Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 17−24
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

Implementasi Algoritma Quicksort Untuk Pembangkitan Kunci Algoritma Elgamal Pada Pengamanan Data File Dokumen

Al-Amansyah

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: alamansyah32@gmail.com

Abstrak–Dokumen secara sahifah yaitu sebuah tulisan penting yang memuat informasi. Biasanya, dokumen ditulis di kertas dan informasinya ditulis memakai tinta baik memakai tangan atau memakai media elektronik. Untuk itu diperlukan sebuah sistem pengamanan untuk mengamankan isi dokumen yang terdapat didalamnya. Dalam penelitian digunakan metode kriptografi yang akan menjamin keamanan dari sebuah file dokumen. Dalam penelitian ini terdapat proses keamanan yang menggunakan metode algoritma Elgamal sebagai penyelesaian pada implementasi metode algoritma Quicksort untuk file dokumen. Hasil penelitian ini menunjukkan bahwa aplikasi visual basic dapat mengetahui bahwa file dokumen aman atau tidak. Pengerjaan metode ini dapat membantu dengan mengamankan isi sebuah file dokumen agar tidak diketahui dengan mudah. Implementasi Quicksort yang menggunakan algoritma Elgamal akan diterapkan dengan menggunakan aplikasi visual basic sebagai bukti dari hasil pengaman file dokumen dengan aman.

Kata Kunci: File Dokumen; Pesan Teks; Kriptografi; Quicksort; Elgamal

Abstract–A valid document is an important article containing information. Typically, documents are written on paper and the information is written in ink either by hand or using electronic media. For that we need a security system to secure the contents of the documents contained therein. In this research, a cryptographic method is used that will ensure the security of a document file. In this research, there is a security process that uses the Elgamal algorithm method as a solution to the implementation of the Quicksort algorithm method for document files. The results of this study indicate that the visual basic application can determine whether document files are safe or not. Using this method can help by securing the contents of a document file so that it is not known easily. Quicksort implementation that uses the Elgamal algorithm will be implemented using the visual basic application as evidence of the results of securing document files safely.

Keywords: Document Files; Text Messaging; Cryptography; Quicksort; Elgamal

1. PENDAHULUAN

Banyaknya informasi yang bisa didapatkan sehingga menyebabkan penyimpanan suatu file haruslah memenuhi aspek keamanan. Baik itu berupa file dokumen dalam bentuk teks, gambar, suara dan video. Agar file-file yang bersifat rahasia tersebut tidak dapat dicuri dan digunakan oleh pihak-pihak yang tidak bertanggung jawab.

Kriptografi yang merupakan suatu seni dan ilmu untuk menulis rahasia yang bertujuan untuk menjaga kerahasiaan suatu pesan. Informasi yang ingin dikirim harus terjaga kerahasian didalamnya dan tetap asli pada sampai tujuan tanpa harus dimodifikasi. Dalam metode algoritma kriptografi terdapat dua proses yaitu enkripsi dan dekripsi.

Teknik kriptografi yang dapat digunakan adalah algoritma Elgamal. Algoritma Elgamal adalah sebuah algoritma untuk kriptografi kunci publik. Algoritma ini memiliki keamanan yang terletak pada kesulitan dalam menghitung logaritma diskrit. Algoritma Elgamal tipe algoritma kriptografi asimetris terdiri atas dua buah kunci yaitu kunci publik untuk melakukan enkripsi sedangkan kunci pribadi untuk melakukan dekripsi.

Dalam algoritma Elgamal, kunci yang didistribusikan adalah kunci publik yang tidak diperlukan kerahasiaanya sedangkan kunci pribadi tetap disimpan atau tidak didistribusikan. Setiap orang yang memiliki kunci publik dapat melakukan proses enkripsi tetapi hasil dari enkripsi tersebut hanya bisa dibaca oleh orang yang memiliki kunci pribadi. Untuk meningkatkan kekuatan dari algoritma tersebut, maka kunci yang digunakan untuk melakukan proses enkripsi dan dekripsi akan dimodifikasi terlebih dahulu menggunakan algoritma pengacakan.

