BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Bertukar informasi merupakan hal yang biasa kita lakukan. Bertukar informasih jarak jauh dapat dilakukan melalui kantor pos, surat, dan surel (surat elektronik). Surel memungkinkan kita untuk bertukar informasi jarak jauh tanpa membutuhkan waktu yang lama,namun keamanan informasi (data) dalam pengiriman informasi melalui surat elektronik (*email*) dipertaruhkan. Oleh karena itu dibutuhkan berbagai cara untuk mengamankan informasi tersebut agar tercapai ketujuan dengan aman. Salah satu metode yang digunakan untuk mengamankan data adalah kriptografi.

Kriptografi adalah sebuah cabang ilmu dalam ilmu komputer yang berfungsi untuk mengamankan data. Secara terminologi, kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya kedalam bentuk yang tidak dapat dipahami maknanya sehingga tidak dapat dibaca oleh orang yang tidak berkepentingan. Dalam kriptografi dibutuhkan kunci yaitu kode untuk melakukan enkripsi dan dekripsi. Berdasarkan kuncinya kriptografi dibagikan menjadi dua tipe yaitu algoritma simetris dan algoritma asimetris.

Algoritma simetris (*symmetric algorithm*) adalah algoritma yang mempunyai kunci enkripsi dan kunci dekripsi yang sama sehingga kunci ini disebut juga single key algorithm sedangkan algoritma asismetris (*asysmmetric algorithm*) merupakan algoritma yang terdiri atas dua buah kunci yaitu kunci

publik untuk melakukan enkripsi dan kunci privat untuk melakukan dekripsi. Kunci publik disebarkan secara umum sedangkan kunci privat disimpan secara rahasia oleh pengguna. Walau kunci publik telah diketahui namun akan sangat sukar mengetahui kunci privat yang digunakan.

Algoritma kriptografi yang dikategorikan kedalam algoritma asismetris salah satunya adalah algoritma RC6 (*Rivest Cipher* 6). Algoritma RC6 merupakan salah satu metode enkripsi yang menggunakan kunci simetris dan berfungsi menjaga kerahasiaan data. Berdasarkan uraian diatas dilakukan penelitian yang lebih mendalam mengenai metode kriptografi RC6 yang diterapkan pada keamanan data teks untuk mencapai tingkat keamanan yang tinggi.

B. Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan masalahnya yaitu " Bagaimana Mengimplementasikan Aplikasi Untuk Keamanan Data Teks Dengan Menggunakan Algoritma RC6,"

C. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah didefenisikan sebelumnya, tujuan dari penelitian ini adalah mengimplementasikan algoritma RC6 (Rivest Chiper) untuk menjaga keamanan pada data teks.

D. Batasan Masalah

Batasan masalah dari penelitian ini adalah :

1. Metode kriptografi yang digunakan yaitu algoritma RC6
2. Memfokuskan penelitian pada keamanan data teks yang bersifat sangat rahasia

3. Aplikasi yang dibuat hanya berlaku untuk data teks berekstensi txt.

E. Manfaat Penelitian

Dari rumusan masalah diatas adapun manfaat yang ingin dicapai penulis ialah untuk membangun aplikasi kriptografi pada teks dengan algoritma RC6 berbasis android agar dapat meningkatkan keamanan informasi pesan.

F. Sistematika Penulisan

Struktur penulisan penelitian ini adalah :

BAB I : PENDAHULUAN

Berisi tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penulisan dari pembuatan proposal ini.

BAB II : TINJAUAN PUSTAKA

Berisi tentang penelitian kajian teori, kajian hasil penelitian dan hasil tinjauan penelitian yang mendukung pembuatan skripsi ini.

BAB III : METODE PENELITIAN

Berisi tentang letak lokasi serta waktu, jenis penelitian, metode pengumpulan data, alat dan bahan penelitian, tahap penelitian, metode pengujian, dan analisa sistem.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Berisi tentang gambaran umum, analisis, meliputi pendefenisian dan pemodelan sistem dalam bentuk *use case, class* diagram, *sequence* diagram serta desain *database*, dan pengujian *system* dengan *whitbox* dan *blackbox.*

BAB V : PENUTUP

Berisi kesimpulan yang dapat di ambil dari penulisan akhir ini dan saran-saran pengembangannya.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

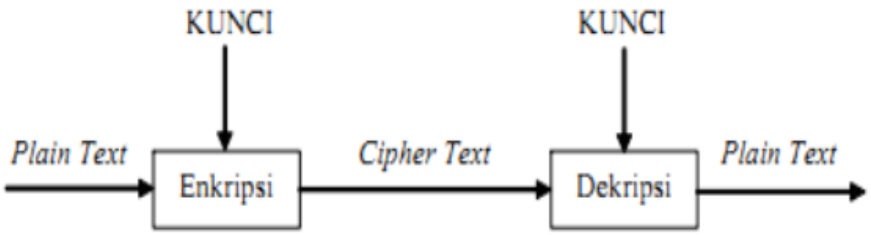
1. Kriptografi

Asal usul kata kriptografi merupakan dari bahasa yunani yang berasal dari dua kata, yaitu cripto dan graphia. *Crypto* dapat diartikan rahasi, dan arti kata *graphia* artinya tulisan, sehingga kriptografi dapat diartikan suatu tulisan yang bersifat rahasia. Menurut istilah kriptografi merupakan ilmu yang digunakan untuk menjaga keaslian sebuah pesan agar orang lain tidak mudah menyalahgunakan. Menurut Menezes, kriptografi merupakan sebuah ilmu yang membahas teknik matematis yang berkaitan dengan topik keamanan informasi (Andika D, 2018). Semakin berkembangnya zaman, keamanan kriptografi bertambah pula. Kegunaan lain dari kriptografi antara lain digunakan untuk mengidentifikasi sebuah pengiriman pesan, mengenali tanda tangan digital dan menguji keaslian pesan dengan menggunakan sidik jari digital. Pada algoritma kriptografi aman tidak suatu algoritma ditentukan oleh bagaimana algoritma tersebut bekerja. Algoritma seperti ini biasa disebut dengan algoritma terbatas (Rsa, 2016). Algoritma terbatas adalah algoritma yang digunakan oleh suatu organisasi atau sekelompok manusia untuk merahasiakan pesan yang mereka kirim. Pesan tersebut hanya akan diketahui oleh sekelompok manusia pada kumpulan tersebut. Jika suatu hari ada salah satu anggota yang keluar dari kumpulan tersebut, maka algoritma yang digunakan untuk mengirim pesan harus

diganti. Jika tidak diganti, akan didapatkan masalah dikemudian hari. Keamanan kriptografi modern terletak pada bagaimana cara kita merahasiakan algoritma tersebut kepada orang lain. Kegunaan dari kunci ini sama dengan kegunaan password. Jika seluruh keamanan algoritma bergantung pada kunci yang akan digunakan, maka algoritma tersebut dapat diumumkan dan dianalisis oleh orang lain (Muhammad Sholeh, 2014). Jika algoritma yang telah diumumkan bisa dipecahkan oleh orang lain dalam waktu yang singkat, maka algoritma tersebut kurang aman untuk digunakan. Kesulitan dalam mengolah data ataupun mengolah pesan yang akan disampaikan bukanlah syarat dari algoritma kriptografi yang baik. Yang lebik penting, algoritma kriptografi yang baik harus memenuhi empat persyaratan berikut:

a. Kerahasiaan. Kerahasiaan yang dimaksud dalam hal ini adalah menjaga informasi dari orang lain, kecuali yang memiliki akses terhadap kunci untuk membuka pesan tersebut.
b. Autentifikasi. Autentifikasi adalah berhubungan dengan pengenalan informasi pengirim dan penerima harus dapat dikenali dengan baik, serta harus memastikan tidak ada penyusup dalam proses pengiriman pesan.
c. Integritas data. Integritas data yang dimaksud yaitu sistem yang digunakan harus dapat mendeteksi bahwa benar-benar tidak ada manipulasi data oleh pihak manapun yang tidak memiliki kepentingan.
d. Non-repudiasi. Non-repudiasi atau disebut juga nirpenyangkalan merupakan usaha untuk mencegah penyangkalan. Penyangkalan yang

dimaksud bisa pada proses pengiriman maupun penerima pesan informasi dalam pesan sulit dipahami.



Gambar 2. 1 Diagram Proses Enkripsi dan Dekripsi

2. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi, (Nazrudin Safaat H, 2012:1). Menurut Akhmad Dharma Khasman (2016:2), "Android adalah sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh (*touchscreen*) yang berbasis linux". Namun seiring perkembangan, android berubah menjadi platfrom yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembang utama dibelakanya yaitu google. Googlelah yang mengakusisi android, kemudian membuatkan sebuah platform. Platform terdiri dari sistem operasi berbasis linux, sebuah *Graphic User Interface* (GUI) sebuah web browser dan aplikasi end-user yang dapat diunduh dan juga para pengembang bisa dengan leluasa berkarya serta menciptakan aplikasi yang terbaik dan terbuka untuk digunakan berbagai macam perangkat.

3. Algoritma *Rivest Code* (RC6)

Eko Juliansyah (2017). Algoritma RC6 merupakan salah satu kandidat *Advanced Encryption Standard* (AES) yang diajukan oleh RSA Laboratoriest

kepada NIST. Dirancang oleh Ronal L Rivest M.J.B.Robshaw, R Sidney dan Y.L.Yin, algoritma ini merupakan pengembangan dari algoritma sebelumnya yaitu RC5 dan telah memenuhi semua kriteria yang diajukan oleh NIST. Algoritma RC6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b, dimana parameter w merupakan ukuran kata dalam satuan bit, r merupakan bilangan bulat bukan negatif yang menunjukan banyaknya iterasi selama proses enkripsi, dan b menunjukan ukuran kunci enkripsi dalam byte. Ketika algoritma ini masuk sebagai kandidat AES, maka ditetapkan nilai parameter w = 32, r = 20 b bervariasi antara 16, 24, dan 32 byte (Maman Abdurohman, 2002).

RC6 merupakan algoritma yang merupakan keturunan dari RC5 yang juga merupakan kandidat Advanced Encryption Standard (AES) . Pada mulanya, perancangan RC6 diawali ketika RC5 dianggap dapat dijadikan kandidat untuk mengikuti kompetisi pemilihan AES. Modifikasi kemudian dibuat untuk meningkatkan keamanan dan performa dan juga untuk dapat memenuhi persyaratan AES. RC6 dirancang untuk menghilangkan segala ketidakamanan yang ditemukan pada RC5, karena analisis pada RC5 menunjukkan bahwa ternyata jumlah rotasi yang terjadi pada RC5 tidak sepenuhnya bergantung pada data yang terdapat dalam blok. Selain itu, serangan kriptanalisis diferensial juga ternyata dapat menembus keamanan yang ditawarkan RC5. RC6 juga dirancang untuk memenuhi persyaratan AES yang diantaranya adalah kemampuan untuk beroperasi pada mode blok 128 bit. Jika besar blok 128 bit langsung dipaksakan

untuk diimplementasikan dengan algoritma RC5, maka akan dibutuhkan register kerja 64 bit.

Spesifikasi arsitektur dan bahasa yang menjadi tempat implementasi algoritma yang ditentukan oleh AES belum mendukung pengoperasian 64 bit yang efisien. Oleh karena itu, daripada menggunakan 2 register 64 bit seperti pada RC5, RC6 menggunakan 4 register 32 bit. Karena menggunakan 4 register maka akan terdapat 2 operasi rotasi pada setiap half-round yang ada, dan juga akan lebih banyak bit-bit yang akan digunakan untuk mempengaruhi banyaknya bit yang dirotasi. RC6 seperti juga RC5 mengeksploitasi penggunaan operasi-operasi primitif yang diimplementasikan secara efisien dalam prosesor-prosesor modern.

RC6 juga selain menggunakan ketiga operasi primitif yang digunakan dalam RC5, juga menggunakan operasi perkalian 32-bit yang telah diimplementasikan secara efisien dalam prosesor modern saat ini.Primitif operasi perkalian ini sangat efektif dalam menghasilkan efek “diffusion” atau penyebaran yang tentu saja mengakibatkan RC6 lebih aman daripada RC5. Operasi perkalian ini digunakan untuk menghitung jumlah bit yang dirotasi sehingga konsep data-dependent rotations dapat dengan lebih sempurna diimplementasikan.RC6-w/r/b memecahkan blok 128 bit menjadi 4 buah blok 32 bit dan mengikuti aturan operasi dasar sebagai berikut:

a. A+B operasi penjumlahan bilangan integer
b. A-B operasi pengurangan bilangan integer
c. A B operasi exclusive –QR (XOR)
d. AxB operasi perkalian bilangan integer

e. A<<<B A dirotasikan kekiri sebanyak variabel kedua B

f. A>>>B A dirotasikan kekanan sebanyak variabel kedua B

Algoritma RC6 seperti juga RC5 merupakan algoritma cipher yang terparameterisasi. RC6 secara tepat ditulis sebagai:

```
RC6 – w / r / b
```

Nilai parameter w, r, dan b menyatakan hal yang sama seperti yang ditunjukkan dalam algoritma RC5. Algoritma RC6 yang dipakai sebagai kandidat AES adalah RC6-32/20/b, yang berarti ukuran word 32 bit, jumlah ronde 20 kali, dengan panjang kunci b ditentukan pengguna.

4. Key Expansion Algorithm

Algoritma untuk membangkitkan kunci internal sama seperti pada RC5. Nilai konstanta Pw dan Qw yang digunakan juga sama, tetapi ukuran array S tidak sama dengan yang seperti RC5. Ukuran t dari array S dalam RC6 adalah t = 2(r+2), yang berarti terdapat lebih banyak kunci internal yang dibangkitkan daripada jumlah kunci internal RC5. Berikut algoritmanya:

```
S[0] = Pw
fori = 1 to (2r + 3) do
  S[i] = S[i – 1] + Qw
  i = 0
  j = 0
  A = 0
  B = 0
  for3 × max(c, (2r + 4)) times do
```

```
S[i] = (S[i] + A + B) <<< 3

A = S[i]

L[i] = (L[j] + A + B) <<< 3

B = L[i]

i = (i + 1) mod (2r+4)

j = (j + 1) mod c
```

5. Enkripsi Algoritma

Karena RC6 memecahkan blok 128 bit menjadi 4 buah blok 32 bit, maka algoritma ini bekerja dengan 4 buah register 32 bit A, B, C, D. Byte yang pertama dari plaintext atau ciphertext ditempatkan pada byte A, sedangkan byte yang terakhirnya ditempatkan pada byte D. Dalam prosesnya akan didapatkan (A, B, C, D) = (D, C, B, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri, (Maman Abdurohman, 2002). Berikut ini adalah algoritma enkripsi RC6:

```
B = B + S [ 0 ] D = D + S [ 1 ]

For i = 1 to 20 do

     {

t = ( B X ( 2B + 1 ) ) <<< 5

u = ( D X ( 2D + 1 ) ) <<< 5

A = ( ( A Å t ) <<< u ) + S [ 2i ]

C = ( ( C Å u ) <<< t ) + S [ 2i + 1 ]

( A, B, C, D ) = ( D, C, B, A )

     }

A = A + S [ 42 ] C = C + S [ 43 ]
```

Algoritma RC6 menggunakan 44 buah sub kunci yang dibangkitkan dari kunci dan dinamakan dengan S[0] hingga S[43]. Masing masing sub kunci panjangnya 32 bit. Proses enkripsi pada algoritma RC6 dimulai dan diakhiri dengan proses whitening yang bertujuan untuk menyamarkan iterasi yang pertama dan yang terakhir dari proses enkripsi dan dekripsi. Pada proses whitening awal nilai B akan dijumlahkan dengan S[0], dan nilai D dijumlahkan dengan S[i]. Pada masing-masing iterasi pada RC6 menggunakan 2 buah sub kunci. Sub kunci pada iterasi yang pertama menggunakan S[2] dan S[3], sedangkan itersi-itersi berikutnya menggunakan sub-sub kunci lanjutannya. Setelah itersi ke-20 selesai, dilakukan proses whitening akhir dimana A dijumlahkan dengan S[42], dan nilai C dijumlahkan dengan S [43], (Maman Abdurohman, 2002). Setiap iterasi pada algoritma RC6 mengikuti aturan sebagai berikut, nilai B dimasukan kedalam fungsi f, yang didefenisikan sebagai f(x) = x (2x+1), kemudian diputar kekiri sejauh 1g-w atau 5 bit. Hasil yang didapat pada proses ini dimisalkan sebagai u. Nilai u kemudian di XOR dengan C dan hasilnya menjadi nilai C. Nilai t juga digunakan sebagai acuan C untuk memutar nilainya kekiri. Begitu pula dengan nilai u juga digunakan sebagai acuan bagi niali A untuk melakukan proses pemutaran kekiri. Kemudian sub kunci S[2i] pada iterasi dijumlahkan dengan A dan sub kunci S[2i+1] dijumlahkan dengan C. Keempat bagian dari blok kemudian akan dipertukarkan dengan mengikuti aturan, bahwa nilai A ditempatkan pada D, nilai B ditempatkan pada A, nilai C ditempatkan pada B, dan nilai D (asli) ditempatkan pada C. Demikian iterasi tersebut akan terus berlangsung hingga 20 kali, (Maman Abdurohman, 2002).

