**Laporan** **Praktikum**  
**P. SISTEM KEAMANAN DATA**

**[KELOMPOK 2]**

* Yuanda Wardana Eka Saputra (V3920062)
* Rifqi Achmad Fadhila (V3920053)
* Ilham Hidayat Sidik (V3920065)
* Tanzil Rahmatul Karim (V3920057)
* Zulfi Masyita Resia Reswati (V3920063)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
**Topik praktikum:**  
Modul 13 - Resume Jurnal RSA  
  
**Dosen** **Pengampu**: Yusuf Fadlila Rachman, S.Kom., M.Kom  
  
**Program Studi D3 Teknik Informatika,   
Sekolah Vokasi, Universitas Sebelas Maret  
2021**

# **RESUME JURNAL I**

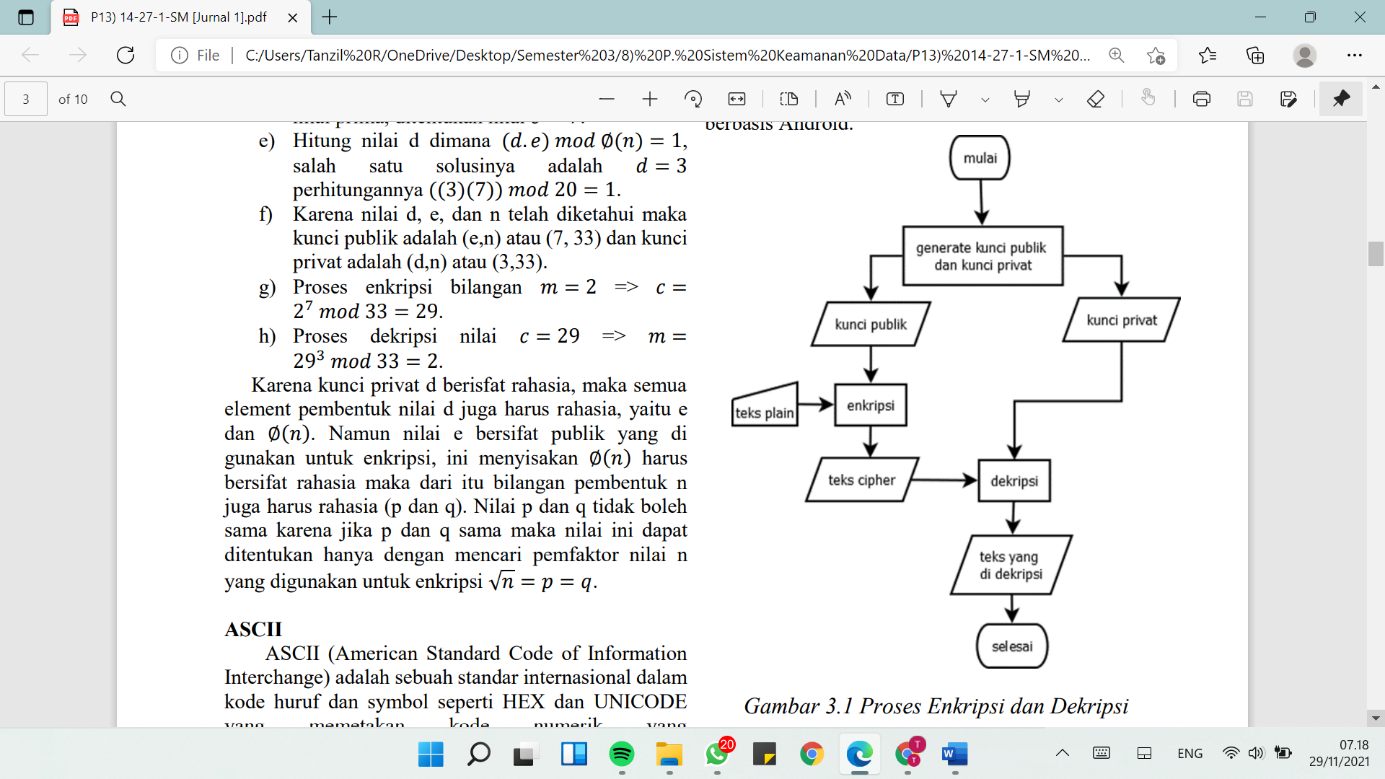
## **Judul & Latar Belakang**

|  |  |
| --- | --- |
| Judul : | PENERAPAN ALGORITMA RSA UNTUK KEAMANAN PESAN INSTAN PADA PERANGKAT ANDROID |
| Latar Belakang : | Pesan instan sering digunakan karena kecepatan pengiriman data dan murahnya biaya yang digunakan saat mengirim pesan, pesan instan ini menggunakan jaringan internet dan menjaga kerahasiaan pesan dianggap penting. Salah satu cara untuk menjaga kerahasiaan data adalah menggunakan kriptografi. Algoritma kriptografi yang banyak digunakan adalah RSA. Alasan peneliti dalam pemilihan RSA untuk permasalahan diatas adalah algoritma RSA memiliki keunggulan sulitnya dalam memfaktorkan bilangan besar menjadi faktor-faktor prima. |

## **Tujuan**

Pada jurnal pertama ini, penelitian dilakukan untuk melindungi data yang tersimpan pada cloud menggunakan keamanan algoritma. Algoritma RSA di implementasikan pada pada Google App menggunakan Cloud SQL. Pada penelitian ini algortima RSA diuji terhadap beberapa serangan. Serangan tersebut antara lain serangan pada chiper text saja, serangan faktorisasi, dan serangan brute force.

## **Tipe Algoritma & Alur Penelitian**

Tipe algoritma yang digunakan adalah algoritma RSA. Untuk alur penelitiannya yaitu sebagai berikut.

