KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 4, No. 4, Oktober (2023), pp. 878-889

Penerapan Teknik Super Enkripsi Untuk Keamanan File Teks Menggunakan Gabungan Algoritma Beaufort Cipher dan Algoritma RSA

Yusuf Ramadhan Nasution[1], Abdul Halim Hasugian[2], Mahyudi[3]
[1,2,3]Prodi Ilmu Komputer Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara, Indonesia
E-mail:[1]ramadhannst@gmail.com,[2]abdulhasugian12@gmail.com, [3]mahyudi997@gmail.com

Abstract

Cryptography is an alternative solution that can be applied to protect and increase data security. This research will implement a combination of the Beaufort Cipher symmetric cryptographic algorithm and the RSA asymmetric cryptographic algorithm in a super encryption scheme to increase data security in text files. The combination of the two cryptographic algorithms is done by first encrypting the message contained in the text file using the Beaufort Cipher algorithm key, then re-encrypting it with the public key of the RSA algorithm to create an encrypted text file. The combination method between the two algorithms aims to obtain superior encryption (ciphertext) results so that they are not easy to crack, and also to handle the use of a single ciphertext which is comparatively weak because it only uses one cryptographic algorithm. This research produces a web-based application that can be used to secure text files using a combination of the Beaufort Cipher algorithm and the RSA algorithm in the super encryption technique.

***Keywords**: Super Encryption, Beaufort Cipher, RSA, Text Files*

Abstrak

Kriptografi ialah salah satu alternatif penyelesaian yang bisa diterapkan guna melindungi serta menambah keamanan suatu data. Pada penelitan ini akan mengimplementasikan perpaduan algoritma kriptografi simetris Beaufort Cipher dan algoritma kriptografi asimetris RSA dalam skema super enkripsi guna meningkatkan keamaman data pada file teks. Kombinasi dua algoritma kriptografi dilakukan dengan cara mengenkripsi pesan yang terdapat dalam file teks terlebih dahulu memakai kunci algoritma Beaufort Cipher, setelah itu mengenkripsi kembali dengan kunci publik algoritma RSA guna menciptakan file teks terenkripsi. Metode penggabungan antara kedua algoritma ini bertujuan untuk memperoleh hasil enkripsi (ciphertext) yang lebih unggul sehingga tidak mudah buat dipecahkan, dan juga untuk menangani pemakaian ciphertext tunggal yang secara komparatif lemah sebab hanya menggunakan satu algoritma kriptografi. Penelitian ini menghasilkan sebuah aplikasi berbasis web yang dapat digunakan untuk mengamankan file teks dengan menggunakan gabungan algoritma Beaufort Cipher dan algoritma RSA dalam teknik super enkripsi.

***Kata kunci**: Super Enkripsi, Beaufort Cipher, RSA, File Teks.*

1. Pendahuluan

Berbagai cara dilakukan untuk menjaga keamanan data yang berkaitan dengan hak milik pribadi berupa dokumen, foto, video, pesan, lokasi, maupun data penting lainnya [1] seperti menyembunyikan data dengan teknik steganografi atau dengan cara menyandikan data menjadi suatu kode-kode yang tidak dimengerti oleh orang yang tidak berkepentingan [2], sehingga apabila dicuri ataupun disadap, orang lain bakal kesusahan untuk mengetahui serta memahami informasi yang sesungguhnya. Proses penyandian

ISSN: 2720-992X

dilakukan dengan menggunakan teknik kriptografi yang merupakan metode matematis yang terpaut dengan aspek keamanan suatu sistem informasi [3]. Kriptografi melingkupi proses transformasi informasi menjadi sebuah tatanan yang tidak bisa dimengerti, sehingga orang-orang yang tidak berkepentingan tidak bisa memahaminya. Kriptografi terdiri dari enkripsi, dekripsi, dan tiga komponen teks (*plaintext*, *ciphertext*, dan kunci) [4]. Enkripsi adalah sebuah proses penyandian yang mengubah sebuah pesan (*plaintext*) menjadi sebuah kode yang tidak bisa dimengerti (*ciphertext*). Dekripsi adalah kebalikan dari enkripsi yaitu proses penyandian yang mengubah sebuah kode yang tidak bisa dimengerti (*ciphertext*) menjadi pesan yang mudah dimengerti (*plaintext*). Kunci terbagi menjadi dua bagian *public key* dan *private key* yang dipakai untuk melakukan proses enkripsi dan dekripsi [5]. Terdapat banyak algoritma kriptografi yang sudah ada dengan kelebihan dan kekurangannya. Pada penelitan ini akan mengimplementasikan Super Enkripsi menggunakan gabungan algoritma Beaufort Cipher dan algoritma RSA (Rivest Shamir Adleman) untuk keamanan *file* teks. Alasan menggunakan algoritma RSA berdasarkan penelitian terdahulu yang dilakukan oleh [6] menjelaskan bahwa hingga saat ini algoritma RSA sangat susah buat dipecahkan serta akan memerlukan waktu yang sangat lama. Sedangkan alasan memakai algoritma Beaufort Cipher berdasarkan penelitian terdahulu yang dilakukan oleh [7] menjelaskan bahwa kelebihan dari algoritma Beaufort Cipher yaitu jumlah kunci yang digunakan mempunyai panjang yang sama dengan jumlah karakter pesan asli (*plaintext*). Perihal inilah yang bisa membuat pesan hasil enkripsi jadi susah untuk dimengerti oleh pihak lain, sebab masing-masing karakter *plaintext* akan mempunyai pasangan kunci yang berbeda dengan karakter *plaintext* yang lain. Berdasarkan permasalahan yang telah telah diuraikan serta penjelasan dari hasil penelitian terdahulu, maka pembaruan yang dilakukan dalam penelitian ini adalah terletak pada konsep dan algoritma yang digunakan. Jika pada penelitian yang dilakukan oleh [6] hanya memakai satu algoritma kriptografi yaitu RSA, maka pada penelitian ini akan menggabungkan dua algoritma. Jika pada penelitian yang dilakukan oleh [8] dengan menggabungkan dua algoritma kriptografi, namun hanya menggunakan algoritma kriptografi kunci simetris yaitu Beaufort Cipher dan Transposisi Kolom, maka pada penelitian ini akan mengkombinasikan algoritma kriptografi kunci simetris dan kunci asimetris.

