Studi Kompleksitas Algoritma Enkripsi Teks Simetri dan Asimetri

Almeiza Arvin Muzaki - 13519066
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*13519066@std.stei.itb.ac.id*

***Abstract*—Kriptografi secara harfiah merupakan pengarsipan yang perlu dijaga kerahasiaannya. Terdapat berbagai metode kriptografi yang dapat diterapkan pada arsip, salah satunya teks. Secara umum, terdapat 2 jenis algoritma kriptografi, yakni symmetric dan asymmetric. RSA (Rivest Shamir Adleman) yang termasuk algoritma asymmetric memerlukan waktu yang cukup lama untuk mengenkripsi teks.. Perlu sebuah algoritma lain yang lebih cepat dalam eksekusi kriptografi, yakni algoritma simetri. Algoritma simetri disinyalir lebih efektif dibandingkan algoritma asimetri. Salah satu algoritma yang sering digunakan pada kriptografi adalah algoritma Data Encryption Standard (DES). Studi ini memaparkan analisis DES dengan RSA sebagai parameter pembanding.**

***Keywords*—Kriptografi, Enkripsi, Teks, Simetri, Asimetri, RSA, DES.**

I. PENDAHULUAN

Pengetahuan akan teknologi informasi telah mendorong kualitas kehidupan masyarakat secara signifikan. Aktivitas manusia yang semakin padat dan beragam mengakibatkan terjadinya ledakan data yang besar di berbagai aspek kehidupan. Lalu lintas data mengalir dengan sangat cepat di dunia maya. Hal ini juga memengaruhi keberadaan teknologi sebagai fasilitas komunikasi data. Berbagai teknologi data yang sebelumnya tidak pernah terbayangkan oleh manusia sekarang ini benar-benar telah menjadi bagian penting dalam hidup. Secara bertahap, data telah sedemikian rupa menjadi bagian yang penting dalan kehidupan manusia.

Akan tetapi, pertumbuhan data dan teknologinya ini tidak selalu dihinggapi oleh hal-hal yang baik. Penipuan, pemerasan, pencurian data, dan lain sebagainya masih kerap terjadi. Tidak hanya data pribadi, data massif yang jumlahnya jutaan milik perusahaan start-up saja bisa dicuri oleh sembarang pihak yang tidak bertanggung jawab. Sistem keamanan berkomunikasi perlu menjadi perhatian khusus dalam bidang teknologi. Kriptografi adalah salah satu ilmu yang bisa menjawab persoalan ini.

"Kriptografi merupakan studi Teknik matematika yang berkaitan dengan aspek keamanan informasi seperti kerahasiaan, integritas data dan autentikasi (Vanstone, Oorschot & Scott, 1997). "

II. LANDASAN TEORI

2.1. Definisi Kriptografi

Kriptografi, menurut Menezes, Oorschot dan Vanstone (1996), didefiniskan sebagai suatu studi teknik matematika yang berhubungan dengan aspek keamanan informasi seperi kerahasiaan, integritas data, otentikasi entitas dan otentikasi keaslian data.

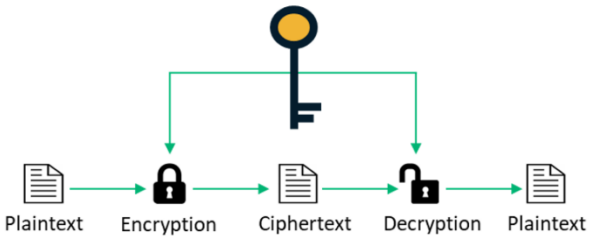
Istilah dalam kriptografi
Beberapa istilah penting dalam kriptografi diantaranya:
1. Kriptologi: Ilmu yang mempelajari kriptografi dan kriptanalsis
2. Kriptanalisis: Ilmu yang mempelajari cara membobol kriptografi
3. Plainteks: Pesan asli yang akan dirahasiakan
4. Chiperteks: Pesan yang telah dirahasiakan menjadi bentuk lain

2.2. Jenis Kriptografi

Kriptografi menurut jenisnya terbagi menjadi dua.

