

BAB IV

HASIL DAN PEMBAHASAN

4.1 Analisis Penerapan Metode

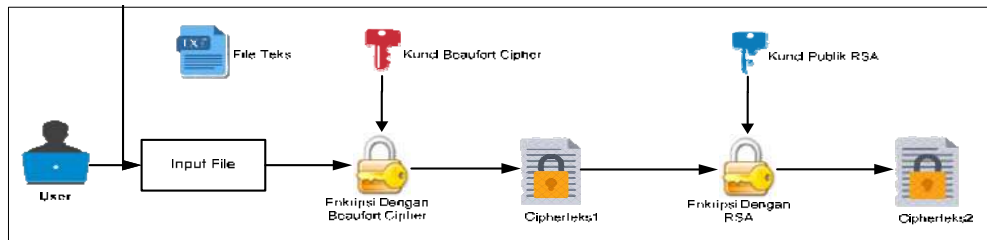
File teks menjadi salah satu bentuk dokumen yang umum digunakan untuk menyimpan informasi yang sifatnya pribadi atau rahasia, sehingga perlu dilakukan tindakan untuk tetap menjaga informasi tersebut hanya dapat diakses oleh pemilik data, baik pada saat disimpan atau pada saat ditransmisikan. Teknik kriptografi merupakan salah satu alternatif solusi yang dapat diterapkan untuk menjaga keamanan data, yaitu dengan cara memanipulasi pesan ke dalam bentuk yang tidak dimengerti oleh banyak orang.

Selain menerapkan algoritma kriptografi yang baru, biasanya banyak peneliti memodifikasi suatu algoritma kriptografi, namun hal ini bukanlah suatu pekerjaan yang mudah. Oleh karena itu, salah satu cara yang dapat diterapkan dalam meningkatkan keamanan pesan dengan menggunakan teknik kriptografi dengan cara mengkombinasikan dua buah algoritma kedalam satu proses. Pada penelitian ini akan mengimplementasikan perpaduan algoritma kriptografi simetris *Beaufort Cipher* dan algoritma kriptografi asimetris RSA guna meningkatkan keamanan pesan pada *file* teks. Metode pengkombinasian antara kedua algoritma yang bertujuan untuk mendapatkan hasil enkripsi (*ciphertext*) yang lebih kuat sehingga sulit untuk dipecahkan, dan juga untuk mengatasi penggunaan *ciphertext* tunggal yang secara komparatif lemah (Syahputra et al., 2021) karena hanya menggunakan satu algoritma kriptografi. Sistem yang dibangun pada penelitian ini adalah berupa aplikasi berbasis *web* untuk menyelesaikan permasalahan keamanan pesan pada *file* teks.

4.1.1 Analisis Proses Enkripsi

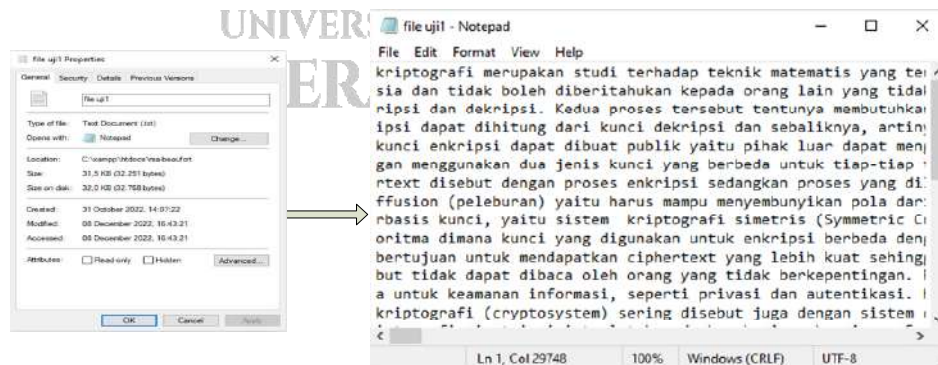
Proses enkripsi merupakan tahap untuk mentransformasi data ke dalam bentuk yang tidak dimengerti oleh banyak orang. Dalam penelitian ini data yang akan di enkripsi adalah berupa pesan yang terdapat dalam *file* teks dengan

menerapkan perpaduan algoritma kriptografi simetris *Beaufort Cipher* dan algoritma asimetris RSA dalam skema super enkripsi. Gambar 4.1 berikut menjelaskan secara umum mengenai proses enkripsi yang diterapkan dalam penelitian ini.



Gambar 4.1 Skema Proses Enkripsi

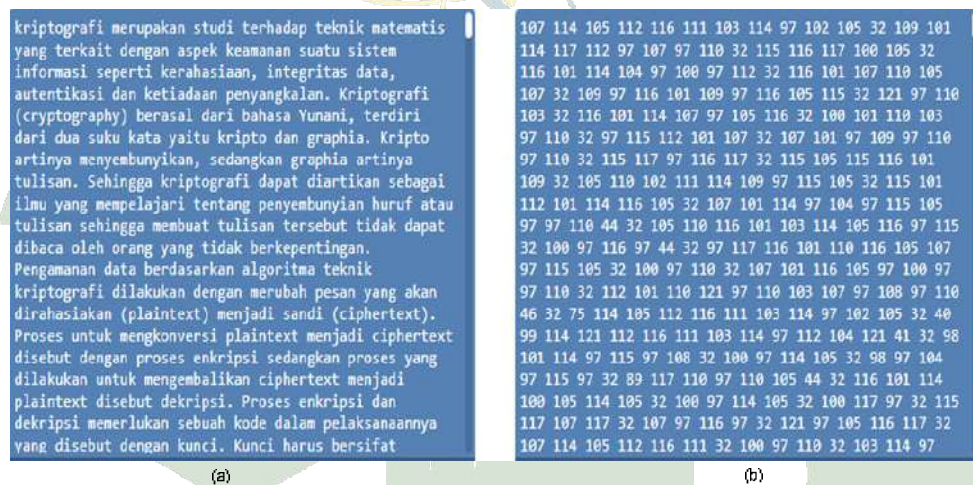
Sesuai gambar 4.1 terdapat dua proses utama yang dilakukan dalam mengamankan data berupa *file* teks dalam skema super enkripsi, proses pertama yaitu melakukan enkripsi menggunakan kunci algoritma *Beaufort Cipher* sehingga menghasilkan *ciphertext1*, lalu hasil dari proses tersebut akan di enkripsi lagi dengan menggunakan kunci publik algoritma RSA sehingga menghasilkan *ciphertext2* sebagai hasil akhir dari proses enkripsi. Pesan yang terdapat dalam *file* teks yang akan di enkripsi hanya menggunakan penyandian karakter A-Z dan karakter a-z yang terdapat dalam tabel ASCII dengan *modulo* 26 untuk algoritma *Beaufort Cipher* dan hanya menggunakan kunci dengan karakter *alphabet*. Sedangkan hasil enkripsi algoritma RSA akan dikonversi kedalam nilai desimal. Berikut ditampilkan contoh sampel data dengan format *file* teks.



Gambar 4.2 Potongan Sampel Data

Sesuai gambar 4.2, sampel data mempunyai *size* sebesar 32 KB (*kilo bytes*) dengan jumlah karakter sebanyak 28.272 karakter. Oleh karena itu, untuk mempermudah perhitungan dalam proses enkripsi maka hanya mengambil 11 huruf pertama sebagai inputan dalam melakukan proses perhitungan enkripsi secara manual.

Sebelum melakukan proses enkripsi menggunakan algoritma *Beaufort Cipher*, maka terlebih dahulu pesan yang terdapat dalam *file* teks dikonversi kedalam bentuk desimal dalam tabel ASCII. Proses konversi teks kedalam bentuk desimal pada penelitian ini dengan memanfaatkan tools yang dapat diakses secara *online* melalui alamat <https://onlinetexttools.com/>. Adapun hasil konversinya dapat disajikan pada gambar 4.3.



Gambar 4.3 Potongan Sampel Data (a) Karakter (b) Nilai Desimal

Gambar 4.3 menampilkan nilai desimal dari sampel data yang diperoleh dengan menggunakan alat bantu (*tools*) dari <https://onlinetexttools.com/>. Setelah nilai desimal dari pesan yang terdapat dalam *file* teks didapatkan, maka proses enkripsi dengan algoritma *Beaufort Cipher* dapat 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

Berdasarkan skema enkripsi pada gambar 4.1 maka tahap pertama yang dilakukan yaitu mengenkripsi pesan dengan menggunakan algoritma *Beaufort Cipher*. Enkripsi *Beaufort Cipher* merupakan teknik substitusi kriptografi yang menggunakan operasi *modulo* bilangan bulat sebagai proses utama. Kunci (K) pada *Beaufort Cipher* adalah urutan karakter-karakter $K = k_1, \dots, k_n$ dimana k_1 didapat dari banyaknya pergeseran dari karakter ke- i . Artinya bahwa jumlah kunci yang digunakan harus sama dengan jumlah karakter *plaintext* yang diamankan. Algoritma ini melakukan proses enkripsi secara *stream* (masing-masing karakter *plaintext* harus memiliki pasangan kunci).