2. Metodologi Penelitian

2.1. Teknik Pengumpulan Data

Pengumpulan data yang dipakai pada penelitian ini merupakan teknik studi literatur. Bahan serta alat yang digunakan dalam penelitian ini terdiri dari perangkat keras dan perangkat lunak. Perangkat keras yang digunakan ialah Prosesor Intel Core i3, RAM 4GB, serta Hard Disk 500GB. Sedangkan perangkat lunak yang digunakan ialah Sistem Operasi Windows 10 Pro 64bit, PHP, HTML, Javascript, XAMPP, Visual Studio. Fokus kajian yang diteliti mengenai Super Enkripsi menggunakan gabungan algoritma Beaufort Cipher serta algoritma RSA.

2.2. Analisis

Proses sistem dari penelitian ini dirancang dengan menggunakan skema Super Enkripsi dengan tujuan untuk mengamankan pesan yang berada di dalam *file* teks dengan format (**.txt*) yang menggunakan keamanan berganda sehingga dapat menanggulangi kebocoran data [9].

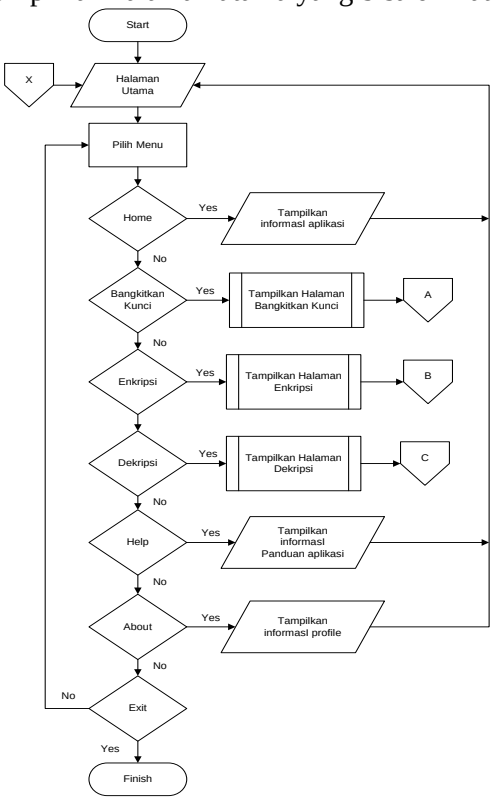
2.3. Perancangan

Perancangan sistem merupakan wujud dari implementasi sistem secara teknis, Perancangan sistem bertujuan untuk menggambarkan semua kondisi dan bagian-bagian yang berperan dalam sistem yang dirancang. Tujuan dari perancangan yaitu membuat

tampilan sistem yang sederhana dan mudah digunakan (*user friendly*) sehingga *user* dapat lebih mudah dalam menggunakan sistem. Dalam perancangan sistem ini menggunakan *flowchart* sistem yang merupakan penggambaran secara grafik dari prosedur program. Selain itu juga akan dijelaskan rancangan antarmuka sistem (*system interface*) yang akan dibangun.

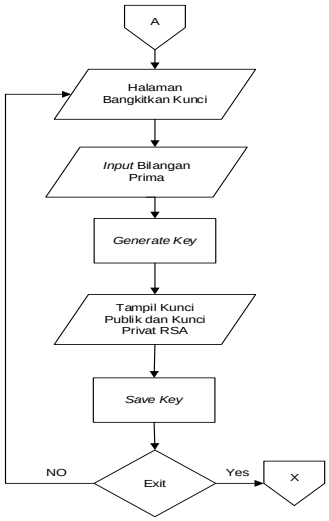
a) Flowchart Sistem

Dimulai dengan menampilkan halaman utama yang bisa dilihat pada Gambar 1.



