

## BAB IV

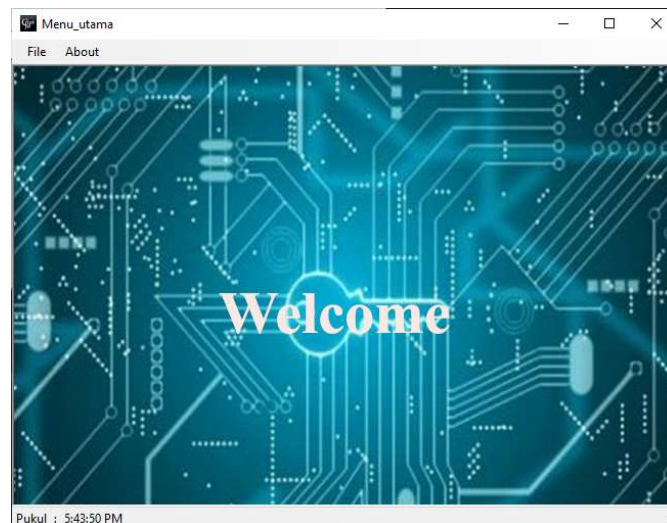
### HASIL DAN UJI COBA

#### IV.1. Tampilan Hasil

Berikut ini dijelaskan mengenai tampilan hasil dari *Pemanfaatan Kriptografi Algoritma Blowfish Kriptografi Algoritma Rijndael dalam Pengamanan Data File* yang dapat dilihat sebagai berikut :

##### 1. Tampilan Form Utama

*Form* ini merupakan tampilan awal sistem yang menampilkan aktifitas sistem dalam bentuk menu-menu sehingga mempermudah *user* dalam penggunaannya. *User* dapat memilih menu-menu terdapat 2 (dua) *Menu file*, *File* dan *About*, menu *file* berisi sub menu (enkripsi,dekripsi dan exit) dan menu *about* berisi sub menu ( Info profile dan Info about) yang tersedia pada sistem, yang ditunjukkan pada gambar IV.1 :



**Gambar IV.1. Tampilan Form Utama**

##### 2. Tampilan Form Enkripsi

Tampilan *form enkripsi* berisi tentang pengamanan data file. Pada *form* ini data *file* akan di *enkripsi* dengan Algoritma *Blowfish* kolaborasi dengan Algoritma *Rijndael* sehingga menjadi *cipher file*. Berikut halaman *Enkripsi* pada gambar IV.2 :

The screenshot shows a Java application window titled "Menu\_utama" with a menu bar containing "File" and "About". The main content area has a header banner with the word "ENKRIPSI" in white text on a blue circuit background. Below the banner is a form for encryption. It includes a text field labeled "File Untuk di Enkripsi" with a placeholder "Klik Browser Untuk Memilih File" and a "Browser" button. Below that is a "File Tujuan" text field and a "Change" button. Further down are two text fields for "Key" and "Konfirmasi Key", and a large "Proses" button. The status bar at the bottom shows the time "Pukul : 5:56:26 PM".

**Gambar IV.2. Tampilan Form Enkripsi**

### 3. Tampilan From Dekripsi

Tampilan *Form dekripsi* untuk menghasilkan *plaint file blowfish* yang didapat dari hasil *dekripsi cipherfile* . kolaborasi antara algoritma *Blowfis* dan *Algoritma Rijndael* yang ditunjukkan pada gambar IV.3 :

The screenshot shows a Java application window titled "Menu\_utama" with a menu bar containing "File" and "About". The main content area has a header banner with the word "DEKRIPSI" in white text on a blue circuit background. Below the banner is a form for decryption. It includes a text field labeled "File Untuk di Dekripsi" with a placeholder "Klik Browser Untuk Memilih File" and a "Browser" button. Below that is a "File Tujuan" text field and a "Change" button. Further down are two text fields for "Key" and "Konfirmasi Key", and a large "Proses" button. The status bar at the bottom shows the time "Pukul : 6:00:57 PM".