Secara lebih detil, proses enkripsi dengan RC6 dapat dibagi dalam beberapa langkah. Dalam penjelasan berikut, notasi (A,B,C,D) = (B,C,D,A) berarti adalah operasi assignment yang dilakukan paralel (bersamaan) untuk setiap elemen di ruas kanan ke ruas kiri yang berkorespondensi. Langkah-langkahnya adalah sebagai berikut:

a. Mula-mula lakukan half-round loop:

```
For i = 1 to r do
     A = ((AB) <<< B) + S[i] ⊕
     (A,B) = (B,A)
```

b. Lakukan dua proses RC5 secara paralel, yang satu untuk register A, B dan yang lain untuk register C, D.

```
For i = 1 to r do
     A = ((AB) <<< B) + S[2i] ⊕
     C = ((CD) <<< D) + S[2i+1] ⊕
     (A,B) = (B,A)
     (C,D) = (D,C)
```

c. Pada tahap pertukaran, daripada menukar A dengan B, dan C dengan D, lakukan permutasi antar keempat register (A,B,C,D) = (B,C,D,A), sehingga komputasi AB bercampur dengan komputasi CD.

```
For i = 1 to r do
     A = ((AB) <<< B) + S[2i] ⊕
     C = ((CD) <<< D) + S[2i+1] ⊕
     (A,B,C,D) = (B,C,D,A)
```

d. Campurkan komputasi AB dengan CD lebih jauh, yaitu dengan mempertukarkan kedua nilai yang menyatakan jumlah rotasi pada masing-masing komputasi.

```
For i = 1 to r do
      A = ((A⊕B) <<< D) + S[2i]
      C = ((C⊕D) <<< B) + S[2i+1]
      (A,B,C,D) = (B,C,D,A)
```

e. Daripada menggunakan nilai B dan D secara langsung, RC6 menggunakan hasil transformasi kedua register ini. Hal ini dilakukan untuk tidak mengulangi masalah rotasi seperti pada RC5 di mana tidak seluruh bit dalam data yang berpengaruh dalam rotasi. Oleh karena itu, fungsi transformasi yang dipilih harus dapat memanfaat seluruh bit di dalam data untuk mengatur jumlah bit yang dirotasikan. Fungsi yang dipilih adalah f(x) = x(2x + 1) (mod 2w) yang kemudian diikuti dengan rotasi ke kiri sebanyak 5 bit. Transformasi ini terpilih karena fungsi f(x) yang merupakan fungsi satu-ke-satu memiliki bit-bit orde atas yang menentukan jumlah rotasi yang akan digunakan yang sangat bergantung pada x.

```
For i = 1 to r do
      p = (B × (2B + 1)) <<< 5
      q = (D × (2D + 1)) <<< 5
      A = ((A⊕p) <<< q) + S[2i]
      C = ((C⊕q) <<< p) + S[2i+1]
```

```
(A,B,C,D) = (B,C,D,A)
```

f. Setelah loop di atas selesai, akan terdapat hasil di mana plaintext bisa menunjukkan bagian input ronde pertama dalam enkripsi dan ciphertext bisa menunjukkan bagian input ronde terakhir dalam enkripsi. Oleh karena itu perlu ditambahkan langkah-langkah di awal dan di akhir loop untuk menyamarkan hubungan ini. Sehingga, terbentuklah algoritma enkripsi RC6 yang sebagai berikut:

```
B = B + S[0]
D = D + S[1]
For i = 1 to r do
     p = (B × (2B + 1)) <<< 5
     q = (D × (2D + 1)) <<< 5
     A = ((A⊕p) <<< q) + S[2i]
     C = ((C⊕q) <<< p) + S[2i+1]
     (A,B,C,D) = (B,C,D,A)
A = A + S[2r + 2]
C = C + S[2r + 3]
```

Perlu diketahui juga, dalam varian baru RC6 jumlah rotasi ke kiri yang mengikuti fungsi kuadrat bukan 5 bit tetapi adalah 2log(w) bit.

6. Dekripsi Algoritma RC6

Proses dekripsi *ciphertext* pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses whitening, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub

kunci yang digunakan pada proses whitening setelah iterasi terakhir diterapkan sebelum iterasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses whitening sebelum iterasi pertama digunakan pada whitening setelah iterasi terakhir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik, (Maman Abdurohman 2002). Berikut ini adalah algoritma dekripsi RC6:

```
C = C − S[2r + 3]
A = A − S[2r + 2]
For i = r down to 1 do
      (A,B,C,D) = (D,A,B,C)
      p = (D × (2D + 1)) <<< 5
      q = (B × (2B + 1)) <<< 5
      C = ((C − S[2i + 1]) >>> q)p ⊕
      A = ((A − S[2i]) >>> p)q ⊕
D = D − S[1]
B = B − S[0]
```

7. File Teks

File teks merupakan jenis file digital yang hanya berisi teks dan tidak memiliki format khusus seperti gambar, grafik, dan video. File teks memiliki ekstensi yang diakhiri dengan “txt” dan dapat dibuka oleh berbagai macam program, seperti notepad atau word. Fungsi file teks adalah menyimpan data atau

informasi tekstual standar dan terstruktur yang dapat dibaca manusia. Selain teks sederhana file teks juga digunakan untuk menulis dan menyimpan kode untuk semua bahasa pemrograman, seperti java dan PHP. File yang dibuat dapat dikonversi kebahasa pemrograman masing-masing dengan mengubah ekstensi file dari “txt’ menjadi PHP atau cpp.

8. *Apache Cordova*

Apache cordova merupakan kerangka pengembangan mobile open source. Apache cordova memungkinkan untuk menggunakan teknologi web standar seperti HTML5, CSS3 dan javascript untuk pengembangan lintas platform. Aplikasi dijalankan dalam pembungkus yang ditargetkan kemasing-masing platform, dan bergantung pada binding API yang sesuai standar untuk mengakses kemampuan masing-masing perangkat seperti sensor, data, status jaringan, (Rusmala Santi, 2019).

9. *JavaScript*

Javascript adalah bahasa pemrograman website yang bersifat *Client Side Programming Language* (CSPL) merupakan tipe bahasa pemrograman yang pemrosesannya dilakukan oleh client. Aplikasi client yang dimaksud merujuk pada web browser seperti Google Chrome dan Mozilla Firefox.

Jenis bahasa pemrograman client side berbeda dengan bahasa pemrograman server side seperti PHP dimana untuk server side untuk seluruh kode program dijalankan disisi server. Untuk menjalankan javascript,kita hanya membutuhkan aplikasi teks editor, dan web browser (Wahyu Nur Rohim,2015).

10. Kode ASCII

Kode standar Amerika untuk pertukaran informasi atau *American Standard Code for Information Interchange* (ASCII) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter “1”. Kode ini selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukan teks. Kode ASCII sebenarnya memiliki komposisi bilangan biner sebanyak 7 bit. Namun ASCII disimpan sebagai sandi 8 bit dengan menambahkan satu angka 0 sebagai bit significant paling tinggi (Bayu Kurniawan, 2022). Bit tambahan ini sering digunakan uji prioritas. Karakter control pada ASCII dibedakan menjadi 5 kelompok sesuai dengan penggunaan yaitu berturut-turut meliputi logical comunication, device control, information separator, code extention, dan physical communition. Code ASCII ini banyak dijumpai pada papan ketik (keyboard) komputer atau instrument-instrument digital. Jumlah code ASCII adalah 225 code. Code ASCII 0..127 merupakan kode ASCII untuk manipulasi teks sedangkan code ASCII 128.225 merupakan code ASCII untuk memanipulasi grafik.

Binary	Oct	Dec	Hex	Glyph
010 0000	040	32	20	sp
010 0001	041	33	21	!
010 0010	042	34	22	"
010 0011	043	35	23	#
010 0100	044	36	24	$
010 0101	045	37	25	%
010 0110	046	38	26	&
010 0111	047	39	27	'
010 1000	050	40	28	(
010 1001	051	41	29	)
010 1010	052	42	2A	*
010 1011	053	43	2B	+
010 1100	054	44	2C	,
010 1101	055	45	2D	-
010 1110	056	46	2E	.
010 1111	057	47	2F	/
011 0000	060	48	30	0
011 0001	061	49	31	1
011 0010	062	50	32	2
011 0011	063	51	33	3
011 0100	064	52	34	4
011 0101	065	53	35	5
011 0110	066	54	36	6
011 0111	067	55	37	7
011 1000	070	56	38	8
011 1001	071	57	39	9
011 1010	072	58	3A	:
011 1011	073	59	3B	;
011 1100	074	60	3C	<
011 1101	075	61	3D	=
011 1110	076	62	3E	>
011 1111	077	63	3F	?

Binary	Oct	Dec	Hex	Glyph
100 0000	100	64	40	@
100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F
100 0111	107	71	47	G
100 1000	110	72	48	H
100 1001	111	73	49	I
100 1010	112	74	4A	J
100 1011	113	75	4B	K
100 1100	114	76	4C	L
100 1101	115	77	4D	M
100 1110	116	78	4E	N
100 1111	117	79	4F	O
101 0000	120	80	50	P
101 0001	121	81	51	Q
101 0010	122	82	52	R
101 0011	123	83	53	S
101 0100	124	84	54	T
101 0101	125	85	55	U
101 0110	126	86	56	V
101 0111	127	87	57	W
101 1000	130	88	58	X
101 1001	131	89	59	Y
101 1010	132	90	5A	Z
101 1011	133	91	5B	[
101 1100	134	92	5C	\
101 1101	135	93	5D	]
101 1110	136	94	5E	^
101 1111	137	95	5F	_

Binary	Oct	Dec	Hex	Glyph
110 0000	140	96	60	`
110 0001	141	97	61	a
110 0010	142	98	62	b
110 0011	143	99	63	c
110 0100	144	100	64	d
110 0101	145	101	65	e
110 0110	146	102	66	f
110 0111	147	103	67	g
110 1000	150	104	68	h
110 1001	151	105	69	i
110 1010	152	106	6A	j
110 1011	153	107	6B	k
110 1100	154	108	6C	l
110 1101	155	109	6D	m
110 1110	156	110	6E	n
110 1111	157	111	6F	o
111 0000	160	112	70	p
111 0001	161	113	71	q
111 0010	162	114	72	r
111 0011	163	115	73	s
111 0100	164	116	74	t
111 0101	165	117	75	u
111 0110	166	118	76	v
111 0111	167	119	77	w
111 1000	170	120	78	x
111 1001	171	121	79	y
111 1010	172	122	7A	z
111 1011	173	123	7B	{
111 1100	174	124	7C	|
111 1101	175	125	7D	}
111 1110	176	126	7E	~

Gambar 2. 2 ASCII Code

11. *UML (Unified Modelling Language)*

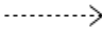
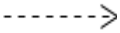
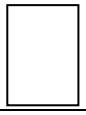
Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. *UML* adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. *UML* merupakan singkatan dari *Unified Modeling Language*. *UML* juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program *UML* juga dapat menjadi alat bantu untuk transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer ke developer lainnya. Tidak hanya antar developer terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya *UML*. *UML* diciptakan oleh *Object Management Group* yang diawali dengan versi 1.0 pada Januari 1997. Dalam pengembangan berorientasi objek ada beberapa prinsip yang harus dikenal:*Object, Class, Abstraction, Encapsulation, Inheritance dan Polymorphism*. Dalam *UML* sendiri terdapat beberapa diagram yaitu :

a. *Use Case Diagram*

Use Case diagram menggambarkan *fungsionalitas* yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use Case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use Case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Adapun simbol-simbol *Use Case Diagram* antara lain :

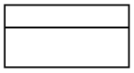
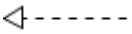
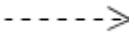
Tabel 2. 1 Simbol *Use Case* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna *Main*kan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *Use Case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *Use Case* target memperluas perilaku dari *Use Case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

b. *Class Diagram*

Adapun simbol-simbol *Class Diagram* antara lain :

Tabel 2.2. Simbol *Class* Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

c. *Sequence Diagram*

Adapun simbol-simbol *Sequence* Diagram antara lain :

Tabel 2.3. Simbol *Sequence* Diagram

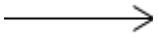
NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.

2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

d. *StateChart Diagram*

Adapun simbol-simbol *StateChart Diagram* antara lain :

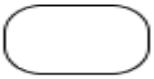
Tabel 2.4. Simbol *StateChart* Diagram

No.	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

e. *Activity Diagram*

Adapun simbol-simbol *Activity Diagram* antara lain :

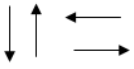
Tabel 2.5. Simbol *Activity* Diagram

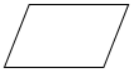
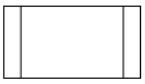
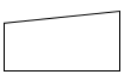
No.	GAMBAR	NAMA	KETERANGAN
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

12. *Flowchart*

Flowchart adalah adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Berikut ini adalah beberapa simbol yang digunakan dalam menggambar suatu *Flowchart* :

Tabel 2.6. Simbol *Flowchart*

Simbol	Nama	Fungsi
	***Terminal Point Symbol***	*Terminal* Menunjukkan permulaan (start) atau akhir (stop) dari suatu proses.
	***Flow Direction Symbol***	Adalah simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (*connecting line*). Simbol ini juga berfungsi untuk Menunjukkan garis alir dari proses.

Simbol	Nama	Fungsi
	***Processing Symbol***	Digunakan untuk Menunjukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi
	***Decision Symbol***	Merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada *Flowchart* program.
	**I*nput-Output***	Masuk menunjukkan proses *input-output* yang terjadi tanpa bergantung dari jenis peralatannya.
	***Predefined Process***	*Terdefinisi* merupakan simbol yang digunakan untuk Menunjukkan pelaksanaan suatu bagian prosedur (*sub*-proses).
	***Connector (On-page)***	Simbol ini fungsinya adalah untuk menyederhanakan hubungan antar simbol yang letaknya berjauhan atau rumit bila dihubungkan dengan garis dalam satu halaman
	***Connector (Off-page)***	Sama seperti *on-page connector*, hanya saya simbol ini digunakan untuk menghubungkan simbol dalam halaman berbeda. label dari simbol ini dapat menggunakan huruf atau angka
	***Preparation Symbol***	Merupakan simbol yang digunakan untuk mempersiapkan penyimpanan di dalam storage.
	***Manual Input Symbol***	Digunakan untuk menunjukkan input data secara manual menggunakan online keyboard.
	***Manual Operation Symbol***	Yang digunakan untuk menunjukkan kegiatan/proses yang tidak dilakukan oleh komputer.

