanfaatkan kesulitan dalam pemfaktoran perkalian dua bilangan prima. Belum adanya algoritma yang dapat memfaktorkan hasil kali dua bilangan prima menyebabkan perkalian dua bilangan prima dapat menjadi basis teknik pengamanan yang ampuh untuk melindungi data dan informasi penting dari serangan peretas.**

**Keywords—Bilangan Prima, Pemfaktoran, RSA.**

## I. PENDAHULUAN

Pada zaman modern ini data merupakan hal yang sangat krusial untuk dilindungi. Penggunaan data-data pribadi pada banyak situs-situs dan jasa-jasa online membuat proteksi terhadap sistem website dan basis data menjadi penting untuk menjaga privasi para pengguna teknologi informasi.

Informasi-informasi seperti kartu kredit, alamat rumah, nomor telepon, dan data-data pribadi merupakan info-info yang sering user simpan di media-media online sehingga apabila sistem rentan akan serangan para peretas maka pengguna-pengguna yang tidak bersalah dapat dirugikan.

Selain informasi pribadi ada juga informasi-informasi yang sangat krusial seperti rahasia bisnis atau rahasia negara.

Sangat pentingnya sebuah informasi menyebabkan informasi tersebut hanya boleh diakses oleh orang-orang tertentu saja. Jatunya informasi kepada pihak lain yang tidak memiliki otoritas untuk mengetahui informasi tersebut dapat merugikan sebuah organisasi secara keseluruhan. Untuk itu keamanan dari sistem informasi yang digunakan haruslah terjamin dalam batas yang dapat diterima.

Salah satu platform dimana kejahatan cyber sangat mungkin terjadi adalah internet. Internet menghubungkan jutaan bahkan milyaran komputer di dunia sehingga sangat mungkin seorang peretas dapat masuk ke sistem keamanan dari komputer lain hanya dengan menggunakan koneksi internet. Tempat dimana kejahatan internet dapat terjadi adalah melalui website.

Website dapat berupa website statis atau dinamis. Website statis tidak menjadi masalah karena website statis hanya berfungsi sebagai sumber informasi konstan yang cenderung tidak berubah. Website dinamis adalah website yang secara berkala informasi di dalamnya berubah dengan berhubungan dengan user dari website tersebut. Data-data milik disimpan dalam sebuah database yang berada di server-server tertentu.

Pada sebuah website yang dinamis biasanya terdapat form yang dapat diisi oleh member sehingga bisa berinteraksi secara penuh dengan website tersebut. Selain itu kita biasanya diharuskan untuk menjadi member dari website tersebut dan sebelumnya kita harus membuat account dengan membuat username dan password yang akan dibutuhkan ketika kita masuk atau login ke website itu.

Untuk menjaga keamanan dari password atau username, biasanya digunakan teknik enkripsi agar kerahasiaan data tersebut terjamin. Jelaslah bahwa teknik enkripsi sangat penting dalam pengamanan data. Karena apabila password yang dikirim tidak dienkripsi maka akibatnya adalah peretas yang menyadap komunikasi dengan server ditengah jalan akan dapat mengetahui password pengguna.

Apabila server tidak diproteksi serta informasi tidak dienkripsi maka peretas dapat membobol server dan dapat langsung memahami informasi yang terdapat di dalam website.

Kegunaan enkripsi adalah meskipun peretas sudah masuk ke dalam server dia tidak bisa langsung memahami informasi karena informasi ter-enkripsi dalam bentuk lain.

Salah satu algoritma enkripsi yang sering digunakan adalah algoritma RSA. RSA adalah sistem sandi yang saat ini praktis menjadi standar de facto dunia. Pada makalah ini fokus pembahasan ada di algoritma RSA karena meskipun konsep matematika yang digunakan cukup sederhana namun kekuatan dari algoritma ini sudah terbukti.

Sandi ini adalah hasil inovasi Ron Rivest, Adi Shamir dan Leonard Adleman di tahun 1987. Mereka kemudian

mendirikan perusahaan RSA Data Security Inc, yang memiliki paten terhadap sandi RSA. Mekanisme kerja RSA cukup sederhana dan mudah dimengerti, tetapi kokoh. Sampai saat ini satu-satunya cara untuk mendobraknya adalah dengan cara mencoba satu persatu kombinasi kunci yang mungkin atau yang biasa disebut brute force attack. Penentuan tingkat keamanan suatu sandi dari kemungkinan dibongkar adalah seberapa panjang dari sandi (ukuran kunci) tersebut. Karena jika semakin panjang suatu kode, maka semakin banyak pula kombinasi kunci yang mungkin ada.