### Gambar IV.3. Tampilan Form Dekripsi

#### 4. Tampilan Form Info Profile

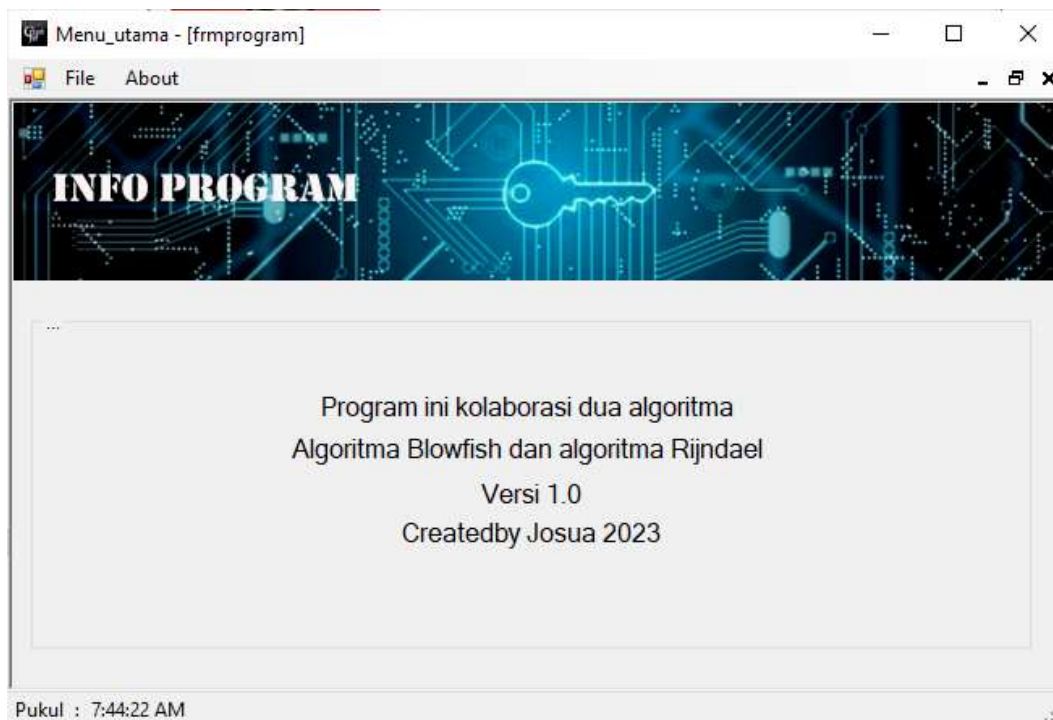
Tampilan *form* tentang yang menampilkan tentang biodata diri yang ditunjukkan pada gambar IV.4 :



### Gambar IV.4. Tampilan Form Info Profile

#### 5. Tampilan From Info Program

Tampilan *form* info program yang menampilkan tentang info program yang ditunjukkan pada gambar IV.5 :



**Gambar IV.5. Tampilan Form Info Program**

## **IV.2. Uji Coba Hasil**

Uji coba terhadap sistem bertujuan untuk memastikan bahwa sistem sudah berada pada kondisi siap pakai. Instrumen yang digunakan untuk melakukan pengujian ini yaitu dengan menggunakan :

1. Satu unit laptop dengan spesifikasi sebagai berikut:
  - a. *Processor Dual Core i3*
  - b. *Memory 4 Gb*
  - c. *Hardisk 1TB*
2. Perangkat Lunak dengan spesifikasi sebagai berikut:
  - a. *Visual Basic 2015*
  - b. *Windows 10 Pro 64 bit*

Tahap pengujian sistem merupakan lanjutan dari tahap implementasi sistem. Fungsi dari tahap pengujian sistem adalah untuk membuktikan bahwa sistem yang telah diimplementasikan dari hasil analisis dan perancangan sistem telah berjalan dengan baik.

