Jurnal UJMC, Volume 2, Nomor 1, Hal. 41 - 49
pISSN : 2460 - 3333 eISSN : 2579-907X

PENERAPAN KODE HUFFMAN PADA ALGORITMA RSA (RIVEST-SHAMIR-ADLEMAN) UNTUK MENYANDIKAN *PASSWORD EMAIL*

Shofwan Ali Fauji[1], Mohammad Syaiful Pradana[2], dan Niken Ayu Azhari[3]

[1] Institut Agama Islam Negeri Jember, shofwan.ali.fauji@gmail.com
[2] Universitas Islam Darul 'Ulum Lamongan, gomasyai@gmail.com
[3] Universitas Islam Darul 'Ulum Lamongan, nikenayuazhari@gmail.com

Abstract. Email password is highly confidential, that the email password can not be seen by other people then the encryption with the science of cryptography. To avoid unwanted things on the email password, the authors will examine and discuss about cryptography or encryption. Encoding using ASCII code and Huffman codewill be applied to the RSA, in order to determine which code is faster in the prosess of encoding the email password. After doing some tests it can be concluded that the Huffman code faster in the encoding process of the ASCII code. The average difference is 45% faster than the ASCII code.

Keywords: *ASCII code, Huffman code, RSA.*

Abstrak. *Password email* merupakan hal yang sangat rahasia, agar password email tidak dapat di lihat oleh orang lain maka harus dilakukan penyandian dengan ilmu kriptografi. Untuk menghindari hal yang tidak diinginkan dalam *password email*, maka penulis akan mengkaji dan membahas tentang kriptografi atau penyandian. Penyandian dengan menggunakan kode ASCII dan kode Huffman akan diterapkan pada RSA, supaya dapat mengetahui kode mana yang lebih cepat dalam proses penyandian *password email*. Setelah melakukan beberapa uji coba maka dapat disimpulkan bahwa kode huffman lebih cepat dalam proses penyandian dari pada kode ASCII. Rata-rata perbedaannya 45% lebih cepat kode Huffman daripada kode ASCII.

Kata Kunci: *kode ASCII, kode Huffman, RSA.*

1 Pendahuluan

Perkembangan ilmu dan teknologi telah mempengaruhi segala aspek kehidupan, tak terkecuali aspek komunikasi, seperti halnya dalam pengiriman pesan. Semakin dengan berkembangnya teknologi, pengiriman pesan atau *password email* kurang aman. Tidak menuntut kemungkinan ada yang ingin merubah *password email* tersebut. Salah satu cara untuk mengamankan *password email* agar tidak diketahui oleh pihak yang tidak bertanggung jawab yaitu dengan cara disandikan dengan kode-kode yang tidak dipahami, sehingga jika ada pihak yang ingin merubah akan kesulitan menerjemahkannya. Selanjutnya, untuk mengatasi permasalahan di atas, dapat diselesaikan dengan kriptografi.
Kriptografi merupakan seni dan ilmu untuk menjaga keamanan data. Dalam menjaga keamanan data, kriptografi mentransformasikan pesan asli (*plaintext*) ke dalam bentuk data sandi (*chipertext*) yang tidak dapat dikenali. *Chipertext* inilah yang kemudian dikirim oleh pengirim (*sender*) kepada penerima

(*receiver*). Setelah sampai ke penerima, *chipertext* tersebut ditransformasikan kembali ke dalam bentuk *plaintext* agar dapat dikenali.

ASCII (*American Standart Code for Information Interchange*) merupakan standar internasional dalam kode huruf dan simbol seperti *hex* dan *unicode*, tetapi ASCII bersifat universal. ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi biner sebanyak 8 bit. Di mulai dari 0000 0000 sampai 1111 1111. Total kombinasi yang dihasilkan sebanyak 256 dimulai dari 0 hingga 255 dalam sistem bilangan desimal.

Algoritma Huffman adalah algoritma yang menggunakan metode statik, yaitu pemetaan kode yang sama. Dalam pembentukannya, kode Huffman menerapkan konsep kode awalan (*prefiks code*), yang merupakan himpunan kode biner, sedemikian sehingga tidak ada anggota himpunan yang merupakan awalan dari anggota yang lain, supaya pada proses *dekoding*, tidak ada keambiguan antara satu simbol dengan simbol yang lain. Kode awalan yang mempresentasikan simbol yang lebih sering muncul menggunakan rangkaian biner yang lebih pendek daripada kode yang digunakan untuk merepresentasikan simbol yang jarang muncul. Dengan demikian, jumlah bit yang digunakan untuk menyimpan informasi pada suatu data bisa lebih pendek.