Simbol	Nama	Fungsi
	***Document Symbol***	Jika Anda menemukan simbol ini artinya input berasal dari dokumen dalam bentuk kertas, atau output yang perlu dicetak di atas kertas.
	***Multiple Documents***	Sama seperti *document symbol* hanya saja dokumen yang digunakan lebih dari satu dalam simbol ini
	***Display Symbol***	Adalah simbol yang menyatakan penggunaan peralatan output, seperti layar monitor, printer, plotter dan lain sebagainya
	***Delay Symbol***	Sesuai dengan namanya digunakan untuk *Menu*njukkan proses delay (*Menu*nggu) yang perlu dilakukan. Seperti *Menu*nggu surat untuk diarsipkan

B. Kajian Hasil Penelitian

Berbagai penelitian sebelumnya hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu pendukung yang menurut peneliti perlu dijadikan bagian tersendiri adalah peneliti terdahulu yang relevan dengan permasalah yang sedang dibahas dalam penelitian ini. Dalam hal ini, fokus penelitian terdahulu yang dijadikan acuan adalah terkait dengan masalah teknologi informasi. Beberapa penelitian sebelumnya yang diambil oleh peneliti sebagai bahan pertimbangan dan sumber referensi yang berhubungan dengan judul penelitian ini diantaranya sebagai berikut:

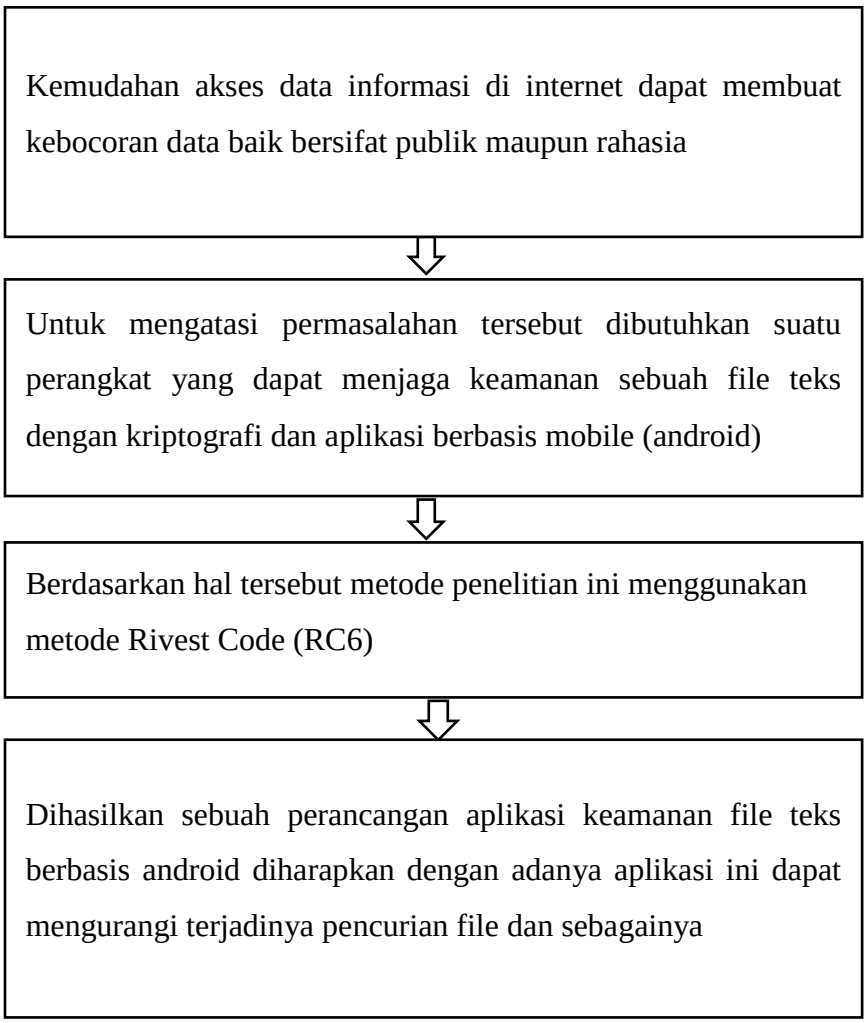
1. Laurentinus (2017) dalam penelitiannya yang berjudul "Implementasi Kriptografi Dan Kompresi Sms Menggunakan Algoritma RC6 Dan Algoritma Huffman Berbasis Android", peneliti merancang sebuah

aplikasi yang dapat menjamin keamanan dan keutuhan data dari transaksi pengiriman pesan SMS. Dengan tujuan agar informasi tidak dapat diakses oleh orang yang berkepentingan terhadap pesan sms tersebut. Algoritma yang digunakan oleh penulis dalam penelitian adalah algoritma RC6 dan algoritma Huffman.

2. Eko juliansyah (2017) dalam penelitiannya yang berjudul "Implementasi Algoritma Kriptografi RC6 Dalam Mengamankan Data Teks", peneliti merancang sebuah aplikasi untuk mengamankan data teks", peneliti merancang sebuah aplikasi untuk mengamankan data teks. Keamanan dari suatu data merupakan hal yang perlu diperhatikan dalam menjaga kerahasiaan informasi terutama bagi informasi yang isinya hanya boleh diketahui oleh pihak yang berhak saja. Algoritma yang digunakan dalam penelitian ini adalah algoritma RC6.
3. Kanneth Mohammad Albany Hakim (2021) pada penelitiannya yang berjudul "Rancang Bangun Aplikasi Enkripsi-Dekripsi Sms Pada Android Dengan Metode RC6", peneliti merancang sebuah aplikasi enkripsi-dekripsi untuk mengamankan pesan singkat (SMS). Short message service merupakan sebuah layanan yang banyak di aplikasikan pada sistem komunikasi tanpa kabel, memunkingkan dilakukannya pengiriman pesan dalam bentuk alphanumeric antara terminal pelanggan dengan sistem eksternal seperti email, paging, dan voice mail. Dalam penelitian ini peneliti menggunakan algoritma RC6.

C. Kerangka Pikir

Untuk lebih memperjelas kerangka pikir maka digambarkan dalam bentuk diagram sebagai berikut :



Gambar 2. 3 Kerangka Pikir

BAB III

METODE PENELITIAN

A. Waktu Penelitian

Penelitian ini dilakuakan dalam kurung waktu yang dipergunakan untuk pelaksanaan penelitian ini berlangsung selama ± 2 bulan tahun 2023.

B. Jenis Penelitian

Untuk membantu penelitian ini dilakukan melaui buku dan internet yang dapat memberikan sumber data dan pengetahuan mengenai kelancaran pengumpulan data, maka penulis menggunakan metode: penelitian pustaka (library research). Sistem yang diteliti, kemudian mencocokan dengan kemungkinan yang terjadi dalam usaha penyelesaian masalah.

C. Metode Pengumpulan Data

Pengumpulan data dilakukan untuk mengumpulkan seluruh informasi yang terkait dan mendukung pelaksanaan penelitian ini.

1. Observasi, pengamatan dan juga pencatatan sistematik atas unsur-unsur yang muncul dalam suatu gejala atau gejala-gejala yang muncul dalam suatu objek penelitian.
2. Wawancara adalah percakapan dengan maksud tertentu. Percakapan itu dilakukan oleh dua pihak, yaitu pewawancara *(interview)* yang mengajukan pertanyaan dan terwawancara *(interview)* yang memberikan jawaban atas pertanyaan itu. Ciri utama wawancara adalah kontak langsung dengan tatap muka antara pencari informasi dan sumber informasi. Dalam wawancara

sudah disiapkan berbagai macam pertanyaanpertanyaan tetapi muncul berbagai pertanyaan lain saat meneliti.

D. Alat dan Bahan Penelitian

Dalam pelaksanaan penelitian ini digunakan beberapa alat dan bahan yang terdiri dari Perangkat Keras (*Hardware*) dan Perangkat Lunak (*Software*). Perangkat keras yang digunakan terdiri dari Laptop ACER. Perangkat lunak yang digunakan adalah *Windows 10 Home Single Language*, *JavaScript, dan Android Studio.* Sedangkan bahan penelitian berupa data-data yang berasal dari hasil pengumpulan data yang telah dilakukan oleh peneliti yang nantinya akan dianalisis lebih lanjut sebagai landasan untuk merancang sebuah sistem.

E. Tahap Penelitian

Tahap penelitian dilakukan dengan cara :

1. Persiapan Penelitian

Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel tentang topik penelitian serta software yang digunakan selama penelitian.

2. Pengumpulan Data

Pada tahap ini penelitian melakukan observasi dengan peninjauan dan juga studi literatur.

3. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem dan juga data skripsi yang diterapkan berdasarkan data yang didapatkan saat melakukan pengumpulan data, kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

4. Perancangan

Peneliti kemudian merancang game edukasi huruf Aksara Nusantara yang ingin dibuat berdasarakan alternatif pemecahan masalah.

5. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan aplikasi. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

6. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk di gunakan.

F. Metode Pengujian

Beberapa kasus pengujian harus dijalankan menggunakan strategi, query, atau jalur navigasi berbeda yang mewakili penggunaaan sistem yang umum, penting atau tidak biasa. Isu penting dalam pengembangan sistem adalah memiliki rangkaian kasus uji yang tepat, secepat mungkin dan sekecil mungkin, untuk memastikan operasi sistem yang terperinci pengujian harus mencakup pengujian unit yang menvalidasi-validasi prosedural dan fungsional secara independen dari komponen dari sistem lainnya. Selanjutnya pengujian modul berikutnya harus menentukan apakah penggabungan beberapa unit dalam satu mobil berhasil termasuk eksekusi beberapa modul terkait dan apakah itu sesuai dengan properti sistem yang diinginkan.

Jika struktur kendali antar modul dibuktikan dengan sendirinya, maka pengujian yang kurang penting adalah pengujian unit. Tes unit digunakan untuk

menguji setiap modul untuk memastikan bahwa setiap modul menjalankan fungsinya dengan benar.

1. Black box testing

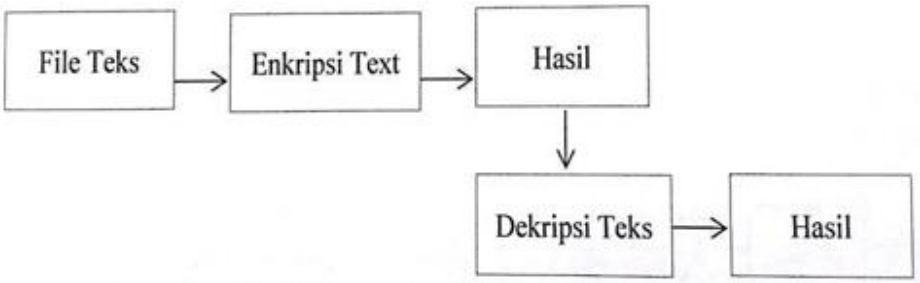
Berdasarkan uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta prose yang dijalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah ditetapkan dan harus diikuti dan apabila dihiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika user belum menginput data yang seharusnya diinput sesuai ketentuan sistem yang dijalankan dan sistem memberikan informasi kepada user karena data yang ingin diproses belum lengkap atau tidak memenuhi ketentuan untuk proses selanjutnya. Pengujian terhadap cara kerja perangkat lunak itu sendiri yaitu prosedur programnya (basis path) atau proses looping (pengulangan) yang berfokus pada efektifitas aplikasi yang dirancang. Black box testing merupakan pengujian yang bertujuan untuk menunjukan fungsi perangkat lunak tentang cara beroperasinya program, jadi semua proses yang ada pada aplikasi akan diuji dengan metode black box apakah perangkat lunak dapat beroperasional, bahwa input diterima dengan baik dan output dihasilkan dengan tepat.

2. White box testing

White box testing merupakan metode perancangan test case yang menggunakan struktur untuk mendapatkan test case, test ini digunakan untuk meramal cara kerja perangkat lunak secara perinci kepada logic path (jalur logika), perangkat lunak ditest dengan kondisi dan perulangan secara fisik.

Pengujian white box testing ini merupakan peringatan ketika user menginput password user yang salah, untuk kesalahan semacam ini akan memberikan suatu informasih kepada user mengenai kesalahan yang dilakukan.

G. Diagam Alir



Gambar 3. 1. Sistem Yang Dusulkan

Sistem pengenkripsi file teks ini merupakan sistem keseluruhan dimana sistem dimulai dari user yang menginput file teks pada aplikasi untuk dienkripsi, lalu menghasilkan teks tersandikan kemudian didekripsikan lalu menghasilkan file teks yang tidak tersandikan.

BAB IV

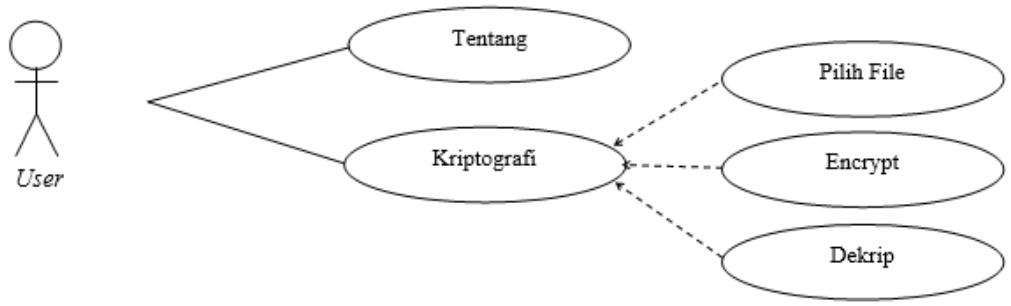
HASIL PENELITIAN DAN PEMBAHASAN

A. Analisis Aliran Data Dengan UML

Dalam analisis sistem ini, penulis menggunakan *Use Case* Diagram, *Activity* Diagram dan *Sequence* Diagram. Adapun aliran data

1. *Use Case* Diagram

Use Case Diagram berfungsi untuk menjalankan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*).