Algoritma pengacakan adalah yang menghasilkan permutasi acak dari suatu himpunan terhingga, dengan kata lain untuk mengacak suatu himpunan Adapun salah satu algoritma pengacakan yang di gunakan adalah Algoritma

QuickSort. Algoritma QuickSort merupakan suatu algoritma pengurutan data yang menggunakan teknik pemecahan data menjadi partisi-partisi, sehingga metode ini disebut juga dengan nama partition exchange sort. Algoritma QuickSort bekerja menurut prinsip bagi-dan-pecahkan (divide-and-conquer). Dengan demikian hal mendasar dalam algoritma Quicksort adalah pemilihan poros (pivot) dan pembuatan partisi sedemikian rupa sehingga elemen dengan nilai kecil ada di bagian kiri poros dan elemen dengan nilai besar ada di bagian kanan poros.

2. METODOLOGI PENELITIAN

2.1 Kriptigrafi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern *kriptografi* adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian *kriptografi* modern adalah tidak saja berurusan hanya dengan menyembunyikan pesan, namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi.

Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 17−24
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

2.2. Algoritma ElGamal

Algoritma *ElGamal* merupakan algoritma dalam kriptografi yang termasuk dalam kategori algoritma asimetris. Keamanan algoritma *ElGamal* terletak pada kesulitan penghitungan logaritma diskrit pada bilangan modulo prima yang besar sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sukar. Algoritma *ElGamal* mempunyai kunci publik berupa tiga pasang bilangan dan kunci rahasia berupa satu bilangan. Algoritma ini mempunyai kerugian pada cipherteksnya yang mempunyai panjang dua kali lipat dari plainteksnya. Akan tetapi, algoritma ini mempunyai kelebihan pada enkripsi. Untuk plainteks yang sama, algoritma ini memberikan cipherteks yang berbeda (dengan kepastian yang dekat) setiap kali plainteks di enkripsi. Algoritma *ElGamal* terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Algoritma ini merupakan *cipher* blok, yaitu melakukan proses enkripsi pada blok-blok plainteks dan menghasilkan blok-blok *cipherteks* yang kemudian dilakukan proses dekripsi dan hasilnya digabungkan [1].

2.3. Algoritma Quicksort

Algoritma *quicksort* bekerja menurut prinsip bagi-dan-pecahkan (*divide-and-conquer*). Dengan demikian hal mendasar dalam algoritma *quicksort* adalah pemilhan poros (*pivot*) dan pembuatan partisi sedemikian rupa sehingga elemen dengan nilai kecil ada dibagian kiri poros dan elemen dengan nilai besar ada di bagian kanan poros. Berbagai varian dari *quicksort* pada intinya berupaya untuk mengembangkan teknik-teknik pembuatan partisi yang efektif untuk berbagai macam masukan. Pada tahun 1998 *M.D Mcllroy* membuat tulisan berjudul *"A Killer Adversary for Quicksort"* yang menguraikan cara untuk membuat susunan data tertentu (dalam *array*) hingga operasi pengurutan menggunakan *quicksort* mendekati kuadratik O(n^2). Cara ini berlaku untuk setiap varian dari *quicksort* dengan syarat tertentu.

Kebutuhan waktu dari *quicksort* bergantung pada pembuatan partisi, seimbang atau tidak, yang bergantung juga pada elemen yang digunakan sebagai poros. Dalam menghitung kompleksitas ini, perlu dilihat pula perhitungan *recurrence*, karena terdapat fungsi rekursif untuk penyelesaian sub-masalah. Terdapat 3 jenis kompleksitas waktu dari *quicksort* [5].

