PENGAMANAN DATA USER LOGIN DENGAN ALGORITMA KRIPTOGRAFI TEA DAN NOTIFIKASI SMS PT. TELKOM INDONESIA DIVISI HIGH SPEED INTERNET.

Ulil Abshor[1)], Ir. Siswanto, M.M [2)]
[1]Program studi, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : ulil.abshor93@gmail.com [1)], siswantobl@budiluhur.ac.id [2)]

Abstrak

Aplikasi pengamanan data user login ini dirancang untuk mengamankan data user login karyawan pada PT. TELKOM Indonesia TBK, Divisi HSI pada aplikasi Minitools yang berisi data dari nomor internet pelanggan. Keamanan data user logni sangat rentan terhadap pencurian dari berbagai pihak yang tidak bertanggung jawab yang akan berimbas pada manipulasi data pelanggan di aplikasi Minitools tersebut. Permasalahan tersebut dapat dihadapi dengan menambahkan aplikasi untuk mencegah pihak yang tidak bertanggung jawab dapat membaca password dari data user login di database. Selain itu dengan menambahkan fitur notifikasi SMS untuk mencegah pihak yang tidak bertanggung jawab bebas menggunakan data user login untuk mengakses dan merubah data pelanggan pada aplikasi Minitools. Dalam penulisan ini algoritma yang digunakan dalam kriptografi, yaitu algoritma kriptografi Tiny Encryption Algorithm (TEA). Untuk notifikasi SMS sendiri, dibangun menggunakan modul Gammu sebagai software bantu khusus SMS Gateway, PHP, dan juga modem Wavecom GSM M1306B. Hasil dari pengujian kriptografi ini, data password dapat diamankan berupa teks tersandi (ciphertext). Dengan adanya aplikasi pengamanan data user login kriptografi dan notifikasi SMS untuk sistem login ini maka akan membuat sistem informasi lebih aman karena password akan dienkrip dan kode otentikasi untuk login ke aplikasi Minitools dikirimkan ke nomor handphone user. Dari hasil percobaan didapatkan rata-rata durasi proses encrypt adalah 0.08361488 ms dan durasi proses descrypt adalah 0.166120556 ms. Untuk pengiriman sms rata-rata durasi adalah tidak lebih dari 10 second

Kata kunci : Kriptografi, Minitools, *Login*, SMS *Gateway*, Gammu, *PHP*

1. PENDAHULUAN

IndiHome merupakan layanan internet dari Telkom. Menggunakan teknologi Fiber Optik (FO) IndiHome mampu menyediakan koneksi internet yang lebih. Teknologi fiber merupakan media yang tidak diragukan untuk menyediakan *bandwidth* dengan jumlah besar. Ini karena tidak dipengaruhi interferensi gelombang elektromagnetik.

Divisi HSI sendiri memiliki pelanggan yang cukup banyak, dimana pada setiap jaringan memungkinkan akan terjadinya kesalahan. Dikarenakan adanya masalah tersebut, maka solusi yang tepat adalah dengan memonitor gangguan internet yang masuk dari pelanggan , lalu kemudian gangguan dilaporkan kepada para teknisi di lapangan untuk memperbaiki gangguan yang terjadi pada jaringan sisi pelanggan. Berdasarkan permasalahan yang ada, PT. TELKOM TBK, Divisi HSI membutuhkan aplikasi yang mampu memberikan hasil audit penanganan gangguan jaringan *High Speed Internet* yang cepat dan tepat serta bisa diakses oleh *user* di internal Telkom.

Minitools yang dikembangkan oleh Pihak PT. TELKOM Indonesia TBK, memiliki Fungsi umum yakni untuk mengetahui Data Pemakaian (*Usage*) pelanggan, melihat status Persiapan untuk Pasang Baru, mengetahui Status Nomor Internet Pelanggan dan melihat data IP pelanggan. Dengan adanya aplikasi Minitools ini mampu membantu pihak PT. TELKOM Indonesia TBK, dalam mengelola data dari nomor internet pelanggan dan menampilkannya dalam laporan pada aplikasi Minitools.