Gambar 4. 1 *Use Case* Diagram

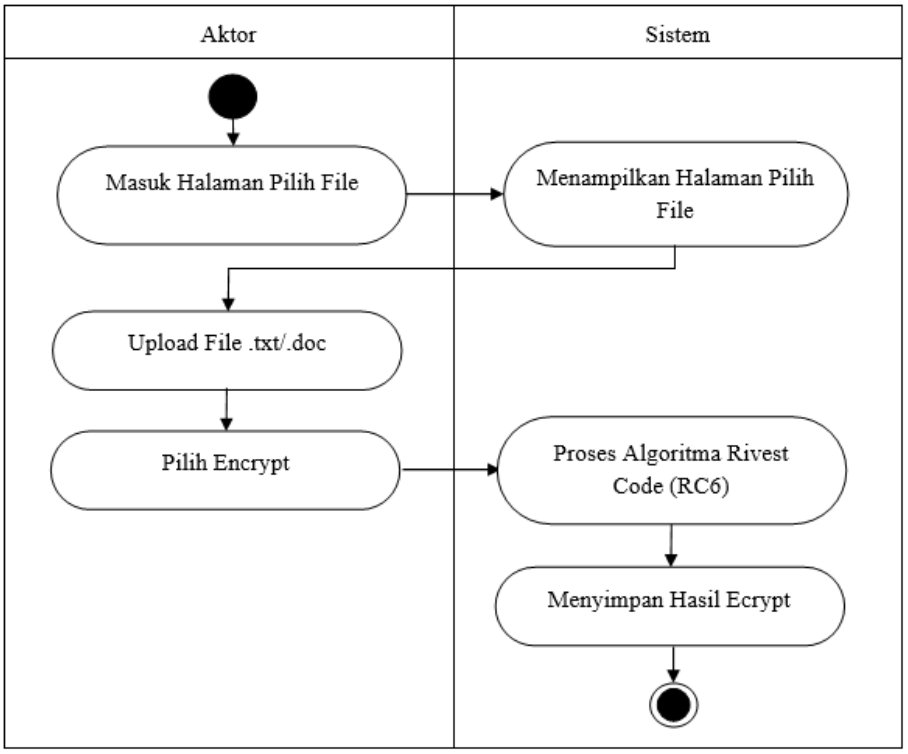
Tabel 4. 1 Penjelasan *Use Case* Diagram Aktor

Nama	**Deskripsi**
Tentang	Merupakan halaman untuk melihat tentang aplikasi
Kriptografi	Merupakan halaman untuk melakukan proses enkrip dan dekrip
Pilih File	Merupakan proses *user* memilih file
Encrypt	Merupakan proses *user* melakukan enkrip file
Dekrip	Merupakan proses *user* melakukan dekrip file

2. *Activity Diagram*

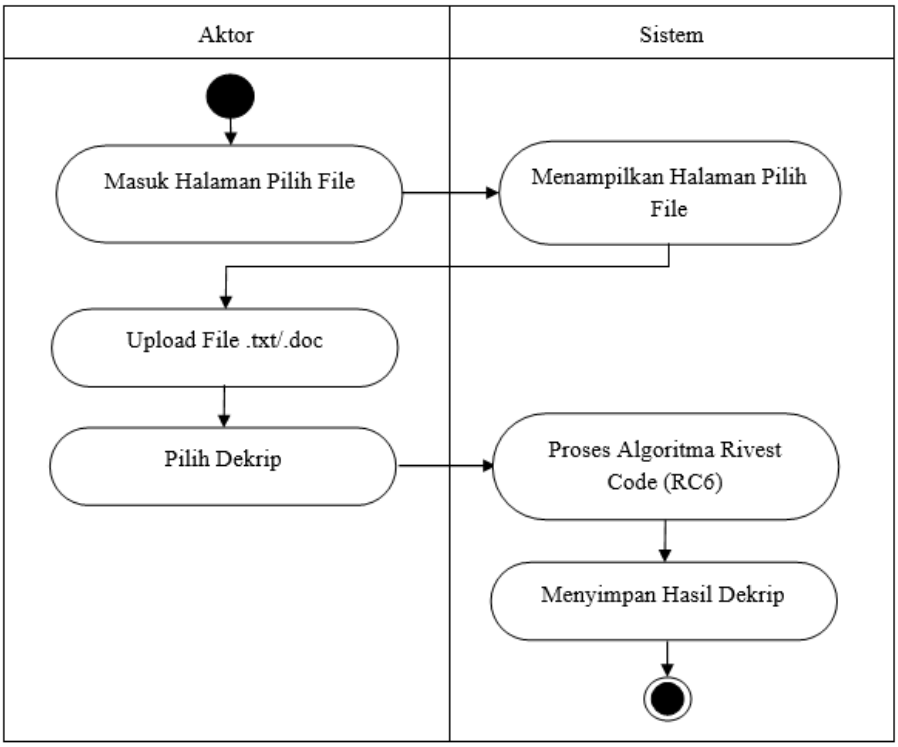
Activity Diagram ini menjelaskan tentang aktivitas-aktivitas yang terjadi dalam sebuah aliran proses pada sistem.

a. Diagram *Activity* Encrypt File



Gambar 4.2 Diagram *Activity* Encrypt File

b. Diagram *Activity* Dekrip File

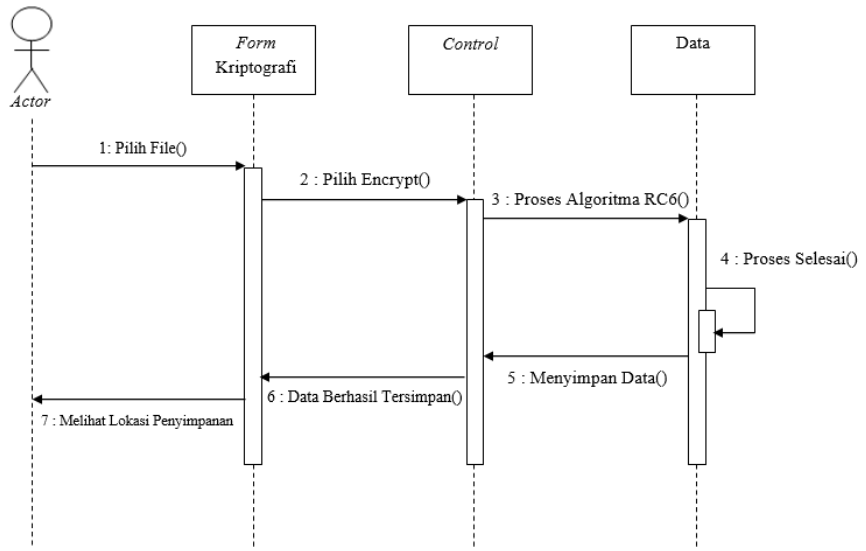


Gambar 4.3 Diagram *Activity* Dekrip File

3. *Sequence* Diagram

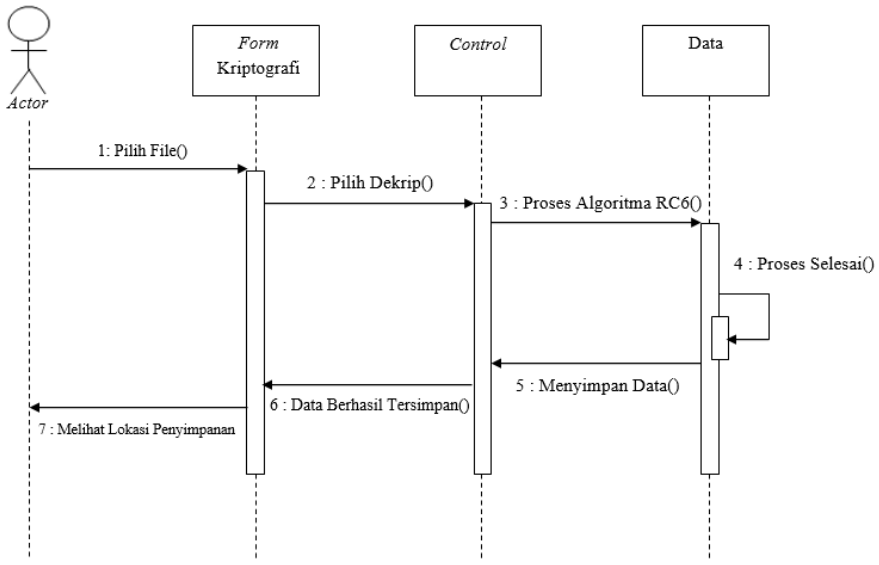
Sequence Diagram merupakan aliran antara objek yang membentuk proses, berikut adalah diagram *Sequencenya*.

a. Diagram *Sequence* Encrypt File



Gambar 4.4 *Sequence* Diagram Encrypt File

b. Diagram *Sequence* Dekrip File

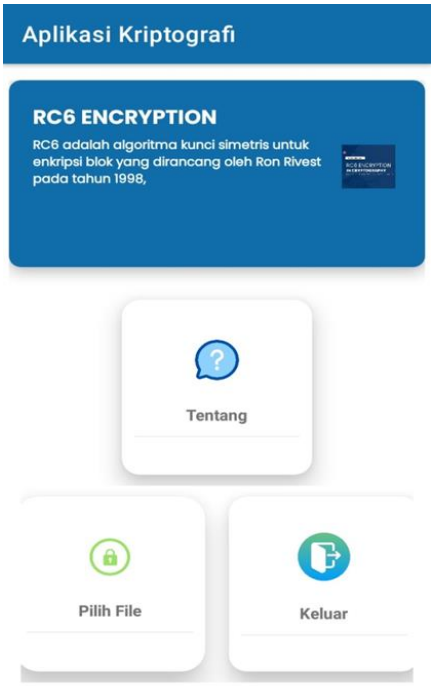


Gambar 4.5 *Sequence* Diagram Dekrip File

B. Rancangan Aplikasi

1. Halaman Utama

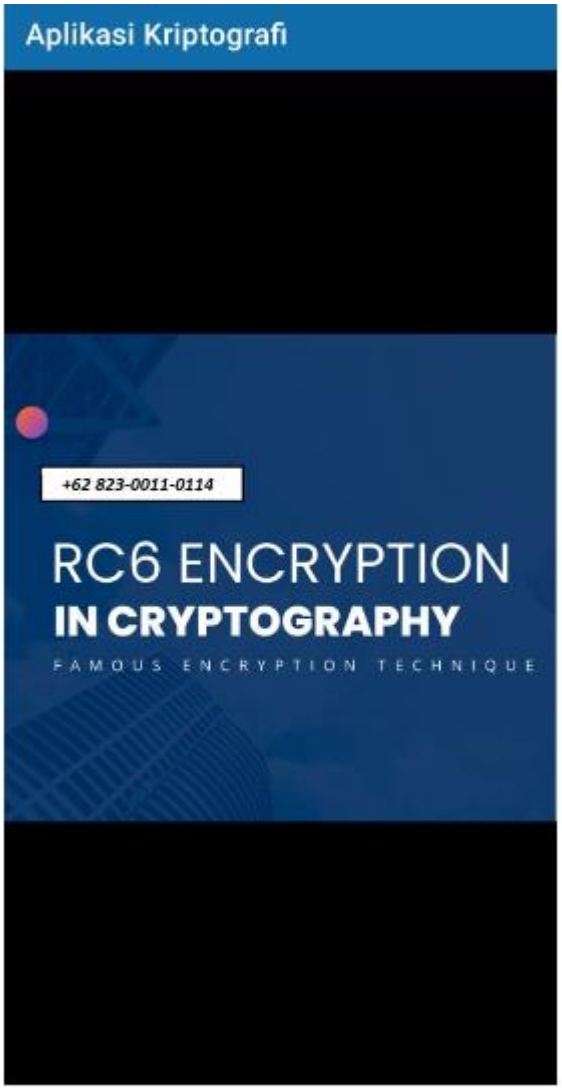
Merupakan tampilan halaman utama yang digunakan untuk menjalankan fungsi aplikasi.



Gambar 4.6. Halaman Utama

2. Halaman Tentang

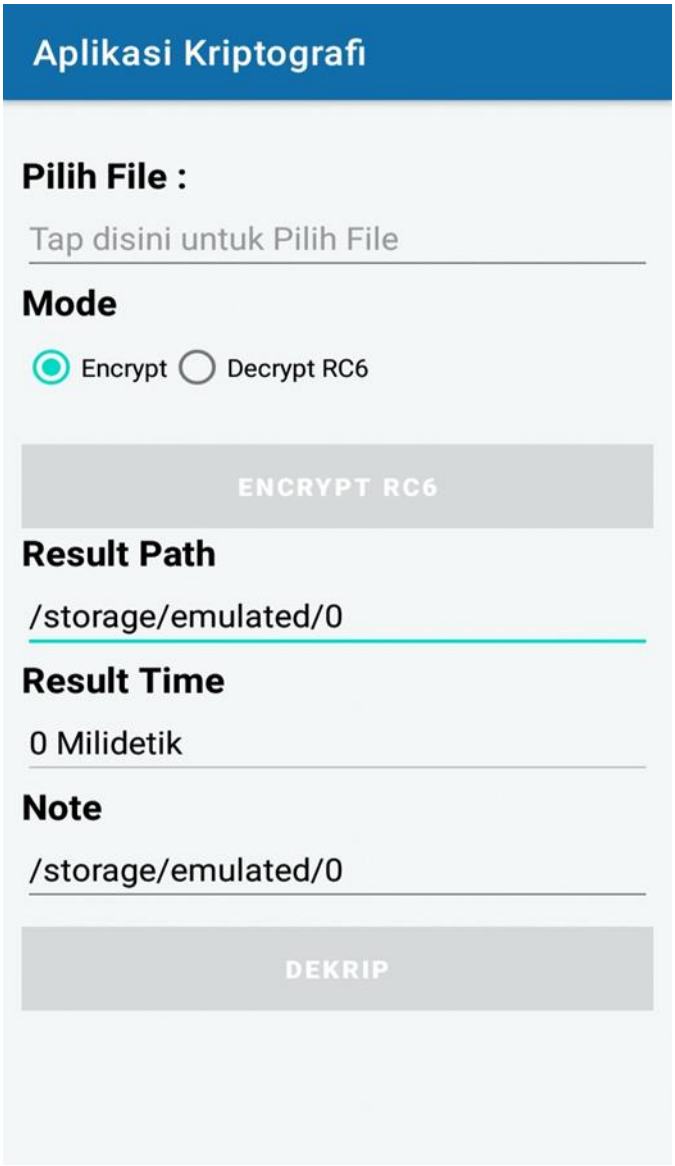
Merupakan tampilan halaman tentang yang digunakan untuk melihat algoritma yang dipakai dalam aplikasi.



Gambar 4.7. Halaman Tentang

3. Halaman Kriptografi

Merupakan tampilan halaman kriptografi yang digunakan untuk melakukan encrypt dan dekrip file.



Gambar 4.8. Halaman Kriptografi

C. Pengujian Algoritma

Pada tahap pengujian algoritma, penulis menguji algoritma yang digunakan yaitu algoritma RC6.

Contoh Perhitungan:

Untuk menerapkan algoritma RC6 dengan teks "selamat pagi" dan password "ABCDEFGHIJKLMNOP", kita perlu mengikuti beberapa langkah.

Algoritma RC6 adalah blok cipher yang menggunakan kunci simetris dan beroperasi pada blok data 128-bit. Berikut ini adalah langkah-langkah dasar perhitungan manual RC6:

Langkah 1: Konversi Teks dan Password ke dalam Bentuk Biner

Teks asli: "selamat pagi"

- Ubah setiap karakter menjadi representasi biner ASCII:

  s: 01110011

  e: 01100101

  l: 01101100

  a: 01100001

  m: 01101101

  a: 01100001

  t: 01110100

  : 00100000

  p: 01110000

  a: 01100001

  g: 01100111

  i: 01101001

Gabungkan semua biner tersebut:

01110011 01100101 01101100 01100001 01101101 01100001
01110100 00100000 01110000 01100001 01100111 01101001

Password: "ABCDEFGHIJKLMNOP"

- Ubah setiap karakter menjadi representasi biner ASCII:

A: 01000001

B: 01000010

C: 01000011

D: 01000100

E: 01000101

F: 01000110

G: 01000111

H: 01001000

I: 01001001

J: 01001010

K: 01001011

L: 01001100

M: 01001101

N: 01001110

O: 01001111

P: 01010000

Gabungkan semua biner tersebut:

```
01000001 01000010 01000011 01000100 01000101 01000110
01000111 01001000 01001001 01001010 01001011 01001100
01001101 01001110 01001111 01010000
```

Langkah 2: Pad Data Blok

RC6 menggunakan blok 128-bit. Data kita (teks asli) harus dipad hingga mencapai kelipatan 128-bit. Dalam contoh ini, kita perlu menambahkan padding. Teks asli biner saat ini adalah 96-bit (12 karakter x 8-bit). Artinya kita perlu menambahkan 32-bit padding (Agar Genap 128Bit):

01110011 01100101 01101100 01100001 01101101 01100001
01110100 00100000 01110000 01100001 01100111 01101001
00000000 00000000 00000000 00000000

Langkah 3: Kunci Ekspansi

RC6 memerlukan kunci yang diperluas dari kunci asli. Untuk kunci 128-bit, kita mengubah kunci ASCII ke biner:

01000001 01000010 01000011 01000100 01000101 01000110
01000111 01001000 01001001 01001010 01001011 01001100
01001101 01001110 01001111 01010000

Selanjutnya, lakukan ekspansi kunci menggunakan algoritma RC6 untuk menghasilkan array kunci yang diperluas.