1. Kasus terburuk (*worst case*), yaitu terjadi bila terbentuk partisi dengan komposisi sub-masalah antara n – 1 elemen dan 0 elemen. Dengan demikian pemanggilan fungsi secara rekursif dengan array berukuran 0 akan langsung kembali, T(0) = O(1), sehingga berlaku: T*(n) = T(n-1) + cn = O(n^2).*
2. Kasus terbaik *(best case),* yaitu terjadi bila terbentuk partisi dengan komposisi seimbang, dengan ukuran masing-masing tidak lebih dari n/2. Sehingga didapat: *T(n) = 2T(n/2) + cn = na + cn log n = O(n log n).*
3. Kasus rata-rata *(average case),* yaitu terjadi dari perimbangan poros antara terbaik dan terburuk, yang dalam prakteknya lebih mendekati kasus terbaik ketimbang terburuk. Sehingga didapat: $T_{avg}(n)$*=O(n log n).*

3. HASIL DAN PEMBAHASAN

Analisa terdapat suatu algoritma dapat bertujuan untuk melihat faktor efesiensi dan efektifitas dari algoritma yang sedang dianalisa, dapat dilakukan dengan melihat sisi waktu tempuh dari suatu algoritma, proses, langkah-langkah atau satuan waktu yang ditempuh dari suatu algoritma dalam menyelesaikan suatu masalah. Kriptografi merupakan metode dengan menyandikan *file* teks menjadi yang sulit atau bahkan tidak dipahami melalui proses enkripsi. Untuk memperoleh kembali informasi yang dapat dengan proses enkripsi, untuk memperoleh kembali informasi yang asli dan dapat dilakukan dengan proses enkripsi yang tentunya dapat digunakan dengan kunci yang benar. Untuk melindungi *file* teks dari pihak-pihak yang tidak berkepintangan tersebut maka diperlukan enkripsi dan denkripsi agar dapat dilakukan dengan baik. Dibutuhansuatu algoritma untuk enkripsi dan dekripsi salah satunya algoritma *Elgamal*.

Algoritma *Elgamal* mempunyai dua kunci, yaitu kunci publik dan kunci rahasia. Algoritma ini memiliki keamanan yang terletak pada kesulitan dalam menghitung logaritma diskrit. Baik kunci enkripsi maupun dekripsi keduanya merupakan bilangan bulat. Algoritma *Elgamal* tipe algoritma kriptografi asimetris terdiri atas dua buah kunci yaitu kunci publik untuk melakukan enkripsi sedangkan kunci pribadi untuk melakukan dekripsi. Dalam algoritma *Elgamal*, kunci yang didistribusikan adalah kunci publik yang tidak diperlukan kerahasiaanya sedangkan kunci pribadi tetap disimpan atau tidak didistribusikan. Setiap orang yang memiliki kunci publik dapat melakukan proses enkripsi tetapi hasil dari enkripsi tersebut hanya bisa dibaca oleh orang yang memiliki kunci pribadi. Untuk meningkatkan kekuatan dari algoritma tersebut, maka kunci yang digunakan untuk melakukan proses enkripsi dan dekripsi akan dimodifikasi terlebih dahulu menggunakan algoritma *QuickSort*.

3.1 Pembentukan Kunci Elgamal Dengan Algoritma QuickSort

Adapun proses modipikasi kunci algoritma *Elgamal* dengan menggunkan algoritma *Quicksort* adalah sebagai berikut:

1. Pilih sembarang bilangan prima dengan melakukan 15 kali pengacakan
2. Pilih dua buah bilangan acak guntuk nilai minimum dan x untuk nilai maksimum berdasar kanan goritma *Quicksort*
3. Pembentukan kunci berdasarkan bilangan prima g dan x.