Sehubungan dengan itu, penulis akan menerapkan kode ASCII maupun kode Huffman untuk penyandian *password email* agar bisa dipahami oleh pembaca. Oleh karena itu, judul yang diangkat penulis adalah **Penerapan Kode Huffman pada Algoritma RSA untuk Menyandikan *Password Email***.

2 Kajian Teori

2.1 *American Standart Code for Information Interchange* (ASCII)

ASCII merupakan standar internasional dalam kode huruf dan simbol seperti *hex* dan *unicode*, tetapi ASCII bersifat universal. ASCII selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Kode ASCII sebenarnya memiliki komposisi biner sebanyak 8 bit. Di mulai dari 0000 0000 sampai 1111 1111. Total kombinasi yang dihasilkan sebanyak 256 dimulai dari 0 hingga 255 dalam sistem bilangan desimal [1]. Setiap kode ASCII dikodekan dalam 8 bit biner. Contoh kode ASCII terlihat dalam Tabel 1,

Tabel 1: Karakter dalam kode ASCII

Karakter	Kode ASCII	Karakter	Kode ASCII
A	01000001	N	01001110
B	01000010	O	01001111
C	01000011	P	01010000
D	01000100	R	01010010
E	01000101	S	01010011
I	01001001	”	00100010
M	01001101	Spasi	00100000

2.2 Kode Huffman

Kode Huffman dibuat pertama kali oleh Prof. David A. Huffman (1925-1999) pada tahun 1952 sebagai disertasi Ph. D dengan publikasi berjudul *A Menthod for the Contruction of Minimum-Redundancy Codes*. Algoritma Huffman adalah algoritma yang menggunakan metode statik, yaitu pemetaan kode yang sama. Dalam metode ini ada dua fase yang harus dilalui, yaitu: fase pertama adalah menghitung probabilitas kemunculan setiap karakter dan menentukan pemetaan kodenya, dan fase kedua adalah mengubah data menjadi sekumpulan kode yang akan ditransmisikan.

Kode Huffman menggunakan tabel dengan variasi kode panjang untuk melakukan *encoding* dari sebuah simbol. Tabel variasi kode panjang telah dibuat terlebih dahulu secara terpisah berdasarkan nilai kekerapan munculnya suatu simbol. Biasanya kode Huffman digunakan pada aplikasi kompresi teks, data atau citra digital.

Metode yang umum digunakan untuk kompresi data adalah kode Huffman. Yang berfungsi untuk beberapa program yang digunakan pada komputer pribadi. Beberapa dari mereka menggunakan kode Huffman, sementara yang lain sebagai salah satu langkah dalam proses kompresi bertingkat. Metode Huffman agak mirip dengan metode Fano Shannon.

2.3 Kriptografi

Kriptografi (*chrytography*) berasal dari bahasa yunani: *chrytos* artinya *secret* (rahasia), sedangkan *graphein* artinya *writing* (tulisan). Jadi, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan [4].

Didalam mempelajari ilmu kriptografi terdapat beberapa istilah atau termologi antara lain, pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Keamanan pesan diperoleh dengan menyandikannya menjadi pesan yang tidak mempunyai makna pesan yang dirahasiakan dinamakan plainteks (plainteks, artinya teks jelas yang dapat dimengerti), sedangkan pesan hasil penyandian disebut cipherteks (cipherteks, artinya teks tersandi). Pesan yang telah disandikan dapat dikembalikan lagi kepesan aslinya hanya oleh orang yang berhak (orang yang berhak adalah orang yang mengetahui metode penyandian atau memiliki kunci penyandian). Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) dan proses membalikkan cipherteks menjadi plainteksnya disebut dekripsi (*decryption*) [4].

2.4 RSA (Rivest-Shamir-Adleman)

Algoritma RSA diperkenalkan oleh tiga peneliti dari MIT (*Massachussets Institute of Technology*), yaitu Ron Rivest, Adi Shamir, dan Len Adleman, pada tahun 1976. RSA mendasarkan proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmatika modulo. Baik kunci enkripsi maupun dekripsi keduanya berupa bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diketahui umum (sehingga dinamakan kunci publik), namun kunci untuk dekripsi bersifat rahasia.