Pengujian akan dilakukan pada *file document* (\*.pdf, \*.doc, \*.txt), Gambar (\*.jpeg, \*.png, \*.bmp), Audio (\*.acc, \*.mp3, \*.wav). Data *file* yang akan diuji akan di *enkripsi* menggunakan Algoritma *Blowfish* dan Algoritma *Rijndael*.

#### 1. Proses *Enkripsi*

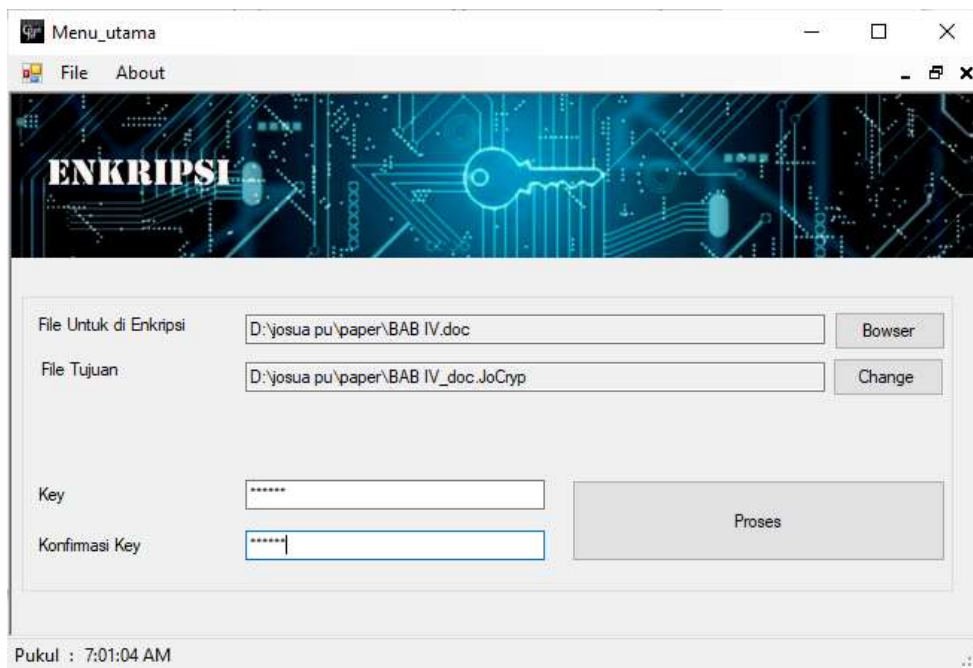
Proses *enkripsi* merupakan proses di mana data *file* yang asli akan mengalami perubahan isi (konten data) sehingga tidak memungkinkan dibaca. Proses *enkripsi* sendiri adalah sebuah proses yang memanfaatkan penggunaan *password* dan juga Algoritma Kriptografi (Algoritma Kriptografi yang dibahas adalah *Blowfish* dan *Rijndael*) dalam upaya untuk pengubahan *plainfile* menjadi *cipherfile* atau *enkripsi*.

#### 2. Proses Dekripsi

Proses *dekripsi* sendiri adalah sebuah proses yang bertolak belakang dengan proses *enkripsi*. Proses *dekripsi* adalah proses yang akan mengembalikan kondisi data *file* yang semula dalam kondisi yang tidak bisa dibaca, kembali menjadi kondisi semula sebelum data *file* mengalami proses *enkripsi*. Proses *dekripsi* dilakukan dengan menggunakan *password* yang sama dengan *password* yang digunakan dalam proses *enkripsi* dan juga tentunya dengan menggunakan Algoritma yang sama.