# II. TEORI BILANGAN

Teori Bilangan adalah cabang ilmu matematika yang mempelajari sifat-sifat bilangan bulat. Beberapa dasar teori bilangan akan dibahas di subbagian-subbagian dari bagian ini.

## 2.1 PBB

PBB adalah faktor pembagi terbesar dari dua buah bilangan bulat. atau dengan kata lain PBB dari sebuah bilangan bulat a dan b adalah sebuah bilangan bulat k dengan k adalah bilangan terbesar dengan properti yaitu k membagi a dan k membagi b. Notasi yang umum dipakai untuk menyatakan PBB adalah gcd. Untuk angka-angka diatas maka ditulis gcd(a,b)=k.

Contoh PBB dari 10 dan 5 adalah 5 karena 5 adalah bilangan bulat terbesar yang membagi 10 dan membagi 5.

## 2.2 Relatif Prima

Dua buah bilangan a dan b disebut relatif prima apabila gcd(a,b)=1.

Contoh 3 dan 5 relatif prima karena angka 1 adalah bilangan bulat terbesar yang membagi 3 dan membagi 5.

## 2.3 Algoritma Euclid

Algoritma euclid adalah algoritma yang memungkinan menghitung PBB dari dua buah angka dengan mudah. Algoritma euclid secara rekursi dinyatakan sebagai berikut:

1. apabila a dan b tidak sama dengan 0. Misal c adalah angka terbesar diantara a dan b dan d adalah angka terkecil diantara a dan b maka berlaku gcd(a,b) = gcd(c,d) = gcd(d,c mod d)

2. Apabila salah satu dari a dan b adalah 0 maka gcd a dan b adalah bilangan yang tidak 0.

simbol "mod" pada algoritma diatas adalah operasi sisa pembagian. Detail mengenai "mod" akan dibahas di subbab selanjutnya. Contoh penggunaan algoritma euclid: gcd(12,8)=gcd(8,4)=gcd(4,0), maka gcd(12,8)=4.
*Pseudo code* untuk algoritma euclid adalah sebagai berikut

```
function euclid(a,b:integer) ->integer

ALGORITMA
  if (a=0) ->b
  else if (b=0) ->a
  else if (a>b) then
        ->gcd(b,a mod b)
  else
        ->gcd(a,b mod a)
```

## 2.4 Aritmatika Modulo

Aritmatika modulo memiliki peranan yang sangat penting dala kriptografi. Dalam aritmatika modulo terdapat operasi yang sangat dominan yaitu operasi mod. Operasi mod adalah operasi sisa pembagian.

notasi yang digunakan untuk aritmatika modulo adalah a mod b=c, ini ekivalen dengan a=kb+c.

contoh: 5 mod 4=1 karena 5=4x1+1

## 2.5 Kekongruenan

dua angka yang berbeda dikatakan kongruen dalam modulo m apabila sisa pembagian keduanya terhadap m sama.

Notasi untuk kekongruenan adalah sebagai berikut.

$$a \equiv b \bmod m$$

contoh: $5 \equiv 25 \bmod 4$ karena 25 dan 5 sama-sama menghasilkan sisa 1 apabila dibagi dengan 4.

## 2.6 Inverse Modulo

sebuah bilangan bulat a dikatakan memiliki inverse dalam modulo m apabila terdapat bilangan bulat x sehingga

$$ax \equiv 1 \bmod m$$

contoh: 5 memiliki inverse yaitu 4 dalam modulo 19 karena $5x4 \equiv 1 \bmod 19$

## 2.7 Bilangan Prima

Sebuah bilangan p dikatakan prima jika ia bukan 1 dan tidak bisa dibagi bilangan selain 1 dan dirinya sendiri.

contoh bilangan prima: 2,3,5,7,dst.

