# Prediksi Waktu Dekripsi Chipper Text Algoritma Rsa Tanpa Kunci Private Menggunakan Pendekatan Statistik

Raden Bagus Dimas Putra

***Abstract*— RSA was an asymmetric cryptosystem that is widely used today. Its strength depended on the presumed difficulty of the factorization problem. Currently there were some bits size of RSA that very unsafe to use. This study would predict a factorization time on certain bits of RSA. The processed data was a process time for factoring RSA using multiple polynomials quadratic sieve(MP-QS). The number of bits RSA that factored were 64, 96, 128, 160, 192, 224, and 256. Experiments were conducted 10 times for each bit and the results were averaged. The average result for each bit became a training data for prediction algorithm. The prediction algorithm were Double Exponential Smoothing and GROWTH function in Ms. Excel. The conclusion of this study were : don't use 256 bits RSA for securing communication, 512 bits RSA still save for securing private communication, and use 1024 bit or 2048 bit RSA for secret, confidential, sensitive, or classified communication.**

**Intisari— RSA adalah metode kriptografi asimetris yang banyak digunakan saat ini. Kekuatannya tergantung pada kesulitan dari masalah faktorisasi. Saat ini ada beberapa bit ukuran RSA yang sangat tidak aman untuk digunakan. Penelitian ini akan memprediksi waktu faktorisasi pada bit RSA tertentu. Data yang diolah adalah waktu proses memfaktorisasi RSA menggunakan multiple polynomials quadratic sieve(MP-QS). Jumlah bit RSA yang diperhitungkan adalah 64, 96, 128, 160, 192, 224, dan 256. Percobaan dilakukan 10 kali untuk setiap bit dan hasilnya dirata-ratakan. Hasil rata-rata untuk setiap bit menjadi data training untuk algoritma prediksi. Algoritma prediksi yang digunakan adalah Double Exponential Smoothing dan fungsi GROWTH di Ms. Excel. Kesimpulan dari penelitian ini adalah: faktorisasi RSA 256 bit dengan tools hanya perlu 205 detik yang berarti jangan gunakan RSA 256 bit untuk mengamankan komunikasi, Prediksi faktoorisasi RSA 512 bit masih cukup aman untuk mengamankan komunikasi pribadi tetapi masih dapat didekripsi dengan sumber daya yang sangat besar, dan gunakan 1.024 bit atau 2048 bit RSA untuk komunikasi rahasia.**

***Kata Kunci— Double exponential smoothing*, Faktorisasi, MP-QS RSA,**

## I. PENDAHULUAN

Pada tulisan ini, penulis akan membahas teknologi dan keamanan komputasi awan yang diselenggarakan oleh salah satu perusahaan besar di Indonesia yaitu Telkomcloud dan membandingkannya dengan jika kita menggunakan server sendiri. Dari tulisan ini akan terlihat prospek komputasi awan dalam perkembangan e-commerce di Indonesia.

Penelitian mengenai RSA, MP-QS, dan double exponential smooting telah banyak dilakukan oleh para peneliti sebelumnya, adapun penelitian terkait penelitian ini diantaranya adalah :

A. *Hardware Acceleration*

Penelitian [4] berisi tentang memparalelkan algoritme quadratic sieve(QS). Hasil dari penelitian tersebut menunjukan bahwa memparalelkan QS memiliki efisiensi 92% yang artinya penambahan prossesor sebanyak 100 akan meningkatkan kecepatan 92 kali lipat. Berdasarkan hasil tersebut kita dapat mengetahui kebutuhan resource yang digunakan untuk memfaktorisasi suatu bit RSA berdasarkan hasil penelitian pada tulisan ini.

B. Implementasi RSA pada Handphone

Penelitian [5] berisi tentang mengimplementasikan RSA pada gadget berkemampuan rendah yang berakibat harus memilih bit yang rendah untuk pengimplementasiannya. Pada saat ini bit rendah tidak bisa digunakan kembali sebagai proses pengamanan yang akan dibuktikan pada tulisan ini.

C. Implenetasi Double Exponential Smoothing untuk Meramalkan Penjualan

Isi dari [2] yaitu membandingkan metode double exponential smoothing satu parameter dan dua parameter. Hasilnya adalah double exponential smoothing dengan satu parameter lebih baik dalam peramalan untuk kasus data PT. NEW RED & WHITE. Penelitian pada tulisan ini menggunakan double exponential smoothing untuk memprediksi lama faktorisasi pada bit yang besar.