Dapat diketahui bahwa panjang *plaintext* = 11, sedangkan panjang kunci = 7, karena panjang kunci algoritma *Beaufort Cipher* < panjang *plaintext*, maka kunci algoritma *Beaufort Cipher* tersebut akan diulang secara periodik sehingga panjang kunci tersebut sama dengan panjang *plaintext*-nya, yaitu sebagai berikut:

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

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 4.1.

Tabel 4.1 Proses Mengubah *Plaintext* Menjadi Desimal

<i>Plaintext</i> (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 dengan menggunakan rumus pada persamaan (2.1) sehingga diperoleh hasilnya sebagai berikut:

Untuk karakter $(M_1) = k$ dan kunci $(K_1) = m$

$$\begin{aligned} C_1 &= (K_1 - M_1) \bmod 26 = (m - k) \bmod 26 + 97 \\ &= (109 - 107) \bmod 26 + 97 \\ &= 99 \text{ (huruf "c" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(M_2) = r$ dan kunci $(K_2) = a$

$$\begin{aligned} C_2 &= (K_2 - M_2) \bmod 26 = (a - r) \bmod 26 + 97 \\ &= (97 - 114) \bmod 26 + 97 \\ &= 106 \text{ (huruf "j" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(M_3) = i$ dan kunci $(K_3) = h$

$$\begin{aligned} C_3 &= (K_3 - M_3) \bmod 26 = (h - i) \bmod 26 + 97 \\ &= (104 - 105) \bmod 26 + 97 \\ &= 122 \text{ (huruf "z" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(M_4) = p$ dan kunci $(K_4) = y$

$$\begin{aligned} C_4 &= (K_4 - M_4) \bmod 26 = (y - p) \bmod 26 + 97 \\ &= (121 - 112) \bmod 26 + 97 \\ &= 106 \text{ (huruf "j" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(M_5) = t$ dan kunci $(K_5) = u$

$$\begin{aligned} C_5 &= (K_5 - M_5) \bmod 26 = (u - t) \bmod 26 + 97 \\ &= (117 - 116) \bmod 26 + 97 \\ &= 98 \text{ (huruf "b" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(M_6) = o$ dan kunci $(K_6) = d$

$$\begin{aligned} C_6 &= (K_6 - M_6) \bmod 26 = (d - o) \bmod 26 + 97 \\ &= (100 - 111) \bmod 26 + 97 \\ &= 112 \text{ (huruf "p" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(M_7) = g$ dan kunci $(K_7) = i$

$$\begin{aligned} C_7 &= (K_7 - M_7) \bmod 26 = (i - g) \bmod 26 + 97 \\ &= (105 - 103) \bmod 26 + 97 \\ &= 99 \text{ (huruf "c" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(M_8) = r$ dan kunci $(K_8) = m$

$$C_8 = (K_8 - M_8) \bmod 26 = (m - r) \bmod 26 + 97$$

$$= (109 - 114) \text{ mod } 26 + 97$$

$$= 118 \text{ (huruf "v" dalam tabel ASCII)}$$

Untuk karakter $(M_9) = a$ dan kunci $(K_9) = a$

$$C_9 = (K_9 - M_9) \text{ mod } 26 = (a - a) \text{ mod } 26 + 97$$

$$= (97 - 97) \text{ mod } 26 + 97$$

$$= 97 \text{ (huruf "a" dalam tabel ASCII)}$$

Untuk karakter $(M_{10}) = f$ dan kunci $(K_{10}) = h$

$$C_{10} = (K_{10} - M_{10}) \text{ mod } 26 = (h - f) \text{ mod } 26 + 97$$

$$= (104 - 102) \text{ mod } 26 + 97$$

$$= 99 \text{ (huruf "c" dalam tabel ASCII)}$$

Untuk karakter $(M_{11}) = i$ dan kunci $(K_{11}) = y$

$$C_{11} = (K_{11} - M_{11}) \text{ mod } 26 = (y - i) \text{ mod } 26 + 97$$

$$= (121 - 105) \text{ mod } 26 + 97$$

$$= 113 \text{ (huruf "q" dalam tabel ASCII)}$$

Berdasarkan hasil perhitungan diatas maka diperoleh hasil enkripsi pertama (*ciphertext1*) dengan menggunakan algoritma *Beaufort Ciphert* yaitu:

<i>Plaintext</i> (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
<i>Ciphertext1</i> (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 dan dekripsi dengan menggunakan algoritma RSA, maka terlebih dahulu membangkitkan sepasang kunci, yaitu kunci publik (*public key*) dan kunci privat (*private key*). Adapun algoritma untuk membangkitkan kunci publik (*public key*) dan kunci privat (*private key*) algoritma RSA yaitu sebagai berikut:

1. Pilih dua buah bilangan prima sembarang untuk p dan q , misalkan $p = 11$ dan $q = 23$.
2. Hitung nilai n sehingga diperoleh hasilnya:

$$n = p * q$$

$$n = 11 * 23 = 253$$

3. Hitung nilai *totient* (φ) sehingga diperoleh hasilnya yaitu:

$$\varphi(n) = (p - 1)(q - 1)$$

$$\varphi(n) = (11 - 1)(23 - 1)$$

$$\varphi(n) = 10 * 22 = 220$$

4. Pilih sembarang bilangan e sebagai kunci publik yang relatif prima terhadap $\varphi(n)$ yaitu $1 < e < \varphi(n)$ dan $\gcd(e, \varphi(n)) = 1$. Karena e mempunyai ketentuan $e > 1$ dan $e < \varphi(n)$, maka e dimulai dari $e = 2, 3, \dots, n$

Misalkan dipilih $e = 3$ karena relatif prima dengan $\varphi(n) = 220$.

Pembuktian, $\gcd(3, 220) = 1$ sehingga nilai e yang digunakan yaitu $e = 3$

5. Hitung kunci privat, disebut namanya d sedemikian agar $d * e \bmod \varphi(n) = 1$

Dengan mencoba nilai-nilai $d = 1, 2, 3, \dots, n$

Misalkan dipilih nilai $d = 147$ yang memenuhi syarat $d * e \bmod \varphi(n) = 1$

Pembuktian, $d * e \bmod \varphi(n) = 147 * 3 \bmod 220 = 1$ sehingga nilai d yang digunakan yaitu $d = 147$.

Berdasarkan hasil perhitungan diatas maka diperoleh pasangan kunci publik dan kunci privat algoritma RSA sebagai berikut:

1. Kunci enkripsi (*public key*) adalah pasangan $(n, e) = (253, 3)$
2. Kunci dekripsi (*private key*) adalah pasangan $(n, d) = (253, 147)$

Dengan mengambil hasil enkripsi *Beaufort Cipher* (*ciphertext1*) maka proses enkripsi dengan menggunakan kunci publik algoritma RSA dapat dilakukan dengan menggunakan persamaan (2.4) adalah sebagai berikut:

1. Ambil kunci publik (*public key*) algoritma RSA yang telah dibangkitkan sebelumnya, yaitu pasangan $(n, e) = (253, 3)$.
2. Ambil *plaintext* (m_i) yang merupakan hasil enkripsi *Beaufort Cipher* (*ciphertext1*) kemudian susun menjadi blok-blok $m_1, m_2, m_3, \dots, m_n$, lalu konversi bentuk desimal dalam tabel ASCII sehingga:

$$m_1 = c = 99$$

$$m_2 = j = 106$$

$$m_3 = z = 122$$

$$m_4 = j = 106$$

$$m_5 = b = 98$$

$$m_6 = p = 112$$

$$m_7 = c = 99$$

$$m_8 = v = 118$$

$$m_9 = a = 97$$

$$m_{10} = c = 99$$

$$m_{11} = q = 113$$

3. Enkripsi *plaintext* (m_i) menggunakan persamaan (2.4), sehingga hasilnya:

$$c_i = m_i^e \bmod n$$

$$c_1 = 99^3 \bmod 253 = 44$$

$$c_2 = 106^3 \bmod 253 = 145$$

$$c_3 = 122^3 \bmod 253 = 67$$

$$c_4 = 106^3 \bmod 253 = 145$$

$$c_5 = 98^3 \bmod 253 = 32$$

$$c_6 = 112^3 \bmod 253 = 19$$

$$c_7 = 99^3 \bmod 253 = 44$$

$$c_8 = 118^3 \bmod 253 = 50$$

$$c_9 = 97^3 \bmod 253 = 102$$

$$c_{10} = 99^3 \bmod 253 = 44$$

$$c_{11} = 113^3 \bmod 253 = 38$$

Berdasarkan hasil perhitungan diatas maka diperoleh hasil enkripsi (*ciphertext*₂) 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 disajikan pada tabel 4.2.