## 2.8 Teorema Fermat

Teorema fermat adalah teorema yang sangat penting untuk algoritma RSA. Teorema fermat adalah sebagai berikut: jika a dan p relatif prima dan p adalah bilangan prima maka berlaku.

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

## 2.9 Pemfaktoran

Pemfaktoran adalah cara untuk memfaktorkan sebuah bilangan menjadi faktor-faktor primanya. Sampai saat ini belum ada algoritma pemfaktoran yang efektif dapat memfaktorkan sebuah angka dengan cepat.

Ini adalah faktor kunci dalam kesuksesan algoritma RSA karena properti ini menjamin bahwa untuk memecahkan sebuah enkripsi RSA peretas harus melakukan brute force.

# III ALGORITMA RSA

Algortima RSA dijabarkan pada tahun 1977 oleh tiga orang : Ron Rivest, Adi Shamir dan Len Adleman dari Massachusetts Institute of Technology. Huruf RSA itu sendiri berasal dari inisial nama mereka (Rivest — Shamir — Adleman).

Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima. Pemfaktoran dilakukan untuk memperoleh kunci pribadi. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang mangkus, maka selama itu pula keamanan algoritma RSA tetap terjamin.

Sekedar trivia, Clifford Cocks, seorang matematikawan Inggris yang bekerja untuk GCHQ, menjabarkan tentang sistem yang equivalen pada dokumen internal di tahun 1973. Penemuan Clifford Cocks tidak terungkap hingga tahun 1997 karena alasan top-secret classification.

Inti dari RSA adalah menghasilkan dua buah kunci yaitu kunci public dan kunci privat. Kunci publik akan dibagikan untuk men-enkprisi informasi. Pesan yang di enkripsi dengan kunci public hanya bisa dipecahkan dengan kunci privat.

## 3.1 Menghasilkan kunci publik dan kunci privat

Ada beberapa langkah untuk menghasilkan kunci privat. Langkah-langkah itu adalah sebagai berikut

1. Pilih dua bilangan prima p dan q secara random. Agar susah di pecahkan, p dan q haruslah berupa bilangna prima yang besar.
2. hitung n=pq.
3. hitung r=(p-1)(q-1)
4. pilih bilangan baru, katakanlah e yang memenuhi syarat $1<e<r$ dan gcd(e,r)=1. e akan di publish sebagai kunci publik.
5. Cari d sehingga $de \equiv 1 \pmod{r}$ atau dengan kata lain d adalah inverse dari e di modulo r. d disimpan sebagai kunci privat dan tidak dishare.

## 3.2 Enkripsi Pesan

Setelah kunci dihasilkan maka langkah selanjutnya adalah mengenkripsi sebuah pesan yang ingin diproteksi. Untuk kemudahan dari bagian ini maka diasumsikan pesan berupa sebuah bilangan bulat m dengan m kurang dari n. Cara untuk mengenkripsi m adalah dengan memangkatkan m sebanyak e kali dimana e adalah kunci publik yang telah dihitung sebelumnya. Misalkan kunci yang dihasilkan adalah c.

$$c = m^e \bmod n$$

Jadi c adalah informasi m yang telah di enkripsi. c dapat dilihat oleh publik namun tanpa kunci privat informasi sesungguhnya dari pesan ini tidak bisa didapatkan.

## 3.3 Dekripsi Pesan

Tahap selanjutnya adalah memecahkan sandi yang telah dibuat sebelumnya dengan bantuan kunci privat.

Cara yang mudah untuk melakukan ini adalah dengan memangkatkan c dengan d. Misalkan angka baru itu f.

$$f = c^d \bmod n$$

Setelah itu kita tinggal membuktikan kalau f itu sama dengan m. Pembuktian dapat dilakukan dengan mengganti c dengan $m^e$.

$$f = m^{ed} \bmod n$$

perhatikan bahwa $ed \equiv 1 \bmod((p-1)(q-1))$.

$$ed \equiv 1 \bmod (p-1)$$
$$ed \equiv 1 \bmod (q-1)$$

sekarang saatnya kita mengaplikasikan teorema fermat.

$$m^{ed} \equiv m^{k*(p-1)+1} \equiv m \bmod p$$
$$m^{ed} \equiv m^{k*(q-1)+1} \equiv m \bmod q$$

dengan chinese remainder theorem didapat

$$m^{ed} \equiv m \bmod n$$
$$f \equiv m \bmod n$$

karena f dan m kurang dari n

$$f = n$$

pembuktian ini membuktikan bahwa pesan asli bisa didapat kembali dengan kunci privat meskipun telah mengalami enkripsi oleh kunci publik.