D. Penjelasan Fungsi Statistik pada Excel

Isi dari penelitian [3] adalah menjelaskan fungsi-fungsi statistic yang ada pada Ms. Excel, cara penggunaan dan contoh, serta kelemahan pada fungsi-fungsi tersebut. Penelitian pada tulisan ini menggunakan fungsi "GROWTH" sebagai pembanding fungsi double exponential smoothing dan mean absolute percentage error (MAPE) untuk menghitung nilai kesalahan algoritma tersebut.

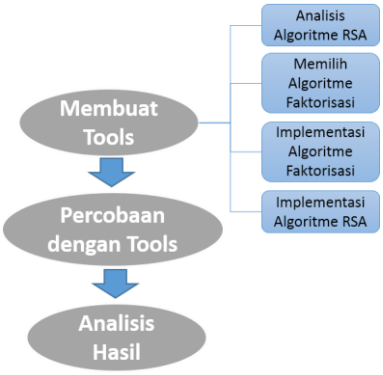
E. Penggunaan Nilai Trend pada Excel

Isi dari tulisan [1] adalah penjelasan tentang mencari nilai trend pada Ms. Excel beserta kelemahannya. Nilai trend digunakan untuk mencari nilai terbaik atau nilai yang memiliki eror rendah pada kasus tertentu. Nilai trend dapat di implementasikan pada nilai alpha dan beta pada double exponential smoothing yang dilakukan pada penelitian ini.

[1]*Program Studi Teknik Informatika, STMIK Nusa Mandiri Jakarta, Jl. Kramat Raya No.18 Jakarta Pusat (Telp.021-3100413; e-mail : raden.rbd@nusamandiri.ac.id)*

## II. METODE PENELITIAN

Penelitian ini akan dikerjakan dalam beberapa tahap. Secara garis besar tahapan tersebut dapat dilihat pada Gbr 1.




*Gbr 1. Langkah metode penelitian*

### A. Membuat Tools

Pada tahapan ini akan dibuat tools yang digunakan untuk proses faktorisasi yang menghasilkan sebuah output berupa lama faktorisasi. Hasil dari tools ini akan dijadikan data untuk diterapkan pada metode ekstrapolasi atau prediksi. Adapun tahapan membuat tools adalah:

#### 1) Analisis *Algoritme RSA*

Tahap ini bertujuan untuk mengetahui struktur algoritma RSA seperti bagaimana cara enkripsi dan cara dekripsi, serta menganalisis bagai mana cara mendapat kunci private(d) hanya berdasarkan pasangan kunci public(N,e). Secara umum kunci private(d) bisa didapatkan jika kita bisa memfaktorisasi (N) menjadi (p) dan (q) dimana p*q=N. Setelah mendapat (p) dan (q) kita dapat mencari (M) yaitu M=(p-1)*(q-1). Jika (M) telah didapatkan kita bisa menghitung (d) dengan cara $d=e^{-1}$ mod M. Yang menjadi pokok permasalahan disini yaitu bagaimana cara memfaktorisasi (N) menjadi (p) dan (q). Pokok permasalahan ini juga yang menjadi kekuatan dari algoritma kunci publik RSA

#### 2) *Memilih Algoritme Faktorisasi*

Berdasarkan [4], Saat ini waktu tunggu yang mungkin untuk mendekripsi RSA dengal single computer yaitu 300bit RSA. Untuk ukuran tersebut, algoritme yang tercepat dapat memfaktorisasi (N) menjadi (p) dan (q) adalah *Multiple Polinomials Quadratic Sieve* atau yang biasa disingkat MP-QS.

Pada penelitian ini proses memfaktorisasi (N) menjadi (p) dan (q) akan menggunakan algoritme MP-QS karene merupakan algoritme yang tercapat untuk ukuran bit tersebut. Waktu hasil percobaan pun dapat di ekstrapolasi untuk memprediksi berapa lama waktu yang dibutuhkan untuk memfaktorisasi bit yang lebih besar.

#### 3) Implementasi *Algoritme Faktorisasi*

Algoritme MP-QS diimplementasi-kan dengan menggunakan bahasa pemrograman C dengan *library* GMP yang memiliki banyak fungsi matematis yang sangat membantu sekali dalam implementasi. Hasil implementasi ini akan dijalankan pada program yang dibuat dengan Bahasa pemrograman java.

