ISSN 2085-4552

Analisa Dan Implementasi Algoritma Enkripsi Simetris *Data Encryption Standard* (DES) Pada Raspberry Pi

Erick Fernando[1], Fachruddin[2], Dina Fitra Murad[3], Hetty Rohayani AH[4], Pandapotan S[5]

[1] Information Systems Department, School of Information Systems, Bina Nusantara University, Jakarta, Indonesia 11480
[2]Computer Science Department, STIKOM Dinamika Bangsa, Jambi, Indonesia
[3] Information Systems Department, BINUS ONLINE LEARNING, Bina Nusantara University, Jakarta, Indonesia 11480
[4]Information Technology Department, Computer science, Adiwangsa Jambi University, Jambi, Indonesia
[5]Faculty of Informatics and Electrical Engineering, Institute Technology Del, North Sumatera, Indonesia, 22381
erick.fernando001@binus.ac.id[1], fachruddin@stikom-db.ac.id[2] , dmurad@binus.edu[3], setty_mna@yahoo.com[4], siagian.p@gmail.com[5]

Diterima 28 Mei 2019
Disetujui 20 Desember 2019

***Abstract*— Tujuan artikel ini untuk menyajikan enkripsi DES pada pc mini Raspberry Pi. Implementasi ini juga bertujuan untuk menggambarkan bahwa algoritma DES ini dapat diterapkan dengan sumber daya yang kecil. Penelitian ini dilakukan dengan pendekatan eksperimental, yang melakukan proses implementasi dalam perangkat keras pc mini dan perangkat lunak xampp. Algoritma DES di implementasikan berbasis web dengan pemrograman PHP dan server web Apache dengan menggunakan inputan data teks. Hasil penelitian, bahwa algoritma DES dapat berjalan dengan baik dengan perangkat keras minimum, seperti raspberry mini pc dengan waktu yang sangat cepat dalam proses, kecepatan dalam proses dan banyak data teks dari proses. Jadi, algoritma DES dapat diadopsi secara luas untuk berbagai aplikasi dari raspberry PI mini dengan menghasilkan informasi yang kuat dalam keamanan dan keandalan.**

***Index Terms*—Enkripsi, Simetris, DES (Data Encryption Standard) , Raspberry Pi**

I. PENDAHULUAN

Perkembangan teknologi informasi begitu cepat didalam kehidupan manusia dan memberikan dampak positif dan negatif[1]. Salah satu dampak negatif yang ditimbulkan permasalahan yang dapat mengancam pengguna komputer atas privasi sebuah data sampai kepada kerahasiaan informasi [1],[2],[3]. Salah satu solusi yang dapat ditawarkan untuk dapat menjaga kerahasiaan data dengan melakukan enkripsi data[4]. Enkripsi adalah sebuah proses perubahan sebuah pesan atau data yang berupa plain text (pesan yang dapat dimengerti oleh manusia) menjadikan sebuah pesan yang berbentuk cipther text (pesan yang tidak terstruktur/acak dan tidak dimengerti oleh manusia) [4]. Penggunaan enkripsi telah banyak diterapkan dalam berbagai penggunaan berbagai kegiatan di pengiriman pesan dengan komputer diantaranya metode RSA, Blowfish, Rijndael, DES, Serpent, RC4, dll.

Penelitian ini akan membahas salah satu enkripsi yang digunakan yaitu DES (*Data Encryption Standard*). DES merupakan salah satu standar enkripsi yang diterapkan oleh *Federal Information Processing Standard* (FIPS) Amerika Serikat dan menjadi acuan dalam pembuatan enkripsi lainnya[5], [6],[7].

Proses enkripsi terkadang banyak menggunakan *resource hardware* yang dibutuhkan untuk proses pengelolaan data dan dipengaruhi dengan metode enkripsi yang digunakan. Akan tetapi perkembangan pesat *hardware* dan *software* memberikan kecepatan proses yang dilakukan komputer sehingga mempengaruhi waktu proses dilakukan.

