SNESTIK

Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika

https://ejurnal.itats.ac.id/snestik dan https://snestik.itats.ac.id

Informasi Pelaksanaan :
SNESTIK IV - Surabaya, 27 April 2024
Ruang Seminar Gedung A, Kampus Institut Teknologi Adhi Tama Surabaya

Informasi Artikel:
DOI : 10.31284/p.snestik.2024.5713

Prosiding ISSN 2775-5126

Fakultas Teknik Elektro dan Teknologi Informasi-Institut Teknologi Adhi Tama Surabaya
Gedung A-ITATS, Jl. Arief Rachman Hakim 100 Surabaya 60117 Telp. (031) 5945043
Email : snestik@itats.ac.id

Implementasi Mengamankan Pesan Teks Menggunakan Metode GOST (Gosundarstevenny Standard)

Mario Franko Ezra Hasiholan Ritonga, Muchamad Kurniawan, Siti Agustini
Institut Teknologi Adhi Tama Surabaya
E-Mail : muchamad.kurniawan@itats.ac.id

ABSTRACT

The increasing development of technology has made it easier for people to communicate with each other. Sometimes, some of the information sent must be kept confidential so that it is not misused by irresponsible people. This research deals with securing messages using the GOST method during information exchange. The output messages of chats converted into ciphertext served as the research materials. The research results indicated that the application to secure text messages successfully implemented encryption and decryption techniques through the GOST method. The more bit changes occurred, the more difficult the cryptographic algorithm was to solve.

Keywords: *GOST, Ciphertext, Cryptography*

ABSTRAK

Meningkatnya perkembangan teknologi pada saat ini, membawa kemudahan untuk orang-orang dapat berkomunikasi satu sama lain. Terkadang, beberapa informasi yang dikirimkan harus dirahasiakan agar tidak di salah gunakan oleh orang yang tidak bertanggung jawab. Pada penelitian kali ini memiliki tujuan dimana dalam melakukan pertukaran informasi tersebut dibutuhkan pengamanan pesan yang menggunakan metode GOST dengan output pesan berupa chatting yang telah di ubah menjadi ciphertext sebagai bahan melakukan penelitian.

Hasil yang diperoleh dari penelitian ini adalah aplikasi pengamanan pesan teks berhasil mengimplementasikan teknik enkripsi dan dekripsi dengan menggunakan metode GOST dalam

mengamankan pesan serta semakin banyak perubahan bit yang terjadi maka akan sulit algoritma kriptografi tersebut dipecahkan.

Kata kunci: Kriptografi,Gosudarstvennyi Standard (GOST), Pesan Teks,Chatting

PENDAHULUAN

Perkembangan teknologi sekarang ini memberikan kemudahan agar manusia dapat saling berkomunikasi dan mengirimkan informasi secara cepat. Namun terkadang informasi yang dikirimkan bersifat sangat penting dan harus dirahasiakan agar informasi tesebut tidak disalahgunakan oleh pihak yang tidak bertanggung jawab. Dari hal tersebut, maka dibuatlah sebuah keamanan bagi seluruh aset-asetnya dan data-data penting demi menjaga kerahasiaan informasi tersebut. Dari segi keamanan data tersebut mensyaratkan adanya sistem keamanan data yang lebih baik dalam melindungi data dari berbagai potensi ancaman,ini menjadi dasar pengembangan sistem keamanan data yang dirancang untuk melindungi data yang dikirimkan atau dikirim melalui jaringan komunikasi. [1]

Ada beberapa cara untuk mengamankan data atau pesan,diantaranya adalah dengan menggunakan teknik penyamaran data yang dikenal dengan Kriptografi.

Kriptografi adalah suatu metode penyajian pesan agar pesan dari pengirim (sender) tidak dapat dibaca oleh pihak lain kecuali (receiver) secara aman. Dalam kriptografi, data atau pesan yang dikirimkan akan disamarkan kedalam bentuk yang tidak dapat dimengerti lagi maknanya. Sehingga seandainya data tersebut dapat dibaca oleh orang lain, maka pihak yang tidak berwenang tidak akan bisa mengerti isi dari data atau pesan tersebut.

Pada kriptografi terdapat dua konsep yang sangat penting yaitu enkripsi dan dekripsi. Enkripsi atau enciphering adalah suatu metode penyandian plainteks menjadi cipherteks. Sedangkan dekripsi merupakan kebalikan dari enkripsi. Deskripsi atau deciphering merupakan suatu metode yang berfungsi mengubah chiperteks menjadi plainteks.

Oleh karena itu akan dibuat suatu aplikasi penyandian pesan teks (chatting) menggunakan algoritma GOST (Gosudarstvenny Standard). Aplikasi ini dibuat dengan menggunakan bahasa pemrograman yaitu Java.

Adapun permasalahan yang dirumuskan adalah sebagai berikut:

a. Bagaimana cara mengimplementasikan algoritma GOST ke dalam aplikasi mengamankan sebuah pesan teks (chatting)?
b. Bagaimana cara melakukan pengamanan terhadap informasi atau pesan chat yang dikirimkan sehingga informasi tersebut bisa terjaga keamanannya?