Karena banyaknya pihak yang kurang bertanggung jawab ingin bebas mengakses aplikasi Minitools, dan merubah data pada aplikasi Minitools. Maka keamanan data *user login* kru HSI pada aplikasi Minitools menjadi sangat rentan terhadap pencurian oleh pihak yang kurang bertanggung jawab. Permasalahan tersebut dapat diatasi dengan menambahkan aplikasi untuk mencegah pihak yang tidak bertanggung jawab dapat membaca *password* dari data *user login* di *database*. Selain itu dengan menambahkan pengiriman sms kode otentikasi pada saat melakukan *login* ke handphone user. Dengan itu, diharapkan tidak adanya lagi pencurian data user login kru HSI pada aplikasi Minitools dan pemakaian aplikasi Minitools untuk karyawan/kru PT. Telkom Indonesia sesuai dengan peran divisi masing-masing.

Tiny Encryption Algorithm (TEA) adalah algoritma sandi yang diciptakan oleh David Wheeler dan Roger Needham dari Computer Laboratory, Cambridge University, England pada bulan November tahun 1994.

Sistem penyandian TEA menambahkan fungsi matematik berupa penambahan dan pengurangan sebagai operator pembalik selain XOR. Pergeseran dua arah (ke kiri dan ke kanan) dimaksudkan agar

semua bit kunci dan data bercampur secara berulang-ulang. Algotima TEA memproses 64-bit input sekali waktu pemrosesan dan menghasilkan 64-bit output. TEA menyimpan 64-bit input tersebut kedalam L0 dan R0 yang masing masing berjumlah 32-bit. Sedangkan 128-bit kunci disimpan kedalam k(0), k(1), k(2), dan k(3) yang masing masing berisi 32-bit. eknik ini diharapkan cukup dapat mencegah penggunaan teknik *exshautive search* secara efektif. Hasil outputnya akan disimpan dalam L16 dan R16.

Berikut beberapa langkah penyandian dengan algoritma TEA dalam dua ronde (satu cycle) :

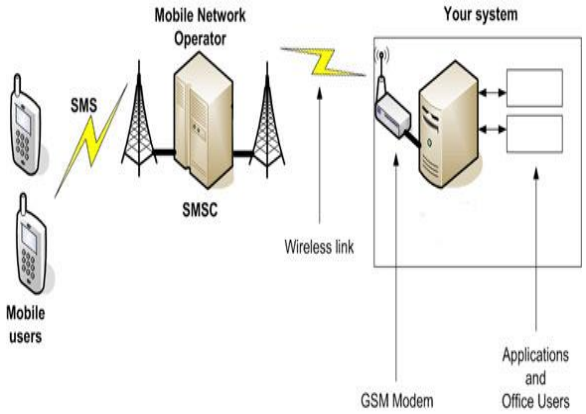
a. Pergeseran (shift)
   Blok teks terang yang masing masing sebanyak 32-bit pada kedua sisi yang akan digeser kekiri sebanyak 4 kali dan kemudian digeser ke kanan sebanyak 5 kali.
b. Penambahan
   Setelah proses pergeseran kekiri dan kekanan, maka variabel Y dan Z akan ditambahkan dengan kunci k(0)-k(3). Sedangkan untuk variabel Y dan Z awal akan ditambahkan dengan sum (delta).
c. Peng-XOR-an
   Selanjutnya adalah proses peng-XOR-an dengan rumus untuk satu round adalah sebagai berikut :

y[i] = y + (((ROL4 z) + K0) ^ (z + sum) ^ ((ROL5 z) + K1))

z[i] = z + (((ROL4 y) + K2) ^ (y + sum) ^ ((ROL5 y) + K3))