Diketahui :
Dimana:
B= 1 a_n= 41 *C*= 91 *mod*= 255

Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 17−24
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

$$a_1 = (1 \times (41 + 91)) \; mod \; 255$$
$$= 132 \; mod \; 255 = 132$$
$$a_2 = (1 \times (132 + 91)) \; mod \; 255$$
$$= 223 \; mod \; 255 = 223$$
$$a_3 = (1 \times (223 + 91)) \; mod \; 255$$
$$= 314 \; mod \; 255 = 59$$
$$a_4 = (1 \times (59 + 91)) \; mod \; 255$$
$$= 150 \; mod \; 255 = 150$$
$$a_5 = (1 \times (150 + 91)) \; mod \; 255$$
$$= 241 \; mod \; 255 = 241$$
$$a_6 = (1 \times (241 + 91)) \; mod \; 255$$
$$= 332 \; mod \; 255 = 77$$
$$a_7 = (1 \times (77 + 91)) \; mod \; 255$$
$$= 168 \; mod \; 255 = 168$$
$$a_8 = (1 \times (168 + 91)) \; mod \; 255$$
$$= 259 \; mod \; 255 = 4$$
$$a_9 = (1 \times (4 + 91)) \; mod \; 255$$
$$= 95 \; mod \; 255 = 95$$
$$a_{10} = (1 \times (95 + 91)) \; mod \; 255$$
$$= 186 \; mod \; 255 = 186$$
$$a_{11} = (1 \times (186 + 91)) \; mod \; 255$$
$$= 277 \; mod \; 255 = 22$$
$$a_{12} = (1 \times (22 + 91)) \; mod \; 255$$
$$= 113 \; mod \; 255 = 113$$
$$a_{13} = (1 \times (113 + 91)) \; mod \; 204$$
$$= 204 \; mod \; 255 = 204$$
$$a_{14} = (1 \times (204 + 91)) \; mod \; 295$$
$$= 295 \; mod \; 255 = 40$$
$$a_{15} = (1 \times (40 + 91)) \; mod \; 131$$
$$= 131 \; mod \; 255 = 131$$

Sehingga didapatkan nilai acak adalah: 132, 223, 59, 150, 241, 77, 168, 4, 95, 186, 22, 113, 204, 40, 131. Maka proses selanjutnya mencari nilai bilangan prima dari nilai acak diatas menggunakan algoritma *Quicksort* dengan cara menyeleksi dan mengurutkan bilangan yang telah diacak sebagai berikut:

132	223	59	150	241	77	168	4	95	186	22	113	204	40	131

1. Langkah pertama yaitu menentukan privotnya, dalam contoh diatas saya memilih angka dari sebelah kanan kolom

132	223	59	150	241	77	168	4	95	186	22	113	204	40	131

2. Kemudian buat partisi disebelah kiri dan kanan dengan pengurutan angka terkecil sebelum angka terbesar

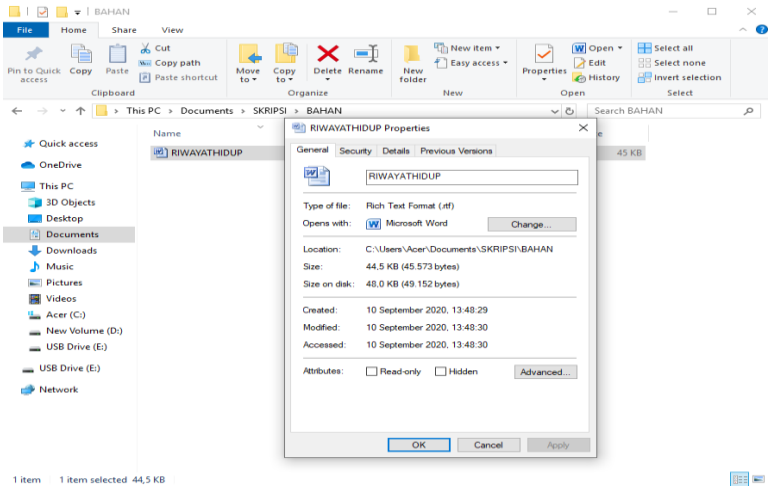
132	223	59	150	241	77	168	4	95	186	22	113	204	40	131
131	132	223	59	150	241	77	168	4	95	186	22	113	204	40