Langkah 4: Proses Enkripsi

Algoritma RC6 terdiri dari langkah-langkah berikut:

1. **Initial Add Round Keys:** Tambahkan kunci pada awal blok data. Tambahkan kunci pada awal blok data. Misalnya, kita bagi blok data menjadi empat bagian A, B, C, D (masing-masing 32-bit):

```
A = 01110011 01100101 01101100 01100001
B = 01101101 01100001 01110100 00100000
C = 01110000 01100001 01100111 01101001
D = 00000000 00000000 00000000 00000000
```

Misalnya kita tambahkan kunci awal (misal S[0], S[1], S[2], S[3] adalah kunci awal hasil ekspansi kunci):

```
A = A + S[0]
B = B + S[1]
C = C + S[2]
D = D + S[3]
```

2. Feistel-like Rounds: Lakukan 20 putaran enkripsi dengan operasi seperti rotasi bit, XOR, dan penambahan menggunakan kunci yang diperluas. Lakukan 20 putaran enkripsi. Tiap putaran melibatkan operasi seperti rotasi bit, XOR, dan penambahan. Contoh operasi pada satu putaran:

```
for i = 1 to 20 do
t = (B * (2B + 1)) <<< lg w
u = (D * (2D + 1)) <<< lg w
A = ((A ^ t) <<< u) + S[2i]
C = ((C ^ u) <<< t) + S[2i+1]
(A, B, C, D) = (B, C, D, A)
```

3. Final Transformation: Terapkan transformasi akhir dan tambahkan kunci, Misalnya:

Proses ini cukup rumit dan melibatkan berbagai operasi bitwise. Kita tidak dapat melakukannya secara manual tanpa bantuan perangkat lunak.

```
B = B + S[42]
D = D + S[43]
```

Contoh Hasil Akir

Setelah melakukan 20 putaran dan transformasi akhir, kita dapatkan hasil akhir dalam bentuk biner, misalnya:

```
A = 11001010 10110110 11001011 10010101
B = 10010001 11100001 01101010 01011110
C = 01111110 01100001 10101011 10011100
D = 01011001 11110001 10001011 10101110
```

Gabungkan A, B, C, dan D untuk mendapatkan ciphertext 128-bit:

```
11001010 10110110 11001011 10010101 10010001 11100001
01101010 01011110 01111110 01100001 10101011 10011100
01011001 11110001 10001011 10101110
```

Ciphertext dalam bentuk heksadesimal:

```
CA B6 CB 95 91 E1 6A 5E 7E 61 AB 9C 59 F1 8B AE
```

Ini adalah contoh representasi hasil akhir dari enkripsi RC6 pada teks "selamat pagi" dengan password "ABCDEFGHIJKLMNOP".

Langkah 5: Hasil Akhir

Setelah melakukan semua langkah di atas, kita mendapatkan ciphertext dalam bentuk biner.Ciphertext ini dapat dikonversi kembali ke bentuk teks atau representasi heksadesimal. Algoritma RC6 sebenarnya cukup kompleks dan lebih baik diimplementasikan menggunakan perangkat lunak kriptografi yang andal untuk memastikan keakuratan.

Selanjutnya, hasil adalah dalam bentuk HEXA…Karena mengubah ciphertext hasil dari algoritma RC6 menjadi bentuk ASCII dapat menghasilkan karakter yang tidak dapat ditampilkan atau karakter yang tidak dapat dibaca karena hasil enkripsi adalah data biner yang tidak selalu sesuai dengan karakter ASCII yang dapat ditampilkan. Namun, kita bisa mencoba mengonversi hasil biner dari ciphertext menjadi string ASCII untuk melihat hasilnya. Berikut adalah prosesnya:

Ubah setiap byte biner ke bentuk desimal:

```
11001010 -> 202
10110110 -> 182
11001011 -> 203
10010101 -> 149
10010001 -> 145
11100001 -> 225
01101010 -> 106
```

01011110 -> 94

01111110 -> 126

01100001 -> 97

10101011 -> 171

10011100 -> 156

01011001 -> 89

11110001 -> 241

10001011 -> 139

10101110 -> 174

Konversi ke Karakter ASCII: Ubah setiap nilai desimal ke karakter ASCII (bisa menggunakan tabel ASCII untuk referensi):

202 -> Ê

182 -> ¶

203 -> Ë

149 -> •

145 -> ‘

225 -> á

106 -> j

94 -> ^

126 -> ~

97 -> a

171 -> «

```
156 -> œ

89 -> Y

241 -> ñ

139 -> ‹

174 -> ®
```

Gabungkan semua karakter ASCII tersebut menjadi string:

```
Ê¶Ë•‘áj^~a«œYñ‹®
```

Mengonversi hasil ciphertext ke dalam bentuk ASCII dapat menghasilkan karakter yang tidak dapat dibaca atau karakter yang spesial. Inilah sebabnya seringkali ciphertext ditampilkan dalam bentuk heksadesimal atau base64 untuk memastikan bahwa semua data dapat diwakili dalam format yang dapat dibaca dan diolah.

Praktek kode:

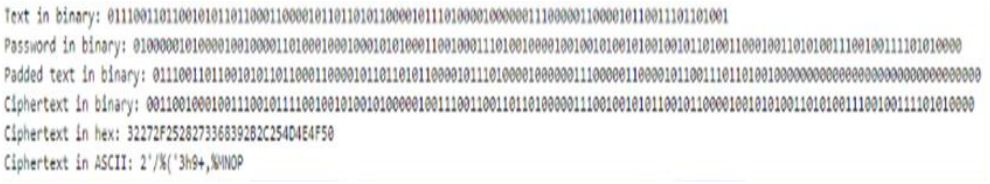
Berikut adalah contoh kode Python yang menunjukkan proses tersebut:

```
import binascii

def text_to_bin(text):

# Convert text to binary

return ''.join(format(ord(c), '08b') for c in text)

def bin_to_text(binary):

# Convert binary to text

binary_values = [binary[i:i+8] for i in range(0, len(binary), 8)]
```

```
ascii_text = ''.join(chr(int(bv, 2)) for bv in binary_values)
return ascii_text
def pad_text(bin_text, block_size=128):
# Pad text to fit into 128-bit blocks
padding = block_size - (len(bin_text) % block_size)
bin_text += '0' * padding
return bin_text
def xor(a, b):
# XOR two binary strings
return ''.join('1' if x != y else '0' for x, y in zip(a, b))
def encrypt_rc6_step(text_bin, key_bin):
# Basic example of RC6 encryption step
A = text_bin[:32]
B = text_bin[32:64]
C = text_bin[64:96]
D = text_bin[96:128]
S = [key_bin[i:i+32] for i in range(0, len(key_bin), 32)]
A = xor(A, S[0])
B = xor(B, S[1])
C = xor(C, S[2])
D = xor(D, S[3])
return A + B + C + D
# Input text and password
```

```
text = "selamat pagi"
password = "ABCDEFGHIJKLMNOP"
# Step 1: Convert text and password to binary
text_bin = text_to_bin(text)
password_bin = text_to_bin(password)
print("Text in binary:", text_bin)
print("Password in binary:", password_bin)
# Step 2: Pad text to 128-bit block size
padded_text_bin = pad_text(text_bin)
print("Padded text in binary:", padded_text_bin)
# Step 3: Split text into blocks (if needed)
# (Here we assume a single block for simplicity)
# Step 4: Perform RC6 encryption step (simplified)
cipher_bin = encrypt_rc6_step(padded_text_bin, password_bin)
print("Ciphertext in binary:", cipher_bin)
# Step 5: Convert binary ciphertext to hex
cipher_hex = hex(int(cipher_bin, 2))[2:].upper()
print("Ciphertext in hex:", cipher_hex)
# Step 6: Convert binary ciphertext to ASCII
cipher_text = bin_to_text(cipher_bin)
print("Ciphertext in ASCII:", cipher_text)
```

List Source Code RC6



Gambar Hasil Pyton Rc6

Penjelasan pada kode:

1. **Konversi Teks dan Password ke Biner:**
   - Menggunakan text_to_bin untuk mengubah teks dan password ke biner.
2. **Pad Teks ke Ukuran Blok 128-bit:**
   - Menambahkan padding untuk membuat panjang biner menjadi kelipatan 128-bit.
3. **Enkripsi Sederhana dengan XOR:**
   - Melakukan operasi XOR antara blok teks dan kunci (S).
4. **Konversi Ciphertext ke Heksadesimal dan ASCII:**
   - Mengonversi hasil biner ke heksadesimal dan ASCII untuk ditampilkan.

PROSES DEKRIP:

Berikut adalah contoh kode Python yang menunjukkan proses enkripsi dan dekripsi sederhana menggunakan algoritma RC6, dengan menampilkan prosesnya langkah demi langkah.

List Source Code Enkrip Dekrip RC6

```
import binascii

def text_to_bin(text):

# Convert text to binary

return ''.join(format(ord(c), '08b') for c in text)

def bin_to_text(binary):

# Convert binary to text

binary_values = [binary[i:i+8] for i in range(0, len(binary), 8)]

ascii_text = ''.join(chr(int(bv, 2)) for bv in binary_values)

return ascii_text

def pad_text(bin_text, block_size=128):

# Pad text to fit into 128-bit blocks

padding = block_size - (len(bin_text) % block_size)

bin_text += '0' * padding

return bin_text

def xor(a, b):

# XOR two binary strings

return ''.join('1' if x != y else '0' for x, y in zip(a, b))

def encrypt_rc6_step(text_bin, key_bin):

# Basic example of RC6 encryption step

A = text_bin[:32]

B = text_bin[32:64]

C = text_bin[64:96]
```

```
D = text_bin[96:128]

S = [key_bin[i:i+32] for i in range(0, len(key_bin), 32)]

A = xor(A, S[0])

B = xor(B, S[1])

C = xor(C, S[2])

D = xor(D, S[3])

return A + B + C + D

# Input text and password

text = "selamat pagi"

password = "ABCDEFGHIJKLMNOP"

# Step 1: Convert text and password to binary

text_bin = text_to_bin(text)

password_bin = text_to_bin(password)

print("Text in binary:", text_bin)

print("Password in binary:", password_bin)

# Step 2: Pad text to 128-bit block size

padded_text_bin = pad_text(text_bin)

print("Padded text in binary:", padded_text_bin)

# Step 3: Split text into blocks (if needed)

# (Here we assume a single block for simplicity)

# Step 4: Perform RC6 encryption step (simplified)

cipher_bin = encrypt_rc6_step(padded_text_bin, password_bin)

print("Ciphertext in binary:", cipher_bin)
```

```
# Step 5: Convert binary ciphertext to hex
cipher_hex = hex(int(cipher_bin, 2))[2:].upper()
print("Ciphertext in hex:", cipher_hex)
# Step 6: Convert binary ciphertext to ASCII
cipher_text = bin_to_text(cipher_bin)
print("Ciphertext in ASCII:", cipher_text)
```

List source kode dekrip file

```
Text in binary: 01110011011001010110110001100001011011010110000101110100001000000111000001100001011001110110100
Password in binary: 01000001010000010010000110100010001000101010001100100011101001000010010010100101001001011010011000100110101001110010011110101000
Padded text in binary: 01110011011001010110110001100001011011010110000101110100001000000111000001100001011001110110100000000000000000000000000000000000
Ciphertext in binary: 00110010000100111001011110010010100101000001001110011001110110100000111001001010110010110000100101010011010010111001001111010000
Ciphertext in hex: 32272F2528273368392B2C25404E4F50
Ciphertext in ASCII: 2'/%('3h9+,%@NOP
Decrypted binary: 01110011011001010110110001100001011011010110000101110100001000000111000001100001011001110110100000000000000000000000000000000000
Decrypted text: selamat pagi
```

Gambar Hasil Dekrip RC6

Metodologi Enkripsi dan Dekripsi RC6 Sederhana

1. **Konversi Teks dan Password ke Biner**

   Langkah:

   - Ubah setiap karakter dalam teks dan password menjadi representasi biner menggunakan tabel ASCII. Contoh:
   - Teks: "selamat pagi"
   - Hasil biner: 01110011 01100101 01101100 01100001 01101101 01100001 01110100 00100000 01110000 01100001 01100111 01101001

2. **Padding Teks ke Ukuran Blok 128-bit**

   Langkah:

● Tambahkan padding berupa nol untuk membuat panjang teks biner menjadi kelipatan 128-bit.

Contoh:

● Teks biner asli: 01110011 01100101 01101100 01100001 01101101 01100001 01110100 00100000 01110000 01100001 01100111 01101001

● Teks biner setelah padding: 01110011 01100101 01101100 01100001 01101101 01100001 01110100 00100000 01110000 01100001 01100111

01101001 00000000 00000000 00000000 00000000

3. Pembagian Blok Data

Langkah:

● Pisahkan teks biner yang sudah dipadding menjadi blok-blok 128-bit. Pada contoh ini, hanya ada satu blok.

4. Ekspansi Kunci

Langkah:

● Konversi password menjadi biner dan bagi menjadi bagian 32-bit untuk setiap kunci (S).

Contoh:

● Password: "ABCDEFGHIJKLMNOP"

● Password biner: 01000001 01000010 01000011 01000100 01000101 01000110 01000111 01001000 01001001 01001010 01001011 01001100 01001101 01001110 01001111 01010000

- S = [S[0], S[1], S[2], S[3]]: ['01000001 01000010 01000011 01000100', '01000101 01000110 01000111 01001000', '01001001 01001010
01001011 01001100', '01001101 01001110 01001111 01010000']

5. Enkripsi dengan XOR

Langkah:

- Lakukan operasi XOR antara setiap bagian dari blok teks biner dengan kunci yang sesuai.

Contoh:

- Blok teks biner: 01110011 01100101 01101100 01100001 01101101 01100001 01110100 00100000 01110000 01100001 01100111 01101001 00000000 00000000 00000000 00000000
- Hasil enkripsi: 00110010 00100111 00101111 00100100 00101000 00100111 00101111 00101000 00101000 00101111 00100111 00101100 01001101 01001110 01001111 01010000

6. Konversi Ciphertext ke Heksadesimal dan ASCII

Langkah:

- Konversi hasil biner ke bentuk heksadesimal dan ASCII untuk ditampilkan.

Contoh:

- Ciphertext biner: 00110010 00100111 00101111 00100100 00101000 00100111 00101111 00101000 00101000 00101111 00100111 00101100 01001101 01001110 01001111 01010000

- Ciphertext heksadesimal: 32272F24 28272F28 282F272C 4D4E4F50
- Ciphertext ASCII: 2'/$('/('/(,'MNO

7. Dekripsi dengan XOR

Langkah:

- Lakukan operasi XOR antara setiap bagian dari blok ciphertext biner dengan kunci yang sesuai untuk mendapatkan kembali teks asli.

Contoh:

- Ciphertext biner: 00110010 00100111 00101111 00100100 00101000 00100111 00101111 00101000 00101000 00101111 00100111 00101100 01001101 01001110 01001111 01010000
- Hasil dekripsi biner: 01110011 01100101 01101100 01100001 01101101 01100001 01110100 00100000 01110000 01100001 01100111 01101001 00000000 00000000 00000000 00000000
- Hasil dekripsi teks: selamat pagi����

Metodologi di atas memberikan panduan langkah demi langkah untuk enkripsi dan dekripsi sederhana menggunakan algoritma RC6.

D. Implementasi

Implementasi sistem merupakan tahap penerapan dari suatu teknologi yang didesain untuk siap dioperasikan. Tahap ini merupakan terjemahan perancangan dari bab hasil analisis sebelumnya dalam suatu bahasa pemrograman. Bahasa pemrograman yang digunakan untuk membangun Implementasi Kriptografi Pada Teks Menggunakan Algoritma Rc6 Berbasisi Android menggunakan bahasa pemrograman *JavaScript* dengan *framework* Android Studio.