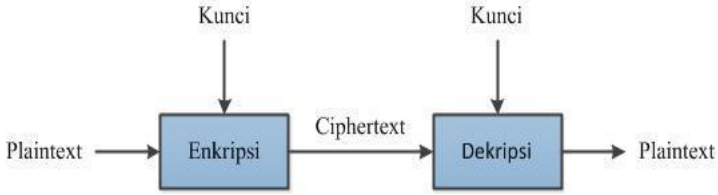
Tujuan yang diambil dari perancangan aplikasi ini adalah sebagai berikut:

a. Mengembangkan sistem aplikasi pengamanan pesan teks dengan menggunakan algoritma kriptografi GOST.
b. Mengamankan pesan teks yang dikirim agar tidak dapat diketahui oleh orang yang tidak berwenang.
c. Menghasilkan sistem aplikasi chat yang menggunakan teknik kriptografi yang mudah dimengerti dan digunakan oleh pengguna.

Pada sistem aplikasi ini hanya pesan teks (chatting) yang dapat di enkripsi dan dekripsi.

METODE

Kriptografi adalah ilmu yang mempelajari bagaimana supaya pesan atu dokumen kita aman, dan tidak bisa dibaca oleh pihak yang tidak bersangkutan. Kriptografi juga bisa digunakan untuk identifikasi pengiriman pesan dengan tanda tangan digital dan keaslian pesan dengan sidik jari digital (fingerprint). Kriptografi adalah ilmu menyembunyikan pesan, selain itu kriptografi juga merupakan ilmu yang bersandarkan pada teknik matematika yang berhubungan dengan keamanan informasi seperti kerahasiaan data, keutuhan data dan otentikasi data.



Gambar 1. Algoritma GOST

Pada gambar 1 dapat dijelaskan bahwa pesan asli (plaintext) dienkripsi melalui kunci. Hasil enkripsi adalah berupa (ciphertext) yaitu, pesan yang tidak terbaca. Untuk membuka pesan yang tidak terbaca tersebut (ciphertext) maka pesan tersebut didekripsi dengan kunci yang sama sehingga menghasilkan pesan teks yang dapat dibaca (plaintext). Pada dasarnya, kriptografi memiliki beberapa komponen, yaitu:

1. Enkripsi merupakan hal yang sangat penting dalam kriptografi sebagai pengamanan atas data yang akan dikirimkan agar rahasianya terjaga. Pesan aslinya disebut plaintext yang diubah menjadi pesan yang tidak bisa dimengerti. Enkripsi bisa juga diartikan sebagai cipher atau kode.

2. Dekripsi merupakan kebalikan dari enkripsi, dimana pesan yang telah dienkripsi akan dikembalikan menjadi pesan aslinya (plaintext). Algoritma yang digunakan untuk deskripsi tentu berbeda dengan yang digunakan untuk enkripsi.

3. Kunci adalah kunci yang dipakai untuk melakukan enkripsi dan deskripsi. Kunci terbagi menjadi dua bagian, kunci private dan kunci public.

4. Ciphertext merupakan suatu pesan yang sudah melalui proses enkripsi. Pesan yang ada pada ciphertext tidak bisa dibaca karena berisi karakter karakter yang tidak memiliki makna.

5. Plaintext atau sering disebut cleartext merupakan suatu pesan bermakna yang ditulis atau diketik dan plaintext itulah yang akan diproses menggunakan algoritma kriptografi agar menjadi ciphertext

6. Pesan bisa berupa data atau informasi yang akan dikirim atau disimpan di dalam media perekaman seperti kertas,storage, dan sebagainya.

7. Kriptanalisis dan Kriptologi: kriptanalisis (cryptanalysis) bisa diartikan sebagai analisis sandi tau ilmu untuk mendapatkan plaintext tanpa harus mengetahui kunci secara wajar. Jika suatu ciphertext berhasil menjadi plaintext tanpa menggunakan kunci yang sah, maka proses tersebut dinamakan breaking code yang dilakukan oleh cryptanalyst. Analisis sandi juga mampu menemukan kelemahan dari suatu algoritma kriptografi dan akhirnya bisa menemukan kunci atau plaintext dari ciphertext yang dienkripsi menggunakan algoritma tertentu.[2]

CHATTING

Chatting umumnya merupakan aktivitas komunikasi yang dilakukan oleh dua pengguna atau lebih aplikasi chatting dan internet. Aplikasi chatting sekarang sudah sangat canggih. Jangan hanya teks. Tentunya, aktivitas chatting kini juga bisa mengirim emoji, pesan suara, bahkan video. Chatting itu salah satu ciri dari kecanggihan teknologi informasi saat ini. Dari anak-anak hingga orang dewasa sudah tidak asing lagi dengan istilah chatting. Singkatnya, konsep obrolan adalah program yang melibatkan koneksi ke Internet bertukar pesan dari satu orang ke orang lain. Obrolan adalah bentuk komunikasi yang paling efektif dan

sata yang efektif ini sata yang efektif ini.[3]

AVALANCHE EFFECT

Avalanche effect adalah salah satu karakteristik yang menjadi acuan untuk menentukan baik atau tidaknya ketahanan suatu algoritma kriptografi (khususnya block cipher dan hash). Avalanche effect dalam kriptografi dapat dilihat ketika dilakukan perubahan kecil pada plaintext maupun key yang akan menyebabkan perubahan signifikan terhadap ciphertext yang dihasilkan. Dengan kata lain, perubahan satu bit pada plaintext maupun key akan menghasilkan perubahan banyak bit pada ciphertext. Suatu avalanche effect dikatakan baik jika perubahan bit yang dihasilkan berkisar antara 45- 60% (sekitar separuhnya, 50 % adalah hasil yang sangat baik) [4]. Hal ini dikarenakan perubahan tersebut berarti membuat perbedaan yang cukup sulit untuk kriptanalis melakukan serangan.

$$Avalanche_{-Effect(AE)} = \frac{\sum bit_{-berubah}}{\sum bit_{-total}} * 100\%$$