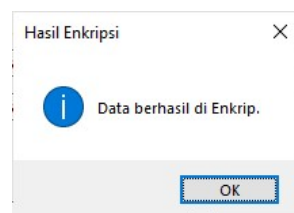
### IV.2.1 Proses Enkripsi Blowfish dan Rijndael

Proses *enkripsi* pada sistem yang dirancang oleh penulis menyediakan hasil dari masing-masing proses *enkripsi*, di mana masing-masing proses *enkripsi* Algoritma akan melakukan proses *enkripsi* dengan Algoritma *Blowfish* dan Algoritma *Rijndael*. ditunjukkan pada gambar IV.6 :



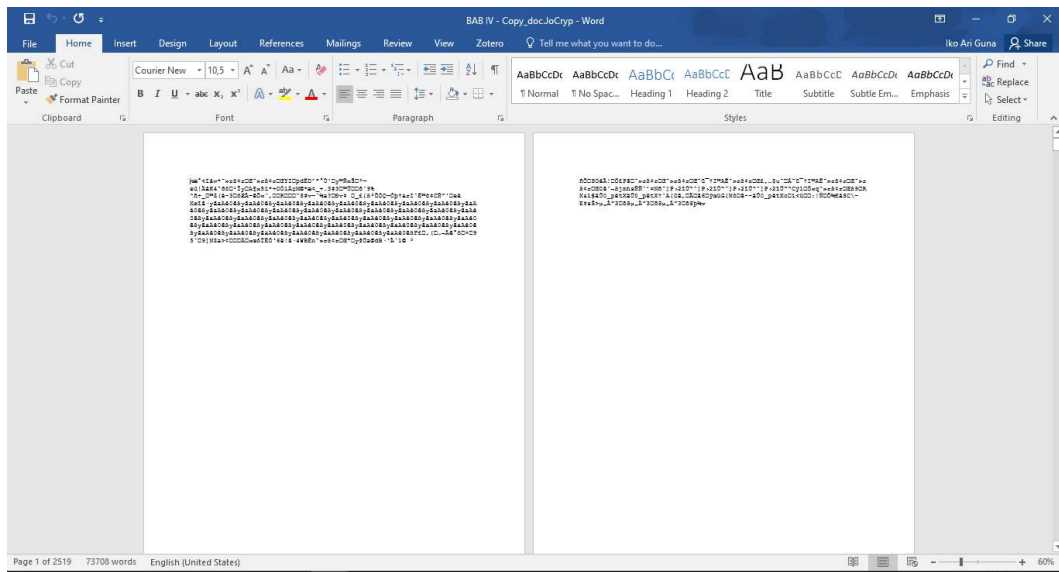
**Gambar IV.6. Antarmuka *Enkripsi***

selanjutnya adalah proses *enkripsi* menggunakan Algoritma *Blowfish* dan *Rijndael* ditunjukkan pada gambar IV.7 :



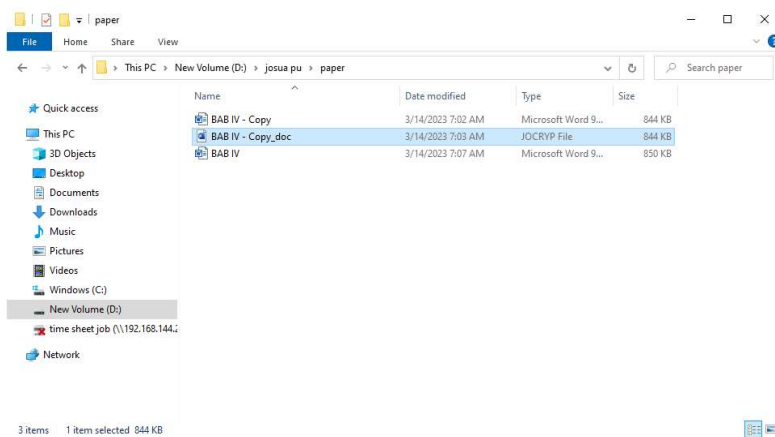
**Gambar IV.7. Poses *Enkripsi***

selanjutnya adalah hasil *enkripsi* menggunakan Algoritma *Blowfish* dan *Rijndael* dari file *extensiton \*.doc* ditunjukkan pada gambar IV.8 :