### B. Implementasi *Algoritme RSA*

Algoritme RSA diimplementasikan dengan menggunakan bahasa pemrograman java. Form akan dibuat sebanyak lima buah yang diantaranya adalah :

1) Form menu utama
   Form yang merupakan tampilan awal sistem yang merupakan penghubung dengan form lainnya.
2) Form Generate Kunci
   Form yang berfungsi membangkitkan nilai-nilai random yang dibutuhkan untuk proses enkripsi dan dekripsi sesuai dengan jumlah bit yang dimasukan.
3) Form Enkripsi
   Form ini bertujuan untuk merubah pesan plain text menjadi chipper text sesuai dengan nilai pasangan kunci publik(N,e).
4) Form Dekripsi
   Form ini bertujuan untuk merubah pesan chipper text menjadi plain text sesuai dengan nilai kunci private(d). Hal ini bertujuan untuk pengecekan error apakah kunci private(d) yang dibentuk dari hasil faktorisasi telah sesuai atau tidak.
5) Form Dekripsi Tanpa Kunci
   Form ini bertujuan untuk merubah pesan chipper text menjadi plain text hanya dengan menggunakan nilai pasangan kunci public (N,e). Faktorisasi (N) akan menggunakan hasil dari implementasi algoritme MP-QS. Output program diambil menggunakan thread. Hasil tersebut diolah untuk mendapatkan nilai kunci private(d) dan chipper text di dekripsi dengan kunci private(d) yang didapatkan. Selama proses berlangsung waktu dihitung dan dijadikan output bersamaan dengan hasil plain text.

### C. Percobaan dengan Tools

Dalam percobaan dengan menggunakan tools, langkah pertaman yang harus dilakukan adalah membangkitkan panjang bit RSA yang digunakan untuk mengunci pesan yaitu: 64, 96, 128, 160, 192, 224, dan 256. Masing-masing kunci RSA(N,e) akan di bangkitkan secara random sebanyak 10 kali.

Langkah selanjutnya yaitu mengenkripsi pesan menggunakan setiap kunci yang di bangkitkan. Adapun pesan yang dienkripsi adalah “BSI OKE”. Hasil enkripsi pesan akan menghasilkan chipper text yang berbeda pada setiap kunci yang dibangkitkan.

*Chipper text* hasil enkripsi kemudian didekripsi menggunakan tools hanya dengan menggunakan kunci publik(N,e). Hasil setiap dekripsi disimpan dalam file Ms. Excel. Lama proses dekripsi untuk ukuran bit yang sama dirata-ratakan.

D. Analisis Hasil

Pada tahap analisis, Hasil waktu rata-rata dekripsi akan diolah menggunakan metode double exponential smoothing yang akan dilakukan pada Ms. Excel dan sebagai pembanding digunakan metode forecast dengan fungsi "GROWTH" pada Ms. Excell. Hasil dari dari kedua algoritma tersebut diukur tingkat error berdasarkan nilai mean absolute percentage error(MAPE) yaitu tingkat kesalahan pada nilai prediksi. Perhitungan MAPE akan menggunakan fungsi yang ada pada Ms. Excel.

Dari hasil menghitung nilai MAPE, yang memiliki MAPE paling kecil digunakan untuk memprediksi nilai bit yang lebih besar. Berdasarkan penelitian (Putra, 2012) memparalel data pada algoritma QS dan turunannya memiliki effisiensi 92% yang artinya dengan kenaikan resource sebanyak 100 maka kecepatan akan bertambah 92 kali lipat. Dari hasil tersebut dihitung berapa banyak prossesor yang dibutuhkan untuk memfaktorisasi bit tersebut agar waktunya terjangkau.

## III. HASIL PEMBAHASAN

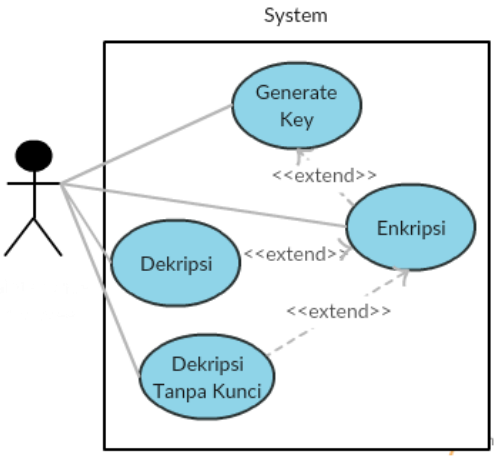
Sekilas tentang faktorisasi N, sebelum patten dari RSA labs dicabut, faktorisasi N merupakan penelitian populer dikarenakan ketika kita bisa memfaktorisasi data yang lebih tinggi, kita akan mendapatkan hadiah dari RSA labs. Sehingga memfaktorisasi RSA merupakan kegiatan ilmiah, bukan merupakan suatu kejahatan. Hasil dan pembahasan berdasarkan setiap langkah pada metode penelitian adalah:

A. Tools yang Dibuat

Algoritme RSA diimplementasikan dengan menggunakan IDE Netbean 8.1 serta JDK dan JRE versi 1.8.0_66. Project java ini diberi nama "Thesis" yang terdiri dari 9 kelas yang berupa 1 kelas main, 5 kelas beserta form, 1 kelas thread, dan 2 kelas biasa. Gbr project ini dapat dilihat pada Gbr 3 dan use case diagramnya pada Gbr 2 Adapun kelas-kelas tersebut adalah

1) Kelas *"Thesis"*
   Kelas ini adalah main project dari aplikasi yang dibuat. Kelas ini hanya digunakan untuk memanggil main form yang bernama kelas "Simulasi".
2) Kelas *"Simulasi"*
   Kelas ini adalah form utama yang menghubungkan user dengan form lainnya. Adapun tampilannya dapat dilihat pada Gbr 4.
3) Kelas *"RSA"*
   Kelas yang merupakan algoritme RSA itu sendiri. Memiliki fungsi untuk enkripsi, dekripsi, membangun objek RSA, serta fungsi umum lainnya seperti setter dan getter.
4) Kelas *"ProcessResultReader"*
   Kelas ini merupakan kelas thread yang berfungsi untuk membangkitkan thread lain yang nantinya dapat digunakan untuk membuka program lain seperti program faktorisasi yang dibuat menggunakan bahasa pemrograman lain, memberi inputan pada program lain, serta mengambil output dari program lain.
5) Kelas *"GetKey"*
   Kelas Ini adalah kelas yang memiliki objek dari kelas thread "ProcessResultReader". Dari kelas ini objek diperintahkan untuk berkomunikasi dengan program faktorisasi untuk memfaktorisasi nilai N yang didapat dari form "Bobol", lalu mengambil outputnya yang kemudian disimpan dalam variabel p dan q yang nantinya dapat diambil melalui fungsi getter.
6) Kelas *"GenerateKey"*
   Kelas ini dapat diakses ketika user mengklik button "GENERATE KUNCI" pada menu utama. Kelas ini berfungsi mendapatkan input jumlah bit dari user, dari jumlah bit tersebut dibangkitkan nilai-nilai yang dibutuhkan dalam penelitian seperti N, e, d serta nilai-nilai lain yang berguna untuk melakukan pengecekan terhadap error. Nilai tersebut dibangkitkan secara random berdasarkan nilai bit yang diinputkan. Adapun tampilan dari kelas ini dapat dilihat pada Gbr 5.
7) Kelas *"Enkrip"*
   Fungsi kelas ini yaitu mengenkripsi pesan plain text menjadi chipper text berdasrakan N, e, serta pesan yang diinputkan oleh user. Kelas ini dapat diakses ketika user mengklik button "ENKRIPSI" pada menu utama. Adapun tampilannya dapat dilihat pada Gbr 6.
8) Kelas *"Dekrip"*
   Kelas ini dibentuk hanya untuk mengecek apakah kelas-kelas lain telah berjalan dengan semestinya. Kelas ini dapat mendekripsi pesan chipper text menjadi plain text dengan inputan p, q, d, dan chipper text. Pada proses sebenarnya, dekripsi hanya membutuhkan d dan N karena fungsi dekripsi yaitu $m = c^d \pmod N$, dimana m = plain text, dan c = chipper text. Menggunakan p dan q karena builder RSA dengan 2 input sudah dibuat untuk keperluan enkripsi sehingga dekripsi menggunakan 3 input dan nantinya N bisa dibentuk padi perkalian p dan q. Tampilan form dekripsi dapat dilihat pada Gbr 7 Kelas ini dapat diakses ketika user mengklik button "DEKRIPSI" pada menu utama.
9) Kelas *"Bobol"*
   Kelas ini adalah kelas utama yang digunakan dalam penelitian ini. Kelas ini dapat diakses ketika user mengklik button "DEKRIPSI TANPA KUNCI PRIVATE" pada menu utama. Kelas ini menerima input N, e, dan chipper text. Nilai N dikirim pada objek dari kelas GetKey yang mempunyai objek thread yang berhubungan dengan program faktorisasi. Nilai N akan difaktorisasi oleh objek thread dan menghasilkan nilai p dan q. Nilai p dan q pada objek dari kelas GetKey diambil menggunakan fungsi getter. Setelah kelas ini memiliki nilai p, q, dan e, nilai d bisa dibentuk. Seperti pada kelas Dekrip, objek dari kelas RSA bisa dibentuk dengan input p, q, dan d. Dari objek tersebut, pesan chipper text dapat di dekripsi menjadi plain text. Waktu dari awal proses hingga akhir proses dihitung. Output dari Kelas ini berupa pesan plain text dan lama proses.