   Dalam hal ini sum= sum+delta. Hasil penyandian dalam satu cycle satu blok teks 64-bit menjadi 64-bit teks sandi adalah dengan cara menggabungkan variabel y dan variabel z. Untuk proses penyandian pada cycle selanjutnya, variabel y dan variabel z ditukar posisinya, sehingga y1 menjadi z1 dan z1 menjadi y1 lalu dilanjutkan proses seperti langkah-langkah diatas tadi sampai dengan 16 cycle (32 round).
d. *Key Schedule*
   KeySchedule pada algoritma TEA sangat sederhana. Untuk kunci k(0) dan k(1) konstan digunakan untuk round ganjil saja, sedangkan untuk kunci k(2) dan k(3) konstan digunakan untuk round genap. [3]

Cara kerja SMS *gateway* pada dasarnya hampir sama dengan mengirimkan SMS melalui *handphone* pada umumnya. Hanya saja, bedanya adalah perangkat pengirimnya bukan lagi *handphone*, tetapi modem GSM. Dan moden inilah yang dikendalikan oleh PC menggunakan aplikasi SMS *gateway* yang akan dibuat. [1]



Gambar 1: Cara Kerja SMS Gateway

GAMMU (GNU *All Mobile Management Utilities*) merupakan *software* yang digunakan sebagai *tool* untuk mengembangkan aplikasi berbasis SMS *Gateway*, cukup mudah untuk diimplementasikan, dan juga tidak berbayar. Kelebihan GAMMU dari tool SMS *gateway* lainnya adalah :

a. GAMMU dapat dijalankan di sistem operasi Linux maupun Windows.
b. Banyak *device* yang kompatibel di GAMMU.
c. GAMMU menggunakan *database* MySQL untuk menyimpan SMS yang ada pada kotak masuk (*inbox*) maupun untuk mengirim pesan, sehingga dapat dibuat *interface* yang berbasis web maupun desktop.
d. Baik kabel data USB maupun serial, semuanya kompatibel di GAMMU. [1]

2. METODE PENILITIAN

Dalam penyusunan laporan tulisan ilmiah ini telah dilakukan penelitian untuk memperoleh fakta dan juga data-data yang diperlukan. Adapun metode yang digunakan adalah metode waterfall dengan langkah – langkah sebagai berikut :

a. Studi kasus. Menganalisis masalah, kebutuhan, keperluan, dan penggunaan apa saja yang akan diperlukan untuk pengamanan data *user login* di PT. TELKOM Indonesia TBK, Divisi High Speed Internet
b. Metode pengumpulan data yang digunakan, diantaranya yaitu :
   1) Perencanaan, mengidentifikasi masalah-masalah keamanan data *user login* pada aplikasi Minitools di PT. TELKOM Indonesia TBK, Divisi High Speed Internet.
   2) Penelitian lapangan, yaitu melakukan observasi atau praktek lapangan secara langsung di perusahaan terkait guna mendapatkan data yang akurat dan dapat dipertanggung jawabkan keabsahannya.

Adapun teknik pengumpulan data yang digunakan yaitu:

a) Studi lapangan, yaitu penelitian langsung di PT. TELKOM Indonesia TBK, Divisi *High Speed Internet* (HSI) untuk mendapatkan data serta informasi yang dibutuhkan.
b) Pengamatan, yaitu teknik pengumpulan data dengan mengamati langsung cara kerja dari aplikasi Minitools PT. TELKOM Indonesia TBK, Divisi High Speed Internet.
c) Metode wawancara, ialah proses tanya jawab langsung kepada orang yang memahami dan mengetahui secara langsung tentang permasalahan yang sedang diamati .