**Gambar IV.8. Hasil *Enkripsi***

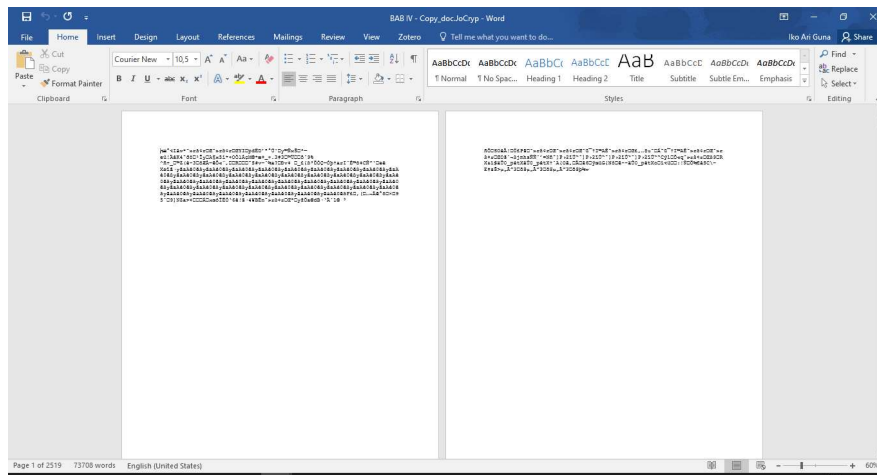
Selanjutnya hasil *enkripsi* atau *cipherfile* akan disimpan pada direktori tertentu dengan \*.JoCryp untuk nanti nya akan digunakan kembali dalam proses *dekripsi*. ditunjukkan pada gambar IV.8:



**Gambar IV.8. Hasil *Enkripsi***

selanjutnya adalah hasil *enkripsi* menggunakan Algoritma *Blowfish* dan *Rijndael* dari file *extensi* \*.doc mendi *extensi* \*.JoCrip ditunjukkan pada gambar IV.9 :

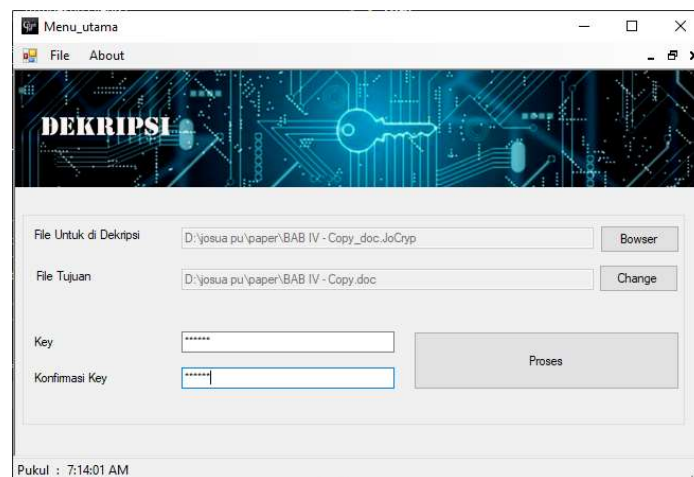




**Gambar IV.9. Hasil *Enkripsi***

#### **IV.2.2 Proses Dekripsi Blowfish dan *Rijndael***

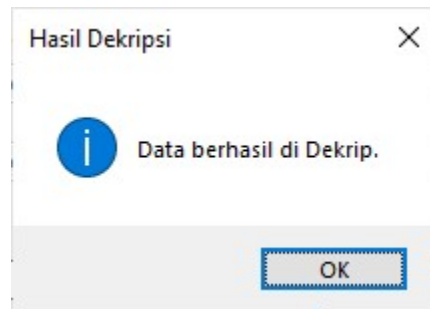
Proses *dekripsi* pada sistem yang dirancang oleh penulis menyediakan hasil dari masing-masing proses *dekripsi*, di mana masing-masing proses *dekripsi* Algoritma akan melakukan proses *dekripsi* dengan Algoritma *Blowfish* dan Algoritma *Rijndael*. ditunjukkan pada gambar IV10:



**Gambar IV.10. Antarmuka *Dekripsi***

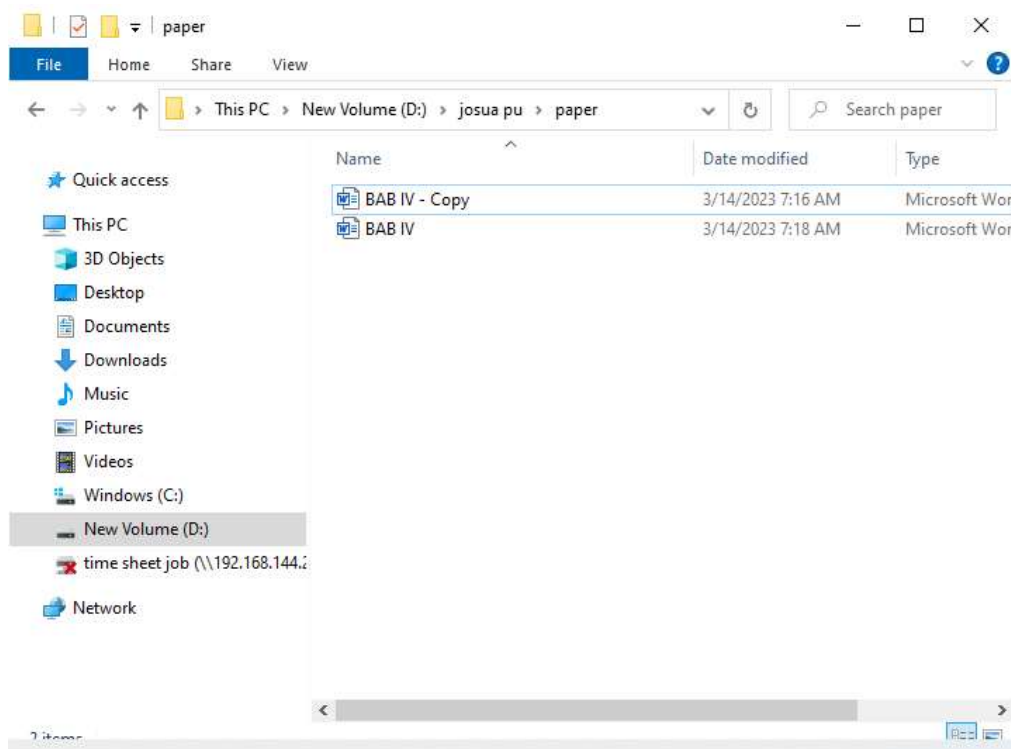
selanjutnya adalah proses *dekripsi* menggunakan *Blowfish* dan *Rijndael* ditunjukkan pada gambar IV.11:





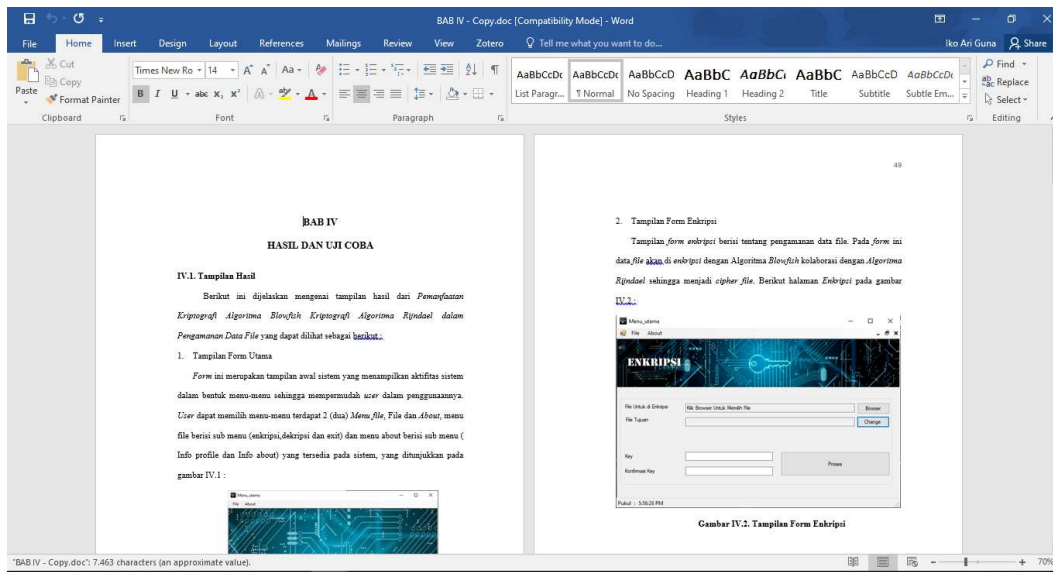
**Gambar IV.10. Antarmuka Dekripsi**

Hasil akhir dari proses *dekripsi file* yang telah di *enkripsi* dapat dikembalikan pada kondisi awal sama seperti sebelum *file* melewati proses *dekripsi*. ditunjukan pada gambar IV.11:



**Gambar IV.11. Hasil Dekripsi**

selanjutnya adalah hasil *dekripsi* menggunakan Algoritma *Blowfish* dan *Rijndael* dari file *extensi \*.JoCrip* mendi *extensi \*.doc* ditunjukan pada gambar IV.12 :



**Gambar IV.12. Hasil Dekripsi**

### IV.3. Pengujian Modul

Pengujian modul ini merupakan hasil pengujian implementasi aplikasi secara detail mengenai item-item yang terdapat pada setiap tampilan proses *enkripsi file*. Pengujian modul meliputi pengujian modul *enkripsi file* dan *dekripsi file*.

#### IV.3.1. Pengujian Modul Enkripsi File

Merupakan hasil pengujian implementasi aplikasi secara detail mengenai item-item yang terdapat pada setiap tampilan proses *enkripsi file*:

##### 1. Pengujian Tahap Enkripsi

Sudah ada sumber *file* didalam perangkat penyimpanan yang akan dilakukan pengujian, ditunjukkan pada tabel IV.1.:

**Tabel.IV.1. Pengujian Tahap Enkripsi**

Dekripsi	Perenkondisi	Prosedur Pengujian	Maskuan	Keluaran yang di harapkan	Hasil yang di dapat	kesimpulan

Pengujian tahap 1 mencari sumber <i>file</i>	Tampilan layar Enkripsi, ada 4 fasilitas option, Browser,	1. Cari file 2. Menentukan letak file yg telah dienkripsi 3. pengimputan kunci 4. Proses Enkripsi	Sumber File	Proses pembacaan file berhasil, tidak ada instruksi error	Proses pembacaan file berhasil, tidak ada instruksi error	Diterima
--	---	--	-------------	---	---	----------

#### IV.3.2. Pengujian Modul Dekripsi File

Merupakan hasil pengujian implementasi aplikasi secara detail mengenai *item-item* yang terdapat pada setiap tampilan proses *dekripsi file*:

##### 1. Pengujian Tahap Dekripsi

Sudah ada sumber *file* didalam perangkat penyimpanan yang akan dilakukan pengujian, ditunjukkan pada tabel IV.2.:

**Tabel.IV.2. Pengujian Tahap *Dekripsi***

Dekripsi	Perenkondisi	Prosedur Pengujian	Maskuan	Keluaran yang diharapkan	Hasil yang dapat	kesimpulan
Pengujian tahap 1 mencari sumber <i>file</i>	Tampilan layar dekripsi, ada 4 fasilitas	1. Cari file 2. Menentukan letak file yg telah	Sumber File	Proses pembacaan file berhasil, tidak ada	Proses pembacaan file berhasil,	Diterima

	option, Browser,	dienkrip si 3. Peinputa n kata kunci dan konfir kunci		instruksi erro	tidak ada instruksi erro	
--	---------------------	--	--	-------------------	--------------------------------	--

### IV.3.3. Pengujian Terhadap Jenis Data File

Pengujian terhadap data *file* yang dibutuhkan untuk proses *enkripsi* dan *dekripsi* dilakukan membuktikan dan memberikan *list* hasil dari pengujian. Sehingga dari *list* tersebut dapat ditentukan untuk memperoleh hasil *enkripsi* dan *dekripsi* terhadap besarnya ukuran *file*. Pengujian dilakukan dengan melakukan percobaan terhadap *file* antara lain :

1. Dokumen (\*.pdf, \*.doc, \*.txt)

Pengujian pada data dokumen (\*.pdf, \*.doc, \*.txt) dapat ditunjukkan pada tabel IV.3.