1. **Generate kunci publik dan kunci privat**

Berikut adalah proses dimana kunci publik dan kunci privat diciptakan.

* 1. Tentukan dua nilai p dan q, dimana p dan q adalah bilangan prima. Nilai p dan q tidak boleh sama karena jika p dan q sama maka nilai ini dapat ditentukan hanya dengan mencari pemfaktor nilai n yang digunakan untuk enkripsi √𝑛 = 𝑝 = 𝑞.

Contoh: 𝑝 = 3 𝑑𝑎𝑛 𝑞 = 11.

* 1. Hitung nilai n dari hasil p \* q.

𝑛 = 𝑝 \* 𝑞 = (3)(11) = 33. Maka nilai n = 33.

* 1. Hitung nilai

∅(𝑛) = (𝑝 − 1)(𝑞 − 1)

= (3 − 1)(11 − 1) = (2)(10) = 20.

* 1. Pilih nilai e dimana 1 < 𝑒 < ∅(𝑛) dan e adalah nilai prima, ditentukan nilai 𝑒 = 7.
  2. Hitung nilai d dimana (𝑑. 𝑒) 𝑚𝑜𝑑 ∅(𝑛) = 1, salah satu solusinya adalah 𝑑 = 3 perhitungannya ((3)(7)) 𝑚𝑜𝑑 20 = 1.
  3. Karena nilai d, e, dan n telah diketahui maka kunci publik adalah (e,n) atau (7, 33) dan kunci privat adalah (d,n) atau (3,33).
  4. Proses enkripsi bilangan 𝑚 = 2 => 𝑐 = 2 7 𝑚𝑜𝑑 33 = 29.
  5. Proses dekripsi nilai 𝑐 = 29 => 𝑚 = 293 𝑚𝑜𝑑 33 = 2.

Kunci publik dan kunci privat di buat dengan menggunakan pustaka pada bahasa pemograman Java. Pustaka yang di gunakan adalah:

* + - * java.security.KeyPairGenerator,
      * java.security.KeyPair,
      * java.security.PublicKey, dan
      * java.security.PrivateKey.

1. **Pengumpulan dan Pengujian Data**

Pengumpulan data teks yang akan di lakukan sebagai salah satu variabel dalam proses pengujian enkripsi dan dekripsi. Contoh: teks dengan ukuran sedang dan panjang teksnya (PT) adalah 25.

1. **Proses Data Pengujian Algoritma RSA**

Dalam pengujian setiap proses dicatat pada log aplikasi, berdasarkan log ini maka diketahui variabel yang akan diuji. Variabel tersebut adalah sebagai berikut.