## 3.4 Kelebihan algoritma RSA

Kekuatan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor primanya. Penemu algoritma RSA menyarankan nilai a dan b panjangnya lebih dari 100 digit. Dengan demikian hasil kali n = ab akan berukuran lebih dari 200 digit.

Bayangkanlah berapa besar usaha kerja yang diperlukan untuk memfaktorkan bilangan bulat 200 digit menjadi factor primanya. Mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun (dengan asumsi bahwa algoritma pemfaktoran yang

digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai
mempunyai kecepatan 1 mildetik).

### 3.5 Kelemahan algoritma RSA

Sekali n berhasil difaktorkan menjadi a dan b, maka r = (a - 1)(b - 1) dapat dihitung. Selanjutnya karena kunci enkripsi diutamakan e diumumkan (tidak rahasia), maka kunci deskripsi d dapat dihitung dari persamaan ed = 1 (mod r). Ini berarti proses deskripsi dapat dilakukan oleh orang yang tidak berhak.

Untunglah algoritma yang paling mampu untuk memfaktorkan bilangan yang besar belum ditemukan. Inilah yang membuat algoritma RSA tetap dipakai hingga saat ini. Selagi belum ditemukan algoritma yang bisa untuk memfaktorkan
bilangan bulat menjadi faktor primanya, maka algoritma RSA masih direkomendasikan untuk penyandian pesan.

## IV. PENGAPLIKASIAN ALGORITMA RSA PADA STRING

Kebanyakan dari informasi penting yang digunakan oleh manusia ada dalam tipe string. Jadi harus dipikirkan cara merubah informasi dalam bentuk string kedalam bentuk integer agar dapat diproses dengan algoritma RSA. Cara untuk merubah string dapat dengan merubahnya karakter per karakter ke dalam kode ASCII-nya.

### 4.1 Kode ASCII

Kode ASCII(American Standard Code for Information Interchange) enkoding standar untuk merubah karakter menjadi integer. Kode ASCII telah digunakan di bahasa pemrogramman untuk memproses karakter karena komputer hanya mengenal 1 dan 0 dan tidak mengenal karakter. Daftar kode ASCII dapat dilihat di

http://www.asciitable.com/

contoh translasi dari string ke ASCII:
misal kalimat “AKUKAMUMAKANIKAN"

“AKU” menjadi 657585
“KAMU” menjadi 75657785
“MAKAN" menjadi 7765756578
“IKAN” menjadi 73756578

### 4.2 Penggunaan RSA untuk enkripsi string

Setelah pesan berupa string dirubah kedalam kode ASCII maka dapat di enkripsi menggunakan algoritma RSA seperti telah dijelaskan sebelumnya.

Agar hasil konversi string ke integer tetap kurang dari n maka harus dilakukan pemisahan terhadap hasil enkripsi. Pemisahan dilakukan sesuai dengan besarnya n. Misalkan dipilih memisahkan per dua karakter, hasil pemisahan terhadap string pada contoh sebelumnya menjadi:

1. “AKU” dipisah menjadi “AK” dan “U”, kode ASCII menjadi 6575 dan 85.

2. “KAMU” dipisah menjadi “KA" dan “MU”, kode ASCII menjadi 7565 dan 7785.

3. “MAKAN” dipisah menjadi “MA”,”KA”, dan ”N”, kode ASCII menjadi 7765, 7565, dan 78.

4. “IKAN" dipisah menjadi “IK" dan “AN”, kode ASCII menjadi 7375 dan 6578.

Setelah string dipisah seperti ini kita dapat melakukan algoritma RSA seperti pada bab sebelumnya. Perlu diperhatikan bahwa bilangan hasil dua bilangan prima yang dipakai harus lebih besar dari semua pesan yang akan di enkripsi, hal ini untuk menjamin bahwa pesan didekripsi dengan kunci privat akan sama seperti pesan semula.