1. Kriptografi simetri

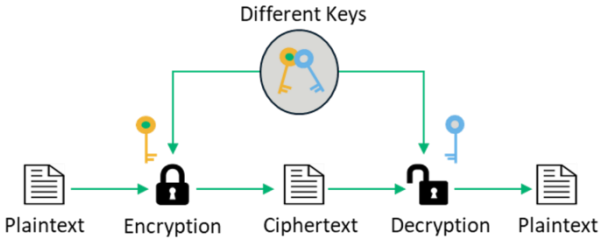
Algoritma simetri menjadi algoritma konvensional kriptografi secara umum. Algoritma ini hanya menggunakan satu buah kunci saja. Dengan kata lain, pengirim maupun penerima informasi hanya menggunakan satu kunci untuk enkripsi sekaligus dekripsi. Beberapa contoh algoritma kunci simetri ini diantaranya DES (Data Encryption Standard), MARS, AES (Advanced Encryption Standard) IDEA dan 3DES.



Gambar 2.2. Algoritma Kriptografi Simetri (sectigostore.com)

2. Kriptografi asimetri

Algoritma asimetri merupakan kebalikan dari simetri yang menggunakan dua buah kunci (publik dan privat). Kunci publik merupakan kunci yang disebarkan kepada umum sebagai kunci enkripsi, sedangkan kunci privat merupakan kunci dekripsi yang hanya dipegang oleh pendekripsi pribadi saja. Beberapa algoritma asimetri tersebut diantaranya RSA (Rivest Shamir Adleman) dan ECC (Elliptic Curve Cryptography).



Gambar 2.2. Algoritma Kriptografi Asimetri (sectigostore.com)

3. Kompleksitas Algoritma

a. Definisi

Kompleksitas algoritma merupakan cabang matematika diskrit yang membahas mengenai perhitungan kebutuhan waktu eksekusi faktual algoritma dalam satuan detik.

Kompleksitas waktu dibagi menjadi tiga, yaitu:

1. Tmax(n) : kompleksitas waktu kasus terburuk (worst case),
2. Tmin(n) : kompleksitas waktu kasus terbaik (best case),
3. Tavg(n): kompleksitas waktu untuk kasus rata-rata (average case)

Dengan nilai Tmax(n)>Tavg(n)> Tmin(n).

b. Kompleksitas Waktu Asimptotik

Data kebutuhan waktu tumbuh algoritma lebih mudah dianalisis ketika ukuran masukannya (n) meningkat. Kinerja algoritma baru akan tampak untuk n yang sangat besar, bukan pada n yang kecil, melainkan n yang besar dan semakin membesar. Oleh karena itu, diperlukan suatu notasi kompleksitas algoritma yang merepresentasikan kinerja n pada waktu yang semkain meningkat. Notasi kompleksitas waktu algoritma untuk n yang besar dinamakan kompleksitas waktu asimptotik.

$$T(n) = O(f(n))$$

artinya “T(n) adalah O(f(n)), yang artinya T(n) berorde paling besar f(n)) bila terdapat konstanta C dan n0 sedemikian sehingga T(n) <= C f(n) untuk n >= n0.

Nilai O(n) sangat berpengaruh terhadap input dan output data yang diproses.

$\log n$	n	$n \log n$	n^2	n^3	2^n	$n!$
0	1	0	1	1	2	1
1	2	2	4	8	4	2
2	4	8	16	64	16	24
3	8	24	64	512	256	362880
4	16	64	256	4096	65536	20922789888000
5	32	160	1024	32768	4294967296	(terlalu besar untuk ditulis)

Gambar 3.1. Perbandingan nilai O(n) terhadap data (informatika.stei.itb.ac.id)

- O(1)
  Kompleksitas O(1) adalah kompleksitas yang stagnan. Artinya, tidak tergantung pada ukuran input. Kompleksitas O(1) terdapat pada algoritma yang instruksinya dijalankan satu kali (tidak ada pengulangan) seperti penjumlahan, pengurangan ,if-else, dan lain sebagainya.