* + - 1. Plain teks: yaitu teks yang akan di enkripsi, teks ini di dapat berdasarkan maukkan data dari user.
      2. Panjang teks (PT): pajang teks adalah banyaknya karakter pada teks.
      3. Waktu enkripsi (WE): waktu enkripsi di dapatkan dengan mencatat log sebelum dan setelah proses enkripsi.
      4. Teks cipher: adalah teks yang di hasilkan setelah proses enkripsi.
      5. Panjang dari teks cipher (PC).
      6. Waktu dekripsi (WD): adalah waktu yang di butuhkan untuk proses dekripsi. Seperti waktu enkripsi, waktu dekripsi didapatkan dari selisih waktu pada saat mulai proses dekripsi sampai selesai proses dekripsi.
      7. Hasil dekripsi adalah hasil dari teks cipher yang di dekripsi.

Contoh:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Plaintext | PT | WE | Teks Cipher | PC | WD | Hasil Dekripsi |
| teks dengan ukuran sedang | 25 | 25 ms | 56071A ECE918 E2316E AE6CA 177067 CB0577 010DE9 D74748 91431F 546FEA F4DD60 7E2712 234F62 DB7025 79E61D 886BA6 9DCC5 D48C4 DCF494 8EEBD BBEE3 B47D5 B4D13 DDBA4 77317A 77E5F7 7721267 23196D A2D6F5 E4FF03 40F9DC 60C132 3C362E 3C1918 519B92 08D947 B81D1 CB2920 2A29F5 | 256 | 31 ms | teks dengan ukuran sedang |

## **Hasil Penelitian & Kesimpulan**

Perbandingan panjang teks, waktu enkripsi, dan waktu dekripsi:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variabel | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| PT | 25 | 25 | 67 | 33 | 54 | 127 | 38 |
| WE | 21 | 25 | 7 | 9 | 10 | 7 | 9 |
| WD | 25 | 31 | 24 | 21 | 24 | 29 | 35 |

Nilai rata-rata dihitung hanya pada sampel yang berhasil dilakukan enkripsi dan dekripsi, maka **berikut hasil penelitiannya**.

Diketahui N = 8

Rata-rata panjang teks =

Rata-rata waktu enkripsi =

Rata-rata waktu dekripsi =

Berdasarkan pengujian dan pengolahan data pada jurnal maka **dapat** **disimpulkan**:

* Algoritma RSA adalah algoritma asimetris yaitu algoritma yang mempunyai dua kunci berbeda untuk proses enkripsi dan dekripsi yaitu kunci publik dan kunci privat.
* Kunci publik di gunakan untuk mendekripsi teks biasa menjadi teks yang terenkripsi atau disebut teks cipher.
* Kunci privat digunakan untuk mengembalikan teks cipher menjadi teks biasa.
* Penggunaan algoritma RSA untuk aplikasi pesan instan adalah untuk mengenkripsi pesan sebelum dikirim kepada penerima.
* Untuk kebutuhan enkripsi, pengirim pesan meminta kunci publik dari penerima yang di distribusikan secara bebas.
* Pesan yang diterima kemudian di dekripsi menggunkan kunci privat yang hanya dimiliki oleh penerima pesan.

## **Kelebihan & Kekurangan**

**Kelebihan dari RES**

* Kelebihannya terletak pada sulitnya memecahkan kunci dan penggunaan kunci yang lebih efektif.
* Pada jurnal pembahasannya lengkap yang disertai alur penggunaan algoritma RSA pada permasalahan yang diangkat.

**Kelemahan dari RES**

* Kelemahan pada jurnal terdapat tabel yang tidak rapi sehingga data tidak terlihat dengan jelas dan sulit untuk dipahami.
* Terdapat kata dan angka yang typo atau salah dalam penulisan dan perhitungan, juga terdapat icon yang tidak bisa muncul.
* Kelemahan dari algoritma kriptografi RSA adalah waktu yang dibutuhkan untuk melakukan proses pembangkitan kunci, enkripsi dan dekripsi lambat.

# **RESUME JURNAL II**

## **Judul & Latar Belakang**

|  |  |
| --- | --- |
| Judul : | Desain Steganografi untuk Keamanan Gambar dengan Algoritma RSA dan LSB Berbasis Android |
| Latar Belakang : | Dalam penelitian ini dengan menggunakan metode RSA untuk mengeknripsi file plaintext dan juga menggunakan kunci sebelum dikirim ke sipenerima, plaintext tersebut akan diubah kedalam bentuk ASCII kemudian diubah ke dalam bentuk biner agar dapat disisip kedalam gambar dengan menggunakan LSB. Dengan menggunakan teknik kriptografi dengan algoritma RSA menjadi pilihan untuk mengubah pesan menjadi tidak terbaca lagi karena sampai saat ni algoritma RSA dinilai yang masih bagus tingkat keamanan nya. |

## **Tujuan**

Membuat Desain untuk keamanan Gambar menggunakan algoritma RSA dan LSB.