1. Kebutuhan perangkat keras. Spesifikasi minimum perangkat keras sebagai berikut :

Tabel 4.2 Kebutuhan Perangkat keras

Jenis	Spesifikasi
Laptop/PC	ACER
Processor	Intel(R) Celeron(R) N4000 CPU @ 1.10Ghz 1.10Ghz
Memory	4GB RAM
SSD	1TB

2. Kebutuhan perangkat lunak. Spesifikasi minimum perangkat lunak sebagai berikut :

Tabel 4.3 Kebutuhan Perangkat Lunak

Jenis	Spesifikasi
Sistem Operasi	*Windows 10 Home Single Language*
Framework	*Android Studio*

E. Pengujian Sistem

1. *BlackBox*

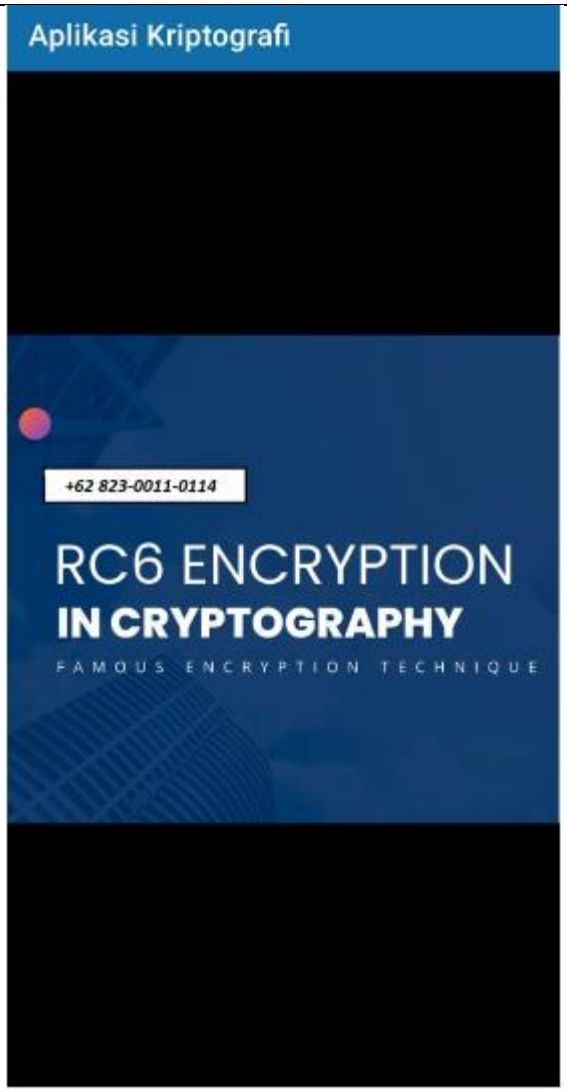
Pengujian sistem dilakukan dengan cara pengujian *BlackBox*. *BlackBox* adalah metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi tanpa mengintip ke dalam struktur atau cara kerja internalnya. Metode pengujian ini dapat diterapkan secara *virtual* ke setiap tingkat pengujian perangkat lunak: unit, integrasi, sistem, dan penerimaan.

Tabel 4.4. *BlackBox* Halaman Utama

Test Faktor	Hasil	Kesimpulan
Jika pertama kali membuka aplikasi.	✓	Informasi, tampil halaman utama.
Screen Shot		
		

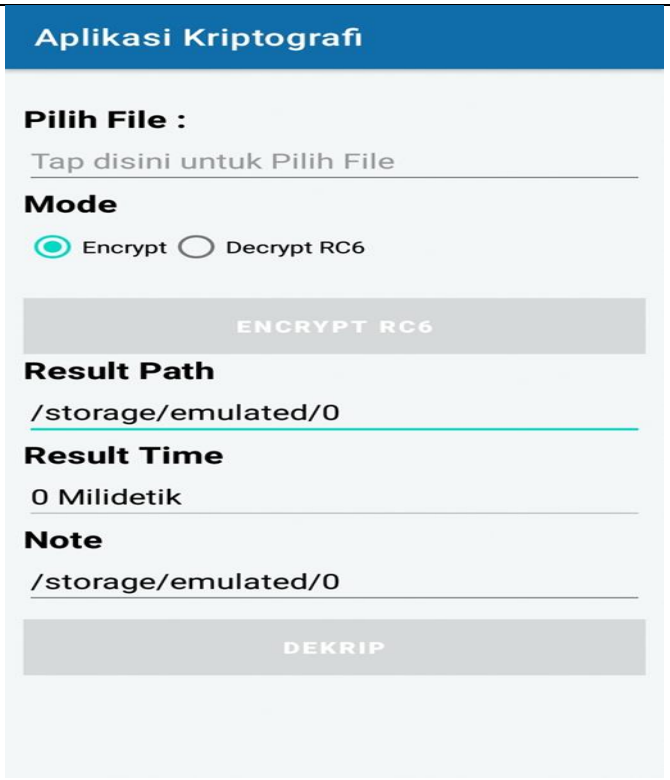
Tabel 4.5. *BlackBox Form* Tentang

Test Faktor	Hasil	Kesimpulan
Jika memilih menu tentang pada halaman utama.	✓	Informasi, tampil form tentang.
Screen Shot		



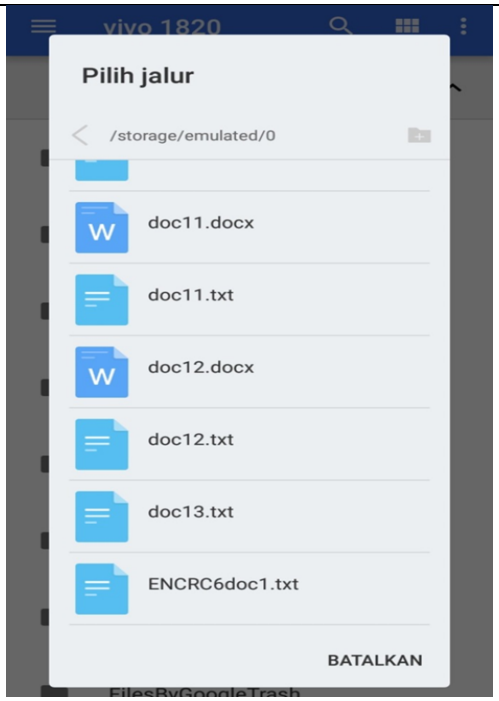
Tabel 4.6 *BlackBox* Form Pilih File

Test Faktor	**Hasil**	**Kesimpulan**
Jika menekan menu pilih file pada halaman utama.	✓	Informasi, tampil form pilih file.
Screen Shot		



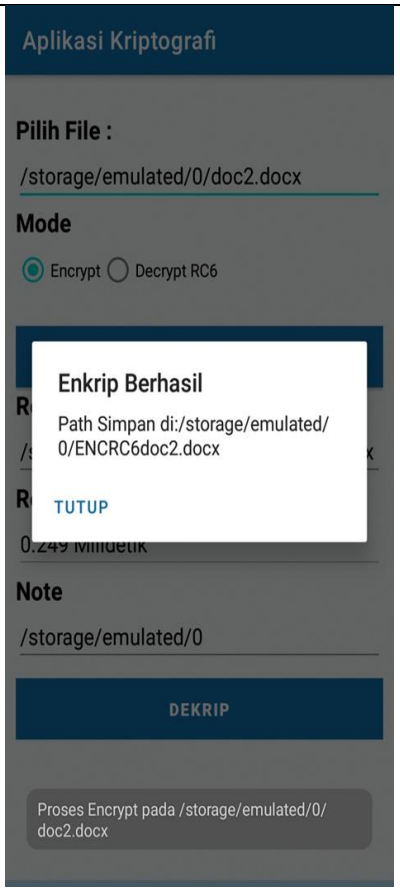
Tabel 4.7 BlackBox Form *Tambah Lapangan*

Test Faktor	Hasil	Kesimpulan
Jika menekan double tap pilih file pada form pilih file.	✓	Informasi, tampil open device manager.
Screen Shot		



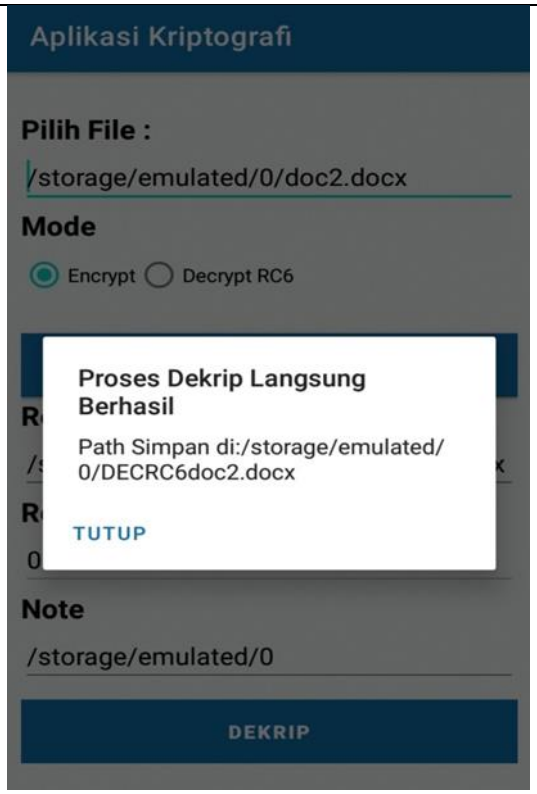
Tabel 4.8. *BlackBox* Proses Encrypt

Test Faktor	**Hasil**	**Kesimpulan**
Jika telah memilih file kemudian menekan tombol encrypt.	✓	Informasi, tampil pesan proses encrypt.
Screen Shot		

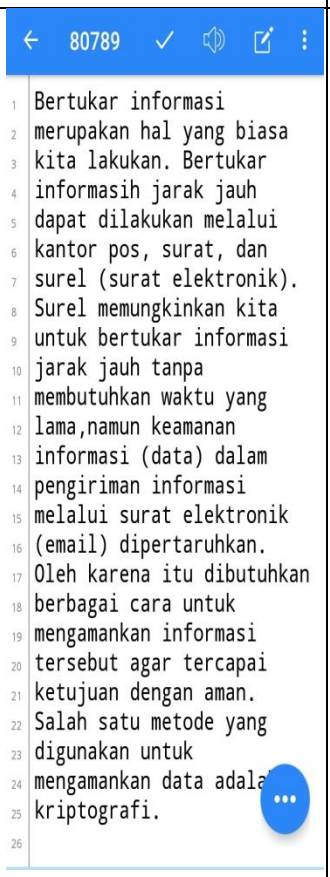
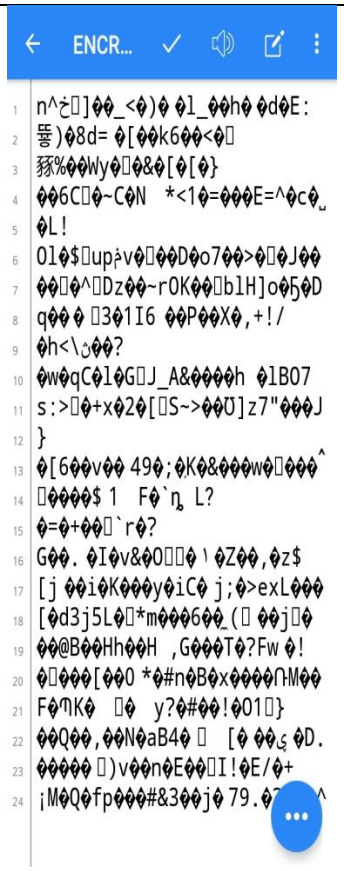
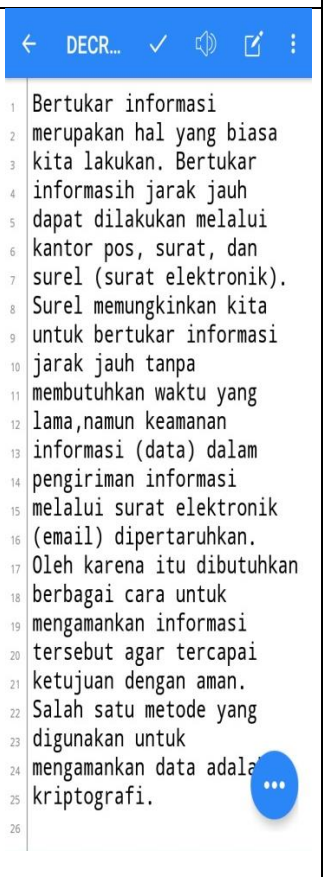


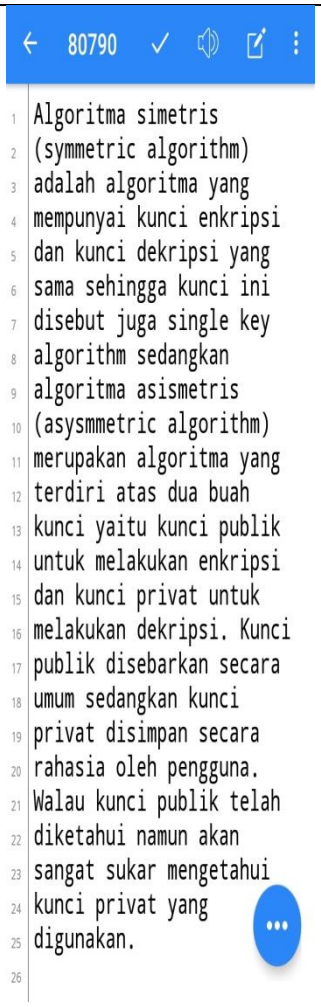
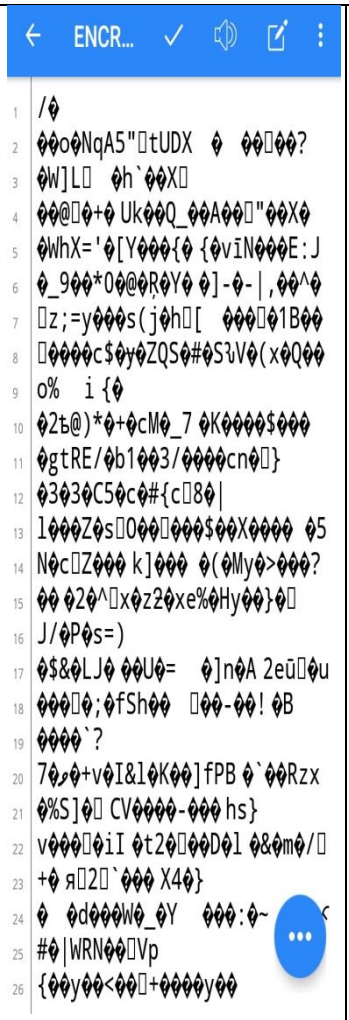
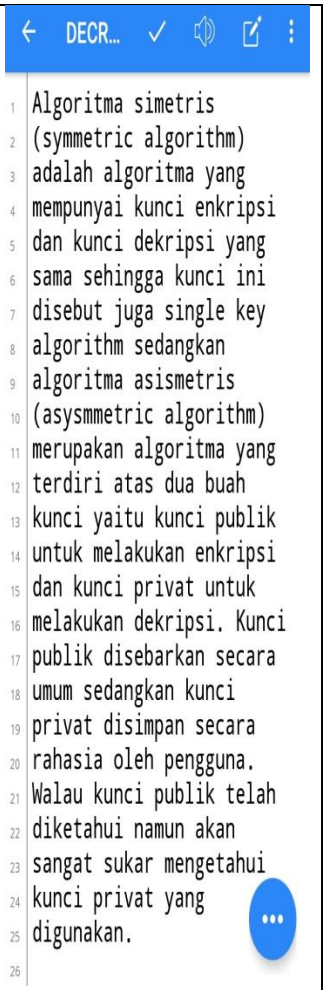
Tabel 4. 9. BlackBox Proses Dekrip