c. Studi Literatur, Mempelajari referensi atau sumber-sumber yang berkaitan dengan algoritma kriptografi TEA, SMS, SMS *Gateway*, *GAMMU*.
d. Desain Sistem, membuat desain system yang akan dibuat, dari desain awal hingga akhir agar memudahkan dalam merelisasikan Aplikasi SMS Gateway pada form *login* aplikasi Minitools.
e. Implementasi, mengimplementasikan rancangan yang telah dibuat pada tahap perancangan sistem ke dalam perangkat lunak komputer dengan menggunakan bahasa pemrograman PHP.
f. Ujicoba program, menguji kinerja program, apakah program berjalan dengan baik atau belum. Jika belum, maka akan dilakukan perbaikan pada tahap implementasi.
g. Melakukan Simulasi, kegiatan simulasi berupa pengujian program secara nyata yang melibatkan kru PT.TELKOM Indonesia TBK, Divisi High Speed Internet.
h. Dokumentasi, melakukan penulisan hasil sistem yang telah dibangun kedalam sebuah laporan.

3. HASIL DAN PEMBAHASAN

3.1. *Tiny Encryption Algorithm* (TEA)

TEA adalah fiestel cipher yang menggunakan modus operasi XOR, ADD dan SHIFT. TEA menggunakan 64 bits block size dan menggunakan 128-bits key dan melakukan 32 putaran proses yang sama. TEA adalah cipher yang menggunakan iterasi dengan variable i, tiap putaran mempunyai input y [i-1] dan z [i-1] yang didapat dari putaran sebelumnya. Untuk subkey K[i] didapatkan dari 128 bits key dan menggunakan delta (∂). Untuk delta ini didapat dari rasio emas (golden number ratio) untuk memastikan subkey sulit diketahui.

Nilai dari delta (rasio emas) didapat dari fungsi berikut ini :

$\partial = (\sqrt{5} - 1) * 231 = 9E3779B9$ (dalam bentuk hexadecimal)

Pseudo code proses *encrypt* algoritma TEA adalah seperti pada gambar 2 :

```
1   START
2       INSERT plaintext
3       INSERT secret key (16 chr)
4       DO key schedule    //split key menjadi 4
    subkey [K0 –K3]
5       i = 1, str = panjang dari plaintext
6       SPLIT (plaintext/8 chr ) ← block//text
    dikelompokan per 8 chr
7       IF ( i ≤ str )
8       SPLIT (block / 4 chr ) ← P
9           P = y,z
10          delta = 9E3779B9, n = 1, sum  =
            delta
11          IF ( n ≤ 32 )
12          y+(((ROL4 z)+K0) XOR (z+sum)
            XOR ((ROR5 z) + K1)) ← y
            z+(((ROL4 y)+K2) XOR (y+sum)
13          XOR ((ROR5 y) + K3)) ← z
            n = n + 1, sum = sum + delta
        ELSE
14              i = i + 8
15              JOIN all cipher ←  ciphertext
16          ENDIF
17      ELSE
            PRINT ciphertext
18      ENDIF
19  END
20
```

Gambar 2: Pseudo code encrypt TEA

Sedangkan untuk *pseudo code decrypt TEA* seperti pada gambar 3 berikut :

```
1
2   START
3       INSERT ciphertext
4       INSERT secret key (16 chr)
        DO key schedule      //split key menjadi 4
5   subkey [K0 –K3]
6       i = 1, str = ciphertext length
        SPLIT ciphertext/8 chr ← block      //text
7   dikelompokan per 8 chr
8       IF ( i ≤ str )
9           SPLIT block / 4 chr ← C
            C = y,z
10          delta = 9E3779B9, n = 1, sum =
11  C6EF3720
12          IF ( n ≤ 32 )
                z-(((ROL4 y)+K2) XOR (y+sum)
                XOR ((ROR5 y) + K3)) ← z
13              y-(((ROL4 z)+K0) XOR (z+sum)
                XOR ((ROR5 z) + K1)) ← y
                n = n + 1, sum = sum - delta
14          ELSE
15              i = i + 8
16              JOIN all plain ←  plaintext
17          ENDIF
18      ELSE
19          PRINT plaintext
20      ENDIF
21  END
22
```

Gambar 3 : Pseudo code decrypt TEA

Contoh perhitungan proses *encrypt TEA* satu putaran dengan :

Plaintext = uL1l4bSh
Secret key = T3LK0M1ND0N3S1@!