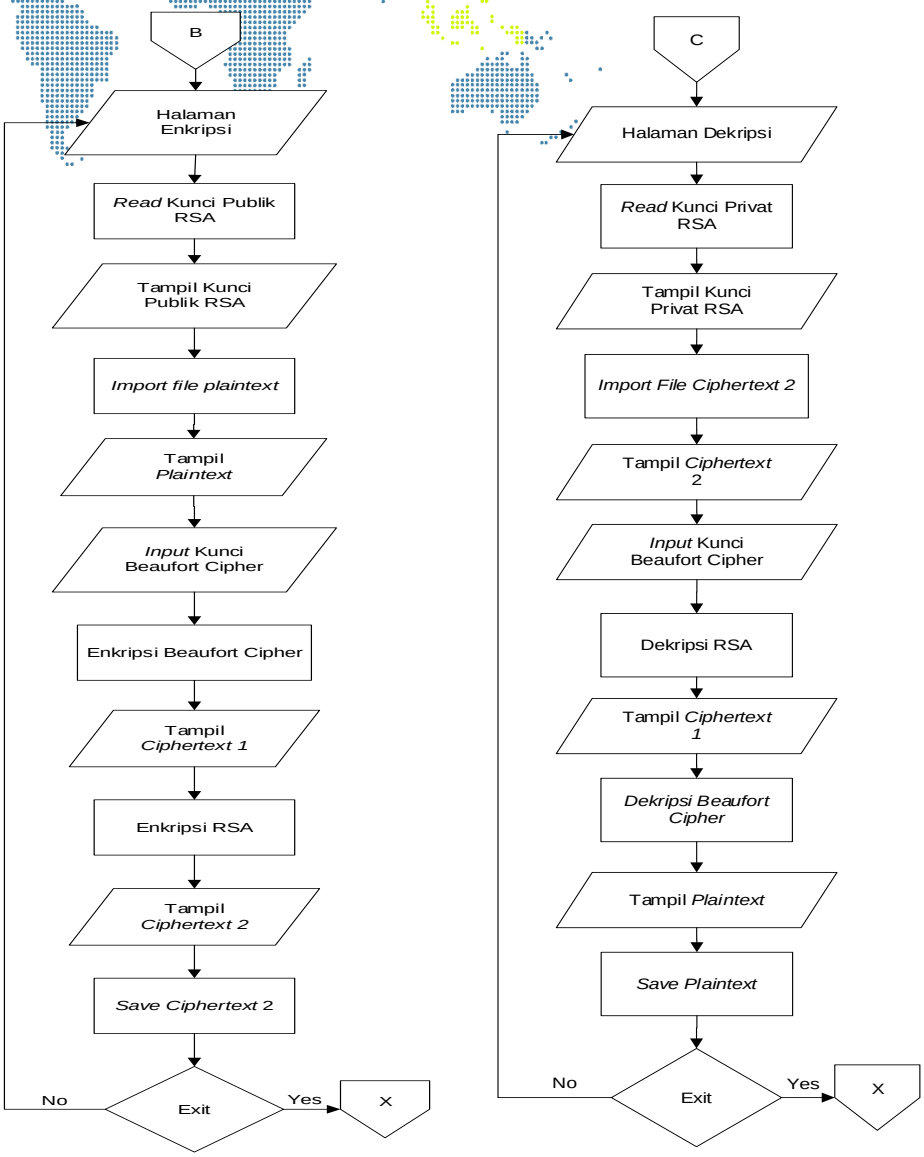
Gambar 1. *Flowchart* Halaman Utama

Adapun flowchart dari halaman bangkitkan kunci publik dan privat pada algoritma asimetris RSA bisa dilihat pada Gambar 2.



Gambar 2. *Flowchart* Bangkitkan Kunci

Adapun *flowchart* pada halaman enkripsi dan dekripsi dapat dilihat pada Gambar 3.