## **Tipe Algoritma & Alur Penelitian**

Tipe algoritma yang digunakan adalah RSA dan LSB. Untuk alur penelitiannya yaitu sebagai berikut.

1. **Bangkitkan kunci dari dua bilangan prima**

Proses enkripsi RSA dilakukan dengan membangkitkan kunci dari dua buah bilangan prima untuk mendapatkan nilai n, Θn, e, dan d. Sebagai contoh digunakan dua buah bilangan prima p = 31 dan q = 37. Rumus pembangkit kunci algoritma RSA adalah sebagai berikut:

* 1. Mencari nilai n, yaitu 𝑛 = 𝑝 x 𝑞.
  2. Mencari nilai Θn, yaitu 𝑛 = (𝑝 − 1)𝑥 (𝑞 − 1).
  3. Mencari nilai e, yaitu

e = 2

*While Θ*n mod e ≠ 0

e = e + 1

*End While*

* 1. Mencari nilai d, yaitu

U1 = 1

U2 = 0

U3 = Θn

V1 = 0

V2 = 1

V3 = e

*While* V3 = 0

Q = Int(U3/V3)

N1 = U1 - (Q x V1)

N2 = U2 - (Q x V2)

N3 = U3 - (Q x V3)

U1 = V1

U2 = V2

U3 = V3

V1 = N1

V2 = N2

V3 = N3

*End While*

Kemudian untuk melakukan proses enkripsi diperlukan algoritma enkripsi untuk mengubah plainteks menjadi cipherteks. Berikut adalah algoritma enkripsi RSA.

𝐶𝑖 = 𝑃 𝑒 𝑖 𝑚𝑜𝑑 𝑛

𝐶𝑖 = Cipherteks hasil enkripsi

𝑃𝑖 = Plainteks atau pesan asli

e = Kunci enkripsi

n = Nilai n

Setelah proses enkripsi berhasil tentu harus ada cara agar cipherteks bisa dikembalikan menjadi plainteks, yaitu dengan cara dekripsi. Berikut adalah algoritma dekripsi RSA:

𝑃𝑖 = 𝐶 𝑑 𝑖 𝑚𝑜𝑑 𝑛

𝑃𝑖 = Plainteks atau pesan asli

𝐶𝑖 = Cipherteks hasil enkripsi

d = Kunci dekripsi

n = Nilai n

1. **Menggunakan LSB**

Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai byte satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Sebagai contoh sebuah segmen pixel gambar sebelum disisipkan bit pesan adalah:

11001010 01101110 11100111 10011001

00110111 10110011 10101011 00011111

Akan disisipkan sebuah pesan rahasia yang sudah dikonversikan menjadi bentuk biner misalkan “10110011”, setiap bit dari pesan rahasia tersebut akan menggantikan bit – bit terakhir dari pixel gambar menjadi:

11001011 01101110 11100111 10011001

00110110 10110010 10101011 00011111

## **Hasil Penelitian & Kesimpulan**

Pesan yang digunakan merupakan kumpulan karakter yang terdapat pada tabel ASCII, dimana pesan tersebut kemudian di enkripsi dengan kriptografi algoritma RSA yang kemudian disisipkan kedalam citra gambar dan akan di ekstrak kembali dari citra gambar menjadi cipherteks dengan metode LSB yang selanjutnya akan di dekripsi menjadi plainteks. Citra gambar yang digunakan sebagai cover object adalah citra gambar dengan format file \*.jpg dengan ukuran citra 3888x2592.