Maka hasil data acak yang telah diurutkan dengan algoritma *Quicksort* adalah sebagai berikut: 4, 22, 40, 59, 77, 95, 113, 131, 132, 150, 168, 186, 204, 223, 241. Maka proses selanjutnya adalah mencari bilangan prima terbesar dan terkecil dengan cara mengurutkan dengan algoritma *Quicksort.* Adapun bilangan prima yang didapatkan dari hasil pengurutan algoritma *Quicksort* adalah: 59, 113, 131, dan 223. Maka nilai bilangan terkecil 59 dan nilai bilangan nilai terbesar 223 dimana 59 adalah *g*, 223 adalah *p* dan nilai 113 adalah nilai *x*.
Maka proses selanjutnya menentukan kunci:
y = g^x mod p
$y = 59^{133} \bmod 223$ y = 22
Jadi kunci *public* yang digunakan adalah y= 22, g= 11, p= 223 dan kunci *private* adalah x= 113, p= 223.

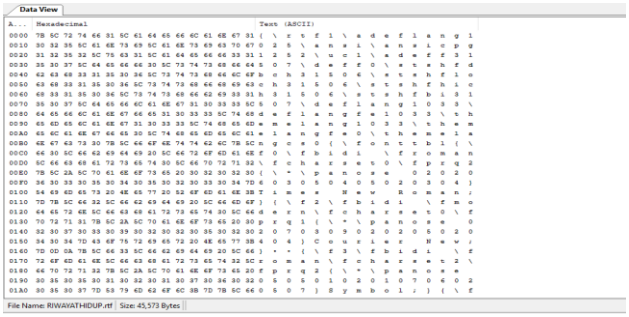
3.2 Proses Enkripsi Algoritma Elgamal

Adapun proses implementasi enkripsi algoritma *Elgamal* yaitu contoh kasus dalam proses ini adalah *file* dokumen RTF "RIWAYATHIDUP". Data yang diambil hanya sebanyak 10 *bytes* untuk plainteks, cara pengambilan nilai text *ASCII* data dokumen menggunakan aplikasi *Binary Viewer*, seperti dibawah ini:

Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 17−24
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

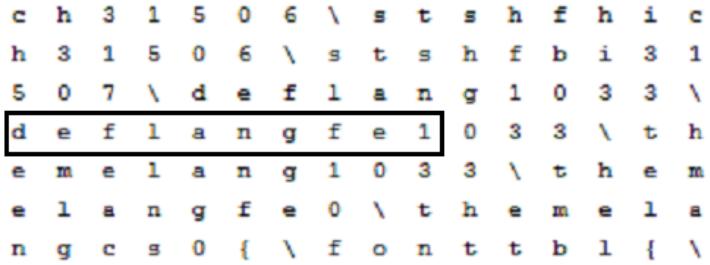


Gambar 1. *File* Dokumen Yang Akan Diimplementasi



Gambar 2. Isi *file* Dokumen Dari Hasil *Binary Viewer*

Dari data tersebut diambil sebanyak 10 *bytestext ASCII* dan dikonversi ke *ASCII decimal* sebagai sampel metode, yang berguna untuk mengetahui nilai biner dari bilangan tersebut.