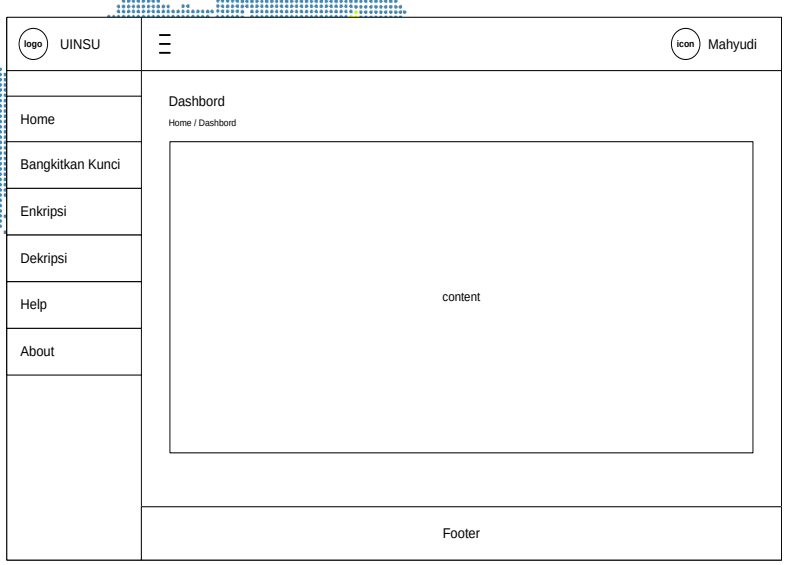


Gambar 3. *Flowchart* (B) Enkripsi dan (C) Dekripsi

b) Perancangan Antar Muka

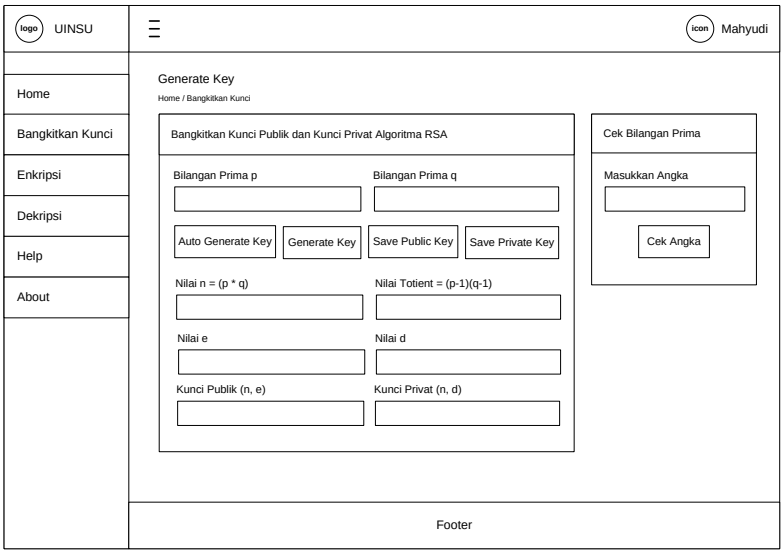
Proses perancangan interface sistem (antarmuka sistem) merupakan hal yang sangat penting karena diperlukan perancangan yang sederhana dan efisien agar pengguna tidak sulit pada saat menjalankannya, Sistem dibangun berbasis web dengan menggunakan PHP sebagai bahasa pemrograman.

Perancangan halaman utama merupakan halaman pembuka yang akan ditampilkan pertama kali saat aplikasi dijalankan dapat dilihat pada Gambar 4.



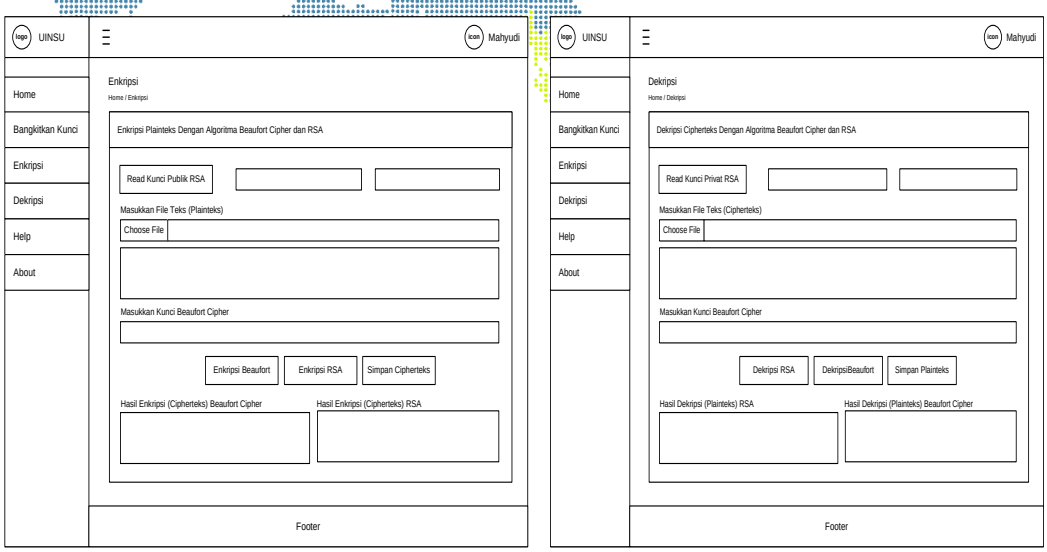
Gambar 4. Rancangan Halaman Utama

Perancangan halaman bangkitkan kunci merupakan sebuah form yang dirancang untuk membangkitkan kunci dari algoritma RSA, kunci publik yang digunakan untuk mengenkripsi pesan dan kunci privat yang digunakan untuk mendekripsi pesan yang dapat dilihat pada Gambar 5.



Gambar 5. Rancangan Halaman Bangkitkan Kunci

Perancangan Halaman Enkripsi dan Dekripsi adalah sebuah form yang dirancang guna melaksanakan proses enkripsi pesan dalam file teks dengan memakai kombinasi algoritma kriptografi simetris Beaufort Cipher dan algoritma kriptografi asimetris RSA dalam skema Super Enkripsi yang bisa dilihat pada Gambar 6.



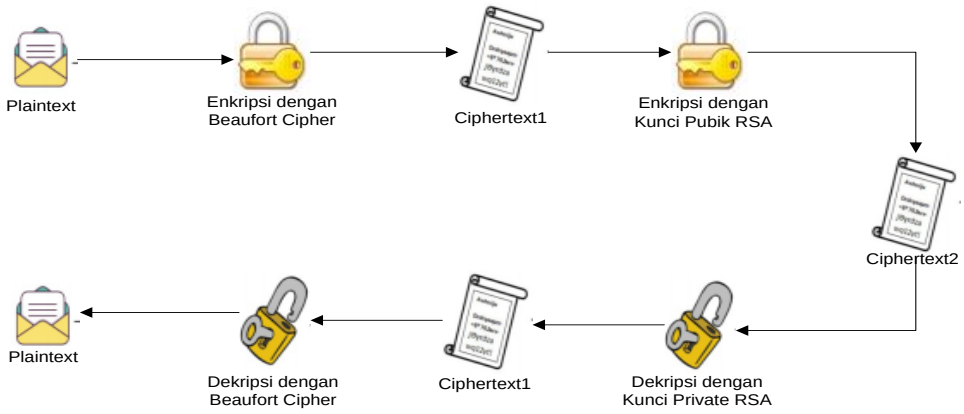
(a) (b)