Gambar 2. Rumus Avalanche Effect

GOSUDARSTVENNYI STANDARD (GOST)

Algoritma GOST merupakan blok kode dari Uni Soviet, yang merupakan singkatan dari Gosudarstvennyi Standard atau standart pemerintah. GOST memiliki blok kode 64bit dengan Panjang kunci 256bit. Algoritma ini mengiterasi algoritma enkripsi sederhana sebanyak 32 putaran. Untuk mengenkripsi pertama yang harus dilakukan adalah plainteks 64bit dipecah menjadi 2 bagian, yang pertama 32 bit bagian kiri (L) dan 32 bit bagian kanan (R). subkunci (subkey) untuk putran I adalah Ki. Pada satu putaran ke-I operasinya adalah sebagai berikut [5]:

Li = R i-1
Ri = L i-1 xor f(R i-1, Ki)

Metode Gosundarstevennyi Standard digunakan untuk menyembunyikan informasi teks asli dan dibuat teks tersebut tidak tampak seperti teks aslinya. Konsep kerja pada algoritma Government Standart itu sendiri memiliki jumlah proses sebanyak 32 round (putaran) menggunakan 64bit block chiper pada setiap kali prosesnya dan 256bit kunci atau 32 karakter. Metode GOST juga menggunakan 8 buah S-BOX yang permanen dan operasi XOR serta RLS (Rotate Left Shift).

A. Proses Pembangkitan Kata Kunci

Proses Pembangkitan Kunci Kunci internal pada algoritma GOST dibangkitkan dari kunci eksternal yang diberikan oleh pengguna [3]. Pembangkitan kunci internal dilakukan dengan membagi kunci eksternal 256 bit (k1, k2, k3, k4, …, k256) ke dalam delapan bagian yang masingmasing panjangnya 32 bit. Pembagiannya adalah sebagai berikut :

K0 = (k32, ……. ,k1)
K1 = (k64, ……. ,k33)
K2 = (k96, ……. ,k65)
K3 = (k128, ……. ,k97)
K4 = (k160, ……. ,k129)
K5 = (k192, ……. ,k161)
K6 = (k224, ……. ,k193)
K7 = (k256, ……. ,k225)

B. Proses Enkripsi

Proses Enkripsi pada metode Gosundarstevennyi Standard untuk satu putaran (iterasi), adalah sebagai berikut :

1) 64 bit plainteks dibagi menjadi 2 buah bagian
32 bit, yaitu Li dan Ri.
Caranya :
Input a1(0), a2(0), ….. ,a32(0) ; b1(0), b2(0),
….. ,b32(0)
R0 = a32(0), a31(0), ….. ,a1(0)
L0 = b32(0), b31(0), ….. ,b1(0)
2) (Ri + Ki) mod 232. Hasil dari penjumlahan modulo 232 berupa 32 bit.
3) Hasil dari penjumlahan modulo 232 dibagi menjadi 8 bagian, dimana masing-masing bagian terdiri dari 4 bit. Setiap bagian dimasukkan ke dalam table S-box yang berbeda, 4 bit pertama menjadi input dari SBox 0, 4 bit kedua menjadi S-Box 1 dan seterusnya.

Tabel 1. S-Box Algoritma GOST

Tabel S-Box	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S-Box 0	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
S-Box 1	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
S-Box 2	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
S-Box 3	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
S-Box 4	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
S-Box 5	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
S-Box 6	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
S-Box 7	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

4) Hasil yang didapat dari substitusi ke S-Box kemudian digabungkan kembali menjadi 32 bit dan kemudian dilakukan RLS (Rotate Left
Shift) pergeseran ke kiri sebanyak 11 bit.
5) Ri + 1 = RLS XOR Li
6) Li + 1 = Ri sebelum dilakukan proses

Langkah nomor 2 sampai 6 dilakukan sebanyak 32 kali (putaran). Pada langkah nomor 2 penggunaan kunci dijadwalkan penggunaannya sesuai dengan putarannya.

Tabel 2. Penjadwalan Kunci Internal Enkripsi

Putaran	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Kunci Internal	K7	K6	K5	K4	K3	K2	K1	K0

Untuk putaran ke-31, langkah nomor 5 dan 6 sedikit berbeda. Langkah 5 dan 6 untuk putaran 31 adalah sebagai berikut :

R32 = R31 sebelum dilakukan proses
L32 = L31 XOR R31
Sehingga cipherteks yang dihasilkan adalah,
L32 : b(32), b(31), ….. ,b(1)
R32 : a(32), a(31), ….. ,a(1)
Chiperteks = a(1), ….. ,a(32); b(1), ….. ,b(32).

C. Proses Dekripsi

Proses dekripsi merupakan kebalikan dari proses enkripsi. Penggunaan kunci pada masing-masing putaran pada proses dekripsi adalah sebagai berikut:

Tabel 3. Penjadwalan Kunci Internal Dekripsi

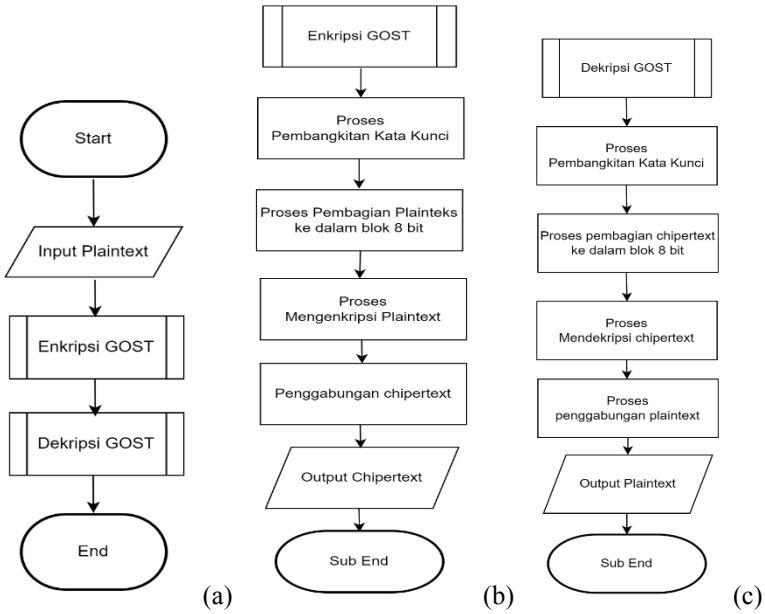
Putaran	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Kunci Internal	K7	K6	K5	K4	K3	K2	K1	K0
Putaran	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
Kunci Internal	K7	K6	K5	K4	K3	K2	K1	K0
Putaran	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Kunci Internal	K7	K6	K5	K4	K3	K2	K1	K0

Pada proses dekripsi terdapat aturan sama dengan proses enkripsi yaitu untuk langkah ke-5 dan ke-6 pada putaran ke-31 sebagai berikut :

R32 = R31 sebelum dilakukan proses
L32 = L31 XOR R31
Sehingga, plainteks yang dihasilkan pada proses
dekripsi adalah [4], L32 : b(32), b(31), ….. ,b(1) R32 : a(32), a(31), ….. ,a(1)
Plainteks = a(1), ….. ,a(32); b(1), ….. ,b(32).

D. Perancangan Sistem

Berikut ini adalah gambaran untuk flowchart sistem proses mengamankan pesan teks chatting:



Gambar 3.(a) Flowchart Proses Kriptografi GOST, (b) Flowchart subproses enkripsi, (c) Flowchart subproses dekripsi

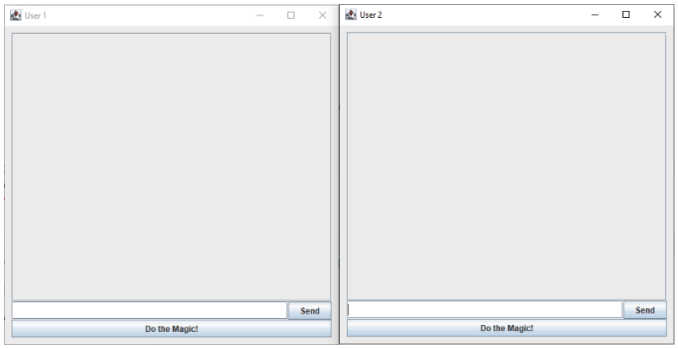
HASIL DAN PEMBAHASAN

Tampilan Utama Enkripsi dan Dekripsi

Pada gambar 4 merupakan tampilan ketika user menjalankan aplikasi enkripsi dan dekripsi pertama kali. User 1 adalah pengirim pesan sedangkan user 2 penerima pesan. Walaupun peran pengirim dan penerima pada halaman sistem ini mempunyai hak akses yang bisa digunakan setiap tampilan. Ada terdapat tombol pada menu enkripsi dan dekripsi ini, yaitu:

1. Tombol Send (Enkripsi) yang digunakan untuk mengirim pesan antara user pertama dan user kedua,sekaligus melakukan enkripsi pada pesan tersebut menjadi chipertext.

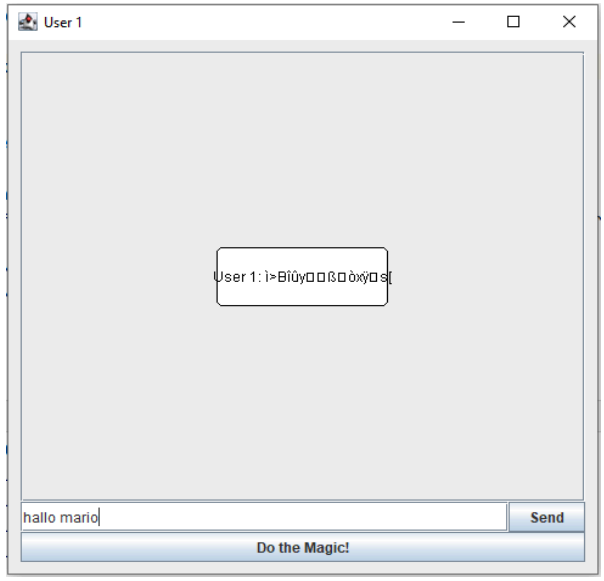
2. Tombol Do the magic (Dekripsi) yang digunakan untuk memproses pesan yang sudah di enkripsi, lalu di dekripsi agar menampilkan pesan teks yang asli (plaintext).