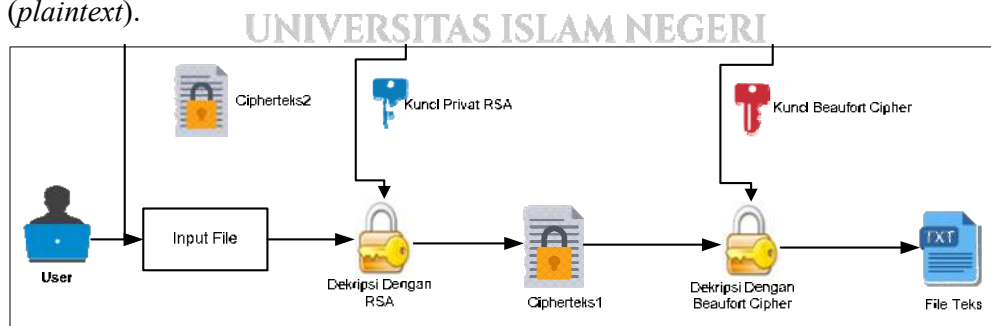
Tabel 4.2 Perbandingan Hasil Enkripsi

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

Sesuai tabel 4.2, maka dapat disimpulkan bahwa setelah di enkripsi melalui dua tahapan proses enkripsi maka diperoleh *ciphertext* yang lebih acak serta tidak memperlihatkan hubungan antara *plaintext* dengan *ciphertext*, karena hasil enkripsi kedua dengan algoritma RSA akan dikodekan kedalam nilai desimal sehingga tingkat keamanan pesan dapat ditingkatkan, akan tetapi akan menghasilkan *ciphertext* yang lebih besar dari teks aslinya, yaitu 200 bit (25 karakter), oleh karena itu perlu dikompresi untuk memperkecil ukurannya.

4.1.2 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*).

**Gambar 4.4** Skema Proses Dekripsi

Sesuai gambar 4.4 terdapat dua proses utama yang dilakukan dalam melakukan proses dekripsi. Proses pertama yaitu melakukan dekripsi menggunakan kunci privat algoritma RSA sehingga menghasilkan *ciphertext1*, lalu hasil dari proses tersebut akan di dekripsi lagi dengan menggunakan kunci algoritma *Beaufort Cipher* sehingga menghasilkan *plaintext* sebagai hasil akhir dari proses dekripsi.

Proses dekripsi dengan menggunakan kunci privat algoritma RSA dapat dilakukan dengan menggunakan persamaan (2.5) yaitu 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 *ciphertext2*) kemudian nyatakan menjadi blok-blok $c_1, c_2, c_3, \dots, c_n$.

$$c_1 = 44$$

$$c_2 = 145$$

$$c_3 = 67$$

$$c_4 = 145$$

$$c_5 = 32$$

$$c_6 = 19$$

$$c_7 = 44$$

$$c_8 = 50$$

$$c_9 = 102$$

$$c_{10} = 44$$

$$c_{11} = 38$$

3. Setiap blok c_i akan didekripsi menjadi blok m_i dengan menggunakan persamaan (2.5), sehingga hasilnya:

$$m_i = c_i^d \text{ mod } n$$

$$m_1 = 44^{147} \text{ mod } 253 = 99$$

$$m_2 = 145^{147} \text{ mod } 253 = 106$$

$$m_3 = 67^{147} \text{ mod } 253 = 122$$

$$m_4 = 145^{147} \text{ mod } 253 = 106$$

$$m_5 = 32^{147} \bmod 253 = 98$$

$$m_6 = 19^{147} \bmod 253 = 112$$

$$m_7 = 44^{147} \bmod 253 = 99$$

$$m_8 = 50^{147} \bmod 253 = 118$$

$$m_9 = 102^{147} \bmod 253 = 97$$

$$m_{10} = 44^{147} \bmod 253 = 99$$

$$m_{11} = 38^{147} \bmod 253 = 113$$

Berdasarkan hasil perhitungan diatas maka diperoleh hasil dekripsi (*ciphertext*) algoritmaRSA yang jika dikonversi kedalam karakter pada tabel ASCII akan diperoleh hasil seperti pada tabel 4.3.

Tabel 4.3 Konversi Desimal Kedalam Tabel 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 4.3 maka diperoleh hasil dekripsi pertama dengan algoritma RSA yaitu “cjzjbpcvacq” 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*). Adapun *ciphertext1* dan kunci *Beaufort Cipher* untuk proses dekripsi tahap kedua yaitu:

Ciphertext1 (C_i) c j z j b p c v a c q
 Kunci *Beaufort* (K_i) m a h y u d i

Panjang *ciphertext1* = 11, sedangkan panjang kunci = 7, karena panjang kunci *Beaufort Cipher* < panjang *ciphertext1*, maka kunci *Beaufort Cipher* tersebut akan diulang secara periodik hingga panjang kunci tersebut sama dengan panjang *ciphertext1*, yaitu:

Ciphertext1 (C_i) c j z j b p c v a c q
 Kunci *Beaufort* (K_i) m a h y u d i m a h y

Pada contoh diatas kunci *Beaufort Cipher* “mahyudi” diulang sedemikian rupa hingga panjang kunci sama dengan panjang *ciphertext1*. Kemudian setelah panjang kunci sama dengan panjang *ciphertext1*, proses enkripsi dilakukan terlebih dahulu akan dikonversi ke desimal pada tabel ASCII sehingga hasilnya dapat disajikan pada tabel 4.4.

Tabel 4.4 Proses Mengubah *Ciphertext1* Menjadi Desimal

<i>Ciphertext1</i> (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 (2.2) sehingga diperoleh hasilnya sebagai berikut:

Untuk karakter (C_1) = c dan kunci (K_1) = m

$$\begin{aligned} M_1 &= (K_1 - C_1) \bmod 26 = (m - c) \bmod 26 + 97 \\ &= (109 - 99) \bmod 26 + 97 \\ &= 107 \text{ (huruf "k" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter (C_2) = j dan kunci (K_2) = a

$$\begin{aligned} M_2 &= (K_2 - C_2) \bmod 26 = (a - j) \bmod 26 + 97 \\ &= (97 - 106) \bmod 26 + 97 \end{aligned}$$

$$= 114 \text{ (huruf "r" dalam tabel ASCII)}$$

Untuk karakter $(C_3) = z$ dan kunci $(K_3) = h$

$$\begin{aligned} M_3 &= (K_3 - C_3) \bmod 26 = (h - z) \bmod 26 + 97 \\ &= (104 - 122) \bmod 26 + 97 \\ &= 105 \text{ (huruf "i" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(C_4) = j$ dan kunci $(K_4) = y$

$$\begin{aligned} M_4 &= (K_4 - C_4) \bmod 26 = (y - j) \bmod 26 + 97 \\ &= (121 - 106) \bmod 26 + 97 \\ &= 112 \text{ (huruf "p" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(C_5) = b$ dan kunci $(K_5) = u$

$$\begin{aligned} M_5 &= (K_5 - C_5) \bmod 26 = (u - b) \bmod 26 + 97 \\ &= (117 - 98) \bmod 26 + 97 \\ &= 116 \text{ (huruf "t" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(C_6) = p$ dan kunci $(K_6) = d$

$$\begin{aligned} M_6 &= (K_6 - C_6) \bmod 26 = (d - p) \bmod 26 + 97 \\ &= (100 - 112) \bmod 26 + 97 \\ &= 111 \text{ (huruf "o" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(C_7) = c$ dan kunci $(K_7) = i$

$$\begin{aligned} M_7 &= (K_7 - C_7) \bmod 26 = (i - c) \bmod 26 + 97 \\ &= (105 - 99) \bmod 26 + 97 \\ &= 103 \text{ (huruf "g" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(C_8) = v$ dan kunci $(K_8) = m$

$$\begin{aligned} M_8 &= (K_8 - C_8) \bmod 26 = (m - v) \bmod 26 + 97 \\ &= (109 - 118) \bmod 26 + 97 \\ &= 114 \text{ (huruf "r" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(C_9) = a$ dan kunci $(K_9) = a$

$$\begin{aligned} M_9 &= (K_9 - C_9) \bmod 26 = (a - a) \bmod 26 + 97 \\ &= (97 - 97) \bmod 26 + 97 \\ &= 97 \text{ (huruf "a" dalam tabel ASCII)} \end{aligned}$$