Gambar 3. Nilai *ASCII* Berdasarkan *Binary Viewer*

Dari gambar data dokumen di atas diambil sebanyak 10 *bytes* untuk plainteks, yaitu :
Text ASCII : d, e, f, l, a, n, g, f, e, 1
Kunci *PRIVATE* = (x,y) $K_{private}$ = (113, 22)
Kunci *PUBLIC* = (p,g,y) K_{public} = (223,11,22)

Tabel 1. Hasil Enkripsi Dari Bilangan *ASCII*

Plainteks	Char	ASC	K	a=(g^k)mod p	b=((y^k)*M)) mod p	(a,b)
M1	D	100	1	11	193	(11,193)
M2	E	101	21	105	18	(105,18)
M3	F	102	19	186	97	(186,97)
M4	L	108	17	102	47	(102,47)
M5	A	97	15	141	192	(141,192)
M6	N	110	13	97	222	(97,222)
M7	G	103	11	198	127	(198,127)
M8	F	102	9	103	89	(103,89)
M9	E	101	7	93	111	(93,11)
M10	1	49	5	45	92	(45,92)

Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 17−24
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

Susun plainteks menja diblok-blok m1 , m2 , …, nilai setiap blok di dalam selang $[0, \mathrm{p} - 1]$. Pilih bilangan acak k, yang dalam hal ini$1 \leq \mathrm{k} \leq \mathrm{p} - 2 \bmod \mathrm{M}$.
Pembentukan Nilai $K_n = 1 \leq \mathrm{k} \leq \mathrm{p} - 2$.

$K_1 = 223 - 2\ mod\ 22 = 1$
$a_1 = (g\hat{}k)\ mod\ p$
$= 11^1\ mod\ 223$
$= 11$
$b_1 = \left((y\hat{}k) * M\right) mod\ p$
$= \left((22^1) * 100\right) mod\ 223$
$= 193$
$K_2 = 223 - 4\ mod\ 22 = 21$
$a_2 = (g\hat{}k)\ mod\ p$
$= 11^{21}\ mod\ 223$
$= 105$
$b_2 = \left((y\hat{}k) * M\right)\ mod\ p$
$= \left((22^{21}) * 101\right)\ mod\ 223$
$= 18$
$K_3 = 223 - 6\ mod\ 22 = 19$
$a_3 = (g\hat{}k)\ mod\ p$
$= 11^{19}\ mod\ 223$
$= 186$
$b_3 = \left((y\hat{}k) * M\right)\ mod\ p$
$= \left((22^{19}) * 103\right)\ mod\ 223$
$= 97$
$K_4 = 223 - 8\ mod\ 22 = 17$
$a_4 = (g\hat{}k)\ mod\ p$
$= 11^{17}\ mod\ 223$
$= 102$
$b_4 = \left((y\hat{}k) * M\right)\ mod\ p$
$= \left((22^{17}) * 108\right)\ mod\ 223$
$= 74$
$K_5 = 223 - 10\ mod\ 22 = 15$
$a_5 = (g\hat{}k)\ mod\ p$
$= 11^{15}\ mod\ 223$
$= 141$
$b_5 = \left((y\hat{}k) * M\right)\ mod\ p$
$= \left((22^{15}) * 97\right)\ mod\ 223$
$= 192$
$K_6 = 223 - 12\ mod\ 22 = 13$
$a_6 = (g\hat{}k)\ mod\ p$
$= 11^{13}\ mod\ 223$
$= 97$
$b_6 = \left((y\hat{}k) * M\right)\ mod\ p$
$= \left((22^{13}) * 110\right)\ mod\ 223$
$= 222$
$K_7 = 223 - 14\ mod\ 22 = 11$
$a_7 = (g\hat{}k)\ mod\ p$
$= 11^{11}\ mod\ 223$
$= 198$
$b_7 = \left((y\hat{}k) * M\right)\ mod\ p$
$= \left((22^{11}) * 103\right)\ mod\ 223$
$= 127$
$K_8 = 223 - 16\ mod\ 22 = 9$
$a_8 = (g\hat{}k)\ mod\ p$
$= 11^9\ mod\ 223$

Bulletin of Artificial Intelligence
Volume 1, No 1, April 2022 Page: 17−24
ISSN 2962-3944 (media online)
https://journal.grahamitra.id/index.php/buai