Test Faktor	**Hasil**	**Kesimpulan**
Jika memilih file yang telah ter encrypt kemudian menakan tombol dekrip.	✓	Informasi, tampil proses dekrip.
Screen Shot		

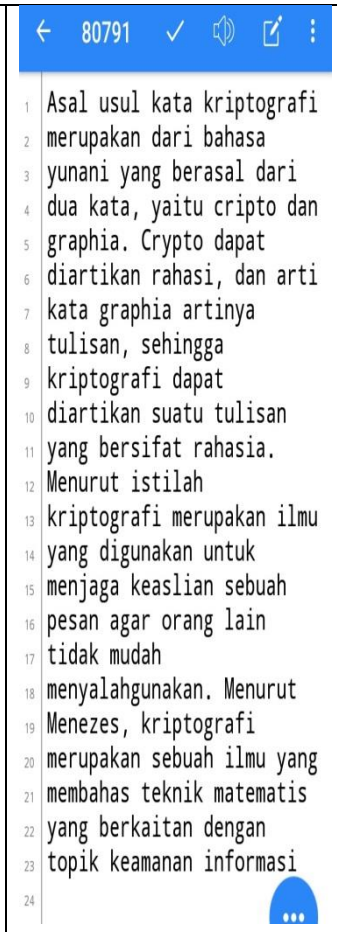

Aplikasi Kriptografi
Pilih File :
/storage/emulated/0/doc2.docx
Mode
Encrypt
Decrypt RC6
Proses Dekrip Langsung Berhasil
Path Simpan di:/storage/emulated/0/DECRC6doc2.docx
TUTUP
Note
/storage/emulated/0
DEKRIP

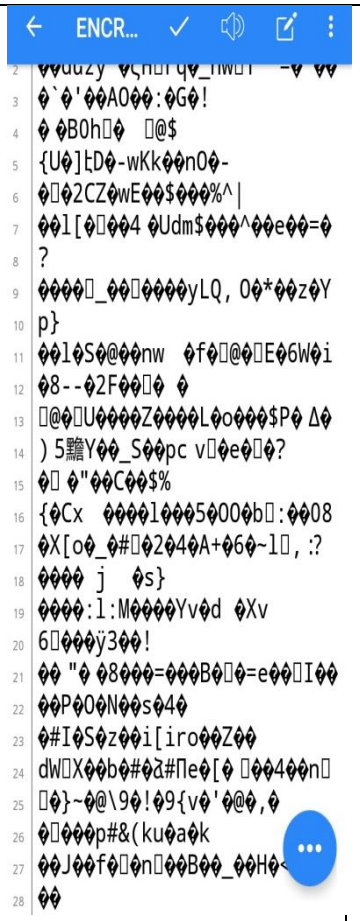
Tabel 4.8. Proses Enkripsi Dan Dekripsi File Teks

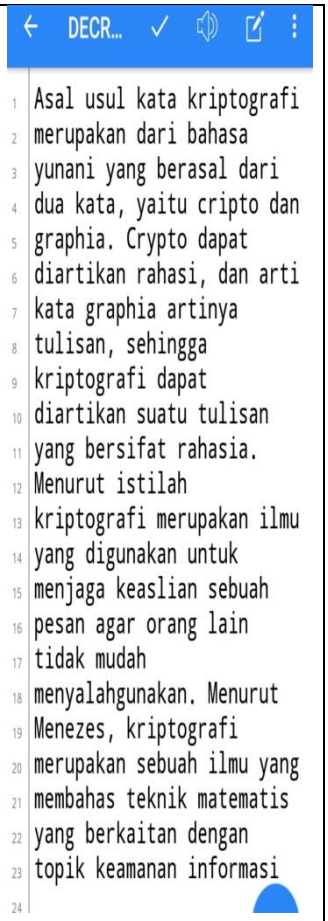
No	Plaintext	Hasil enkripsi (chiphertext)	Hasil dekripsi (plaintext)
1.			

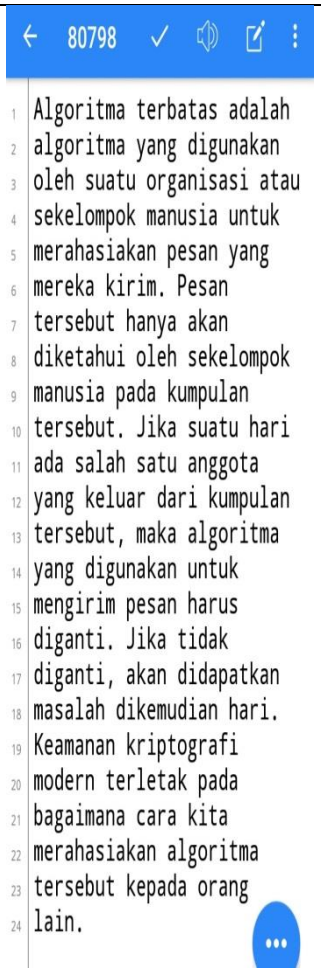
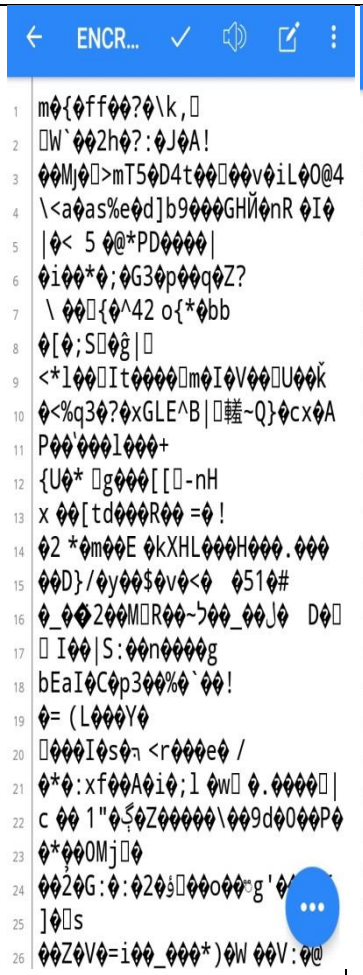
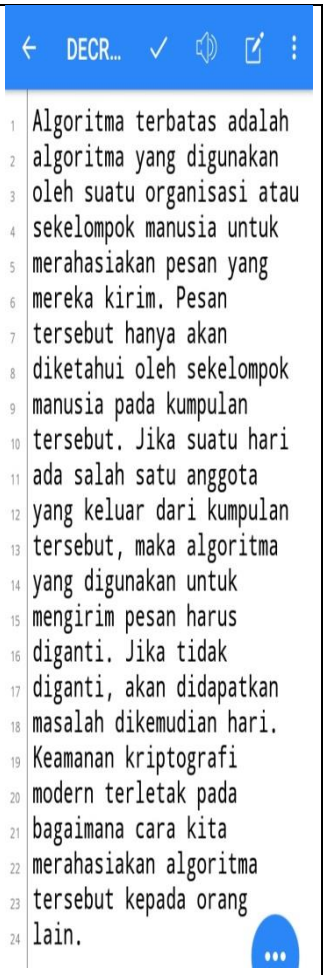
2.	 80790 Algoritma simetris (symmetric algorithm) adalah algoritma yang mempunyai kunci enkripsi dan kunci dekripsi yang sama sehingga kunci ini disebut juga single key algorithm sedangkan algoritma asismetris (asysmmetric algorithm) merupakan algoritma yang terdiri atas dua buah kunci yaitu kunci publik untuk melakukan enkripsi dan kunci privat untuk melakukan dekripsi. Kunci publik disebarkan secara umum sedangkan kunci privat disimpan secara rahasia oleh pengguna. Walau kunci publik telah diketahui namun akan sangat sukar mengetahui kunci privat yang digunakan.	 ENCR...	 DECR... Algoritma simetris (symmetric algorithm) adalah algoritma yang mempunyai kunci enkripsi dan kunci dekripsi yang sama sehingga kunci ini disebut juga single key algorithm sedangkan algoritma asismetris (asysmmetric algorithm) merupakan algoritma yang terdiri atas dua buah kunci yaitu kunci publik untuk melakukan enkripsi dan kunci privat untuk melakukan dekripsi. Kunci publik disebarkan secara umum sedangkan kunci privat disimpan secara rahasia oleh pengguna. Walau kunci publik telah diketahui namun akan sangat sukar mengetahui kunci privat yang digunakan.

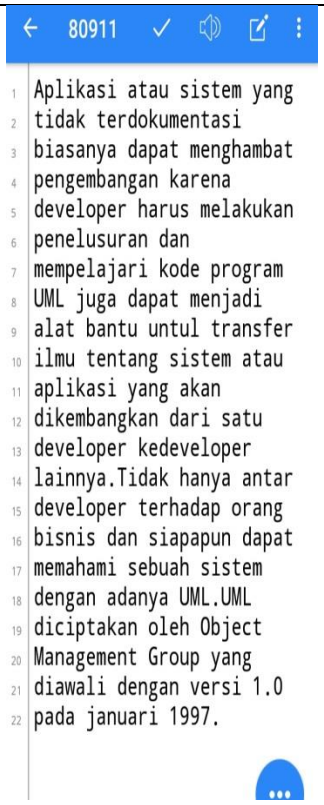
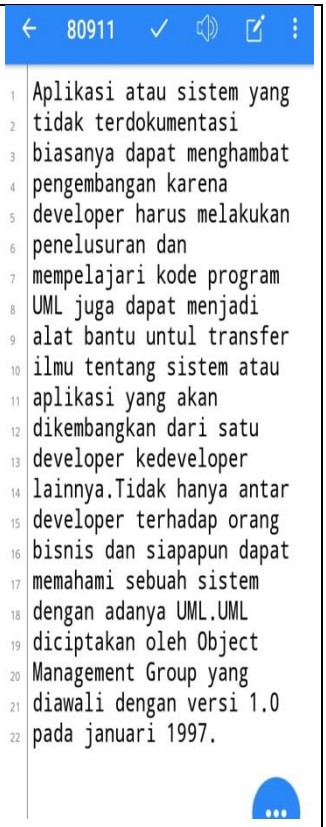
3.

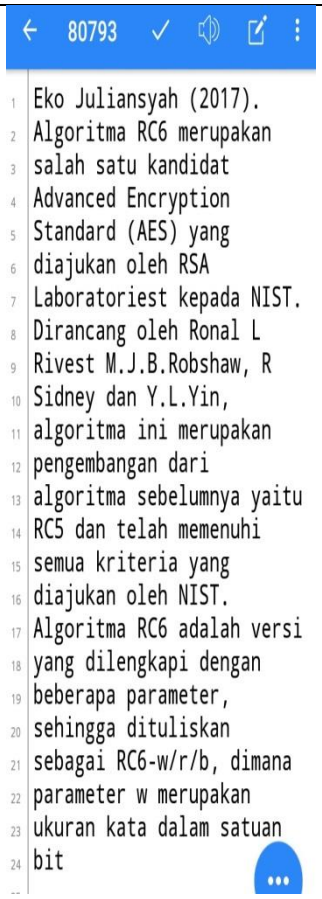
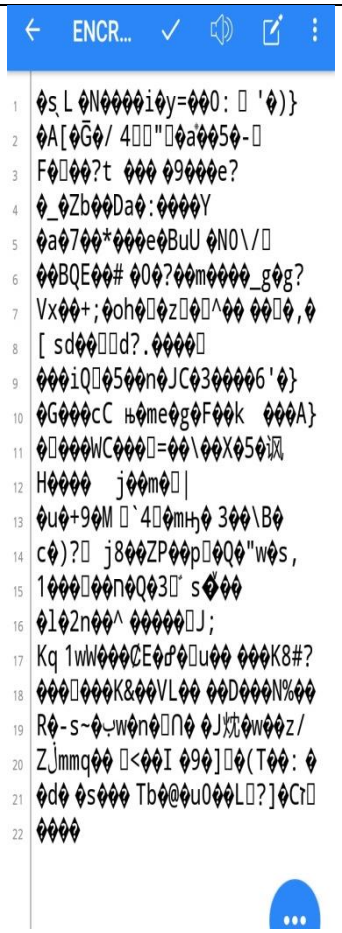
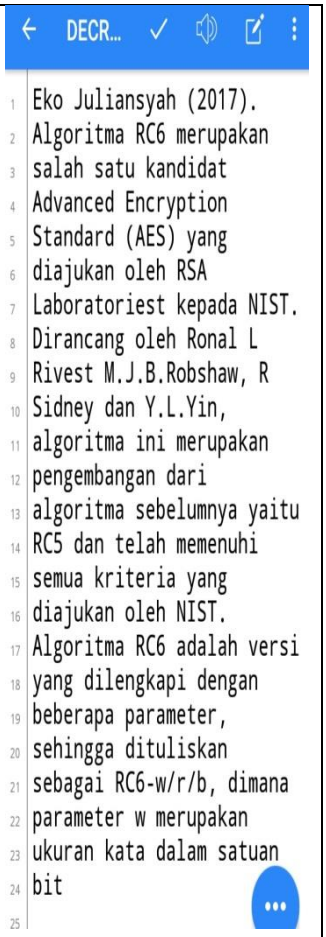

80791
Asal usul kata kriptografi
merupakan dari bahasa
yunani yang berasal dari
dua kata, yaitu cripto dan
graphia. Crypto dapat
diartikan rahasi, dan arti
kata graphia artinya
tulisan, sehingga
kriptografi dapat
diartikan suatu tulisan
yang bersifat rahasia.
Menurut istilah
kriptografi merupakan ilmu
yang digunakan untuk
menjaga keaslian sebuah
pesan agar orang lain
tidak mudah
menyalahgunakan. Menurut
Menezes, kriptografi
merupakan sebuah ilmu yang
membahas teknik matematis
yang berkaitan dengan
topik keamanan informasi

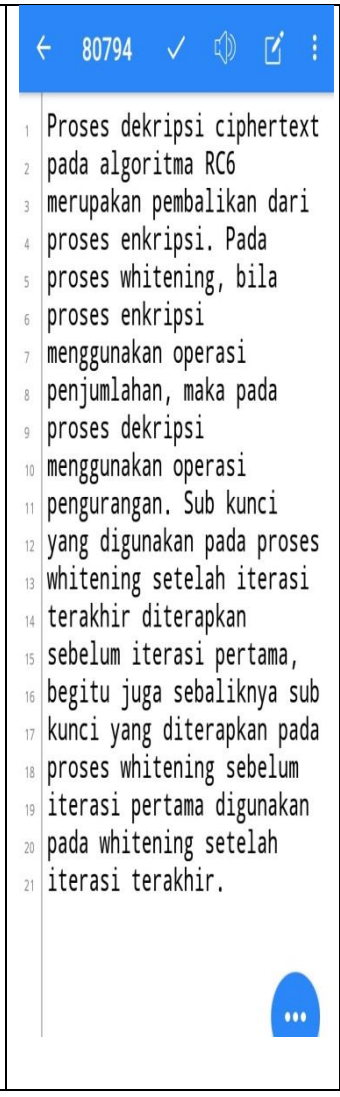
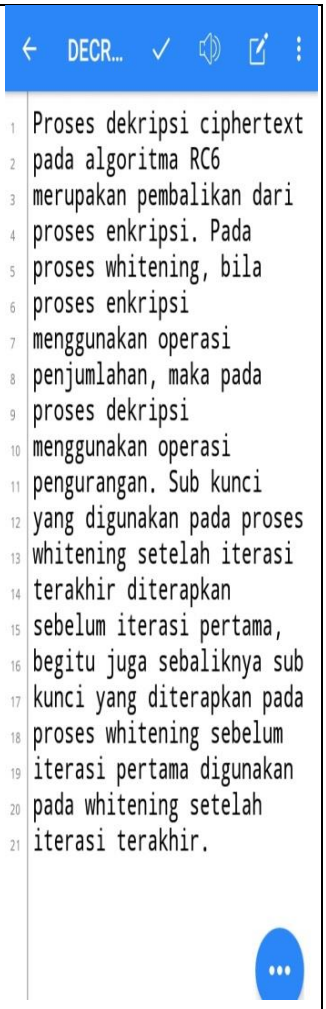

ENCR...