Untuk karakter $(C_{10}) = c$ dan kunci $(K_{10}) = h$

$$\begin{aligned}
 M_{10} &= (K_{10} - C_{10}) \bmod 26 = (h - c) \bmod 26 + 97 \\
 &= (104 - 99) \bmod 26 + 97 \\
 &= 102 \text{ (huruf "f" dalam tabel ASCII)}
 \end{aligned}$$

Untuk karakter $(C_{11}) = q$ dan kunci $(K_{11}) = y$

$$\begin{aligned}
 M_{11} &= (K_{11} - C_{11}) \bmod 26 = (y - q) \bmod 26 + 97 \\
 &= (121 - 113) \bmod 26 + 97 \\
 &= 105 \text{ (huruf "i" dalam tabel ASCII)}
 \end{aligned}$$

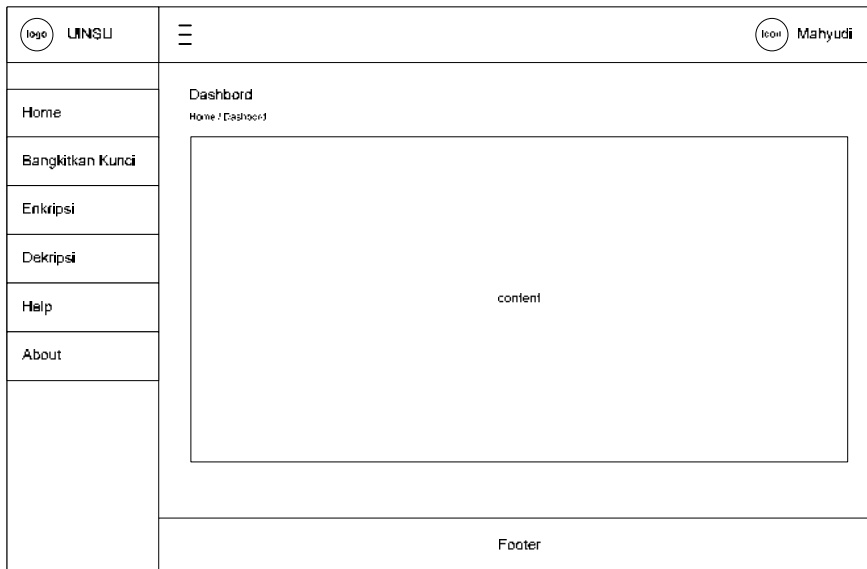
Berdasarkan hasil perhitungan diatas setelah melakukan proses dekripsi dengan menggunakan algoritma *Beaufort Cipher* untuk tahap kedua maka diperoleh kembali *plaintext* aslinya yaitu "kriptografi".

4.2 Perancangan Sistem

Perancangan *interface* atau tampilan antarmuka pengguna sistem pada aplikasi ini berguna sebagai perantara komunikasi pengguna dengan sistem. Sistem dibangun berbasis *web* dengan menggunakan bahasa pemrograman *php*. *Interface* yang akan dirancang pada sistem ini memiliki enam bagian utama, yaitu halaman utama (*home*), halaman bangkitkan kunci, halaman enkripsi, halaman dekripsi, halaman *help*, dan halaman *about*.

4.2.1 Perancangan Halaman Utama

Halaman utama merupakan halaman pembuka yang akan ditampilkan pertama kali saat aplikasi dijalankan. Gambar 4.5 merupakan rancangan *interface* dari halaman utama sistem.



Gambar 4.5 Rancangan *Interface* Halaman Utama

4.2.2 Perancangan Halaman Bangkitkan Kunci

Halaman bangkitkan kunci merupakan sebuah *form* yang dirancang untuk membangkitkan kunci publik (*public key*) dan kunci privat (*private key*) dari algoritma RSA. Kunci publik digunakan untuk mengenkripsi pesan teks, sedangkan kunci privat digunakan untuk mendekripsi pesan teks. Rancangan *interface* dari halaman bangkitkan kunci dapat disajikan pada gambar 4.6.

Gambar 4.6 Rancangan *Interface* Halaman Bangkitkan Kunci

4.2.3 Perancangan Halaman Enkripsi

Halaman enkripsi merupakan sebuah *form* yang dirancang untuk melakukan proses enkripsi pesan dalam *file* teks (*plaintext*) dengan menggunakan kombinasi algoritma kriptografi simetris *Beaufort Cipher* dan algoritma kriptografi asimetris RSA dalam skema super enkripsi. Adapun rancangan *interface* dari halaman enkripsi dapat disajikan pada gambar 4.7.




The image shows a web application interface for encryption. On the left is a sidebar with a logo and the text 'UINSU' at the top, and navigation links: Home, Bangkitkan Kunci, Enkripsi, Dekripsi, Help, and About. The main content area is titled 'Enkripsi' and contains the following elements:

- A sub-header: 'Enkripsi Plainteks Dengan Algoritma Beaufort Cipher dan RSA'
- A button labeled 'Read Kunci Publik RSA' followed by two empty input fields.
- A section titled 'Masukkan File Teks (Plainteks)' with a 'Choose File' button and a large empty text area below it.
- A section titled 'Masukkan Kunci Beaufort Cipher' with a large empty text area below it.
- Three buttons: 'Enkripsi Beaufort', 'Enkripsi RSA', and 'Simpan Cipherteks'.
- Two output areas: 'Hasil Enkripsi (Cipherteks) Beaufort Cipher' and 'Hasil Enkripsi (Cipherteks) RSA', each with a large empty text area below it.
- A 'Footer' section at the bottom.

Gambar 4.7 Rancangan *Interface* Halaman Enkripsi

4.2.4 Perancangan Halaman Dekripsi




Halaman dekripsi dirancang untuk melakukan proses dekripsi *file* teks (*ciphertext*) dengan menggunakan kombinasi algoritma kriptografi *Beaufort Cipher* dan algoritma RSA dalam skema super enkripsi. Adapun rancangan *interface* dari halaman dekripsi dapat disajikan pada gambar 4.8.

 UINSU		 Mahyudi
Home Bangkitkan Kunci Enkripsi Dekripsi Help About	<div style="text-align: center;"> Dekripsi <small>Home / Dekripsi</small> </div> <div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;">Dekripsi Cipherteks Dengan Algoritma Beaufort Cipher dan RSA</p> <div style="display: flex; justify-content: space-around;"> <input type="text" value="Read Kunci Privat RSA"/> <input type="text"/> <input type="text"/> </div> <p>Masukkan File Teks (Cipherteks)</p> <div style="display: flex; align-items: center;"> <input type="button" value="Choose File"/> <input style="width: 200px;" type="text"/> </div> <div style="border: 1px solid black; height: 30px; margin: 5px 0;"></div> <p>Masukkan Kunci Beaufort Cipher</p> <div style="border: 1px solid black; height: 20px; margin: 5px 0;"></div> <div style="display: flex; justify-content: center; gap: 20px; margin: 10px 0;"> <input type="button" value="Dekripsi RSA"/> <input type="button" value="Dekripsi Beaufort"/> <input type="button" value="Simpan Plainteks"/> </div> <div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; width: 45%; height: 30px; margin: 5px 0;"></div> <div style="border: 1px solid black; width: 45%; height: 30px; margin: 5px 0;"></div> </div> </div>	
Footer		

Gambar 4.8 Rancangan *Interface* Halaman Dekripsi

4.2.5 Perancangan Halaman *Help*

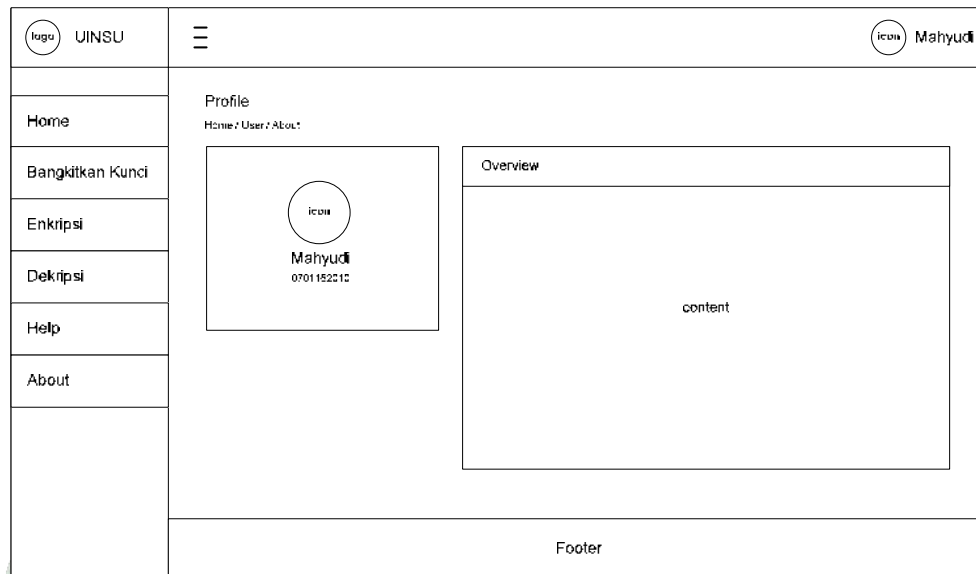
Halaman *help* dirancang untuk menampilkan informasi mengenai proses bangkitkan kunci, proses enkripsi dan proses dekripsi yang terdapat pada aplikasi yang dibuat. Adapun rancangan *interface* dari halaman *help* dapat dilihat pada gambar 4.9.