Kunci dekripsi dibangkitkan dari beberapa buah bilangan prima bersama-sama dengan kunci enkripsi. Untuk menemukan dekripsi, orang-orang memfaktorkan suatu bilangan non prima menjadi faktor primanya. Kenyataannya, memfaktorkan bilangan non prima menjadi faktor primanya bukanlah pekerjaan yang mudah. Belum ada algoritma yang mangkus (efisien) yang ditemukan untuk pemfaktoran itu. Semakin besar bilangan non primanya maka semakin sulit pemfaktorannya. Semakin sulit pemfaktorannya semakin kuat algoritma RSA. Algoritma RSA sebenarnya sederhana sekali. Secara ringkas, algorima RSA terdiri dari tiga bagian, yaitu bagian untuk membangkitkan pasangan kunci, bagian untuk enkripsi, dan bagian untuk dekripsi.

3 Metode

Pada penelitian ini, *password email* yang digunakan memiliki panjang karakter 8 sampai 16, karena *password email* minimal memiliki 8 karakter. Proses penyandian *password* menggunakan algoritma RSA umumnya menggunakan ASCII, tetapi setiap karakter ASCII dipukul rata 8 bit, sehingga dalam penelitian ini dicoba untuk menggunakan kode Huffman. Karena dengan kode Huffman setiap karakter dapat dikompres sehingga kurang dari 8 bit.

Untuk mempermudah perhitungan dalam algoritma RSA, digunakan Matlab. Dengan Matlab pula dapat dihitung waktu yang dibutuhkan dalam penyandian menggunakan algoritma RSA.

3.1 Hasil dan Pembahasan

ASCII (*American Standart Code for Information Interchange*) merupakan standar internasional dalam kode huruf dan simbol seperti *hex* dan *unicode*. Tetapi ASCII selalu bersifat universal. Berikut ini akan ditampilkan karakter alfabet dalam kode ASCII pada Tabel 2,

Tabel 2: Karakter alfabet dalam kode ASCII

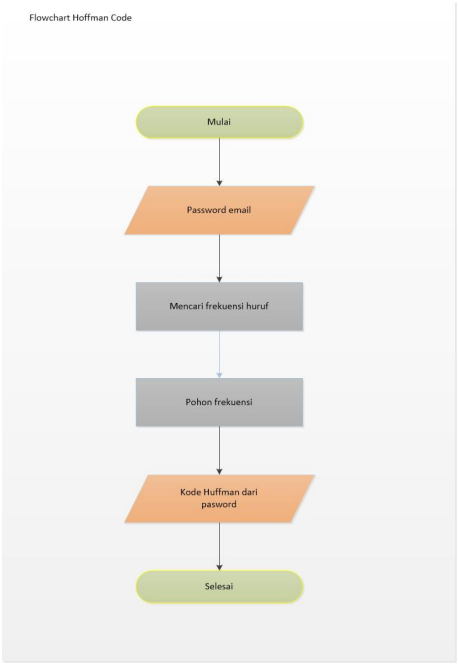
Karakter	Kode ASCII	Karakter	Kode ASCII
A	65	N	78
B	66	O	79
C	67	P	80
D	68	Q	81
E	69	R	82
F	70	S	83
G	71	T	84
H	72	U	85
I	73	V	86
J	74	W	87
K	75	X	88
L	76	Y	89
M	77	Z	90

Berikut ini akan ditampilkan karakter angka dalam kode ASCII pada Tabel 4,

Tabel 3: Karakter alfabet dalam kode ASCII

Karakter	Kode ASCII	Karakter	Kode ASCII
0	48	5	53
1	49	6	54
2	50	7	55
3	51	8	56
4	52	9	57

Kode ASCII di atas akan di rubah menjadi 2 bit menggunakan kode Huffman. Prinsip kode Huffman adalah karakter yang paling sering muncul di dalam data dikodekan dengan kode yang lebih pendek, sedangkan karakter yang jarang muncul di kodekan dengan kode yang lebih panjang. *Flowchart* untuk menerapkan kode Huffman pada *password* ditunjukkan pada Gambar 1 berikut ini,

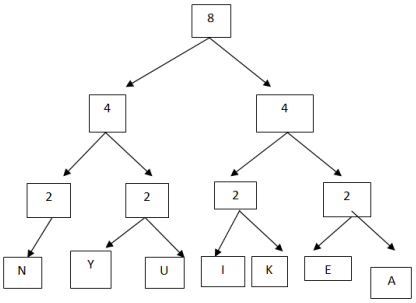


Gambar 1: *Flowchart* kode Huffman

Perbandingan Menggunakan ASCII dan Kode Huffman

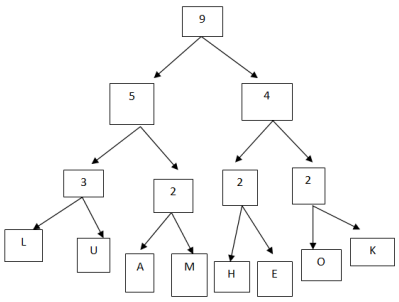
Berikut ini dilakukan percobaan *password email* dari 8 karakter sampai 16 karakter sebagai berikut,