## V. ENKRIPSI DI WEB

Kebanyakan informasi di internet sudah terenkripsi demi menanggulangi kebocoran informasi akibat intervensi pihak luar di tengah-tengah pemindahan data.

Salah satu hal yang penting untuk di enkripsi adalah password pengguna. Password pengguna yang dikirim ke database untuk di verifikasi dienkripsi terlebih dahulu untuk mencegah penyadapan data.

Kunci privat yang harus diberikan kepada user juga di enkripsi dengan algoritma RSA sehingga apabila disadap tidak akan dapat diketahui. Enkripsi ini dilakukan dalam beberapa lapis sehingga akan sulit untuk dipecahkan. Hal yang terpenting adalah setiap kali user mengirimkan file atau mengirim password, sistem akan membuat kunci privat dan kunci publik baru sehingga proses pemindahan file berjalan dengan aman.

## VI. KESIMPULAN

Algoritma RSA adalah contoh klasik dari aplikasi teori bilangan ke kehidupan sehari-hari. Penggunaan enkripsi RSA sudah dilakukan dalam banyak sekali teknologi informasi seperti web, perbankan, dll. Untuk mengenkripsi sebuah string perlu dilakukan perubahan ke integer karena karakter tidak bisa diproses dengan algoritma RSA.

Setelah ditelaah lebih jauh dapat dilihat bahwa konsep matematika yang digunakan dalam algoritma RSA tidaklah terlalu rumit. Hal yang membuat RSA sulit dipecahkan walaupun algoritmanya tidak terlalu rumit adalah karena pemfaktoran bilangan masih merupakan proses yang sulit dan belum ditemukan cara mudah untuk menyelesaikannya. Jika dimasa yang akan datang ditemukan cara untuk memfaktorkan bilangan dengan mudah maka harus dicari cara lain untuk mengamankan informasi dari serangan para peretas.

## VII. SARAN

Saran dari penulis adalah penggunaan algoritma RSA harus diperhartikan oleh website-website baru yang mengharuskan pengguna untuk memasukan data pribadi ke dalam webiste tersebut saat melakukan regristrasi.

RSA mudah diimplementasi dan dimengerti oleh mahasiswa S1 sekalipun sehingga menjadikannya cocok untuk dipakai di startup-startup yang berisi anak muda.

## VIII. UCAPAN TERIMA KASIH

Pertama-tama penulis ingin mengucapkan terima kasih kepada oranguta penulis karena tanpa mereka penulis tidak dapat berkuliah dan mengenyam pendidikan di Insittut Teknologi Bandung. Setelah itu penulis juga ingin mengucapkan terima kasih kepada Bpk. Rinaldi yang sudah mengajar penulis di mata kuliah matematika diskrit. Ilmu yang penulis dapat di mata kuliah ini memungkinkan penulis untuk mengerti bahan yang penulis bahas di makalah ini.

## REFERENSI

1. Artikel tentang dasar dan contoh penerapan algoritma RSA http://math.stackexchange.com/questions/237268/cryptography-rsa-algorithm-basic-question diakses tanggal 7 Desember 2016
2. Kenneth H. Rosen “Elementary Number Theory and its Application” 5th edition.
3. Artikel tentang algoritma RSA dan pembahasan lengkap mengenai pembuatan kunci, enkripsi, dkripsi https://simple.wikipedia.org/wiki/RSA_(algorithm) diakses tanggal 7 Desember 2016.
4. Bagaimana password dienkripsi dan disimpan di internethttp://lifehacker.com/5919918/how-your-passwords-are-stored-on-the-internet-and-when-your-password-strength-doesnt-matter diakses tanggal 7 Desember 2016.
5. Pelajaran mengenai algoritma euclidhttps://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/the-euclidean-algorithm diakses 7 Desember 2016
6. Artikel mengenai kekongruenan http://mathworld.wolfram.com/Congruence.html diakses 7 Desember 2016.
7. Artikel mengenai chinese remainder theorem https://en.wikipedia.org/wiki/Chinese_remainder_theorem diakses 7 Desember 2016.
8. Artikel mengenai pembuktian teorema fermat https://primes.utm.edu/notes/proofs/FermatsLittleTheorem.html diakses 7 Desember 2016.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 7 Desember 2016

Rizki Ihza Parama - 13515104