- O(log n)
  Kompleksitas O(log n) menandakan bahwa jumlah n tidak begitu banyak memengaruhi pemrosesannya. Algoritma yang termasuk jenis ini yakni algoritma yang memecahkan persoalan besar menjadi semakin kecil. Contoh algoritma yang termasuk O(log n) adalah binary search tree.

- O(n)
  Jenis algoritma yang bergradien satu ini menandakan bahwa pemrosesan data yang dilakukan selalu berlaku sama. Algoritma-algoritma sorting manual termasuk ke dalam jenis ini.

- O(n log n)
  Jenis algoritma ini terdapat pada hampir semua jenis algoritma divide and conquer (membagi persoalan ke dalam sekumpulan kasus yang lebih kecil kemudian menggabungkan keseluruhan solusi menjadi sebuah solusi utama). Bisa dilihat bahwa algoritma ini sama dengan n kali dari log n.

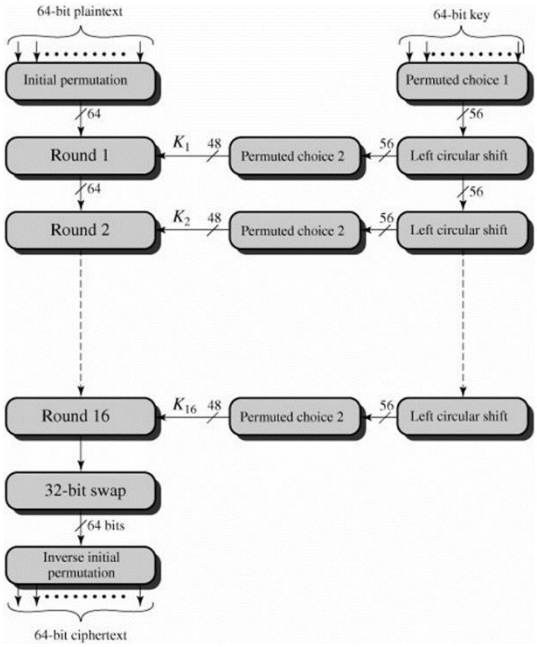
- $O(n^2)$
  Algoritma jenis ini terdapat pada berbagai macam kasus pemrosesan loop ganda (while ganda, for ganda, do-while ganda, dsb)., seperti penjumlahan dan pengurangan matriks sebagai contohnya.

- $O(n^3)$
  Sebagaimana jenis algoritma n kuadratik, jenis algoritma ini menandakan adanya loop lipat tiga. Pemrosesan perkalian matriks adalah salah satu contohnya.

Selain dari jenis-jenis algoritma di atas, kebanyakannya merupakan algoritma yang kurang efektif, seperti 2^n, n!, dan

masih banyak lagi.

4. Algoritma DES



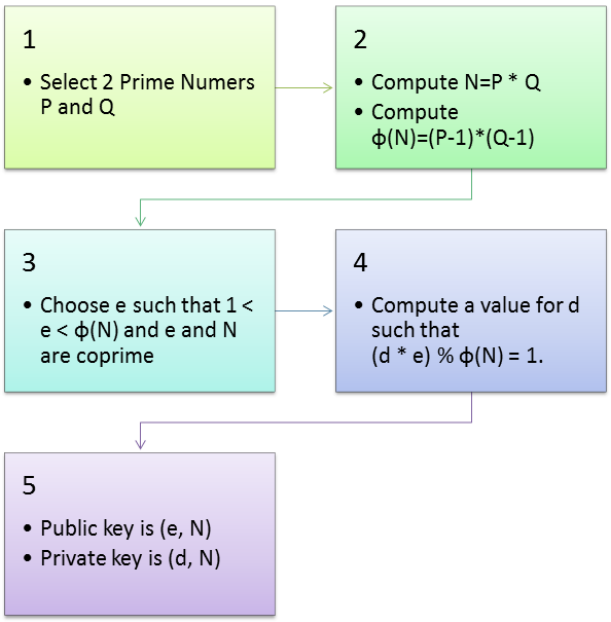
Gambar 4.1. Flowchart DES (thecrazyprogrammer.com)

DES beroperasi pada blok 64 bit. DES akan mengenkripsi 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit subkey. Kunci internal akan diperoleh dari kunci eksternal (external key) 64 bit, tetapi hanya 56 bit yang dipakai (8 bit paritas tidak digunakan).