$= 103$
$b_8 = \left((y\hat{}k) * M\right) mod\ p$
$= \left((22^9) * 102\right)\ mod\ 223$
$= 89$
$K_9 = 223 - 18\ mod\ 22 = 7$
$a_9 = (g\hat{}k)\ mod\ p$
$= 11^7\ mod\ 223$
$= 93$
$b_9 = \left((y\hat{}k) * M\right) mod\ p$
$= \left((22^7) * 101\right)\ mod\ 223$
$= 111$
$K_{10} = 223 - 20\ mod\ 22 = 5$
$a_{10} = (g\hat{}k)\ mod\ p$
$= 11^5\ mod\ 223$
$= 45$
$b_{10} = \left((y\hat{}k) * M\right) mod\ p$
$= \left((22^5) * 49\right)\ mod\ 223$
$= 92$

Hasil *chipertext* dari enkripsi algoritma *Elgamal* adalah= (11,193), (105,18), (186,97), (102,47), (141,192), (97,222), (198,127), (103,89), (93,11), (45,92). Dengan hasil *chipertext* berdasarkan *ASCII* menjadi = (VT, Á), (i, DC2), (º,a), (f,/), (, À), (a, Þ), (Æ,), (g,Y), (],VT), (–,\).

3.3 Proses Dekripsi Algoritma Elgamal

Adapun proses dekripsi algoritma *Elgamal* berdasarkan hasil enkripsi yaitu (VT, Á), (i, DC2), (º,a), (f,/), (, À), (a, Þ), (Æ,), (g,Y), (],VT), (–,\). Maka proses selanjutnya merubah hasil enkripsi ke bilangan *ASCII* desimal sebagai berikut. (11,193), (105,18), (186,97), (102,47), (141,192), (97,222), (198,127), (103,89), (93,11), (45,92).
Kunci dekripsi *PUBLIC* = (p,g,y) K_{public} = (223,11,22)
Mencari plainteks $P_1 = (11{,}193)$
$S_1 = a^x mod\ p$
$= 11^{113}\ mod\ 223$
$= 36$
$P_1 = (b * s^{p-x})\ mod\ p$
$= 193 * 36^{223-113}\ mod\ 223$
$= 100$
Mencari plainteks $P_2 = (105{,}18)$
$S_2 = a^x mod\ p$
$= 105^{113}\ mod\ 223$
$= 78$
$P_2 = (b * s^{p-x})\ mod\ p$
$= 18 * 78^{223-113}\ mod\ 223$
$= 101$
Mencari plainteks $P_3 = (186{,}97)$
$S_3 = a^x mod\ p$
$= 186^{113}\ mod\ 223$
$= 35$
$P_3 = (b * s^{p-x})\ mod\ p$
$= 97 * 35^{223-113}\ mod\ 223$
$= 102$
Mencari plainteks $P_4 = (102{,}47)$
$S_4 = a^x mod\ p$
$= 102^{113}\ mod\ 223$
$= 106$
$P_4 = (b * s^{p-x})\ mod\ p$
$= 47 * 106^{223-113}\ mod\ 223$
$= 108$
Mencari plainteks $P_5 = (141{,}192)$
$S_5 = a^x mod\ p$
$= 141^{113}\ mod\ 223$