 UINSU		 Mahyudi								
Home Bangkitkan Kunci Enkripsi Dekripsi Help About	<div style="text-align: center;"> Help <small>Home / Help / Help</small> </div> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 25%;">Bangkitkan Kunci</td> <td style="width: 25%;">Enkripsi</td> <td style="width: 25%;">Dekripsi</td> <td style="width: 25%;"></td> </tr> <tr> <td colspan="4" style="height: 150px; vertical-align: middle;">content</td> </tr> </table>		Bangkitkan Kunci	Enkripsi	Dekripsi		content			
Bangkitkan Kunci	Enkripsi	Dekripsi								
content										

Gambar 4.9 Rancangan *Interface* Halaman *Help*

4.2.6 Perancangan Halaman *About*

Halaman *about* dirancang untuk menampilkan informasi singkat mengenai *profile user* pada aplikasi yang dibuat. Adapun rancangan *interface* dari halaman *about* dapat dilihat pada gambar 4.10.



Gambar 4.10 Rancangan *Interface* Halaman *About*

4.3 Implementasi Program

Implementasi program 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* yang terdiri dari enam buah halaman, yaitu halaman utama, halaman bagkitkan kunci, halaman enkripsi, halaman dekripsi, halaman *help*, dan halaman *about*. Adapun ruang lingkup spesifikasi kebutuhan perangkat lunak dan perangkat keras yang digunakan saat membangun dan menjalankan aplikasi ini dapat dilihat penjelasannya pada bab tiga.

4.3.1 Implementasi Halaman Utama

Halaman utama atau halaman *home* berfungsi sebagai tampilan utama pada saat aplikasi dijalankan. Pada halaman utama terdapat lima buah menu yang dapat diakses oleh *user* yaitu terdiri dari menu bangkitkan kunci, menu enkripsi, menu dekripsi, menu *help*, dan menu *about*. Gambar 4.11 merupakan tampilan dari halaman utama.



Gambar 4.11 Tampilan Halaman Utama

4.3.2 Implementasi Halaman Bangkitkan Kunci

Halaman bangkitkan kunci berfungsi untuk membangkitkan kunci publik (*public key*) dan kunci privat (*private key*) dari algoritma RSA. Kunci publik digunakan untuk mengenkripsi pesan (*plaintext*) yang terdapat dalam *file* teks dengan format **.txt*, sedangkan kunci privat digunakan untuk mendekripsi pesan (*ciphertext*) yang terdapat dalam *file* teks dengan format **.txt*. Gambar 4.12 merupakan tampilan dari halaman bangkitkan kunci.

Gambar 4.12 Tampilan Halaman Bangkitkan Kunci

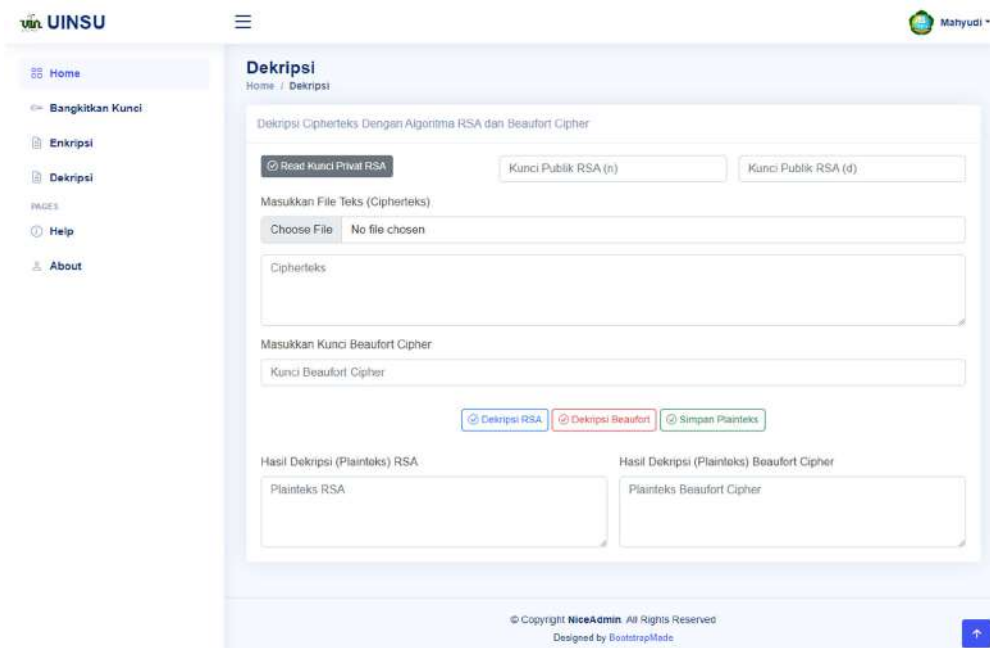
4.3.3 Implementasi Halaman Enkripsi

Halaman enkripsi berfungsi untuk melakukan proses enkripsi pesan (*plaintext*) yang terdapat dalam *file* teks dengan format **.txt* dengan menggunakan kombinasi algoritma kriptografi *Beaufort Cipher* dan algoritma RSA dalam skema super enkripsi. Gambar 4.13 merupakan tampilan dari halaman enkripsi.

Gambar 4.13 Tampilan Halaman Enkripsi

4.3.4 Implementasi Halaman Dekripsi

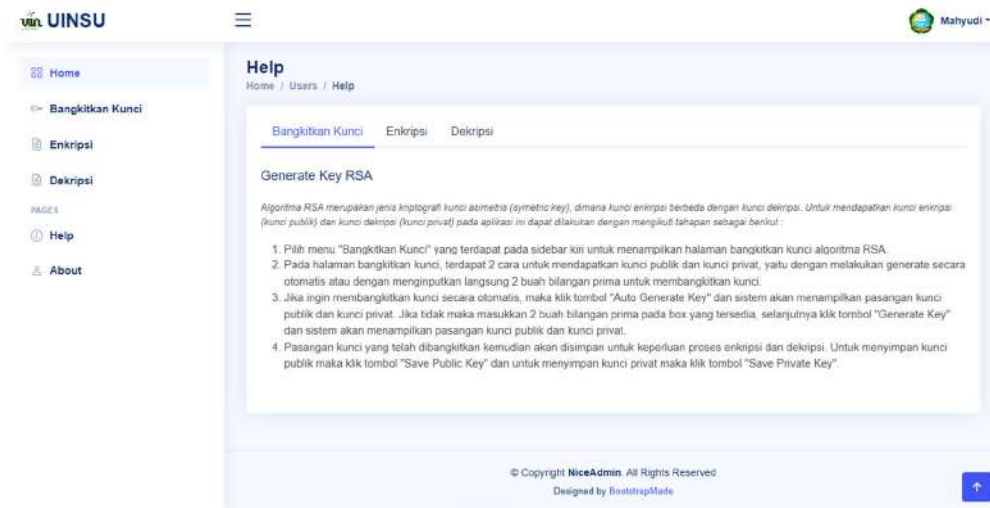
Halaman dekripsi berfungsi untuk melakukan proses dekripsi pesan (*ciphertext*) yang terdapat dalam *file* teks dengan format **.txt* dengan menggunakan kombinasi algoritma kriptografi *Beaufort Cipher* dan algoritma RSA dalam skema super enkripsi. Gambar 4.14 merupakan tampilan dari halaman dekripsi.