Algoritma DES ini pada implementasinya hanya menggunakan beberapa looping saja tanpa nested loop, sehingga kompleksitas dari DES adalah O(n).

5. Algoritma RSA

Algoritma RSA (Rivest Shamir Adleman) adalah algoritma yang memanfaatkan aritmetika modulo dalam penentuan kunci public dan privat untuk proses enkripsi dan dekripsinya. Alur kerja RSA dapat didefinisikan sebagai berikut.



Gambar 5.1. Flowchart RSA (c-sharpcorner.com)

Tiga komponen utama pada RSA adalah perpangkatan, inversi dan modulo . Kompleksitas waktu RSA sangat dipengaruhi oleh tiga proses ini. Pemrosesan modulo pada RSA termasuk algoritma O(log n) . Yang kedua yaitu multiplikasi modular adalah algoritma berjenis $O(\{\log n\}^2)$. Hal ini tecermin dari proses me mod N. Secara keseluruhan, proses ini akhirnya tergolong menjadi $O(\{\log n\}^3)$. Yang ketiga adalah algoritma euclidean (GCD) dan inversnya merupakan $O(\{\log n\}^3)$.

Dengan demikian, notasi Big O untuk enkripsi RSA termasuk ke dalam $O(n^2)$. Selain itu, untuk dekripsi RSA, perlu dilakukan operasi perpangkatan dan perkalian untuk setiap elemen, sehingga $O(n^3)$ adalah representasi Big O yang tepat untuk dekripsi RSA.

III. HASIL DAN PEMBAHASAN

Uji coba dilakukan sebanyak lima kali menggunakan teks sampel "Lorem Ipsum" yang bisa diperoleh pada page ini.

Test 1 : 10.000 kata "Lorem Ipsum"
Test 2 : 100.000 kata "Lorem Ipsum"
Test 3 : 1.000.000 kata "Lorem Ipsum"
Test 4 : 10.000.000 kata "Lorem Ipsum"

Variabel bebas : Plaintext sebanyak empat variasi.
Variabel terikat : Waktu enkripsi, hasil enkripsi dan waktu dekripsi.

Notasi algoritmik pada bab sebelumnya diimplementasikan dengan bahasa C yang terdiri dari run_des.c (main driver), des.c (kumpulan implementasi method) dan des.h (header des.c).

Courtesy : (tarequeh)

a. Test 1

- Enkripsi

```
$ run_des.o -e keyfile.key sample.txt sample.enc
Encrypting..
Finished processing sample.txt. Time taken: 0.046000 seconds.
```

Gambar 3.a.1.

- Hasil Cipherteks

```
sample.enc
4259  d33b 6652 3c4f 6559 530d 4f3a 840b 44d8
4260  49f4 9305 93c3 6e05 3d39 3778 d9e6 cc92
4261  3f83 3223 f984 e93e 1d16 9eb0 e5fe 21a0
4262  6e0c f00e 1a7d dcc4 04ad 3ea1 40d0 21e4
4263  2ffd 88e3 2966 8c27 bc81 777f a5e8 4e38
4264  225c 2b48 ed9f 67b7 e75d 5d45 7f56 2dc9
4265  6b0a aafa 4511 f6b9 0daa f5ab d83c e0d2
4266  44fc d789 6eb0 de08 42f2 1446 815a 5d47
4267  5ad0 21bd 0a58 fb14 ce74 2c6c 60b7 9266
4268  e5d8 28e3 b3e4 59d7 5da0 2782 d7a3 67a2
4269  29aa f967 25b8 8b69 1b80 1c83 a38e 8a0d
4270  5c2c 7611 982a 74ee dcff 22b0 3ef4 2f3b
4271  9163 850a dcc6 5162
```

Gambar 3.a.2 .

Panjang data yang dihasilkan = 4270 x 8 + 4 = 34.164 kata.