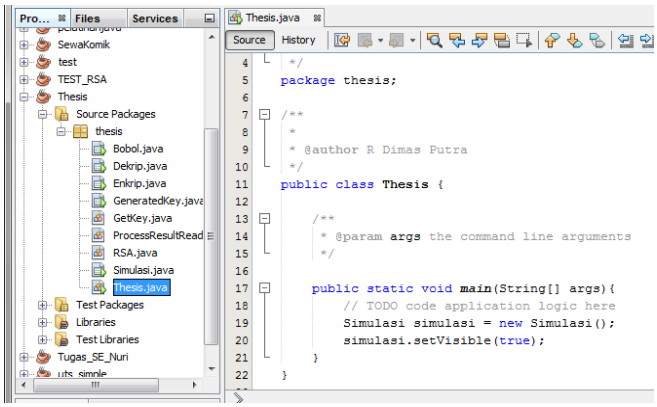
Lama proses akan menjadi bahan analisis selanjutnya. Tampilan form "DEKRIPSI TANPA KUNCI" dapat dilihat pada Gbr 8.




Gbr 2. Use Case Diagram sistem

Dari Gbr 2 terlihat ada empat buah use case. Use case utama pada penelitian ini adalah Dekripsi tanpa kunci yang pseudoconya adalah:

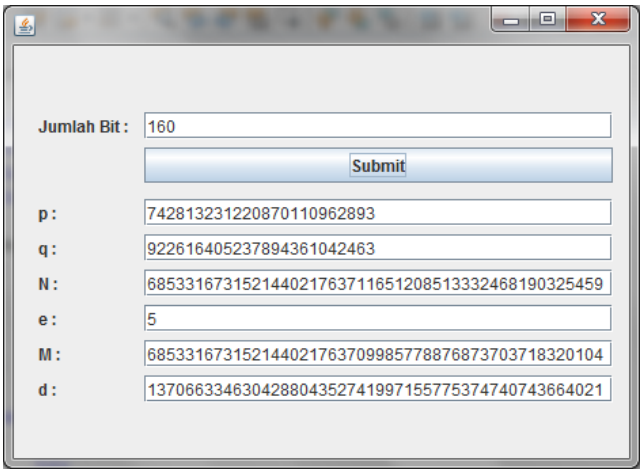
1. Buka form "GENERATE KUNCI"
2. Masukan ukuran kunci yang akan di generate dan klik "Submit".
3. Buka form "ENKRIPSI"
4. Salin nilai N dan e hasil membangkitkan kunci dari form "GENERATE KUNCI" ke form "ENKRIPSI" serta isikan pesan yang akan di enkrip dan klik "Enkrip"
5. Buka form "DEKRIPSI TANPA KUNCI"
6. Salin nilai N, e, dan Chipper text pada form "ENKRIPSI" ke form "DEKRIPSI TANPA KUNCI" lalu klik "Dekripsi Tanpa Kunci"
7. Cek pesan hasil dekripsi dengan pesan sebelum enkripsi, jika sama berarti percobaan berhasil dan catat waktu dekripsi untuk dianalisis




Gbr 3. Hasil implementasi RSA di java

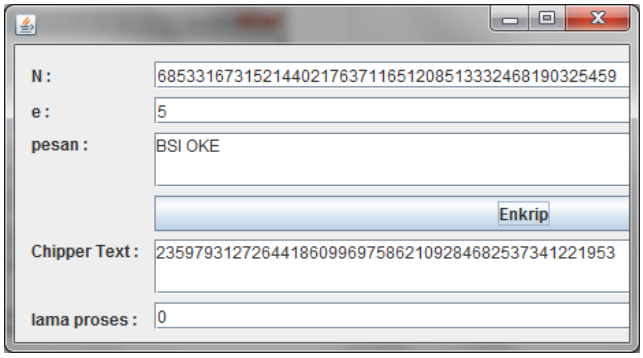

Gbr 4. Tampilan form utama

Dari Gbr 4 terdapat empat button dimana: button "GENERATE KUNCI" berfungsi memanggil form dari kelas "GenerateKey", button "ENKRIPSI" berfungsi memanggil form dari kelas "Enkrip", button "DEKRIPSI" berfungsi memanggil form dari kelas "Dekrip", button "DEKRIPSI TANPA KUNCI" berfungsi memanggil form dari kelas "Bobol".