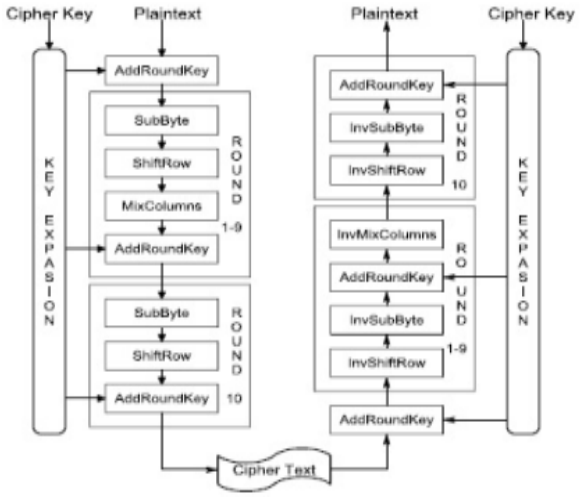
Penelitian ini melakukan implementasi algoritma DES dengan menggunakan perangkat komputer yang mini yaitu Raspberry PI. Dengan penggunaan komputer mini yang menggunakan perangkat pc kecil dengan kapasitas terbatas, diharapkan dapat membantu pengerjaan atau pengolah hal-hal yang dibutuhkan pengguna dengan data yang diinginkan.

II. LANDASAN TEORI

A. Enkripsi dan Dekripsi

Enkripsi merupakan suatu cara di dalam teknologi untuk melindungi data sensitive dengan menggunakan kombinasi kunci privat dan publik untuk menyembunyikan data tersebut[8]-[12]. Demikian pula didefinisikan oleh penulis lain, “Enkripsi adalah sebuah proses merubah *plaintext* yang “tidak tersembunyi” menjadi *ciphertext* “pesan tersembunyi” untuk mengamankan data tersebut dari pencuri data”[13].

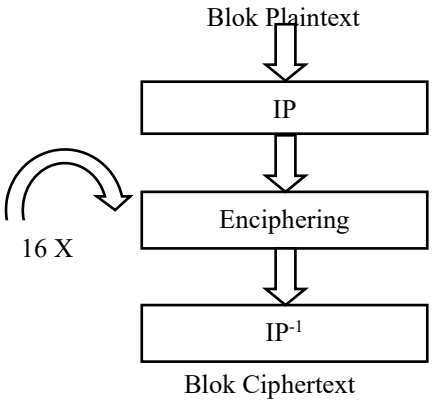
Dekripsi merupakan proses mengembalikan data rahasia ke data aslinya. Dekripsi juga merupakan satu kaidah upaya pengolahan data menjadi sesuatu yang dapat secara jelas dan tetap dengan tujuan agar dapat dimengerti oleh manusia secara langsung.



Gambar 1. *Enkripsi* dan *Dekripsi*

B. DES

DES (*Data Encryption Standard*) merupakan algoritma kriptografi yang termasuk kedalam algoritma simetri. Algoritma ini dijalankan dengan memakai satu buah kunci digunakan untuk melakukan proses enkripsi dan dekripsinya. DES juga dikatakan sebagai salah satu algoritma enkripsi yang sangat sering digunakan di dunia. DES yang sesuai dengan NIST (*National Institute of Standards and Technology*) sebagai standar pengolah informasi Federal AS[14].



Gambar 2. Skema Global Algoritma DES

Keterangan gambar :

1. Blok *plaintext* dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian di *enchipering* sebanyak 16 kali putaran. Setiap putaran menggunakan kunci internal yang berbeda.
3. Hasil *enchipering* kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP^{-1}) menjadi blok *chipertext*.

C. Raspberry Pi

Raspberry Pi adalah perangkat komputer mini seukuran kartu kredit. Raspberry Pi memiliki sistem chip Broadcom BCM2835 (SoC), yang mencakup prosesor ARM1176JZF-S 700 MHz (firmware mencakup sejumlah mode "Turbo" sehingga pengguna dapat mencoba overclocking, hingga 1 GHz, tanpa mempengaruhi garansi), GPU Video Core IV, dan awalnya dikirim dengan 256 megabyte RAM, kemudian ditingkatkan ke 512MB yang sampai sekarang telah berkembang lebih cepat [15],[16].

III. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental, di mana dilakukan implementasi dan dilakukan pengujian langsung pada perangkat keras mini pc, yaitu raspberry PI dan perangkat lunak xampp dengan bahasa pemograman PHP. Dalam proses enkripsi awal yang terjadi dalam algoritma DES, antara lain[7] [17].

1. Ubah pesan *plaintext* kedalam biner.
2. Kelompokkan biner pesan tersebut kedalam kelompok dengan masing-masing 64 bit.
3. Masukkan *key* (kunci yang telah kita/user tentukan) berupa biner.
4. Lakukan proses *initial permutation* menggunakan IP (*Initial Permutation*) sebagai berikut.

IP =

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	52	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Tabel 3 *Initial Permutation*

5. Ubah posisi *key* (kunci yang telah ditentukan pada langkah sebelumnya) kedalam 2 kelompok yaitu C dan D dengan masing-masing berisikan 28 bit dengan ketentuan urutan posisi sebagai berikut : (cara pengerjaan serupa dengan langkah 4).