- Dekripsi

```
$ run_des.o -d keyfile.key sample.enc sample_dec.txt
Decrypting..
Finished processing sample.enc. Time taken: 0.031000 seconds.
```

Gambar 3.a.3 .

b. Test 2

- Enkripsi

```
$ run_des.o -e keyfile.key sample.txt sample.enc
Encrypting..
Finished processing sample.txt. Time taken: 0.593000 seconds.
```

Gambar 3.b.1 .

- Hasil Ciphertext

```
sample.enc
42699  c3d9 4327 0bc1 0739 b18e b68f c334 d003
42700  2a6c e6fc 2e4a 8ae2 f959 1efe 9eed d9f5
42701  d09b 27bc 91a4 1374 9817 205a 8b62 ef6b
42702  841e a9ab e1a4 779b 7636 1f4b 71f5 37e8
42703  54bf 364b 38b4 f6f9 3ce4 6974 c9c5 0224
42704  c3c6 a082 70c6 6e79 1ff9 53b1 e08b 9de0
42705  dec4 86c4 1367 de67 b884 d40d d134 9384
42706  12f4 ff2f 4e02 640a c42a 495d efb6 b4c2
42707  50ac b845 cbb0 71e1 bc4d 18bc 84b5 dbf3
42708  c637 7bd0 9ebc e2ca
```

Gambar 3.b.3 .

Panjang data yang dihasilkan = 42707 x 8 + 4 = 341.660 kata.

- Dekripsi

```
$ run_des.o -d keyfile.key sample.enc sample_dec.txt
Decrypting..
Finished processing sample.enc. Time taken: 0.421000 seconds.
```

Gambar 3.b.3 .

c. Test 3

- Enkripsi

```
$ run_des.o -e keyfile.key sample.txt sample.enc
Encrypting..
Finished processing sample.txt. Time taken: 5.250000 seconds.
```

Gambar 3.c.1 .

- Hasil Ciphertext

```
sample.enc
427067  c3c6 a082 70c6 6e79 1ff9 53b1 e08b 9de0
427068  dec4 86c4 1367 de67 b884 d40d d134 9384
427069  12f4 ff2f 4e02 640a c42a 495d efb6 b4c2
427070  50ac b845 cbb0 71e1 bc4d 18bc 84b5 dbf3
427071  c637 7bd0 9ebc e2ca
```

Gambar 3.c.2 .

Panjang data yang dihasilkan = 427.070 x 8 + 4 = 3.416.560 kata.

- Dekripsi

```
$ run_des.o -d keyfile.key sample.enc sample_dec.txt
Decrypting..
Finished processing sample.enc. Time taken: 4.765000 seconds.
```

Gambar 3.c.3 .

d. Test 4

- Enkripsi

```
$ run_des.o -e keyfile.key sample.txt sample.enc
Encrypting..
Finished processing sample.txt. Time taken: 51.953000 seconds.
```

Gambar 3.d.1 .

- Hasil Ciphertext

```
sample.enc
4270694  d09b 27bc 91a4 1374 9817 205a 8b62 ef6b
4270695  841e a9ab e1a4 779b 7636 1f4b 71f5 37e8
4270696  54bf 364b 38b4 f6f9 3ce4 6974 c9c5 0224
4270697  c3c6 a082 70c6 6e79 1ff9 53b1 e08b 9de0
4270698  dec4 86c4 1367 de67 b884 d40d d134 9384
4270699  12f4 ff2f 4e02 640a c42a 495d efb6 b4c2
4270700  50ac b845 cbb0 71e1 bc4d 18bc 84b5 dbf3
4270701  c637 7bd0 9ebc e2ca
```

Gambar 3.d.2 .

Panjang data yang dihasilkan = 427.070 x 8 + 4 = 34.165.604 kata.

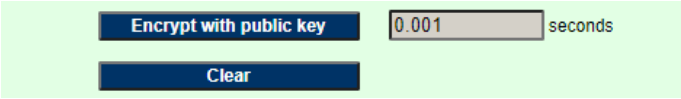
- Dekripsi

```
$ run_des.o -d keyfile.key sample.enc sample_dec.txt
Decrypting..
Finished processing sample.enc. Time taken: 54.858000 seconds.
```

Gambar 3.d.3 .