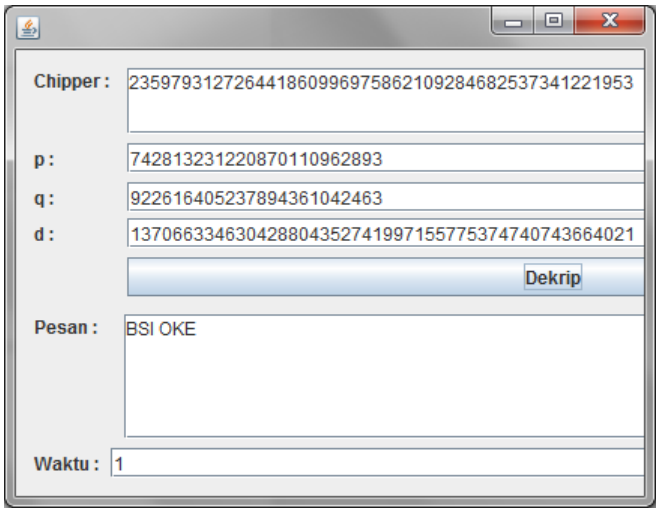

Gbr 5. Tampilan form kelas "GenerateKey"

Pada Gbr 5 button "Submit" akan merubah nilai pada input jumlah bit menjadi nilai-nilai yang dibutuhkan untuk penelitian secara random.



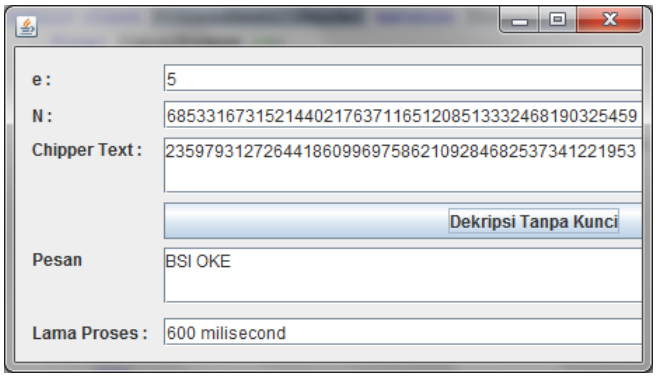

Gbr 6. Tampilan form enkripsi

Pada Gbr 6, fungsi button "Enkrip" yaitu mengenkripsi pesan plain text menjadi chipper text berdasrakan N, e, serta pesan yang diinputkan oleh user.




Gbr 7. Tampilan form dekripsi

Pada Gbr 7, fungsi button "Dekrip" yaitu mendekripsi pesan chipper text menjadi plain text berdasrakan p, q, d, serta Chipper text yang diinputkan oleh user.




Gbr 8. Tampilan form dekripsi tanpa kunci private

Pada Gbr 8, fungsi button "Dekripsi Tanpa Kunci" yaitu mendekripsi pesan chipper text menjadi plain text berdasrakan N, e, serta Chipper text yang diinputkan oleh user.

## B. Hasil Percobaan

Percobaan dilakukan sebanyak 70 kali yaitu untuk masing-masing bit 64, 96, 128, 160, 192, 224, dan 256 dilakukan sebanyak 10 kali. Adapun rata-rata hasil hasil percobaan dapat dilihat pada Tabel 1.

TABEL 1
WAKTU RATA-RATA HASIL PERCOBAAN

| Jumlah Bit | Waktu Rata-Rata (detik) |
|---|---|
| 64 | 0.064 |
| 96 | 0.113 |
| 128 | 0.235 |
| 160 | 0.581 |
| 192 | 3.341 |
| 224 | 31.647 |
| 256 | 205.238 |

Dari Tabel 1 sangat terlihat bahwa RSA 256 bit tidak bisa digunakan lagi untuk pengamanan dikarenakan sangat mudah sekali di dekripsi hanya 205 detik. Jika di dekripsi dengan 100 prossesor dengan efficiency 92% hanya membutuhkan waktu 2.2 detik. Peneliti yang tidak mengetahui hal ini, masih banyak yang membuat penelitian menggunakan RSA bit kecil untuk gadget yang memiliki spesifikasi rendah seperti handphone dan PDA sebagai alat pengaman. Untuk masalah keamanan RSA bit kecil dibawah 300 bit sebaiknya tidak digunakan lagi, namun masih bisa digunakan untuk hal lain seperti watermarking dan tanda tangan digital.

## C. Analisis Hasil Percobaan

Dari hasil waktu rata-rata dibuat persamaan double exponential smoothing dengan alpha = 0.1 dan beta = 0.1. Metode ini terdapat dalam formula Ms. Excel jika menginstall plugin "NumXL" dengan nama formula "=DESMTH" atau dapat dibuat secara manual mengikuti langkah pada kajian pustaka. Hasil dari double exponential smoothing dengan alpha=0.1 dan beta=0.1 dapat dilihat pada Tabel 2.