Gambar 4. Tampilan User 1 dan User 2

Tampilan User 1 (Pengirim)

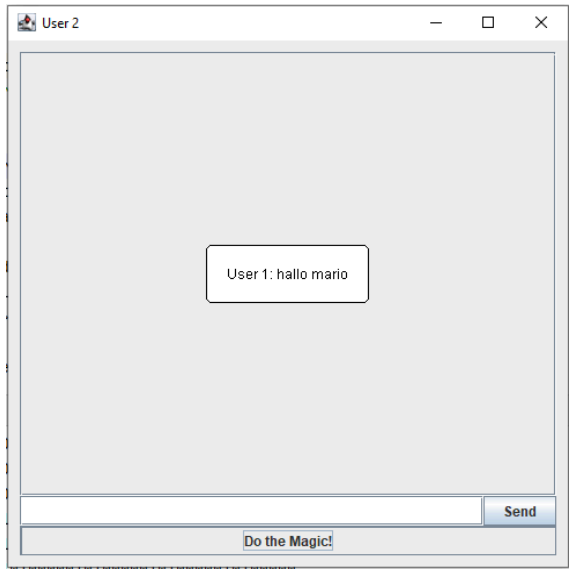
Pada gambar 5 merupakan tampilan saat akan mengiriman pesan kepada penerima. Pesan (plainteks) yang dimasukkan oleh pengirim yaitu "hallo mario". Setelah pengirim mengklik tombol send maka proses enkripsi itu akan berjalan dan menghasilkan tampilan pada halaman pengirim berupa chipertext.



Gambar 5. Tampilan Hasil User 1 sebagai pengirim

Tampilan User 2 (Penerima)

Pada gambar 6 merupakan tampilan hasil dekripsi yang telah dikirimkan oleh pengirim yang dimana sebelumnya berupa chipertext,setelah penerima mengklik tombol "Do the Magic!" maka proses dekripsi itu akan berjalan dan menghasilkan output berupa pesan asli dari pengirim dan dapat dibaca oleh penerima. Pesan asli (plainteks) yaitu "hallo mario".



Gambar 6. Tampilan Hasil User 2 sebagai penerima

PENGUJIAN SISTEM

A. Pengujian Sistem Terhadap Pesan Teks

Pada pengujian ini menggunakan parameter panjang karakter, nilai avalanche effect, dan setiap pesan teks (chatting) yang akan diujikan juga dari waktu proses enkripsi. Ada 30 pesan teks (chatingg) yang akan dibagi menjadi 3 kategori 10 Karakter, 20 Karakter dan 30 Karakter. Tujuan dari uji coba menggunakan parameter tersebut untuk mengetahui pengaruh nilai ketahanan dari pesan teks (chatting) terhadap panjang karakter dan waktu proses enkripsi.[6]

Tabel 4. Pengujian Dengan Pesan 10 Karakter

Pesan Chat	Panjang Karakter	Nilai Avalanche Effect	Waktu Komputasi (milidetik)
Apa Kabar?	10 karakter	27,34%	32 milidetik
Hati-Hati!	10 Karakter	28,12%	8 milidetik
Semangat!!	10 Karakter	26,56%	36 milidetik
LeMineral?	10 Karakter	26,56%	8 milidetik
mau minum?	10 Karakter	22,65%	33 milidetik
dimana lur	10 Karakter	23,43%	11 milidetik
lagi makan	10 Karakter	23,43%	43 milidetik
makan dmn?	10 Karakter	25%	7 milidetik
di cak har	10 Karakter	28,12%	11 milidetik
menu apa?	10 Karakter	21,09%	8 milidetik
Nilai Rata-rata	**25,23%**		

Berdasarkan hasil pengujian pada tabel 4, dapat dilihat bahwa pengamanan menggunakan metode Gosundarstevenny Standard (GOST) menghasilkan nilai rata rata Avalanche Effect sebesar 25,23% menandakan bahwa metode Government Standart (GOST)

memiliki nilai Avalanche Effect yang ***CUKUP***. Menurut sitasi pengujian Avalanche Effect dianggap sulit dibobol jika perubahan bit yang menunjukkan antara 45-60%. Perubahan kecil pada plainteks dapat berdampak ke chipertext sebab terdapat berberapa bit yang berubah yang mengakibatkan perubahan yang tidak terlalu banyak. Pada waktu nilai komputasi yang dihasilkan pesan 9 karakter memiliki waktu yang tidak terlalu signifikan. Dapat di simpulkan panjang karakter tidak terlalu berpengaruh pada waktu komputasi.

Pada pengujian ini menggunakan parameter panjang karakter, sebanyak 20 karakter dengan tambahan simbol-simbol dalam setiap pesan teks, tujuan dari penambahan simbol-simbol itu sendiri untuk mengetahui sejauh mana segi keamanan yang terdapat setiap pesan teks yang ada

Tabel 5. Pengujian Dengan Pesan 20 Karakter

Pesan Teks	Panjang Karakter	Nilai Avalanche Effect	Waktu Komputasi (milidetik)
Selamat.ulang,tahun!	20 karakter	16,66%	44 milidetik
Terima!kasih!banyak!	20 karakter	16,14%	12 milidetik
Berkarya(dengan)hati	20 karakter	17,18%	51 milidetik
Bersama=sampai akhir	20 karakter	17,70%	11 milidetik
Nikmatisetiap>momen	20 karakter	16,14%	11 milidetik
Bersyukur>atas<semua	20 karakter	21,87%	53 milidetik
Semangat!,kamu bisa!	20 karakter	17,18%	13 milidetik
Hari ini{cuaca}cerah	20 karakter	17,18%	52 milidetik
Ingatlah!aku di sana	20 karakter	18,75%	47 milidetik
Mario-Franko-Ezra-HR	20 karakter	19,79%	49 milidetik
Nilai Rata-rata	**17,85%**		

Berdasarkan hasil pengujian pada tabel 5, dapat dilihat bahwa pengamanan menggunakan metode Government Standart (Gost) tidak beda jauh dengan Tabel 4 yang menghasilkan nilai rata rata Avalanche Effect sebesar 17,85 menandakan bahwa metode Government Standart (GOST) memiliki nilai Avalanche Effect yang ***KURANG***, dikarenakan perubahan kecil pada plainteks dapat berdampak ke chipertext sebab terdapat beberapa bit yang berubah yang mengakibatkan perubahan yang tidak terlalu banyak. Pada waktu nilai komputasi yang dihasilkan pesan 20 karakter memiliki waktu yang tidak terlalu signifikan. Dapat disimpulkan panjang karakter dan penambahan simbol-simbol pada setiap pesan teks tidak terlalu berpengaruh pada waktu komputasi.