$= 158$

$P_5 = (b * s^{p-x}) \ mod \ p$

$= 192 * 158^{223-113} \ mod \ 223$

$= 97$

Mencari plainteks $P_6 = (97{,}222)$

$S_6 = a^x mod \ p$

$= 97^{113} \ mod \ 223$

$= 26$

$P_6 = (b * s^{p-x}) \ mod \ p$

$= 222 * 26^{223-113} \ mod \ 223$

$= 110$

Mencari plainteks $P_7 = (198{,}127)$

$S_7 = a^x mod \ p$

$= 198^{113} \ mod \ 223$

$= 75$

$P_7 = (b * s^{p-x}) \ mod \ p$

$= 127 * 75^{223-113} \ mod \ 223 \quad = 103$

Mencari plainteks $P_8 = (103{,}89)$

$S_8 = a^x mod \ p$

$= 103^{113} \ mod \ 223$

$= 31$

$P_8 = (b * s^{p-x}) \ mod \ p$

$= 89 * 31^{223-113} \ mod \ 223$

$= 102$

Mencari plainteks $P_9 = (93{,}11)$

$S_9 = a^x mod \ p$

$= 93^{113} \ mod \ 223$

$= 132$

$P_9 = (b * s^{p-x}) \ mod \ p$

$= 11 * 176^{223-113} \ mod \ 223$

$= 101$

Mencari plainteks $P_{10} = (45{,}92)$

$S_{10} = a^x mod \ p$

$= 45^{113} \ mod \ 223$

$= 114$

$P_{10} = (b * s^{p-x}) \ mod \ p$

$= 92 * 114^{223-113} \ mod \ 223$

$= 49$

Hasil plainteks dari dekripsi algoritma *Elgamal* adalah= d, e, f, l, a, n, g, f, e, 1

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan dapat diambil beberapa kesimpulan Proses penerapan Algoritma *Quicksort* dengan cara melakukan pengacakan dan pengurutan nilai untuk pembentukan kunci berdasarkan metode kriptografi Algoritma *Elgamal*. Proses enkripsi dapat dilakukan serta hasil pengambilan enkripsi dengan cara melakukan proses deskripsi dapat dilakukan dengan baik menggunakana Algoritma *Elgamal* berdasarkan kunci yang telah di modifikasi dengan algoritma *Quicksort*.

REFERENCES

[1] Agus Kurniadi, "Implementasi kriptografi ELGAMAL dalam keamanan pesan," *INFOTEK*, vol. 1, pp. 1–5, 2016.
[2] Listiyono, Hersatoto. "Implementasi Algoritma Kunci Public Pada Algoritma RSA". Fakultas Teknologi Informasi, Universitas Stikubank, Semarang, 2009.
[3] Munir, Rinaldi. Kriptografi, Informatika, Bandung. 2006.
[4] R. Sadikin, *KRIPTOGRAFI UNTUK KEAMANAN JARINGAN*. YOGYAKARTA: C.V ANDI OFFSET, 2012
[5] R. Sedgewick. *Implementing quicksort programs. Comm.* ACM, 21:847–856, 1978.
[6] G.J Renier, Sejarawan dari *Universitas Collage London,(1997; 104).*
[7] jon E. Bella Ariska, suroso, "Rancangan Kriptografi HYBRID kombinasi metode Vigenere Cipher dan ELGAMAL Pada pengamanan pesan Rahasia," *Semin. Nas. Inov. dan Apl. Teknol. di Ind.*, 2018.
[8] R. Zendrato and A. U. Hamdani, "Pemodelan Sistem Informasi Pengadaan Alat dan Bahan Praktikum Menggunakan Unified Modeling Language (Studi Kasus: Program Pendidikan Dokter Gigi Spesialis Konservasi Gigi Fakultas Kedokteran Gigi Universitas XYZ)," *Konf. Nas. Teknol. Inf. dan Komput.*, vol. I, no. 1, pp. 86–95, 2017.

[9] I. Fahmi, *Manajemen Pengambilan Keputusan Teori dan Aplikasi*. Bandung: PT. Alfabeta, 2016.
[10] Z. I, "Pemodelan Berbasis UML (*Unified Modeling Language*) dengan Strategi Teknik Orientasi *User Centered Design* (UCD) dalam Sistem Administrasi Pendidikan," *Univ. Islam Negeri Sumatra Utara Medan*, no. January 2013, 201.
[11] D. Ariyus, *Kriptografi keamanan data dan komunikasi*. Graha Ilmu.
[12] W. Komputer, *Membuat Aplikasi Client Server dengan Visual Basic 2008*. Yogyakarta: Andi, 2010.