Gambar 6. Rancangan Halaman (a) Enkripsi dan (b) Dekripsi

3. Hasil Dan Pembahasan

3.1. Analisis Penerapan Metode

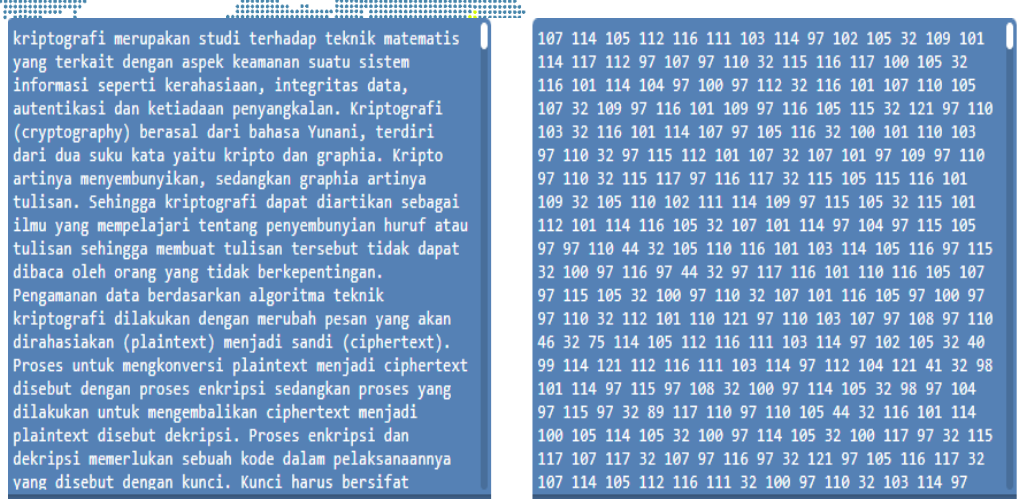
Untuk pengimplementasian pengamanan isi *file* teks menggunakan gabungan algoritma Beaufort Cipher serta algoritma RSA dalam skema Super Enkripsi konsep keamanan tersebut bisa dilihat pada Gambar 7.



Gambar 7. Konsep Pengamanan

a) Analisis Proses Enkripsi

Proses enkripsi merupakan tahap untuk mentransformasi data ke dalam bentuk yang susah dipahami oleh banyak orang. Sesuai gambar 7 ada dua proses enkripsi yang dilakukan dalam mengamankan data, pesan yang terdapat dalam *file* hanya menggunakan karakter *alphabet* A-Z dan a-z yang terdapat dalam Tabel ASCII dengan *modulo* 26. Berikut ditampilkan contoh sampel data.



Gambar 8. Potongan sampel data (a) karakter (b) nilai desimal

Gambar 8 menampilkan nilai desimal dari sampel data yang diperoleh dengan menggunakan alat bantu (*tools*) dari *https://onlinetexttools.com/*. Setelah nilai desimal dari pesan yang berada dalam file teks didapatkan, maka proses enkripsi dengan algoritma Beaufort Cipher bisa dilakukan. Dengan mengambil 11 huruf pertama dari *file* teks serta menentukan kunci enkripsi dan dekripsi sebagai berikut :

Plaintext : k r i p t o g r a f i
Kunci : m a h y u d i

Pada contoh diatas kunci algoritma Beaufort Cipher "mahyudi" diulang sedemikian rupa hingga panjang kunci sama dengan panjang *plaintext*-nya. Kemudian setelah panjang kunci sama dengan panjang *plaintext*, proses enkripsi dilakukan terlebih dahulu akan dikonversi ke desimal pada Tabel ASCII sehingga hasilnya dapat disajikan pada Tabel 1.

Tabel 1. Proses mengubah *plaintext* menjadi desimal

Plaintext (M_i)	k	r	i	p	t	o	g	r	a	f	i
Nilai Desimal	107	114	105	112	116	111	103	114	97	102	105
Kunci (K_i)	m	a	h	y	u	d	i	m	a	h	y
Nilai Desimal	109	97	104	121	117	100	105	109	97	104	121

Setelah *plaintext* dikonversi menjadi desimal, maka proses enkripsi algoritma Beaufort Cipher dapat dilakukan adapun formulasi algoritma Beaufort Cipher dalam proses enkripsi [10] dengan menggunakan rumus pada persamaan berikut : $C_i = E_k(M_i) = (K_i - M_i) \bmod 26 + 97$
Berdasarkan rumus pada persamaan diatas maka diperoleh hasil enkripsi awal (*ciphertext1*) dengan menggunakan algoritma Beaufort Cipher yaitu :

Plaintext (M_i) k r i p t o g r a f i
Kunci Beaufort (K_i) m a h y u d i m a h y
*Ciphertext*1 (C_i) c j z j b p c v a c q

Hasil dari proses enkripsi (*ciphertext1*) kemudian akan di enkripsi lagi dengan menggunakan algoritma kriptografi RSA. Algoritma RSA merupakan algoritma kriptografi asimetris dimana kunci enkripsi tidak sama dengan kunci dekripsinya.