Uji coba e dan f hanya dilakukan sekali hanya untuk memastikan kebenaran parameter. Uji coba dilaksanakan pada platform RSA pada mobilefish.com

e. Enkripsi RSA

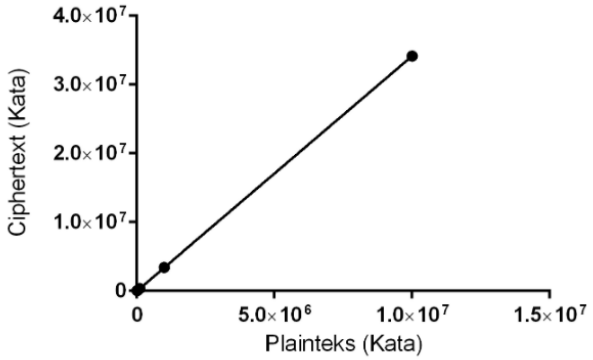


f. Dekripsi RSA

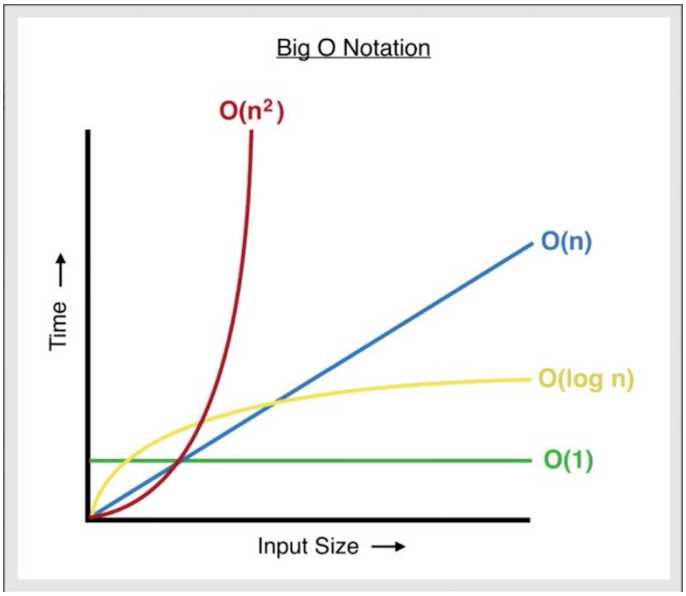
g. Pembahasan

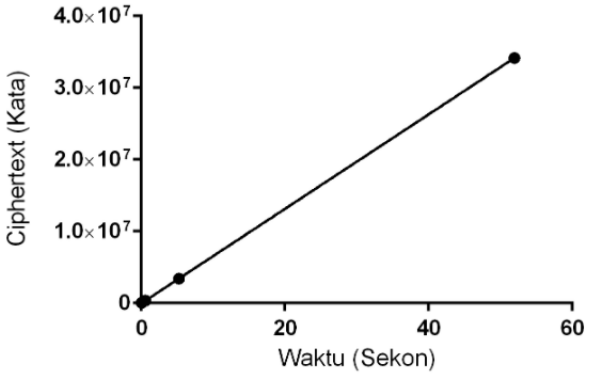
Jumlah kata plaintekt	Waktu enkripsi	Jumlah kata ciphertext	Waktu dekripsi
10.000	0.046 s	34.164	0.031 s
100.000	0.593 s	341.660	0.421 s
1.000.000	5.25 s	3.416.560	4.765 s
10.000.000	51.95 s	34.165.604	54.85 s