Penjelasan : urutan dari baris 1 sampai baris 4 adalah C dan dari baris 5 sampai 8 adalah D untuk lebih jelasnya dapat dilihat pada tabel berikut.

C

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

D

63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Tabel 5. *Permuted Choice* 1

6. Geser posisi biner C dan D dengan ketentuan sebagai berikut :

R#	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tabel 6. *Iteration Process*

Penjelasan : pada round 1 biner digeser kekiri sebanyak 1 langkah, pada round 2 digeser kekiri sebanyak 1 langkah, pada round 3 digeser kekiri sebanyak 2 langkah dan begitu seterusya.

7. Gabungkan bagian C dan D yang telah digeser pada langkah 6 lalu ubah posisi kembali dengan aturan sebagai berikut : (cara kerja merubah posisi serupa pada langkah menukar posisi sebelumnya).

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2

41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Tabel 7. *Permuted Choice* 2

8. Kita akan meng-ekspansi data Ri-1 32 bit menjadi Ri 48 bit sebanyak 16 kali putaran dengan nilai perputaran 1<= i <=16 menggunakan Tabel Ekspansi (E) yang akan menghasilkan Ai.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Tabel 8. E BIT-*Selection Table*

9. Setiap Vektor Ai disubstitusikan kedelapan buah S-Box(Substitution Box), dimana blok pertama disubstitusikan dengan S1, blok kedua dengan S2 dan seterusnya dan menghasilkan output vektor Bi 32 bit. Menggunakan S-Box berikut :

S1 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
01	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
10	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
11	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
01	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
10	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
11	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
01	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
11	3	15	0	6	10	1	13	18	9	4	5	11	12	7	2	14

S5 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
01	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	15
10	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
01	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
10	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
11	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
01	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
10	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
11	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8 :

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
01	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
10	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
11	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Gambar 9. DES S-Boxes

10. Setelah didapatkan nilai vektor Bi, langkah selanjutnya adalah memutasikan bit vektor Bi menggunakan tabel P-Box, kemudian dikelompokkan menjadi 4 blok dimana tiap-tiap blok memiliki 32 bit data.setelah melakukan sampai proses *round* 16, maka gunakan nilai pada *round* 16 dimana nilai pada baris 5-8 menjadi nilai baris 1-4 dan nilai pada baris 1-4 menjadi nilai baris 5-8 lalu lakukan permutasi (perubahan posisi serupa pada langkah sebelumnya) dengan ketentuan sebagai berikut :

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Gambar 10. P-Box

11. Langkah terakhir adalah menggabungkan R16 dengan L16 kemudian dipermutasikan untuk terakhir kali dengan tabel Invers Initial Permutasi(IP-1).

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Gambar 11 Invers Initial Permutation

IV. ANALYSIS AND RESULTS

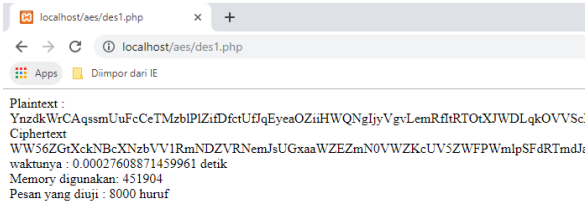
Proses analisis dengan melakukan implementasi menggunakan raspberry PI tipe B berbasis web menggunakan pemrograman PHP dan web server Apache. Proses menganalisis kinerja algoritma DES menyiapkan 6 file yang berisi kumpulan data dengan ukuran dan konten yang berbeda, maka penulis melakukan percobaan eksperimental berdasarkan kriteria kompleks dalam kriptoanalisis menurut Kaisar Siregar [12], yang memiliki 3 kriteria: waktu, memori dan data. Data yang akan digunakan sebagai input adalah 6 file dengan perincian sebagai berikut:

1. Data 1 merupakan sebuah file berukuran 8 kb yang berisikan kombinasi huruf besar (kapital) dan huruf kecil sebanyak 8.000 huruf.

2. Data 2 merupakan sebuah file berukuran 16 kb yang berisikan kombinasi huruf besar (kapital) dan huruf kecil sebanyak 16.000 huruf.
3. Data 3 merupakan sebuah file berukuran 24 kb yang berisikan kombinasi huruf besar (kapital) dan huruf kecil sebanyak 24.000 huruf.
4. Data 4 merupakan sebuah file berukuran 32 kb yang berisikan kombinasi huruf besar (kapital) dan huruf kecil sebanyak 32.000 huruf.
5. Data 5 merupakan sebuah file berukuran 40 kb yang berisikan kombinasi huruf besar (kapital) dan huruf kecil sebanyak 40.000 huruf.
6. Data 6 merupakan sebuah file berukuran 47 kb yang berisikan kombinasi huruf besar (kapital) dan huruf kecil sebanyak 48.000 huruf.