Tabel 6. Pengujian Dengan Pesan 30 Karakter

Pesan Chat	Panjang Karakter	Nilai Avalanche Effect	Waktu Komputasi (milidetik)
Selamat ulang tahun yang ke-30	30 karakter	11,71%	63 milidetik
Saya akan segera tiba di sana.	30 Karakter	11,32%	20 milidetik

Apa kabar hari ini? Apa berita	30 Karakter	12,30%	12 milidetik
Jangan lupa mengambil paket ya	30 Karakter	15,23%	8 milidetik
Semoga harimu menyenangkan ya!	30 Karakter	12,30%	9 milidetik
Kamu mau ikut bermain basket??	30 Karakter	15,23%	7 milidetik
Semangat dan jangan menyerah!!	30 Karakter	13,28%	5 milidetik
sayang, jangan pulang malam ya	30 Karakter	15,23%	3 milidetik
pekerjaan kepin adalah anjelo*	30 Karakter	12,05%	6 milidetik
barang kepin=barang kita juga.	30 Karakter	13,28%	3 milidetik
Nilai Rata-rata	**13,20%**		

Berdasarkan hasil pengujian pada tabel 6, dapat dilihat bahwa pengamanan menggunakan metode Government Standart (Gost) yang menghasilkan nilai rata rata Avalanche Effect sebesar 13,20% menandakan bahwa metode Government Standart (GOST) memiliki nilai Avalanche Effect yang KURANG walaupun sudah ada perubahan disetiap pesan namun hasil yang didapatkan jauh berbeda pada pengujian Tabel 4 dan Tabel 5. Pada waktu nilai komputasi yang dihasilkan pesan 20 karakter memiliki waktu yang tidak terlalu signifikan. Dapat di simpulkan panjang karakter tidak terlalu berpengaruh pada waktu komputasi.

B. Pengujian Sistem Terhadap Kata Kunci

Dalam pengujian ini, akan diuji kata kunci yang dimana satu karakter dalam 32bit diubah menjadi **"1".** Proses ini dilakukan sebanyak 32 kali dalam algoritma dan mencari nilai avalanche effect yang telah di tentukan berdasarkan Chipertext yang dihasilkan.[7]

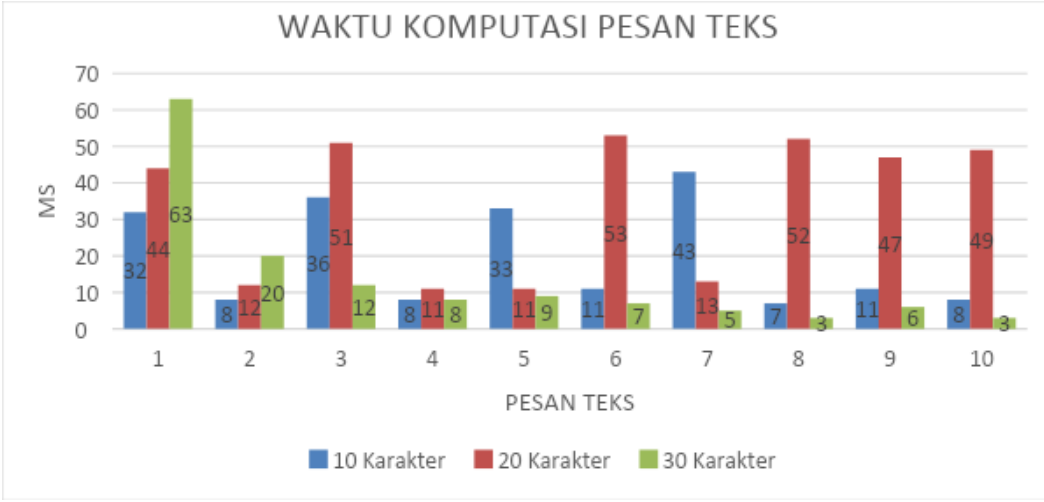
Tabel 7. Pengujian Terhadap Kata Kunci

Key	Pesan Teks	Avalanche Effect (%)
1lgoritma_GOST_Franko_Mario_2017	ITATS'17	39,06%
A**1**goritma_GOST_Franko_Mario_2017	ITATS'17	48,43%