Proses subtitusi key :

T	3	L	K	0	M	1	N	D	0	N	3	S	1	@	!
a		b		c		d		e		f		g		h	

⇓

@	!	T	3	S	1	L	K	N	3	0	M	D	0	1	N
h		a		g		b		f		c		e		d	

Proses *subkey* :

@	!	T	3	S	1	L	K	N	3	0	M	D	0	1	N
K0				K1				K2				K3			
Bentuk ACII (*hexadecimal*) :															
40215433				53314C4B				4E33304D				4430314E			

Split *plaintext* menjadi y dan z :

u	L	1	l	4	b	S	h
75	4C	31	6C	34	62	53	68
y				z			

=> y = 754C316C
z = 34625368

y= 754C316C (11101010100110000110001011011 00)
z= 34625368 (11010001100010010100110110100 0)
K0=40215433(10000000010000101010100001100 11)
K1=53314C4B(10100110011000101001100010010 1)
K2=4E33304D(10011100011001100110000010011 0)
K3 K3=4430314E(10001000011000000110001010011 10)

Proses penghitungan encrypt TEA :

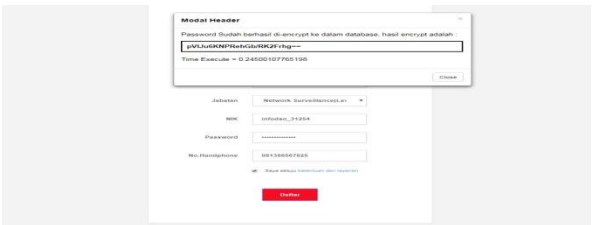
y[1] = y + (((ROL4 z) + K0) ^ (z + sum) ^ ((ROR5 z) + K1))
y[1]= 754C316C + (((ROL4 34625368) + 40215433) XOR (34625368 + 9E3779B9) XOR ((ROL5 34625368) + 53314C4B))
y[1]= 11101010100110000110001011011 00 +
(((ROL411101010100110000110001011011 00) +
10000000010000101010100001100 11)
XOR(11010001100010010100110110100 0+
10011110001101110111100110110 0111)XOR ((ROL5
11010001100010010100110110 0000) +
10100110011000101001100010010 11))
y[1]= 11101010100110000110001011011 00 +
((10101001100001100001011011001 110 +
10000000010000101010100001100 11)XOR
(11010010100110011100110100100001) XOR
((01000110100011000100101001 1011) +
10100110011000101001100010010 11))
y[1]= 11101010100110000110001011011 00 +
(10010100111001000110101100000001 XOR
11010010100110011100110100100001 XOR
11001001101010001011110111001 10)
y[1]= 11101010100110000110001011011 00+
10001010101001111110001100 0110
y[1]= 10010111111101100010101000110010
y[1]= 97F62A32 // hexadecimal
z[1] = z + (((ROL4 y1) + K2) ^ (y1 + sum) ^ ((ROR5 y1) + K3))
z[1]= 34625368 + (((ROL4 97F62A32) + 4E33304D) XOR (97F62A32 + 9E3779B9) XOR ((ROR5 97F62A32) + 4E444949))
z[1]= 11010001100010010100110110100 0 + (((ROL4
10010111111101100010101000110010) +
10011100011001100110000010011 01) XOR
(11010001100010010100110110100 0 +
10011110001101110111100110111 001) XOR ((ROR5
10010111111101100010101000110010) +
10001000011000000110001010011 10))
z[1]= 11010001100010010100110110100 0 +
((01111111011000101010001100101001 +
10011100011001100110000010011 01) XOR
(11010010100110011100110100100001) XOR
(10010100101111111011000101010001 +
10001000011000000110001010011 10))
z[1]= 11010001100010010100110110100 0 +
(11001101100101011110100110111 0110 XOR
11010010100110011100110100100001 XOR
11011000111011111111000101001 1111)
z[1]= 11010001100010010100110110100 0 +
11000111111000111111111000110010 00
z[1]= 11111100010001100101000000110000
z[1]= FC465030 // hexadecimal