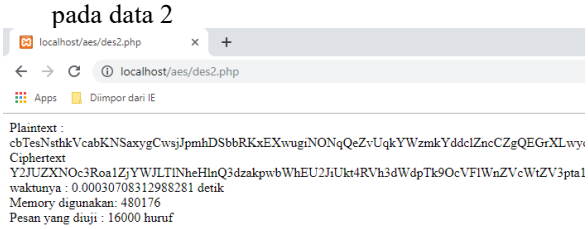
Penelitian ini akan menggunakan data 1 sebagai input kedalam algoritma DES untuk melihat berapa kecepatan dan memori yang digunakan oleh masing-masing algoritma dalam mengenkripsi data 1, data 2, data 3, data 4, data 5 dan data 6. Hal ini bertujuan untuk melihat seberapa cepat dan seberapa besar masing-masing algoritma dalam mengenkripsi sebuah pesan guna mengetahui performa masing-masing algoritma dalam mengenkripsi sebuah pesan. Proses implementasi dapat sebagai berikut:

a. Hasil percobaan menggunakan algoritma DES pada data 1



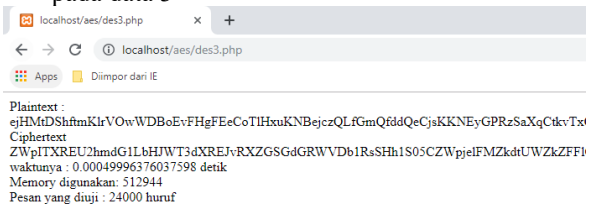
Gambar 12. percobaan menggunakan algoritma DES pada data 1

b. Hasil percobaan menggunakan algoritma DES pada data 2



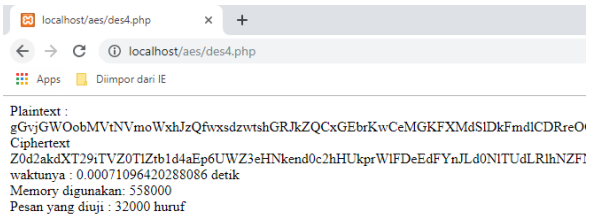
Gambar 13. percobaan menggunakan algoritma DES pada data 2

c. Hasil percobaan menggunakan algoritma DES pada data 3



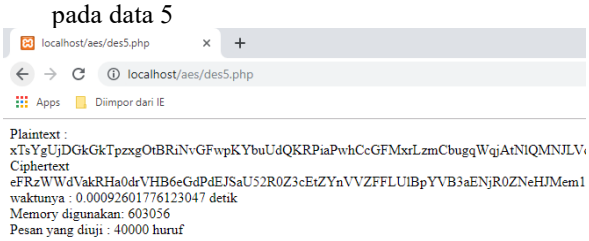
Gambar 14. percobaan menggunakan algoritma DES pada data 3

d. Hasil percobaan menggunakan algoritma DES pada data 4



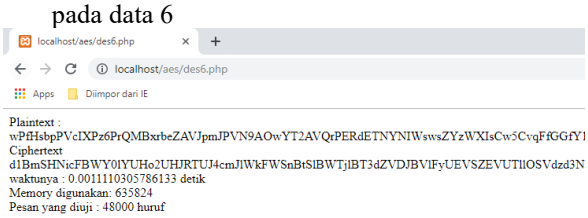
Gambar 15. percobaan menggunakan algoritma DES pada data 1

e. Hasil percobaan menggunakan algoritma DES pada data 5



Gambar 16. percobaan menggunakan algoritma DES pada data 5

f. Hasil percobaan menggunakan algoritma DES pada data 6



Gambar 17. percobaan menggunakan algoritma DES pada data 6

Dari hasil pengujian implementasi algoritma DES dapat dilihat pada tabel berikut ini:

Tabel 1. Hasil pengujian

Data	Kecepatan (detik)	Penggunaan memori (byte)
Data 1	0.00027608	451.904
Data 2	0,00030708	480.176
Data 3	0.00070881	486.752
Data 4	0.00056099	558.000
Data 5	0.00074696	603.056
Data 6	0.00064802	635.824