Gambar 4.14 Tampilan Halaman Dekripsi

4.3.5 Implementasi Halaman *Help*

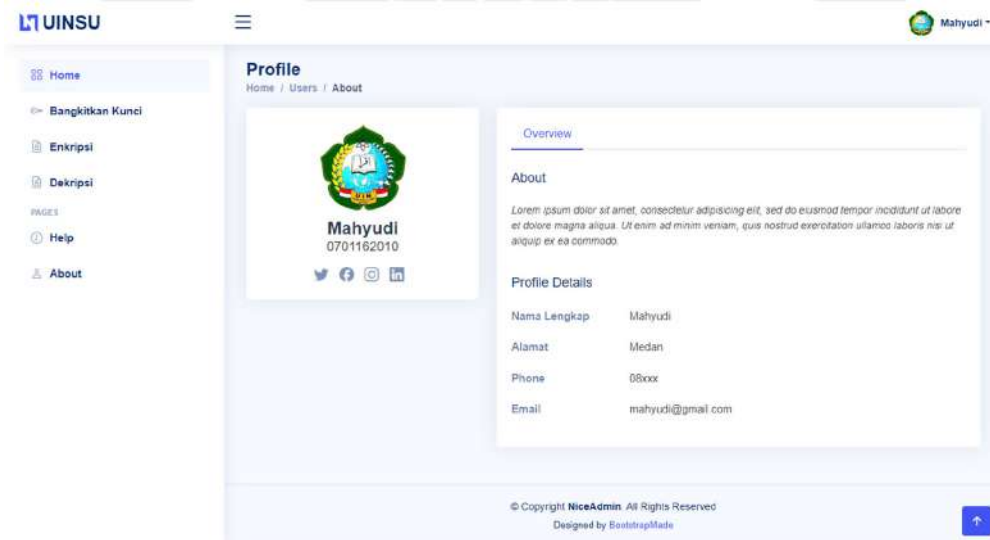
Halaman *help* berfungsi untuk menampilkan informasi panduan kepada *user* mengenai proses dalam membangkitkan kunci algoritma RSA, proses enkripsi dan proses dekripsi yang terdapat pada aplikasi yang dibuat. Gambar 4.15 merupakan tampilan dari halaman *help*.



Gambar 4.15 Tampilan Halaman *Help*

4.3.6 Implementasi Halaman *About*

Halaman *about* berfungsi untuk menampilkan informasi singkat mengenai *profile* pembuat aplikasi yang berisi mengenai nama lengkap, alamat, nomor telepon, dan alamat email. Gambar 4.16 merupakan tampilan dari halaman *about*.



Gambar 4.16 Tampilan Halaman *About*

4.4 Hasil Pengujian

Hasil pengujian merupakan hasil dari pengujian kemampuan atau keakuratan metode yang digunakan dalam menyelesaikan masalah yang diteliti. Dalam hal ini, penerapan skema super enkripsi dengan menggunakan kombinasi algoritma kriptografi *Beaufort Cipher* dan algoritma RSA yang telah dibangun akan diuji dengan menggunakan sampel data atau *file* uji. Objek yang menjadi inputan data sebagai sampel data pada pengujian ini adalah berupa *file* teks berformat **.txt* dengan *size* atau ukuran yang beragam.

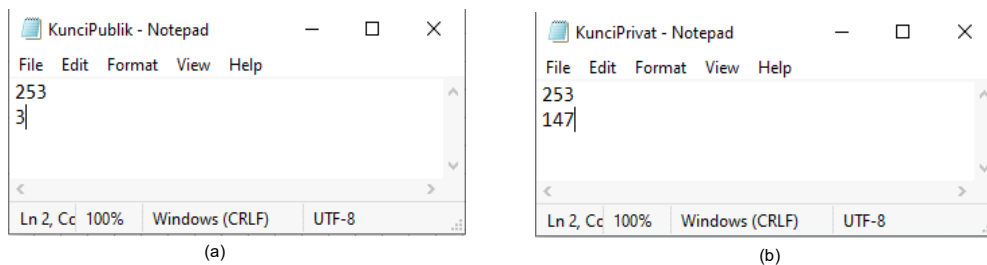
4.4.1 Hasil Pengujian Bangkitkan Kunci

Setelah memilih menu “Bangkitkan Kunci” yang terdapat pada halaman utama, maka sistem akan menampilkan halaman bangkitkan kunci untuk proses membangkitkan kunci publik (*public key*) dan kunci privat (*private key*) 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 secara manual dengan mengisi langsung dua buah bilangan prima p dan q pada *textbox* masing-masing yang tersedia pada halaman bangkitkan kunci.

Dalam membangkitkan kunci algoritma RSA secara manual, maka *user* harus menginputkan terlebih dahulu dua buah bilangan prima p dan q , selanjutnya *user* dapat menekan tombol “Generate Key”. Jika bilangan yang dimasukkan bukan bilangan prima maka sistem akan menampilkan pesan *error* dan *user* harus memasukkan ulang untuk bilangan prima p dan q . Jika bilangan yang dimasukkan merupakan bilangan prima maka sistem akan menampilkan kunci publik (*public key*) dan kunci privat (*private key*) algoritma RSA. Adapun cara kedua untuk membangkitkan kunci yaitu dengan menekan tombol “Auto Generate Key” dan sistem akan mengacak bilangan prima dan akan ditampilkan pada masing-masing *textbox* p dan q serta sistem akan menampilkan kunci publik (*public key*) dan kunci privat (*private key*) algoritma RSA seperti terlihat pada gambar 4.17.

Gambar 4.17 Hasil Pengujian Bangkitkan Kunci

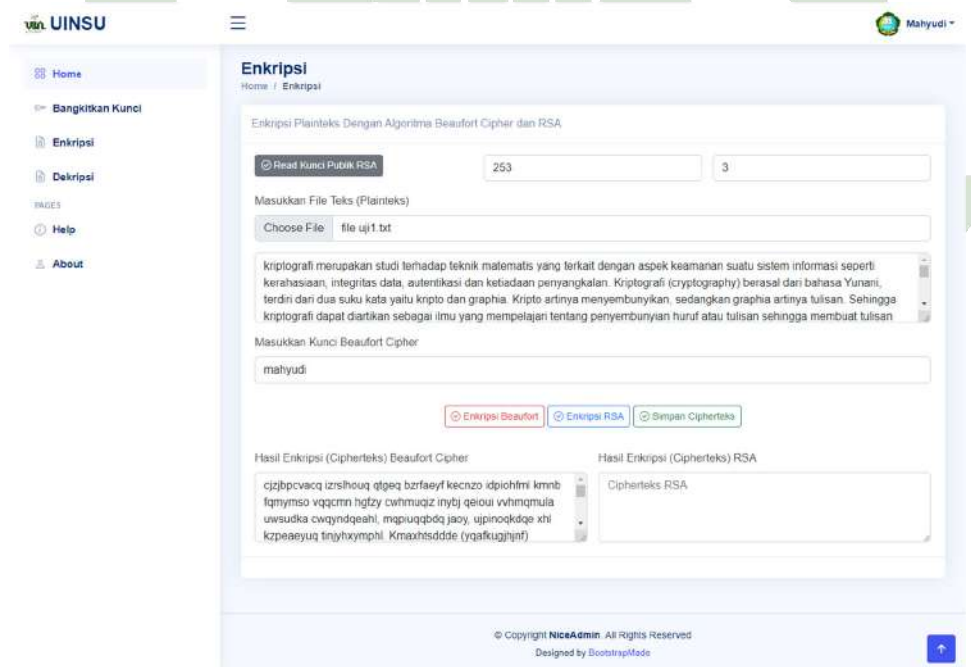
Sesuai gambar 4.17, nilai bilangan prima yang dihasilkan untuk $p = 11$ dan $q = 23$. Sedangkan untuk nilai $n = 253$ yang merupakan hasil dari $p * q$. Adapun kunci publik yang dihasilkan yaitu pasangan n dan e (253, 3) dan kunci privat yang dihasilkan yaitu pasangan n dan d (253, 147). Pasangan kunci yang berhasil dibangkitkan selanjutnya akan disimpan untuk keperluan proses enkripsi dan dekripsi. Untuk menyimpan pasangan kunci dapat dilakukan dengan memilih tombol “*Save Public Key*” untuk menyimpan kunci publik dan tombol “*Save Private Key*” untuk menyimpan kunci privat. Adapun tampilan dari pasangan kunci publik dan kunci privat algoritma RSA setelah berhasil disimpan dapat dilihat pada gambar 4.18.