Untuk mengenkripsi serta mendekripsi dengan menggunakan algoritma RSA, maka terlebih dulu harus membangkitkan sepasang kunci, yakni kunci publik (*public key*) serta kunci privat (*private key*). Adapun algoritma untuk membangkitkan kunci algoritma RSA yaitu sebagai berikut :

1. Pilih dua bilangan prima sembarang untuk p dan q, misalkan p=11 dan q=23
2. Hitung n = p * q
   n = 11 * 23 = 253.
3. Hitung φ(n) = (p – 1)(q – 1)
   φ(n) = (11 – 1) (23 – 1) = 220
4. Pilih kunci publik e, yang relatif prima terhadap φ(n) yaitu 1<e<φ(n) dan gcd(e,φ(n)) = 1. Misalkan dipilih e = 3 karena relatif prima dengan φ(n) = 220
5. Bangkitkan kunci privat dengan menggunakan persamaan berikut : d*e mod φ(n)=1. Misalkan dipilih nilai d = 147.
   147*3 mod 220 = 1
   Sehingga pasangan kunci enkripsi dan dekripsi hasil dari algoritma RSA di atas ialah sebagai berikut:
   a. Kunci enkripsi (*public key*) merupakan pasangan (e,n) = (3 , 253)
   b. Kunci dekripsi (*private key*) merupakan pasangan (d,n) = (147,253)

Dengan mengambil hasil enkripsi Beaufort Cipher (*ciphertext1*) maka proses enkripsi dengan menggunakan kunci publik algoritma RSA dapat dilakukan dengan menggunakan persamaan berikut :

1. Ambil kunci publik (e,n)
2. Nyatakan *plaintext* m menjadi blok-blok $m_1,m_2,m_3,\ldots,m_n$ lalu konversi bentuk desimal dalam Tabel ASCII.
3. Setiap blok m_i akan dienkripsi menjadi blok c_i dengan menggunakan persamaan berikut : $c_i = m_i^e \bmod n$

Maka diperoleh hasil enkripsi (*ciphertext2*) dengan menggunakan kunci publik algoritma RSA dalam bentuk bilangan desimal yang masing-masing hasil enkripsi akan disipsahkan dengan tanda titik adalah 44.145.67.145.32.19.44.50.102.44.38. Adapun perbandingan pesan teks sebelum dan setelah di enkripsi dengan menggunakan algoritma Beaufort Cipher dan algoritma RSA dapat dilihat pada Tabel 2.

Tabel 2. Perbandingan hasil enkripsi

***File* Teks**	*Plaintext*	kriptografi
	Size	11 * 8 = 88 bit
Hasil Enkripsi *Beaufort*	*Ciphertext*	cjzjbpcvacq
	Size	11 * 8 = 88 bit
Hasil Enkripsi RSA	*Ciphertext*	44.145.67.145.32.19.44.50.102.44.38
	Size	25 * 8 = 200 bit

b) Analisis Proses Dekripsi

Setelah *ciphertext* diperoleh, maka selanjutnya akan dilakukan proses dekripsi yang bertujuan untuk mengembalikan pesan sudah di enkripsi pada langkah sebelumnya. Proses dekripsi merupakan kebalikan dari proses enkripsi yaitu untuk mentransformasi *ciphertext* ke dalam bentuk yang dapat dimengerti (*plaintext*). Proses dekripsi dengan menggunakan kunci privat algoritma RSA bisa dilakukan dengan menggunakan persamaan sebagai berikut :

1. Ambil kunci privat (*private key*) algoritma RSA yaitu (n,d) = (253,147).

2. Ambil *ciphertext* (c_i) yang akan didekripsi (dalam hal ini *ciphertext* merupakan hasil dari proses enkripsi sebelumnya yaitu *ciphertext*2) kemudian nyatakan menjadi blok-blok. $c_1,c_2,c_3,\ldots,c_n$.
3. Setiap blok c_i akan didekripsi menjadi blok m_i dengan menggunakan persamaan $m_i = c_i^d \bmod n$

Berdasarkan hasil persamaan diatas maka diperoleh hasil dekripsi (*ciphertext*) algoritma RSA yang jika dikonversi kedalam karakter pada Tabel ASCII akan diperoleh hasil seperti pada Tabel 3.

Tabel 3. Konversi desimal kedalam ASCII

Desimal	Karakter ASCII
99	c
106	j
122	z
106	j
98	b
112	p
99	c
118	v
97	a
99	c
113	q

Berdasarkan Tabel 3 maka diperoleh hasil dekripsi pertama dengan algoritma RSA yang merupakan hasil enkripsi algoritma Beaufort Cipher (*ciphertext1*). Hasil dari proses dekripsi pertama (*ciphertext1*) kemudian akan di dekripsi lagi dengan menggunakan algoritma kriptografi Beaufort Cipher untuk mendapatkan kembali *file* teks aslinya (*plaintext*). Akan dilakukan konversi terlebih dahulu kedalam desimal pada Tabel ASCII sehingga hasilnya bisa dilihat pada Tabel 4.