**Tabel.IV.3. Percobaan Data File Dokumen**

<b>Nama File</b>	<b>Ukuran File (byte)</b>	<b>Kata Kunci (key)</b>	<b>Ekripsi</b>	<b>Dekripsi</b>
85138688.pdf	1,400,832 bytes	12345	Berhasil	Berhasil
Gabung.doc	2,281,472 bytes	12345	Berhasil	Berhasil
Jago.txt	4,096 bytes	12345	Berhasil	Berhasil

2. Audio (\*.acc, \*.mp3, \*.wav)

Pengujian pada data Audio (\*.acc, \*.mp3, \*.wav) dapat ditunjukkan pada tabel IV.4.

**Tabel.IV.4. Percobaan Data *File* Audio**

<b>Nama <i>File</i></b>	<b>Ukuran <i>File</i> (byte)</b>	<b>Kata Kunci (key)</b>	<b><i>Ekripsi</i></b>	<b><i>Dekripsi</i></b>
bb.acc	1,400,832 bytes	12345	Berhasil	Berhasil
Wa bg.mp3	26,247,168 bytes	12345	Berhasil	Berhasil
Sedih.wav	2,281,472 bytes	12345	Berhasil	Berhasil

### 3. *Video* (\*.avi, \*.mp4, \*.wmv)

Pengujian pada data *file* Video (\*.avi, \*.mp4, \*.wmv) dapat ditunjukkan pada tabel IV.5.

**Tabel.IV.5. Percobaan data *file* Video**

<b>Nama <i>File</i></b>	<b>Ukuran <i>File</i> (byte)</b>	<b>Kata Kunci (key)</b>	<b><i>Ekripsi</i></b>	<b><i>Dekripsi</i></b>
Wildlife.wmv	26,247,168 bytes	12345	Berhasil	Berhasil
test.mp4	2,281,472 bytes	12345	Berhasil	Berhasil
coba.avi	1,400,832 bytes	12345	Berhasil	Berhasil

### 4. *Gambar* (\*.jpeg, \*.png, \*.bmp)

Pengujian pada data *file* Gambar (\*.jpg, \*.png, \*.bmp) dapat ditunjukkan pada tabel IV.6.

**Tabel.IV.6. Percobaan data *file* Gambar**

<b>Nama <i>File</i></b>	<b>Ukuran <i>File</i> (byte)</b>	<b>Kata Kunci (key)</b>	<b><i>Ekripsi</i></b>	<b><i>Dekripsi</i></b>
3.jpeg	69,632 bytes	12345	Berhasil	Berhasil
close.png	12,288 bytes	12345	Berhasil	Berhasil
Picture.bmp	53,248 bytes	12345	Berhasil	Berhasil

## **IV.4. Kelebihan Dan Kelemahan Sistem**

### **IV.4.1 Kelebihan Sistem**

Sistem yang dirancang memiliki kelebihan mampu untuk memberikan hasil *enkripsi* yang tidak mudah di bobol maupun diretas . Sistem ini juga dapat memberikan tingkat akurasi *enkripsi* dan *dekripsi* dengan cepat dan tepat.

### **IV.4.2 Kelemahan Sistem**

Selain memiliki kelebihan, sistem juga memiliki kelemahan yaitu :

1. Sistem hanya dapat berjalan di sistem operasi *windows* yang berbasis Aplikasi *dektop*.
2. Masih menggunakan 2 metode Kriptografi yaitu Algoritma *Blowfish* dan *Rijndael*.
3. Untuk *file* yang besar proses pengenkripsian masih membutuhkan waktu yang lama.