Gambar 4.18 Hasil Kunci Algoritma RSA (a) Kunci Publik (b) Kunci Privat

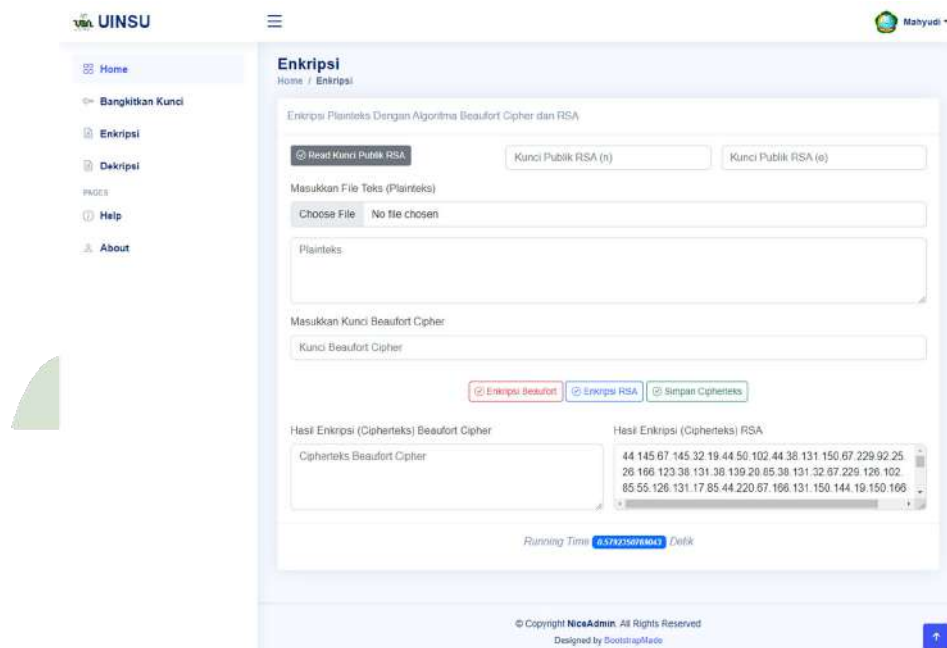
4.4.2 Hasil Pengujian Enkripsi

Setelah memilih menu “Enkripsi” yang terdapat pada halaman utama, maka sistem akan menampilkan halaman enkripsi untuk melakukan proses enkripsi pesan (*plaintext*) yang terdapat dalam *file* teks dengan menggunakan kombinasi algoritma *Beaufort Cipher* dan algoritma RSA dalam skema super enkripsi. Untuk memulai proses enkripsi, maka *user* terlebih dahulu memasukkan *file* kunci publik (*public key*) algoritma RSA yang telah dibangkitkan sebelumnya, yaitu dengan memilih tombol “Read Kunci Publik RSA” dan sistem akan menampilkan kunci publik pada masing-masing *textbox* yang terdapat pada halaman enkripsi. Tahap selanjutnya memasukkan *file* teks yang akan di enkripsi dengan memilih tombol “Choose File” dan sistem akan menampilkan *preview* isi dari *file* teks yang dimasukkan pada *textbox plaintext*. Setelah itu dilanjutkan dengan memasukkan kunci algoritma *Beaufort Cipher* dan dilanjutkan dengan memilih tombol “Enkripsi Beaufort” dan sistem akan menampilkan hasil enkripsi pertama (*ciphertext1*) algoritma *Beaufort Cipher* seperti terlihat pada gambar 4.19.



Gambar 4.19 Hasil Pengujian Enkripsi *Beaufort Cipher*

Gambar 4.19 merupakan hasil pengujian enkripsi pada *file* teks dengan format **.txt* menggunakan kunci algoritma *Beaufort Cipher*. *Input* data yang di uji menggunakan sampel data *file* uji1 dengan *size* 32 KB (*kilo bytes*). Hasil enkripsi pertama dari algoritma *Beaufort Cipher* kemudian akan di enkripsi lagi dengan memilih tombol “Enkripsi RSA” dan sistem akan menampilkan hasil enkripsi kedua (*ciphertext2*) sebagai hasil akhir dari proses enkripsi dalam skema super enkripsi. Adapun hasil enkripsinya dapat dilihat pada gambar 4.20.

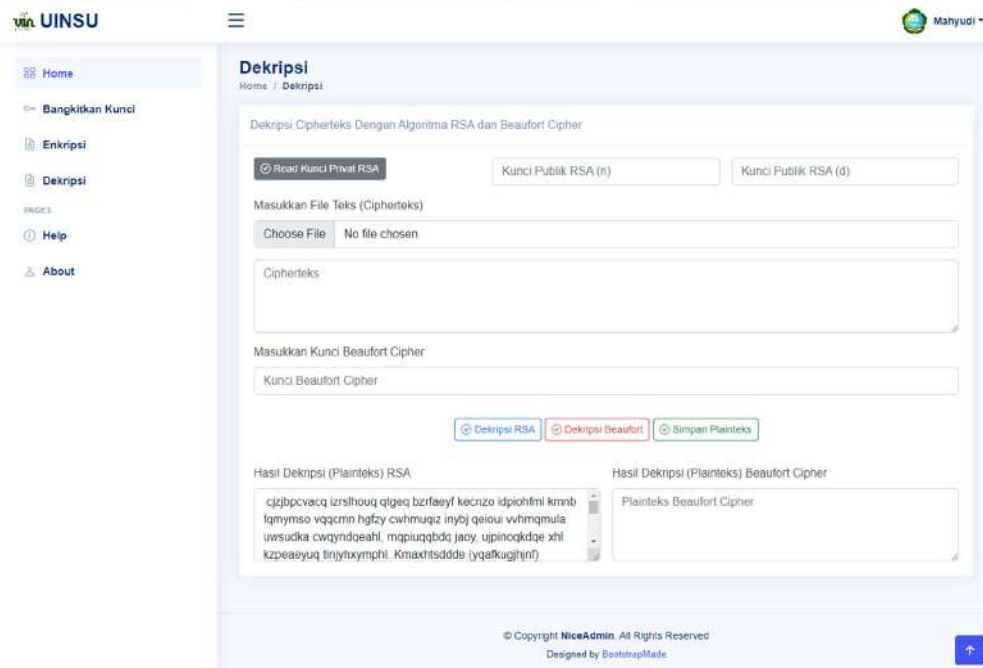


Gambar 4.20 Hasil Pengujian Enkripsi RSA

Gambar 4.20 merupakan hasil pengujian enkripsi kedua dengan menggunakan kunci publik algoritma RSA. Karakter (*string*) yang terdapat pada *file* teks yang telah di enkripsi akan menghasilkan *ciphertext* dalam bentuk bilangan desimal dengan estimasi waktu enkripsi selama 0.579 ms (*millisecond*). *Ciphertext* yang dihasilkan dari proses enkripsi kedua dengan menggunakan algoritma RSA akan terlihat acak dan tidak memiliki makna karena hasil enkripsi dalam bentuk bilangan desimal, sehingga keamanan dan kerahasiaan sebuah informasi pada *file* teks dapat terjaga karena tidak memperlihatkan korelasi antara *plaintext* dengan *ciphertext*.