TABEL 2
HASIL DOUBLE EXPONENTIAL SMOOTHING

| Jumlah Bit | Waktu Rata-Rata (detik) | Double Exponential Smoothing |
|---|---|---|
| 64 | 0.064 | 0.064 |
| 96 | 0.113 | 0.06939 |
| 128 | 0.235 | 0.0880971 |
| 160 | 0.581 | 0.144462519 |
| 192 | 3.341 | 0.503156771 |
| 224 | 31.647 | 3.96802003 |
| 256 | 205.238 | 26.45819676 |

Pada dasarnya nilai DESMTH pada bit 64 tidak melalui proses perhitungan dikarenakan fungsi prediksi memerlukan data sebelumnya. Setelah itu hitung mean absolute percentage error (MAPE) dengan formula pada Ms. Excel yaitu "=SUMSQ(TSSUB([data_prediksi],[data_hasil]))" sehingga didapatkan nilai 32736.611. Error yang dihasilkan terlalu besar. Buat tabel analisis korespondensi nilai alpha dan beta terhadap nilai error. Alpha menjadi baris dan beta menjadi kolom Rancangan tabel korespondensi bisa dilihat pada Tabel 3.

Semakin kecil intervalnya hasilnya akan semakin baik. Pada penelitian ini akan akan digunakan interval = 0.5. Selanjutnya blok seluruh tabel korespondensi lalu pilih menu "What-if Analysis" pada panel data dan pilih "Data Table". Kemudiam masukan data beta dan alpha. Hasilnya akan seperti Tabel 4.

TABEL 3
RANCANGAN TABEL KORESPONDENSI

| MAPE | 0.05 | 0.1 | … | 1 |
|---|---|---|---|---|
| 0.05 | | | | |
| 0.1 | | | | |
| … | | | | |
| 1 | | | | |

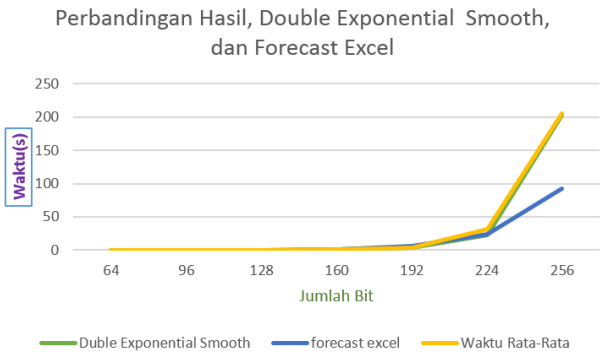
TABEL 4
HASIL TABEL KORESPONDENSI

| 32736.611 | 0.05 | 0.1 | … | 1 |
|---|---|---|---|---|
| 0.05 | 37966.13 | 37694.55 | … | 33014.4 |
| 0.1 | 33234.8 | 32736.61 | … | 24549.91 |
| … | … | … | … | … |
| 1 | 105.925 | 416.515 | … | 30942.81 |

Untuk mempermudah pencarian error terkecil, cari min pada setiap kolom dan baris. Gunakan nilai alpha dan beta yang paling optimal. Untuk memprediksi, nilai optimal pada alpha = 0.4 dan beta = 0.05. Nilai tersebut memiliki error 86.551, jauh lebih kecil dibandingkan error sebelumnya. Masukan nilai tersebut pada persamaan DESMTH, lalu buat juga pembanding dengan fungsi forecast Ms.Excel "=GROWTH([y_train],[x_train],x_dicari,1)". Hasil dari persamaan DESMTH yang baru dan forecast dapat dilihat pada Tabel 5 dan grafiknya pada Gbr 9.

TABEL 5
HASIL DESMTH BARU VS FORECAST

| Jumlah Bit | Waktu Rata-Rata (detik) | Double Exponential Smoothing | Forecast Ms. Excel |
|---|---|---|---|
| 64 | 0.064 | 0.064 | 0.026 |
| 96 | 0.113 | 0.113 | 0.102 |
| 128 | 0.235 | 0.271 | 0.397 |
| 160 | 0.581 | 1.010 | 1.549 |
| 192 | 3.341 | 3.601 | 6.050 |
| 224 | 31.647 | 22.827 | 23.626 |
| 256 | 205.238 | 202.322 | 92.261 |




Gbr 9. Grafik hasil DESMTH baru vs forecast

Berdasarkan Tabel 5 dan Gbr 9, hasil dari DESMTH jauh mendekati nyata dibandingkan dengan forecast. Error dari forecast pun jauh lebih besar yaitu 12836.407. Karena error DESMTH lebih kecil dibandingkan forecast, prediksi waktu eksekusi bit yang lebih besar dilakukan menggunakan DESMTH. Hasil prediksi dapat dilihat pada Tabel 6.