1. *Password email* : NIKEN AYU
   Kode ASCII : 7873756978658985
   Kode Huffman : 000100101110111010011
   Pohon Huffman *password* ini ditampilkan pada Gambar 2 berikut ini,



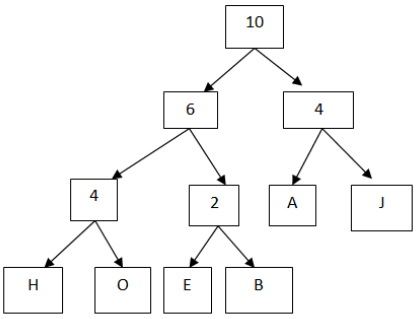
Gambar 2: Pohon Huffman *password* pertama

2. *Password email* : HELLO KAMU
   Kode ASCII : 726976767975657785
   Kode Huffman : 100101000110111010011001
   Pohon Huffman *password* ini ditampilkan pada Gambar 3 berikut ini,



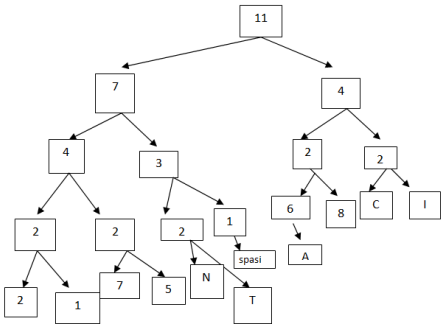
Gambar 3: Pohon Huffman *password* kedua

3. *Password email* : HEBOH AJJAH
   Kode ASCII : 72696679726574746572
   Kode Huffman : 0000100110011011
   Pohon Huffman *password* ini ditampilkan pada Gambar 4 berikut ini,



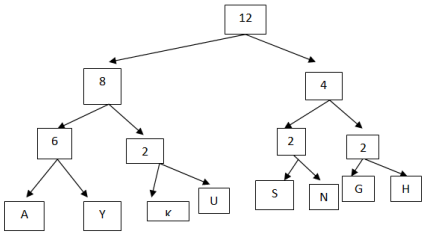
Gambar 4: Pohon Huffman *password* ketiga

4. *Password email* : 217568 CINTA
   Kode ASCII : 5049555354566773788465
   Kode Huffman : 000000010010001110010111011101000101011 0
   Pohon Huffman *password* ini ditampilkan pada Gambar 5 berikut ini,



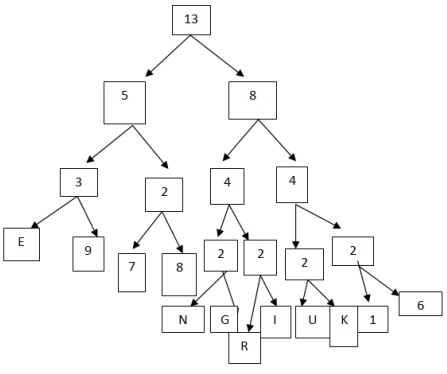
Gambar 5: Pohon Huffman *password* keempat

5. *Password email* : SAYANG AYAHKU
   Kode ASCII : 836589657871658965727585
   Kode Huffman : 100000001101110111010011
   Pohon Huffman *password* ini ditampilkan pada Gambar 6 berikut ini,



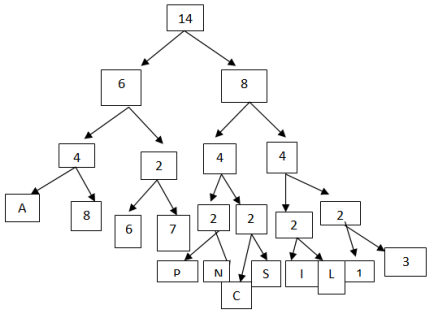
Gambar 6: Pohon Huffman *password* kelima

6. *Password email* : NEGERIKU 16789
   Kode ASCII : 78697169827375854954555657
   Kode Huffman : 1000000100110101011110111001101111010011001
   Pohon Huffman *password* ini ditampilkan pada Gambar 7 berikut ini,