Jadi untuk hasil *encrypt* 1 putaran TEA dengan *Plaintext* = uL1l4bSh dan *Secret key* = T3LK0M1ND0N3S1@! adalah 97F62A32FC465030 (dalam bentuk *hexadecimal*).

Pada bagian ini akan dijelaskan langkah-langkah dalam proses pengujian mulai dari tahap pengujian *register* , proses *encrypt password*, lalu pengujian *login* dengan notifikasi sms kedalam aplikasi yang telah dibuat. Pengujian aplikasi pengamanan data *user login* diawali dengan melakukan *register* di form *register* yang telah disediakan.

Gambar 4: Tampilan proses daftar

Munculnya *alert* notifikasi pada sisi atas halaman web yang memberikan informasi bahwa *password* telah berhasil dienkrip dan hasil dari proses *encrypt* seperti pada gambar 5 di bawah :



Gambar 5: Tampilan notifikasi hasil proses *encrypt*

3.2 Hasil Pengujian

Berikut adalah hasil pengujian dari beberapa proses :

Tabel 1: Tabel hasil uji coba proses encrypt

Pengujian ke-	Nama User	Tipe	Panjang *Password* as *Plaintext (character)*	Durasi (ms)
1	ARIE NALA CANDRA	huruf (A-Z, a-z), Angka (0-9)	14	0.145001984
2	MAOLANA SWARGOASTO	huruf (A-Z, a-z)	14	0.143002129
3	RIZKY TRI ANUGRAH	huruf (A-Z, a-z), Angka (0-9)	16	0.149901867
4	BAYU SEPTI K	huruf (A-Z, a-z), Angka (0-9)	9	0.0330019
5	MUADZIN ARZI	huruf (A-Z, a-z), Angka (0-9)	10	0.050003052
6	SURYO PRASETYO	huruf (A-Z, a-z)	11	0.057003021
7	ULIL ABSHOR	huruf (A-Z, a-z), Angka (0-9)	12	0.049003124
8	BUDI SANTOSO	huruf (A-Z, a-z), Angka (0-9)	12	0.042001963
Rata-rata durasi proses encrypt				0.08361488

Bahwa dengan jumlah karakter yang berbeda proses enkripsi tetap bisa dilakukan dengan waktu rata-rata 0,08361488 ms. Sehingga dapat dikatakan bahwa proses enkripsi password ke database bekerja dengan baik. Dengan waktu tersebut maka user tidak membutuhkan waktu yang lama pada proses enkripsi password.

Tabel 2: Tabel hasil uji coba proses pengiriman sms

Pengujian ke-	Nama User	No. Handphone	Durasi pengiriman sms dr profider TELKOMSEL ke- (s)		
			XL	INDOSAT	TELKOMSEL
1	ARIE NALA CANDRA	0813-8856-7825			5.107293
2	MAOLANA SWARGOASTO	0851-0268-8602			6.510487
3	RIZKY TRI ANUGRAH	0877-7731-4044	8.43513		
4	BAYU SEPTI K	0812-9662-0077			9.182526
5	MUADZIN ARZI	0877-8177-3659	9.312533		
6	SURYO PRASETYO	0815-8615-2001		8.390249	
7	ULIL ABSHOR	0812-4141-0915			5.178296
8	BUDI SANTOSO	0856-9106-4858		8.613642	
Rata-rata durasi pengiriman sms ke user			8.873832	8.501945	6.49465

Diketahui bahwa, meskipun *profider* yang di gunakan oleh user berbeda-beda, proses pengiriman tetap bisa dilakukan dengan waktu rata-rata 6 - 8 detik. Sehingga dapat dikatakan bahwa proses pengiriman sms kode otentikasi ke *user* bekerja dengan baik. Dengan waktu tersebut maka *user* tidak membutuhkan waktu yang lama untuk mendapatkan kode otentikasi dan menyelesaikan proses login.