TABEL 6. WAKTU PREDIKSI DALAM (JAM)

| Bit | 320 | 384 | 448 | 512 |
|---|---|---|---|---|
| Prediksi | 3.3 | 272.4 | 29297.6 | 3985308.5 |

Dari hasil prediksi dapat terlihat bahwa untuk memfaktorisasi 512 bit membutuhkan waktu 3985308.5 jam atau sekitar 455 tahun. Jika dikerjakan menggunakan 10 juta prossesor dengan efisiensi 92% berdasarkan penelitian putra (2012) waktunya menjadi 26 menit. Memiliki jutaan botnet memang bukan perkara yang mudah, namun juga bukan perkara yang tidak mungkin. Seorang *"Black Hat"* professional mungkin saja memilikinya. Workstation penelitian pun sudah mencapai ratusan ribu untuk pengolahan data. Ditambah lagi ada algoritme yang lebih cepat dalam memfaktorisasi RSA berdigit besar seperti algoritme *number field sieve*. Untuk berjaga-jaga, jika data sangat penting gunakanlah RSA 1024 bit. Walaupun akan sangat membebani kinerja, ini akan lebih baik daripada data bocor ke orang yang tidak berhak. RSA 512 memang masih cukup aman diimplementasikan untuk komunikasi data pribadi yang rahasia untuk menghindari orang lain mengetahuinya. Namun untuk data yang sangat rahasia gunakanlah RSA 1024 bit atau lebih besar seperti 2048 bit.

## IV. KESIMPULAN

Kesimpulan dari penelitian ini yaitu: Hasil dekripsi RSA 256 bit menggunakan tools yang dibuat dengan java adalah 205 detik yang artinya RSA yang ukurannya lebih kecil dan sama dengan 256 bit sudah tidak layak lagi untuk proses pengamanan. Hasil prediksi dekripsi RSA 512 dengam menggunakan double exponential smoothing adalah 455 tahun. Dengan kata lain RSA 512 bit masih cukup aman diimplementasikan untuk komunikasi data pribadi untuk menghindari orang lain mengetahuinya karena untuk mendikripsi hal tersebut membutuhkab resource yang sangat besar. Dikarenakan masih bisa di dekripsi walaupun dengan resource yang sangat besar untuk data rahasia gunakan RSA 1024 bit atau lebih besar seperti 2048 bit.

## UCAPAN TERIMA KASIH

Terima kasih saya sampaikan kepada seluruh tim yang terkait dengan jurnal JTI karena dengan adanya jurnal ini para penulis pada umumnya dan saya sendiri pada khususnya sangat terbantu untuk memenuhi kebutuhan BKD semesteran dan meningkatkan mutu kualitas pendidikan. Dengan adanya template yang dibuat juga sangat terbantu untuk menyesuaikan tata tulisan paper yang dibuat. Sekali lagi saya ucapkan terima kasih yang sebanyak-banyaknya.

## REFERENSI

[1] B. R. Hargreaves and T. P. McWilliams, Polynomial Trendline function flaws in Microsoft Excel, Computational Statistics & Data Analysis Volume 54, 2010, p. 1190-1196.
[2] Nixon. (2002) Perancangan Program Aplikasi Peramalan Penjualan Dengan Menggunakan Metode Double Exponential Smoothing. [Online]. Available: http://library.binus.ac.id/Collections/ethesis_detail/LKT2006-0066
[3] H. Pottel, Statistical flaws in Excel, Zwijnaarde, Belgium: Innogenetics NV, 2003.
[4] R. B. D. Putra, "Analysis of Quadratic Sieve Algorithm with Parallel Implementation", Skripsi. Institut Pertanian Bogor, Bogor, Indonesia, 2012.
[5] E. Syahputra, "Pengembangan Aplikasi Pertukaran SMS Rahasia Berbasis Android Menggunakan Algoritme RSA", Skripsi. Institut Pertanian Bogor, Bogor, Indonesia, 2014.

Raden Bagus Dimas Putra, S.Komp, M.Kom. Jakarta 4 Agustus 1990. Seorang Dosen di STMIK Nusa Mandir Jakarta yeng merupakan lulusan Sarjana Komputer dari Ilmu Komputer IPB dan Magister Komputer dari Pasca Sarjana STMIK Nusa Mandiri Jakarta.