Setelah proses enkripsi selesai maka ditambahkan karakter “{” sebagai penanda pada setiap karakter. Maka didapatkan cipherteks dari hasil enkripsi algoritma RSA adalah “1 2 2 1 2 3 1 0 3 5 1 2 3 1 1 7 1 2 3 5 5 2 1 2 3 6 4 1 2 3 1 0 0 3 1 2 3 5 4 8 1 2 3 1 1 7 1 2 3 2 7 7 1 2 3 7 9 1 1 2 3 5 4 8 1 2 3 1 0 5 5 1 2 3 6 6 0 1 2 3 6 2 9 1 2 3 2 7 7 1 2 3 6 4 1 2 3 1 0 3 5 1 2 3 5 4 8 1 2 3 1 1 7 1 2 3 1 0 5 5 1 2 3 1 2 2 1 2 3 2 7 7 1 2 3 7 9 1 1 2 3 1 0 9 3 1 2 3 7 9 1 1 2 3 1 0 1 5”.

Setelah di enkripsi maka sistem akan menyisipkan cipherteks dengan mengubah setiap karakter menjadi biner dan disisipkan kedalam pixel citra gambar menggunakan steganografi LSB. Saat berhasil maka sistem akan memunculkan dialog pesan “Pesan telah di encode”.

Pada layout decode sistem akan melakukan proses ekstraksi dan dekripsi yang mana ekstraksi dilakukan terhadap citra gambar yang sebelumnya sudah disisipi cipherteks hasil enkripsi algoritma RSA. Pertama ambil gambar pada button Ambil Gambar yang kemudian sistem akan membuka galery untuk memilih gambar. Lalu klik Decode dan sistem akan mengekstraksi gambar yang sudah disisipkan gambar.

Dari penelitian ini, dapat ditarik kesimpulan sebagai berikut.

* + - * Perhitungan manual enkripsi dilakukan sebanyak 26 kali perhitungan dan deskripsi juga sama dan hasil perhitungan manual deskripsi dapat mengembalikan pesan yang sudah enkripsi seperti semula dalam bentuk plaintext.
      * Pesan yang akan disisip kedalam sebuah gambar berformat jpg terlebih dahulu di lakukan ke dalam ASCII dan setelah itu diubah kedalam bentuk biner dan disisip ke dalam pixel gambar rgb
      * Aplikasi ini dibangun dengan menggunakan android studio dan diaplikasikan kedalam handphone android.

## **Kelebihan & Kekurangan**

**Kelebihan dari AES**

* RSA: Kekuatan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor primanya, dalam hal ini memfaktorkan n menjadi p dan q. Karena sekali n berhasil difaktorkan, maka menghitung nilai m adalah perkara mudah. Selanjutnya, walau nilai e diumumkan, perhitungan kunci d tidaklah mudah pula karena nilai m yang tidak diketahui.
* Kelebihan lain algoritma RSA terletak pada ketahanannya terhadap berbagai bentuk serangan, terutama serangan brute force. Hal ini dikarenakan kompleksitas dekripsinya yang dapat ditentukan secara dinamis dengan cara menentukan nilai p dan q yang besar pada saat proses pembangitkan pasangan kunci, sehingga dihasilakan sebuah key space yang cukup besar, sehingga tahan terhadap serangan.
* LSB: Algoritma yang sederhana, yaitu algoritma LSB yang langsung mensubstitusi pesan ke cover image. Implementasi yang banyak digunakan pada aplikasi steganografi.Perubahan citra tidak terlalu terlihat oleh mata.

**Kelemahan dari AES**

* RSA: Namun demikian, kelebihan tersebut sekaligus menjadi kelemahan dari sistem ini. Ukuran kunci privat yang terlalu besar akan mengakibatkan proses dekripsi yang cukup lambat, terutama untuk ukuran pesan yang besar. Oleh karena itu, RSA umumnya digunakan untuk meng-enkripsi pesan berukuran kecil seperti kata kunci dari enkripsi simetris seperti DES dan AES yang kemudian kunci tersebut dikirim secara bersamaan dengan pesan utama.
* LSB: Pesan yang disisipkan terbatas. Apabila dibandingkan dengan teknik steganografi yang lainnya.Karena metode yang sangat lama, banyak aplkasi penyerang (aplikasi ekstraksi) yang dapat mengekstrak pesan dari stego image.

# REFERENSI

Zainuddin, M. A., & Mulyana, D. I. (2016). PENERAPAN ALGORITMA RSA UNTUK KEAMANAN PESAN INSTAN PADA PERANGKAT ANDROID. *Teknik Informatika, CKI On SPOT, Vol. 9, No. 2*.