Tabel 3.e.1. Hasil uji coba

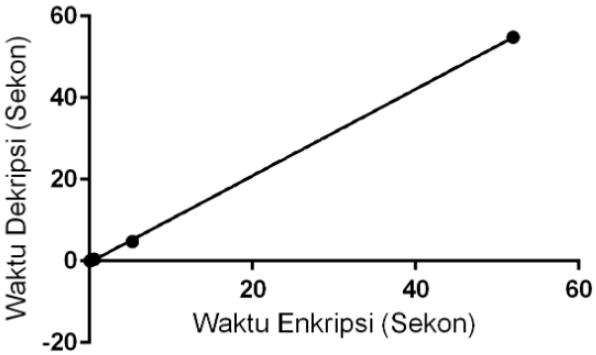


Gambar 3.e.1. Grafik jumlah kata plainteks terhadap hasil cipherteks





Gambar 3.e.2. Grafik waktu enkripsi terhadap hasil cipherteks



Gambar 3.e.2. Grafik waktu dekripsi terhadap hasil cipherteks

Berdasarkan grafik di atas, terlihat bahwa bertambahnya jumlah kata plaintext, waktu enkripsi juga waktu dekripsi, semuanya berbanding lurus dengan jumlah kata cipherteks yang dihasilkan membentuk grafik linier. Hal ini membuktikan bahwa T(n) untuk algoritma enkripsi maupun dekripsi DES adalah O(n) sesuai dengan hipotesis pada bab sebelumnya.

Selain itu, telah diperoleh pula waktu enkripsi RSA dan dekripsi RSA. Waktu dari enkripsi RSA jauh lebih cepat dibandingkan waktu dekripsinya. Hasil tersebut sesuai nilai T(n) enkripsi RSA = $O(n^2)$ dan T(n) dekripsi RSA = $O(n^3)$ yang telah ditemukan pada bab sebelumnya.

IV. KESIMPULAN

Telah dilaksanakan uji coba kebenaran kompleksitas algoritma DES.

Gambar 5.1. Grafik perbandingan nilai Big O terhadap waktu.

Berdasarkan kedua data RSA dan DES yang telah diperoleh, serta grafik perbandingan nilai Big O terhadap waktu, algoritma RSA dapat diposisikan sama dengan garis $O(n^2)$ berwarna merah, sedangkan algoritma DES dapat diposisikan

sama dengan O(n) berwarna biru. Dengan demikian, diperoleh hasil akhir bahwa algoritma kriptografi simetri jauh lebih unggul dibandingkan algoritma kriptografi asimetri.

Saat ini, algoritma DES banyak digunakan dalam berbagai kriptografi. Tidak hanya teks, tetapi juga gambar bahkan suara. Lahirnya algoritma ini pun merupakan hasil pengembangan sekian rupa dari algoritma asimetri yang pernah ada. Ilmu kriptografi saat ini merupakan bidang yang akan terus berkembang seiring berjalannya masa. Tidak hanya ilmu kriptografi, tetapi semua ilmu keinformatikaan yang terkait dengan programming akan selalu berkembang beriringan bersama dengan kemajuan algoritma.

VI. PENUTUP

Puji dan syukur Penulis panjatkan kepada Allah Swt. Karena atas rahmat dan karunia-Nya lah makalah berjudul "Studi Kompleksitas Algoritma Enkripsi Teks Simetri dan Asimetri" ini dapat terwujud. Penulis juga mengucapkan banyak terima kasih kepada seluruh tim dosen IF2120 Matematika Diskrit T.A 2020/2021 (Bpk. Dr. Rinaldi Munir, I. Harlili, M.Sc,, I. Fariska Zakhralativa Ruskanda, S.T., M.T., I. Nur Ulfa Maulidevi, S.T., M.Sc) atas semua ilmu yang telah disampaikan selama perkuliahan ini, khususnya pada bab kompleksitas algoritma dan subbab kriptografi. Tidak lupa pula, penulis mengucapkan terima kasih pada seluruh keluarga, teman-teman, dan semua pihak yang senantiasa mendukung penulis dalam mewujudkan makalah ini.

REFERENCES

[1] Sulistyorini, Agus Prihanto "Perbandingan Efisiensi Algoritma RSA dan RSA-CRT"
[2] (133) What is the complexity of RSA cryptographic algorithm? - Quora. Diakses pada 11 Desember 2020 pukul 16:00.
[3] Cahyo, Nur, Dkk. "Komparasi Waktu Algoritma RSA Dengan RSA-CRT Base"
[4] Kabetta, Herman. "Analisis Kompleksitas Waktu Algoritma Kriptografi Elgamal dan Data Encryption Standard".
[5] Munir, Rinaldi. "Data Encryption Standard"

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, 11 Desember 2020

Almeiza Arvin Muzaki
13519066