Al**1**oritma_GOST_Franko_Mario_2017	ITATS'17	45,31%
Alg**1**ritma_GOST_Franko_Mario_2017	ITATS'17	57,81%
Algo**1**itma_GOST_Franko_Mario_2017	ITATS'17	39,06%
Algor**1**tma_GOST_Franko_Mario_2017	ITATS'17	48,43%
Algori**1**ma_GOST_Franko_Mario_2017	ITATS'17	51,56%
Algorit**1**a_GOST_Franko_Mario_2017	ITATS'17	50%
Algoritm**1**_GOST_Franko_Mario_2017	ITATS'17	46,87%
Algoritma**1**GOST_Franko_Mario_2017	ITATS'17	57,81%
Algoritma_**1**OST_Franko_Mario_2017	ITATS'17	40,62%
Algoritma_G**1**ST_Franko_Mario_2017	ITATS'17	53,12%
Algoritma_GO**1**T_Franko_Mario_2017	ITATS'17	48,43%
Algoritma_GOS**1**_Franko_Mario_2017	ITATS'17	48,43%
Algoritma_GOST**1**Franko_Mario_2017	ITATS'17	43,75%
Algoritma_GOST_**1**ranko_Mario_2017	ITATS'17	42,18%
Algoritma_GOST_F**1**anko_Mario_2017	ITATS'17	48,43%
Algoritma_GOST_Fr**1**nko_Mario_2017	ITATS'17	50%
Algoritma_GOST_Fra**1**ko_Mario_2017	ITATS'17	57,81%
Algoritma_GOST_Fran**1**o_Mario_2017	ITATS'17	53,12%
Algoritma_GOST_Frank**1**_Mario_2017	ITATS'17	56,25%
Algoritma_GOST_Franko**1**Mario_2017	ITATS'17	53,12%
Algoritma_GOST_Franko_**1**ario_2017	ITATS'17	48,43%
Algoritma_GOST_Franko_M**1**rio_2017	ITATS'17	48,43%
Algoritma_GOST_Franko_Ma**1**io_2017	ITATS'17	51,56%
Algoritma_GOST_Franko_Mar**1**o_2017	ITATS'17	50%
Algoritma_GOST_Franko_Mari**1**_2017	ITATS'17	43,75%

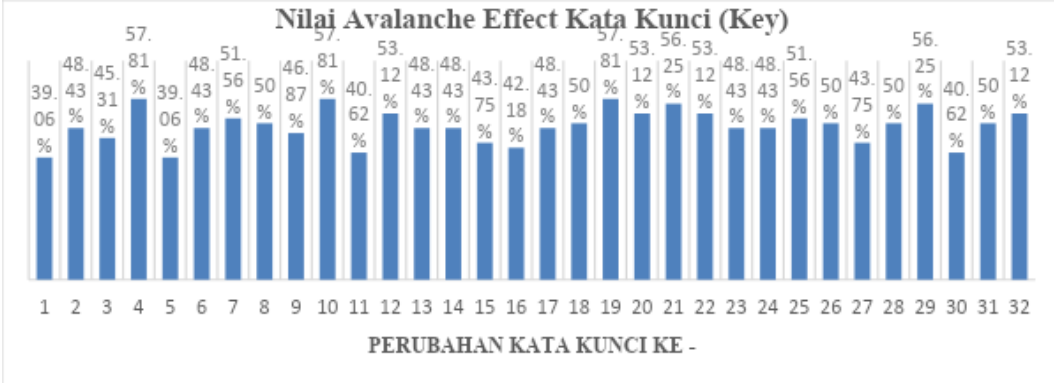
Algoritma_GOST_Franko_Mario12017	ITATS'17	50%
Algoritma_GOST_Franko_Mario_1017	ITATS'17	56,25%
Algoritma_GOST_Franko_Mario_2117	ITATS'17	40,62%
Algoritma_GOST_Franko_Mario_2027	ITATS'17	50%
Algoritma_GOST_Franko_Mario_2011	ITATS'17	53,12%
Nilai Rata-rata	**48,44%**	

Pada tabel diatas, menunjukan hasil pengujian yang diperoleh dari perubahan satu bit kata kunci *(Key)*. Dimana nilai rata-rata avalanche effect yang mendekati **48,44%** dianggap cukup. Nilai terkecil dari avalanche effect adalah **39,06%**, sedangkan nilai tertingginya adalah **57,81%.** Dari pengujian tersebut memiliki avalanche effect yang baik yakni, **48,44%.**



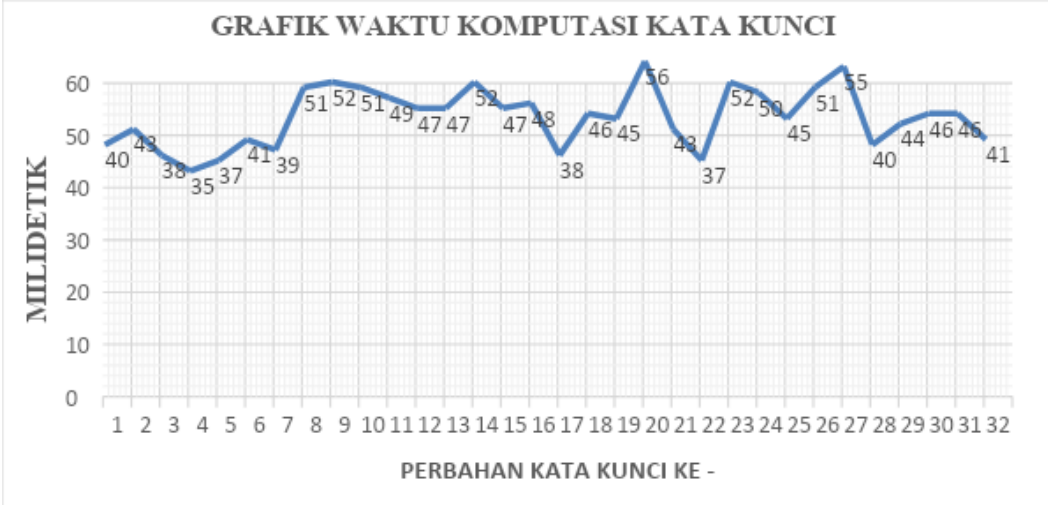
Gambar 7. Grafik Nilai Waktu Komputasi Setiap Pesan Teks