Tabel 3: Tabel hasil uji coba proses decrypt

Pengujian ke-	Nama User	Tipe	Panjang *Ciphertext (character)*	Durasi (ms)
1	ARIE NALA CANDRA	Kode ASCII (A-Z, a-z, 0-9, symbols)	14	0.157003975
2	MAOLANA SWARGOASTO	Kode ASCII (A-Z, a-z, 0-9, symbols)	14	0.197011948
3	BUDI SANTOSO	Kode ASCII (A-Z, a-z, 0-9, symbols)	12	0.166008949
4	RIZKY TRI ANUGRAH	Kode ASCII (A-Z, a-z, 0-9, symbols)	16	0.160889728
5	BAYU SEPTI K	Kode ASCII (A-Z, a-z, 0-9, symbols)	9	0.143020887
6	MUADZIN ARZI	Kode ASCII (A-Z, a-z, 0-9, symbols)	10	0.157008886
7	SURYO PRASETYO	Kode ASCII (A-Z, a-z, 0-9, symbols)	11	0.158009052
8	ULIL ABSHOR	Kode ASCII (A-Z, a-z, 0-9, symbols)	12	0.190011024
Rata-rata durasi proses decrypt				0.166120556

Proses dekripsi tetap bisa dilakukan dengan waktu rata-rata 0,166120556 ms. Sehingga dapat dikatakan bahwa proses enkripsi password ke *database* bekerja dengan baik.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pembahasan serta uji coba sistem dapat disimpulkan sebagai berikut :

a. Pegamanan password dapat diamankan dengan algortima kriptografi *Tea.*
b. Program sistem keamanan data user login pada aplikasi Minitools dengan sistem kriptografi algoritma *Tea* dan notifikasi sms telah diuji coba, sehingga program dinyatakan sudah sesuai.
c. Rata-rata durasi proses *encrypt* adalah 0.08361488 ms
d. Rata-rata durasi proses *descrypt* adalah 0.166120556 ms
e. Rata-rata durasi proses pengiriman sms dari *profider* TELKOMSEL ke sesama TELKOMSEL adalah 6.494650483 *second*
f. Rata-rata durasi proses pengiriman sms dari *profider* TELKOMSEL ke *profider* INDOSAT adalah 8.501945496 *second*
g. Rata-rata durasi proses pengiriman sms dari profider TELKOMSEL ke *profider* XL adalah 8.873831511 *second*
h. Proses pengiriman sms kode otentikasi tergantung dengan sinyal *profider* yang digunakan, sehingga proses pengiriman sms kode otentikasi bisa cepat, lama atau bahkan sms tidak diterima oleh *user*.

5. DAFTAR PUSTAKA

[1] Irmayani 2012, *Implementasi Sms Gateway Untuk Layanan Informasi Absensi Pegawai*, diakses 14 November 2017, <http://repository.usu.ac.id/handle/123456789/33822>

[2] Kumar, Kiran V. G, et al. 2015, *Design And Implementation Of Tiny Encryption Algorithm. International Journal of Engineering Research and Applications*, ISSN : 2248-9622 Vol. 5 Issue 6.

[3] Nugroho, Sandromedo C. 2013, *Algoritma Tea (Tiny Encryption Algorithm)*, diakses 16 November 2017, <https://kriptologi.wordpress.com/2008/10/03/algoritma-tea-tiny-encryption-algorithm/>

[4] Tammam, Aditya Gusti 2014, *Pengertian PseudoCode dan Contohnya*, diakses 3 Desember 2017 < https://www.academia.edu/8929745/Pengertian_PseudoCode_dan_Contohnya>