Tabel 4. Proses mengubah *ciphertext1* menjadi desimal

*Ciphertext*1 (C_i)	c	j	z	j	b	p	c	v	a	c	q
Nilai Desimal	99	106	122	106	98	112	99	118	97	99	113
Kunci (K_i)	m	a	h	y	u	d	i	m	a	h	y
Nilai Desimal	109	97	104	121	117	100	105	109	97	104	121

Setelah *ciphertext1* dikonversi menjadi desimal, maka proses dekripsi algoritma Beaufort Cipher dapat dilakukan dengan menggunakan rumus pada persamaan sebgai berikut :

$$M_i = D_k (C_i) = (K_i - C_i) \bmod 26 + 97$$

Berdasarkan rumus persamaan diatas seteleh melakukan proses dekripsi dengan menggunakan algoritma Beaufort Cipher untuk tahap kedua maka diperoleh kembali *plaintext* aslinya yaitu “kriptografi”.

3.2. Implementasi

Implementasi merupakan tahapan yang dilakukan setelah perancangan sistem dan pembuatan *flowchart* sistem. Setelah selesai menganalisis dan membuat rancangan dari sistem yang akan dibangun, selanjutnya adalah mengimplementasikan hasil analisis dan perancangan ke dalam bentuk aplikasi dengan menggunakan bahasa pemrograman. Implementasi program dalam penelitian dibuat berbasis *web*.

a) Pengujian Bangkitkan Kunci
Setelah memilih menu "Bangkitkan Kunci" yang terdapat pada halaman utama, maka sistem akan menampilkan halaman bangkitkan kunci untuk melakukan proses pembangkitan kunci publik (*public key*) serta kunci privat (*private key*) dari algoritma asimetris RSA. Terdapat dua cara yang dapat dilakukan untuk membangkitkan kunci algoritma RSA pada aplikasi yang dibangun, yaitu dengan membangkitkan kunci secara otomatis atau dengan cara manual dengan mengisikan langsung dua buah bilangan prima p dan q pada setiap *textbox* yang tersedia pada halaman bangkitkan kunci, pasangan kunci yang sukses dibangkitkan berikutnya bakal disimpan untuk keperluan proses enkripsi serta dekripsi. Hasil pengujian bisa dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian Bangkitkan Kunci

No.	Kasus Uji	Langkah Uji	Hasil
1.	Bangkitkan Kunci	Membangkitkan kunci dengan memilih tombol "*Auto Generate Key*" atau memilih tombol "*Generate Key*"	Sistem dapat membangkitkan kunci secara otomatis atau dengan menentukan sendiri bilangan prima untuk membangkitkan kunci
2.	Simpan Kunci	Menyimpan kunci dengan memilih tombol "*Save Public Key*" dan tombol "*Save Private Key*"	Sistem dapat menyimpan kunci publik dan kunci privat kedalam *file* teks dengan format **.txt*

b) Pengujian Enkripsi
Setelah memasukkan *file plaintext* dilanjutkan dengan mengenkripsi menggunakan algoritma Beaufort dan sistem akan menampilkan hasil enkripsi pertama (*ciphertext1*) yang kemudian akan di enkripsi lagi dengan algoritma RSA dan akan menghasilkan (*ciphertext2*) sebagai hasil akhir dari proses enkripsi dalam bentuk bilangan desimal dengan estimasi waktu enkripsi selama 0.579 ms (*millisecond*). Hasil pengujian dapat dilihat pada Tabel 6.

Tabel 6. Hasil Pengujian Enkripsi

No.	Kasus Uji	Langkah Uji	Hasil
1.	*Input* Kunci Publik RSA	Mengimport *file* kunci publik RSA dengan memilih tombol "*Read* Kunci Publik RSA"	Sistem dapat membaca dan menampilkan isi dari *file* kunci publik algoritma RSA
2.	*Input File* Teks	Mengimport *file* teks dengan memilih tombol "*Choose File*"	Sistem hanya dapat menerima format *file .txt* dan sistem dapat menampilkan isi dari *file* teks
3.	*Input* Kunci *Beaufort Cipher*	Memasukkan kunci *Beaufort Cipher*	Sistem dapat menerima kunci
4.	Enkripsi *Beaufort Cipher*	Mengenkripsi *file* teks dengan memilih tombol "Enkripsi *Beaufort*"	Sistem dapat menampilkan hasil enkripsi algoritma *Beaufort Cipher*
5.	Enkripsi RSA	Mengenkripsi hasil *Beaufort Cipher* dengan memilih tombol "Enkripsi RSA"	Sistem dapat menampilkan hasil enkripsi algoritma RSA
6.	Simpan *Ciphertext*	Menyimpan hasil enkripsi dengan memilih tombol "Simpan *Cipherteks*"	Sistem dapat menyimpan hasil enkripsi (*ciphertext*) kedalam *file* baru dengan format **.txt*

c) Pengujian Dekripsi
Setelah memasukkan *file ciphertext2* dilanjutkan dengan mendekripsi menggunakan algoritma RSA dan sistem akan menampilkan hasil dekripsi pertama (*ciphertext1*)

yang kemudian akan di dekripsi lagi dengan algoritma Beaufort dan akan menghasilkan (*plaintext*) yang sama persis dengan *file* teks aslinya sebagai hasil akhir dari proses dekripsi dengan estimasi waktu dekripsi selama 0.433 ms (*millisecond*). Hasil pengujian dapat dilihat pada Tabel 7.