Berdasarkan grafik nilai Avalanche Effect pada pesan teks *(chatting)* telah dilaksanakan dan diketahui rata-rata nilai Avalanche Effect dengan menggunakan metode GOST dengan Panjang 10 karakter **25,23%**, 20 karakter **17,86%** dan 30 karakter **12,02%**. Avalanche effect merupakan salah satu karakteristik yang menjadi salah satu acuan untuk menentukan baik atau tidaknya sebuah algoritma kriptografi tersebut. Dari hasil eksperimen nampak bahwa nilai rata-rata Avalanche Effect yang diperoleh tidak terlalu besar. Penambahan simbol pada pesan dapat menghasilkan perubahan beberapa bit dari chipertext, namun dapat dilihat bahwa semakin panjang suatu pesan teks (chatting) maka hasil persentase dari Avalanche Effect akan semakin menurun



Gambar 8. Grafik Nilai Avalanche Kata Kunci

Berdasarkan grafik nilai avalanche effect pada kata kunci yang berbeda beda dengan pesan teks (chatting) yang sama dapat dilihat bahwa setiap perubahan yang dilakukan dari kata kunci ke-1 hingga kata kunci ke-32 dengan merubah 1bit dari 32bit panjang kata kunci, menghasilkan nilai rata rata avalanche effect yang cukup baik. Nilai rata-rata avalanche effect yang didapatkan adalah **49,11%**



Gambar 9. Grafik Waktu Komputasai Kata Kunci

Berdasarkan grafik menunjukkan bahwa setiap waktu komputasi pada proses enkripsi tidak memiliki perbedaan yang cukup signifikan untuk setiap perubahan kata kunci pertama hingga kata kunci terakhir. Rata – rata waktu komputasi yang dihasilkan dari perubahan bit kata kunci adalah 94,5 milidetik.

KESIMPULAN

Berdasarkan penelitian yang telah dilaksanakan dengan proses pengujian dan melakukan implementasi keamanan pesan dan pengujian dengan menggunakan metode Gosundarstevenny Standard (GOST). Maka dapat ditarik kesimpulan sebagai berikut:

; Aplikasi Mengamankan Pesan Teks berhasil mengimplementasikan teknik enkripsi dan dekripsi dengan menggunakan metode Gosundarstevenny Standard (GOST) dalam mengamankan pesan. Hal ini dibuktikan melalui hasil tampilan bahwa pada user 1 sebagai pengirim dan user 2 penerima bisa saling berkirim pesan dan bisa menampilkan hasil enkripsi dan dekripsi disetiap tampilannya.

; Berdasarkan skenario pengujian, pesan yang telah dikirimkan, kemudian dikelompokkan menjadi 3 kategori untuk mencari nilai ketahanannya. Dari hasil perhitungan Avalance Effect didapatkan nilai rata rata dari 3 kategori tersebut adalah 34,13%. Hal ini menunjukkan bahwa semakin banyak perubahan bit yang terjadi maka akan sulit algoritma kriptografi tersebut dipecahkan.

; Berdasarkan skenario pengujian kata kunci (key) yang telah dirubah 1 bit dari panjang 32bit dengan menggunakan metode Gosundarstevenny Standard GOST mendapatkan hasil rata-rata nilai avalanche effect yang baik dengan nilai 49,11% menunjukkan bahwa metode Gosundarstevenny Standard (GOST) dapat menyandikan pesan dengan baik, pengubahan pada karakter pesan chat dan kata kunci dapat mempengaruhi nilai Avalanche Effect.

; Dan berdasarkan grafik waktu komputasi yang telah didapatkan memiliki nilai waktu yang berbeda beda antara 35 hingga 56 milidetik, khusunya saat pertama kali menginputkan pesan teks (chatting). hal ini disebabkan oleh saat aplikasi pertama kali dijalankan (running) waktu komputasi yang didapatkan akan sangat tinggi.

DAFTAR PUSTAKA

[1] Komalasari, R. (2018). KESADARAN AKAN KEAMANAN PENGGUNAAN. TEMATIK - Jurnal Teknologi Informasi Dan Komunikasi.
[2] Febriana, A. (2017). PENERAPAN TEKNIK KRIPTOGRAFI PADA KEAMANAN SMS ANDROID. JOEICT (Jurnal of Education and Information Communication Technolo, 30-31.
[3] Bahari, M. F. (2022). Analisa Dan Implementasi Keamanan Pesan Chatting Menggunakan Algoritma Challenge Response. JUSSI: Jurnal Sains Dan Teknologi Informasi, 49-53.
[4] [4] Rohit Verma, A. K. (2020). Cryptography: Avalanche effect of AES and RSA. International Journal of Scientific and Research Publications, 120-121
[5] Sri Gustaria, T. F. (2017). IMPLEMENTASI KRIPTOGRAFI MENGGUNAKAN ALGORITMA GOST (GOSUDARSTEVVENYI STANDARD) UNTUK PENGIRIMAN E-MAIL PADA APLIKASI CIRUS-MAIL BERBASIS WEB..
[6] Muslih, L. B. (2022). PENGUJIAN AVALANCHE EFFECT PADA KRIPTOGRAFI TEKS MENGGUNAKAN AUTOKEY CIPHER. SEMINAR NASIONAL TEKNOLOGI DAN MULTIDISIPLIN ILMU, 130-133.
[7] Kamsiah Mohamed, M. N. (2022). ANALYSE ON AVALANCHE EFFECT IN CRYPTOGRAPHY ALGORITHM. European Proceedings of Multidisciplinary Sciences EpMS, 615-617. , 141-144.