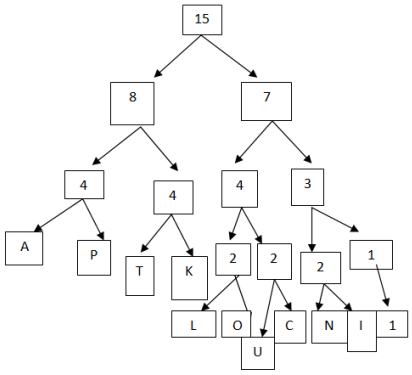
Gambar 7: Pohon Huffman *password* keenam

7. *Password email* : PANCASILA13678
   Kode ASCII : 80657867658373766549515455556
   Kode Huffman : 1000000100110101011110011011110111010011001
   Pohon Huffman *password* ini ditampilkan pada Gambar 8 berikut ini,



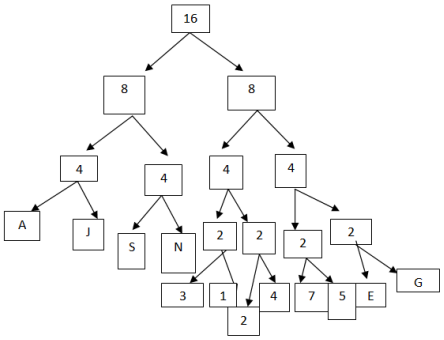
Gambar 8: Pohon Huffman *password* ketujuh

8. *Password email* : LAPTOPKU CANTIK 1
   Kode ASCII : 766580847980758567657884737549
   Kode Huffman : 10100000010101001011101010111100110111110
   Pohon Huffman *password* ini ditampilkan pada Gambar 9 berikut ini,



Gambar 9: Pohon Huffman *password* kedelapan

9. *Password email* : 312475 SENANG SAJA
   Kode ASCII : 514950525553836978657871836574 65
   Kode Huffman : 1000100110101011110011010101110011000111100 1
   Pohon Huffman *password* ini ditampilkan pada Gambar 10 berikut ini,



Gambar 10: Pohon Huffman *password* kesembilan

Password diatas akan menggunakan atribut-atribut sebagai berikut,

Bilangan prima pertama = 211
Bilangan prima kedua = 233
Kunci publik = 241
Kunci deskripsi = 3841
Hasil kali bilangan prima pertama dan kedua = 49163
Hasil kali bilangan prima pertama dan kedua yang sudah dikurangi 1 = 48720

Setelah melakukan 16 kali uji coba yaitu 8 kali menggunakan ASCII dan 8 kali menggunakan kode Huffman didapatkan hasil berikut,

Tabel 4: Karakter alfabet dalam kode ASCII

Percobaan	ASCII	Kode Huffman	Perbedaan (%)
8 huruf	0.017783	0.009808	53%
9 huruf	0.019586	0.015314	21%
10 huruf	0.018548	0.009533	49%
11 huruf	0.021521	0.010939	48%
12 huruf	0.020340	0.011001	52%
13 huruf	0.025532	0.011796	43%
14 huruf	0.021154	0.010271	46%
15 huruf	0.021396	0.011314	50%
16 huruf	0.026159	0.011609	41%

Setelah melakukan 8 huruf sampai 16 huruf, dapat disimpulkan kode Huffman lebih cepat 45% dari pada kode ASCII. Dengan demikian, rata-rata perbedaan menggunakan kode Huffman dan kode ASCII adalah 45%. Kode Huffman lebih menghemat waktu.

4 Kesimpulan

Dalam proses penyandian (enkripsi) *password email* menggunakan kode Huffman berhasil diterapkan pada algoritma RSA. Penerapan algoritma RSA pada Matlab juga berhasil dilakukan. Perbandingan rata-rata menggunakan kode Huffman dan kode ASCII pada RSA adalah 45% , sehingga dapat diketahui bahwa kode Huffman lebih cepat dalam menyandikan *password*.

Daftar Pustaka

[1] A.Saha, N.Manna. 2007. *Digital Principles and Logic Design*. Infinity Science Press LLC. New Delhi.
[2] Gates, E. 2014. *Introduction to Basic Electricity and Electronics Technology*. Delmar Cencage Learning.
[3] Muhsetyo, G. 1995. *Pengantar Ilmu Bilangan*. Malang: IKIP.
[4] Rinaldi, M. 2005. *Matematika Diskrit*. Edisi Tiga. Bandung: Informatika.
[4] Rosen, K. H. 2007. *Discrete Mathematics and Its Aplications*. Sixth Edition. New York.
[5] Salomon, D. 2003. *Data Compression*. Third Edition. California State Univercity. USA.