V. KESIMPULAN

Penelitian yang dilakukan menghasilkan bahwa implementasi algoritma DES pada raspberry PI yang merupakan perangkat mini pc dapat dilakukan dengan baik. Proses ini dilakukan pada Ciphertext dari huruf yang sama dalam plaintext sehingga menghasilkan

output huruf yang berbeda. Semua ini menunjukkan bahwa pesan (*plaintext*) memiliki huruf yang sama tetapi tidak selalu akan menghasilkan output yang sama sehingga sulit bagi kita untuk menebak pesan hanya dengan mengetahui output (*ciphertext*) saja tanpa mengetahui kunci pesan. Proses ini juga bisa dengan aplikasi kecepatan tinggi secara real time. Penggunaan memori dalam algoritma DES berbanding lurus dengan jumlah pesan yang diuji di mana memori yang dibutuhkan akan lebih besar seiring dengan ukuran pesan yang diuji. Jadi algoritma DES dapat digunakan untuk perangkat keras komputer mini seperti raspberry PI dengan kepraktisan yang kuat dalam keamanan dan keandalan informasi serta kecepatan tinggi secara real time.

VI. TERIMAKASIH

Ucapan terima kasih kepada mahasiswa saya Yoga pratama dalam pengolahan data dan membantu menglakukan analisis dalam penelitian ini

REFERENSI

1. G. Chaitanaya, B. Keerthi, A. Saleem, A. T. Rao, and K. T. P. S. Kumar, "An Image Encryption and Decryption using Chaos Algorithm," *IOSR J. Electron. Commun. Eng. Ver. II*, vol. 10, no. 2, pp. 2278–2834, 2015.
2. K. Wu, Y. Zhang, W. Cui, and T. Jiang, "Design and implementation of encrypted and decrypted file system based on USBKey and hardware code," *AIP Conf. Proc.*, vol. 1839, no. May, 2017.
3. M. G. Michael and K. Michael, *Uberveillance and the social implications of microchip implants : emerging technologies*. 2014.
4. W. Stallings, *Cryptography and Network Security (2Nd Ed.): Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.
5. Y. R. A. Kannan, S. A. Prasad, and P. Varalakshmi, "Cognitive symmetric key cryptographic algorithm," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 85, no. PART 2, pp. 50–60, 2012.
6. T. S. Ruprah, "Advance Encryption and Decryption Technique using Multiple Symmetric Algorithm," *J. Inf. Secur. Res.*, vol. 7, no. 2, pp. 62–68, 2016.
7. Kammer Raymond G, "Data Encryption Standard (DES)". *National Institute Of Standards And Technology,*1999,
8. S. P. Singh and R. Maini, "Comparison of Data Encryption Algorithms," *Int. J. Comput. Sci. Commun.*, vol. 2, no. 1, pp. 125–127, 2011.
9. S. M. Seth and R. Mishra, "Comparative Analysis Of Encryption Algorithms For Data Communication," vol. 4333, pp. 292–294, 2011.
10. G. Berad, A. Jaggi, and V. Jagadale, "REVIEW ON IMPLEMENTATION OF AES ALGORITHM FOR," no. 2, pp. 75–78, 2016.
11. Shraddha Dadhich "Performance Analysis of AES and DES Cryptographic Algorithms on Windows & Ubuntu using Java". International Journal of Computer Trends and Technology (IJCTT) V35(4):179-183, May 2016. ISSN:2231-2803. www.ijcttjournal.org.Seventh Sense Research Group.
12. Mahajan, P., & Sachdeva, A. A Study of Encryption Algorithms AES, DES and RSA for Security,2013.
13. Singh, Simar Preet and Raman Maini, *Comparison Of Data Encryption Algorithms. International Journal of Computer Science and Communication.* 2011.
14. Rifkie Primartha, *Penerapan Enkripsi Dan Dekripsi File Menggunakan Algoritma Data Encryption Standard (DES)*. Jurnal Sistem Informasi (JSI), 2011.
15. E. Fernando, "Automatisasi Smart Home Dengan Raspberry Pi Dan Smartphone Android," *Konf. Nas. Ilmu Komput.*, vol. 1, no. December 2014, pp. 1–5, 2014.
16. E. Fernando and Derist Touriono, Experimental Model Nas Dan Cloud Drive Berbasiskan Raspberry-Pi, Jambi: Jurnal Jurnal Processor, pp. 616-621,2017.
17. Ahmad Shofi., Wiyanto., Sulistiyo., 2016, *Enkripsi Dan Deskripsi Dengan Metode Data Encryption Standard (Des) Dengan Menggunakan Bahasa Pemrograman PHP*. UNIROW.