Tabel 7. Hasil Pengujian Dekripsi

No.	Kasus Uji	Langkah Uji	Hasil
1.	*Input* Kunci Privat RSA	Mengimport *file* kunci privat RSA dengan memilih tombol "*Read* Kunci Privat RSA"	Sistem dapat membaca dan menampilkan isi dari *file* kunci privat algoritma RSA
2.	*Input File* Teks	Mengimport *file* teks (*ciphertext*) dengan memilih tombol "*Choose File*"	Sistem hanya dapat menerima format *file* *.*txt* dan sistem dapat menampilkan isi dari *file* teks
3.	Input Kunci *Beaufort Cipher*	Memasukkan kunci *Beaufort Cipher*	Sistem dapat menerima kunci
4.	Dekripsi RSA	Mendekripsi *file ciphertext* dengan memilih tombol "Dekripsi RSA"	Sistem dapat menampilkan hasil dekripsi algoritma RSA
5.	Dekripsi *Beaufort Cipher*	Mendekripsi *file ciphertext* dengan memilih tombol "Dekripsi *Beaufort*"	Sistem dapat menampilkan hasil dekripsi algoritma *Beaufort Cipher*
6.	Simpan *Plaintext*	Menyimpan hasil dekripsi dengan memilih tombol "Simpan *Plaintext*"	Sistem dapat menyimpan hasil dekripsi (*plaintext*) kedalam *file* baru dengan format *.*txt*

4. Kesimpulan

Berdasarkan pada penelitian yang telah dilakukan maka kesimpulan yang bisa diambil setelah melakukan implementasi dan pengujian sistem dengan menerapkan penggabungan algoritma Beaufort Cipher serta algoritma RSA dalam skema Teknik Super Enkripsi untuk keamanan *file* teks adalah kombinasi menggunakan dua algoritma kriptografi dilakukan dengan cara mengenkripsi pesan yang terdapat dalam *file* teks terlebih dahulu menggunakan kunci algoritma Beaufort Cipher, kemudian mengenkripsi kembali dengan kunci publik algoritma RSA untuk memperoleh *file* teks terenkripsi yang lebih aman dan kuat serta dapat mengatasi permasalahan pada penggunaan *ciphertext* tunggal sehingga tidak mudah untuk dipecahkan karena proses enkripsi akan dilakukan sebanyak dua kali. Didasarkan pada penelitian ini untuk mengembangkan serta memanfaatkan penelitian ini ialah dengan menggunakan format *file* yang lebih beragam untuk melakukan proses pengamanan data serta melakukan pengoptimalan dengan menambahkan algoritma kompresi guna memperkecil ukuran hasil *file*.

Daftar Pustaka

[1] Soediro, "Prinsip Keamanan, Privasi, dan Etika dalam Undang-undang Informasi dan Transaksi Elektronik dalam Perspektif Hukum Islam," *Kosmik Huk.*, vol. 18, no. 2, pp. 95–112, 2018.

[2] N. F. Ginting and M. Ginting, "Perbandingan Kriptografi RSA dengan Base64," *J. Tek. Inform. Unika St. Thomas*, vol. 2, no. 2, pp. 47–52, 2017, doi: https://doi.org/10.17605/jti.v2i2.190.

[3] M. Diana and T. Zebua, "Optimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 2, no. 1, pp. 12–22, 2018, doi: 10.30645/j-sakti.v2i1.52.

[4] B. S. Muchlis, M. A. Budiman, and D. Rachmawati, "Teknik Pemecahan Kunci Algoritma Rivest Shamir Adleman (RSA) dengan Metode Kraitchik," *J. Penelit. Tek. Inform.*, vol. 2, no. 2, pp. 49–64, 2017, [Online]. Available: http://jurnal.polgan.ac.id/index.php/sinkron/article/view/75

[5] S. Suhandinata, R. A. Rizal, D. O. Wijaya, P. Warren, and S. Srinjiwi, "Analisis Performa Kriptografi Hybrid Algoritma Blowfish Dan Algoritma RSA," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 6, no. 1, pp. 1–10, 2019, doi: 10.33330/jurteksi.v6i1.395.

[6] R. Kurniawan, "Rancang Bangun Aplikasi Pengaman Isi File Dokumen Dengan Algoritma RSA," *J. Ilmu Komput. dan Inform.*, vol. 1, no. 1, pp. 46–52, 2017, doi: http://dx.doi.org/10.30829/algoritma.v1i01.1309.

[7] M. Fadlan, S. Sinawati, A. Indriani, and E. D. Bintari, "Pengamanan Data Teks Melalui Perpaduan Algoritma Beaufort Dan Caesar Cipher," *J. Tek. Inform.*, vol. 12, no. 2, pp. 149–158, 2019, doi: 10.15408/jti.v12i2.12262.

[8] C. Irawan, E. Hari Rachmawanto, C. Atika Sari, and C. Agus Sugianto, "Super Enkripsi File Dokumen Menggunakan Beaufort Cipher Dan Transposisi Kolom," in *Seminar Nasional LPPM*, 2020, pp. 556–563.

[9] R. A. Megantara and F. A. Rafrastara, "Super Enkripsi Teks Kriptografi menggunakan Algoritma Hill Cipher dan Transposisi Kolom," *Pros. SENDI_U 2019*, pp. 85–92, 2019.

[10] H. H. Naing and Z. M. Aye, "Innovation Security of Beaufort Cipher by Stream Cipher Using Myanmar-Vigenere Table and Unicode Table," in *WCSE 2020: 2020 10th International Workshop on Computer Science and Engineering*, 2020, pp. 55–56. doi: 10.18178/wcse.2020.02.009.