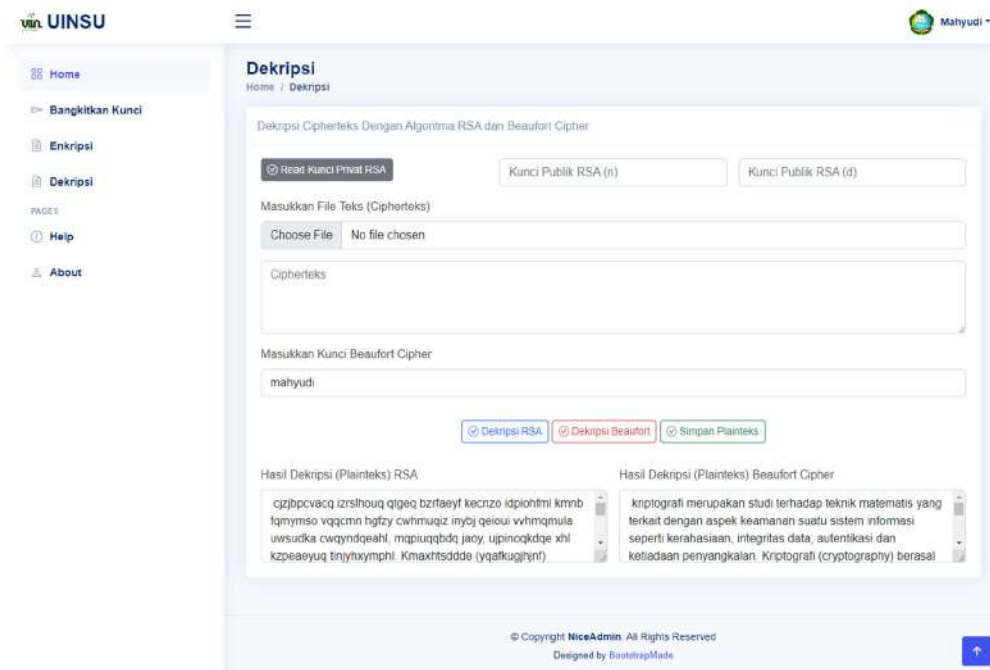
4.4.3 Hasil Pengujian Dekripsi

Setelah memilih menu “Dekripsi” yang terdapat pada halaman utama, maka sistem akan menampilkan halaman dekripsi untuk melakukan proses dekripsi pesan (*ciphertext*) yang terdapat dalam *file* teks dengan menggunakan kombinasi algoritma *Beaufort Cipher* dan algoritma RSA dalam skema super enkripsi. Untuk memulai proses dekripsi, maka *user* terlebih dahulu memasukkan *file* kunci privat (*private key*) algoritma RSA yang telah dibangkitkan sebelumnya, yaitu dengan memilih tombol “Read Kunci Privat RSA” dan sistem akan menampilkan kunci privat pada masing-masing *textbox* yang terdapat pada halaman dekripsi. Tahap selanjutnya memasukkan *file* teks (*ciphertext*) yang akan di dekripsi dengan memilih tombol “Choose File” dan sistem akan menampilkan *preview* isi dari *file* teks yang dimasukkan pada *textbox ciphertext*. Setelah itu dilanjutkan dengan memasukkan kunci algoritma *Beaufort Cipher* dan dilanjutkan dengan memilih tombol “Dekripsi RSA” dan sistem akan menampilkan hasil dekripsi pertama algoritma RSA seperti terlihat pada gambar 4.21.



Gambar 4.21 Hasil Pengujian Dekripsi RSA

Gambar 4.21 merupakan hasil pengujian dekripsi pada *file* teks (*ciphertext*) dengan menggunakan kunci privat algoritma RSA. Hasil dekripsi pertama dari algoritma RSA kemudian akan di dekripsi lagi dengan memilih tombol “Dekripsi *Beaufort*” dan sistem akan menampilkan hasil dekripsi kedua (*plaintext*) sebagai hasil akhir dari proses dekripsi dalam skema super enkripsi. Adapun hasil dekripsinya dapat dilihat pada gambar 4.22.



Gambar 4.22 Hasil Pengujian Dekripsi *Beaufort Cipher*

Gambar 4.22 merupakan hasil pengujian dekripsi kedua dengan menggunakan kunci *Beaufort Cipher*. Karakter (*string*) yang terdapat pada *file* teks yang telah di dekripsi akan menghasilkan *plaintext* yang sama persis dengan *file* teks aslinya dengan estimasi waktu enkripsi selama 0.433 ms (*millisecond*).

4.4.4 Hasil Pengujian *Black box Testing*

Pengujian *black box* (*black box testing*) adalah salah satu metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada *input* dan *output* aplikasi (apakah sudah sesuai dengan apa yang diharapkan atau belum). Pengujian dengan metode *black box testing* dilakukan dengan cara

memberikan sejumlah *input* pada program. *Input* tersebut kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi dapat menghasilkan *output* yang sesuai dengan yang diinginkan dan sesuai pula dengan fungsi dasar dari program tersebut. Apabila dari *input* yang diberikan, proses dapat menghasilkan *output* yang sesuai dengan kebutuhan fungsionalnya, maka program yang dibuat sudah benar, tetapi apabila *output* yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program tersebut, dan selanjutnya dilakukan penelusuran perbaikan untuk memperbaiki kesalahan yang terjadi.

Adapun hasil pengujian *black box testing* sistem dapat diuraikan sebagai berikut:

1. *Black box Testing* Bangkitkan Kunci

Berikut adalah hasil *black box testing* pada proses bangkitkan kunci publik (*public key*) dan kunci privat (*private key*) algoritma RSA yang dapat disajikan pada tabel 4.5.

Tabel 4.5 *Black box Testing* Bangkitkan Kunci

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

Tabel 4.5 pengujian *black box testing* dalam proses bangkitkan kunci, kasus pengujian yang dilakukan antara lain “Bangkitkan Kunci” dan “Simpan Kunci” dengan status hasil pengujian berhasil.

2. *Black box Testing* Enkripsi

Berikut adalah hasil *black box testing* pada proses enkripsi yang dapat disajikan pada tabel 4.6.

Tabel 4.6 *Black box Testing* Enkripsi

No.	Kasus Uji	Langkah Uji	Hasil	Status
1.	<i>Input</i> Kunci Publik RSA	Mengimport <i>file</i> kunci publik RSA dengan memilih tombol “ <i>Read</i> Kunci Publik RSA”	Sistem dapat membaca dan menampilkan isi dari <i>file</i> kunci publik algoritma RSA	
2.	<i>Input</i> File Teks	Mengimport <i>file</i> teks dengan memilih tombol “ <i>Choose</i> File”	Sistem hanya dapat menerima format <i>file</i> . <i>txt</i> dan sistem dapat menampilkan isi dari <i>file</i> teks	
3.	<i>Input</i> Kunci <i>Beaufort</i> <i>Cipher</i>	Memasukkan kunci <i>Beaufort</i> <i>Cipher</i>	Sistem dapat menerima kunci	
4.	Enkripsi <i>Beaufort</i> <i>Cipher</i>	Menkripsi <i>file</i> teks dengan memilih tombol “Enkripsi <i>Beaufort</i> ”	Sistem dapat menampilkan hasil enkripsi algoritma <i>Beaufort</i> <i>Cipher</i>	
5.	Enkripsi RSA	Menkripsi hasil <i>Beaufort</i> <i>Cipher</i> dengan memilih tombol “Enkripsi RSA”	Sistem dapat menampilkan hasil enkripsi algoritma RSA	

6.	Simpan <i>Ciphertext</i>	Menyimpan hasil enkripsi dengan memilih tombol “Simpan <i>Cipherteks</i> ”	Sistem dapat menyimpan hasil enkripsi (<i>ciphertext</i>) kedalam <i>file</i> baru dengan format <i>*.txt</i>	
----	-----------------------------	--	---	--

Tabel 4.6 pengujian *black box testing* dalam proses enkripsi, kasus pengujian yang dilakukan antara lain “*Input Kunci Publik RSA*”, “*Input File Teks*”, “*Input Kunci Beaufort Cipher*”, “*Enkripsi Beaufort Cipher*”, “*Enkripsi RSA*”, dan “*Simpan Ciphertext*”.

3. *Black box Testing* Dekripsi

Berikut adalah hasil *black box testing* pada proses dekripsi yang dapat disajikan pada tabel 4.7.

Tabel 4.7 *Black box Testing* Dekripsi

No.	Kasus Uji	Langkah Uji	Hasil	Status
1.	<i>Input Kunci Privat RSA</i>	Mengimport <i>file</i> kunci privat RSA dengan memilih tombol “ <i>Read Kunci Privat RSA</i> ”	Sistem dapat membaca dan menampilkan isi dari <i>file</i> kunci privat algoritma RSA	
2.	<i>Input File Teks</i>	Mengimport <i>file</i> teks (<i>ciphertext</i>) dengan memilih tombol “ <i>Choose File</i> ”	Sistem hanya dapat menerima format <i>file</i> <i>*.txt</i> dan sistem dapat menampilkan isi dari <i>file</i> teks	
3.	<i>Input Kunci Beaufort Cipher</i>	Memasukkan kunci <i>Beaufort Cipher</i>	Sistem dapat menerima kunci	

4.	Dekripsi RSA	Mendekripsi <i>file ciphertext</i> dengan memilih tombol “Dekripsi RSA”	Sistem dapat menampilkan hasil dekripsi algoritma RSA	
5.	Dekripsi <i>Beaufort Cipher</i>	Mendekripsi <i>file ciphertext</i> dengan memilih tombol “Dekripsi <i>Beaufort</i> ”	Sistem dapat menampilkan hasil dekripsi algoritma <i>Beaufort Cipher</i>	
6.	Simpan <i>Plaintext</i>	Menyimpan hasil dekripsi dengan memilih tombol “Simpan <i>Plaintext</i> ”	Sistem dapat menyimpan hasil dekripsi (<i>plaintext</i>) kedalam <i>file</i> baru dengan format <i>*.txt</i>	

Tabel 4.7 pengujian *black box testing* dalam proses dekripsi, kasus pengujian yang dilakukan antara lain “*Input Kunci Privat RSA*”, “*Input File Teks*”, “*Input Kunci Beaufort Cipher*”, “*Dekripsi RSA*”, “*Dekripsi Beaufort Cipher*”, dan “*Simpan Plaintext*”.

Medan, Februari 2023
Penguji,

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

.....