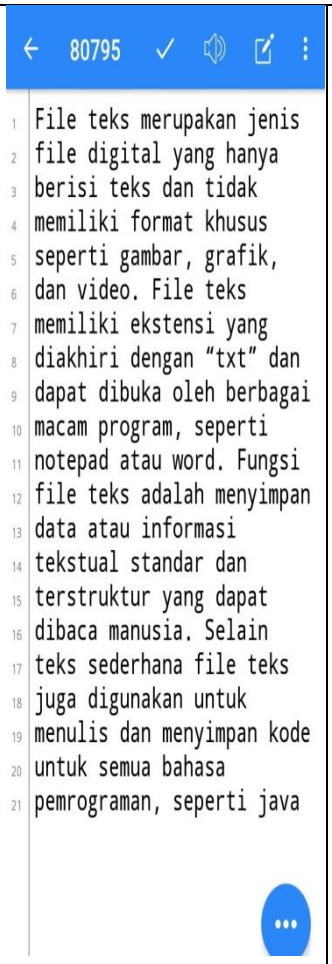
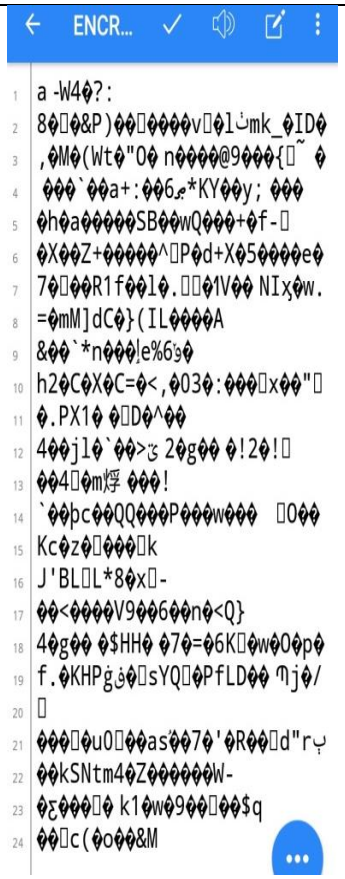
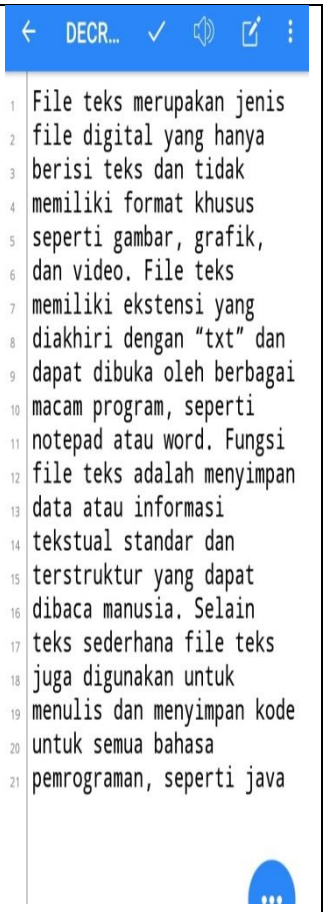

DECR...
Asal usul kata kriptografi
merupakan dari bahasa
yunani yang berasal dari
dua kata, yaitu cripto dan
graphia. Crypto dapat
diartikan rahasi, dan arti
kata graphia artinya
tulisan, sehingga
kriptografi dapat
diartikan suatu tulisan
yang bersifat rahasia.
Menurut istilah
kriptografi merupakan ilmu
yang digunakan untuk
menjaga keaslian sebuah
pesan agar orang lain
tidak mudah
menyalahgunakan. Menurut
Menezes, kriptografi
merupakan sebuah ilmu yang
membahas teknik matematis
yang berkaitan dengan
topik keamanan informasi

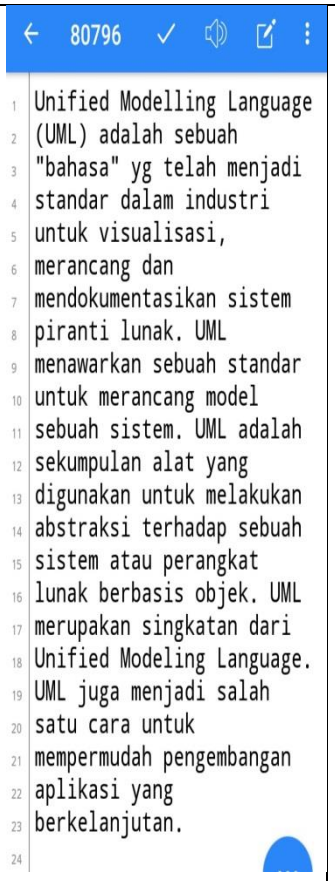
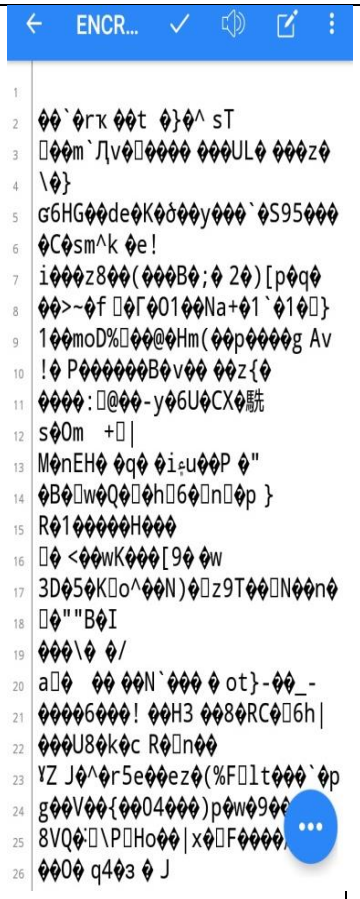
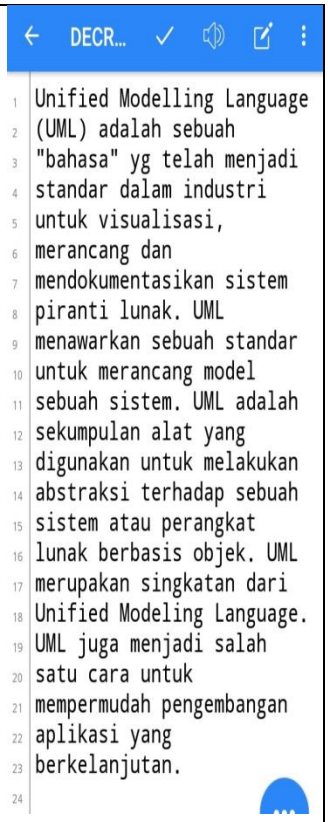
4.	 80798 Algoritma terbatas adalah algoritma yang digunakan oleh suatu organisasi atau sekelompok manusia untuk merahasiakan pesan yang mereka kirim. Pesan tersebut hanya akan diketahui oleh sekelompok manusia pada kumpulan tersebut. Jika suatu hari ada salah satu anggota yang keluar dari kumpulan tersebut, maka algoritma yang digunakan untuk mengirim pesan harus diganti. Jika tidak diganti, akan didapatkan masalah dikemudian hari. Keamanan kriptografi modern terletak pada bagaimana cara kita merahasiakan algoritma tersebut kepada orang lain.	 ENCR...	 DECR... Algoritma terbatas adalah algoritma yang digunakan oleh suatu organisasi atau sekelompok manusia untuk merahasiakan pesan yang mereka kirim. Pesan tersebut hanya akan diketahui oleh sekelompok manusia pada kumpulan tersebut. Jika suatu hari ada salah satu anggota yang keluar dari kumpulan tersebut, maka algoritma yang digunakan untuk mengirim pesan harus diganti. Jika tidak diganti, akan didapatkan masalah dikemudian hari. Keamanan kriptografi modern terletak pada bagaimana cara kita merahasiakan algoritma tersebut kepada orang lain.

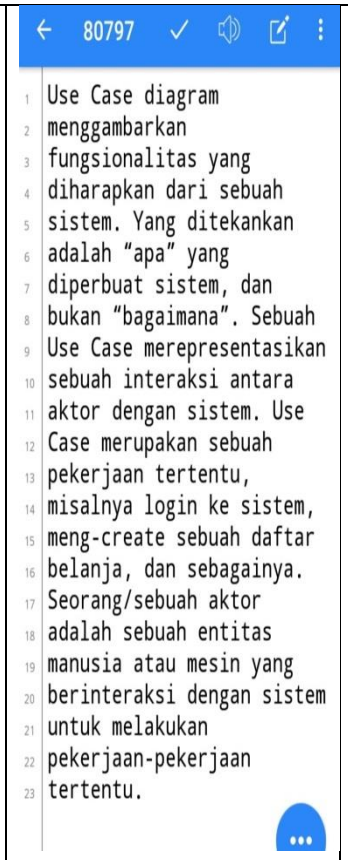
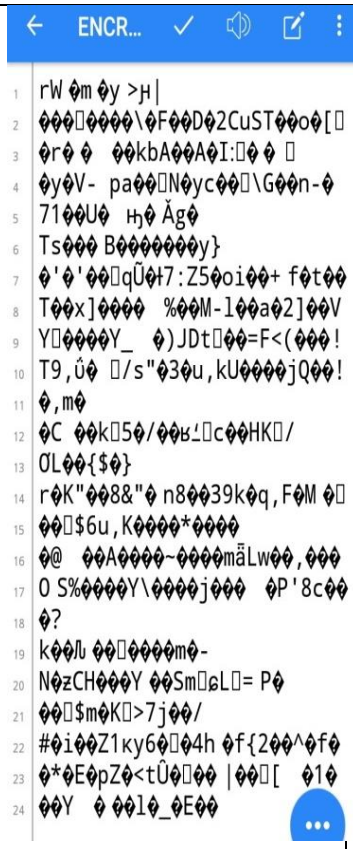
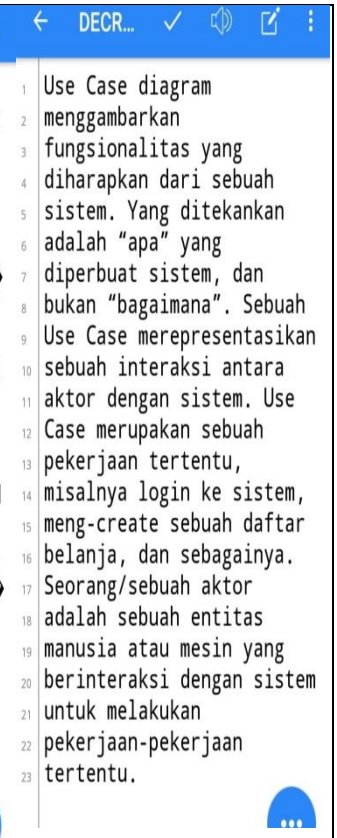
5.	 80911 Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program UML juga dapat menjadi alat bantu untul transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer kedeveloper lainnya.Tidak hanya antar developer terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML.UML diciptakan oleh Object Management Group yang diawali dengan versi 1.0 pada januari 1997.	 ENCR...	 80911 Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena developer harus melakukan penelusuran dan mempelajari kode program UML juga dapat menjadi alat bantu untul transfer ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu developer kedeveloper lainnya.Tidak hanya antar developer terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML.UML diciptakan oleh Object Management Group yang diawali dengan versi 1.0 pada januari 1997.

6.	 80793 Eko Juliansyah (2017). Algoritma RC6 merupakan salah satu kandidat Advanced Encryption Standard (AES) yang diajukan oleh RSA Laboratoriest kepada NIST. Dirancang oleh Ronal L Rivest M.J.B.Robshaw, R Sidney dan Y.L.Yin, algoritma ini merupakan pengembangan dari algoritma sebelumnya yaitu RC5 dan telah memenuhi semua kriteria yang diajukan oleh NIST. Algoritma RC6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b, dimana parameter w merupakan ukuran kata dalam satuan bit	 ENCR...	 DECR... Eko Juliansyah (2017). Algoritma RC6 merupakan salah satu kandidat Advanced Encryption Standard (AES) yang diajukan oleh RSA Laboratoriest kepada NIST. Dirancang oleh Ronal L Rivest M.J.B.Robshaw, R Sidney dan Y.L.Yin, algoritma ini merupakan pengembangan dari algoritma sebelumnya yaitu RC5 dan telah memenuhi semua kriteria yang diajukan oleh NIST. Algoritma RC6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b, dimana parameter w merupakan ukuran kata dalam satuan bit

7.	 80794 Proses dekripsi ciphertext pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses whitening, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses whitening setelah iterasi terakhir diterapkan sebelum iterasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses whitening sebelum iterasi pertama digunakan pada whitening setelah iterasi terakhir.	 ENCR...	 DECR... Proses dekripsi ciphertext pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses whitening, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses whitening setelah iterasi terakhir diterapkan sebelum iterasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses whitening sebelum iterasi pertama digunakan pada whitening setelah iterasi terakhir.

8.	 80795 File teks merupakan jenis file digital yang hanya berisi teks dan tidak memiliki format khusus seperti gambar, grafik, dan video. File teks memiliki ekstensi yang diakhiri dengan “txt” dan dapat dibuka oleh berbagai macam program, seperti notepad atau word. Fungsi file teks adalah menyimpan data atau informasi tekstual standar dan terstruktur yang dapat dibaca manusia. Selain teks sederhana file teks juga digunakan untuk menulis dan menyimpan kode untuk semua bahasa pemrograman, seperti java	 ENCR... a -W4 8 , J'BL 4 f. h2 4 Kc ` = & 7 a+: KY SB Z+ R1f NI dC A e%6 D^ k L*8 V9 n sYQ PfLD j u0 as 7 R d"r kSNtm4 W- k1 $q c M	 DECR... File teks merupakan jenis file digital yang hanya berisi teks dan tidak memiliki format khusus seperti gambar, grafik, dan video. File teks memiliki ekstensi yang diakhiri dengan “txt” dan dapat dibuka oleh berbagai macam program, seperti notepad atau word. Fungsi file teks adalah menyimpan data atau informasi tekstual standar dan terstruktur yang dapat dibaca manusia. Selain teks sederhana file teks juga digunakan untuk menulis dan menyimpan kode untuk semua bahasa pemrograman, seperti java

9.	 80796 Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari Unified Modeling Language. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.	 ENCR...	 DECR... Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. UML adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari Unified Modeling Language. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan.

10.			

2. Persentase validasi

Persentase Validasi Keamanan Data merupakan ukuran seberapa besar data dianggap valid atau aman setelah melalui proses verifikasi atau validasi. Persentase ini penting dalam konteks pengelolaan data, keamanan informasi, dan kepatuhan terhadap standar atau regulasi tertentu.Untuk menghitung persentase validasi keamanan data, Anda biasanya akan mengikuti langkah-langkah berikut:

- Identifikasi Total Data : Tentukan jumlah total data yang akan divalidasi atau diperiksa.

- Identifikasi Data yang Valid : Tentukan jumlah data yang berhasil melewati proses validasi atau dianggap aman.
- Hitung Persentase:
  - Gunakan rumus berikut untuk menghitung persentase validasi keamanan data:
  - Persentase Validasi Keamanan Data = (Jumlah Data yang Valid / Jumlah Total Data) × 100%

Contoh Perhitungan:

Misalkan, Anda memiliki 1000 data yang akan divalidasi, dan setelah proses validasi, ditemukan bahwa 950 data dianggap valid atau aman.

1. Jumlah Total Data: 1000 data
2. Jumlah Data yang Valid: 950 data
3. Hitung Persentase:

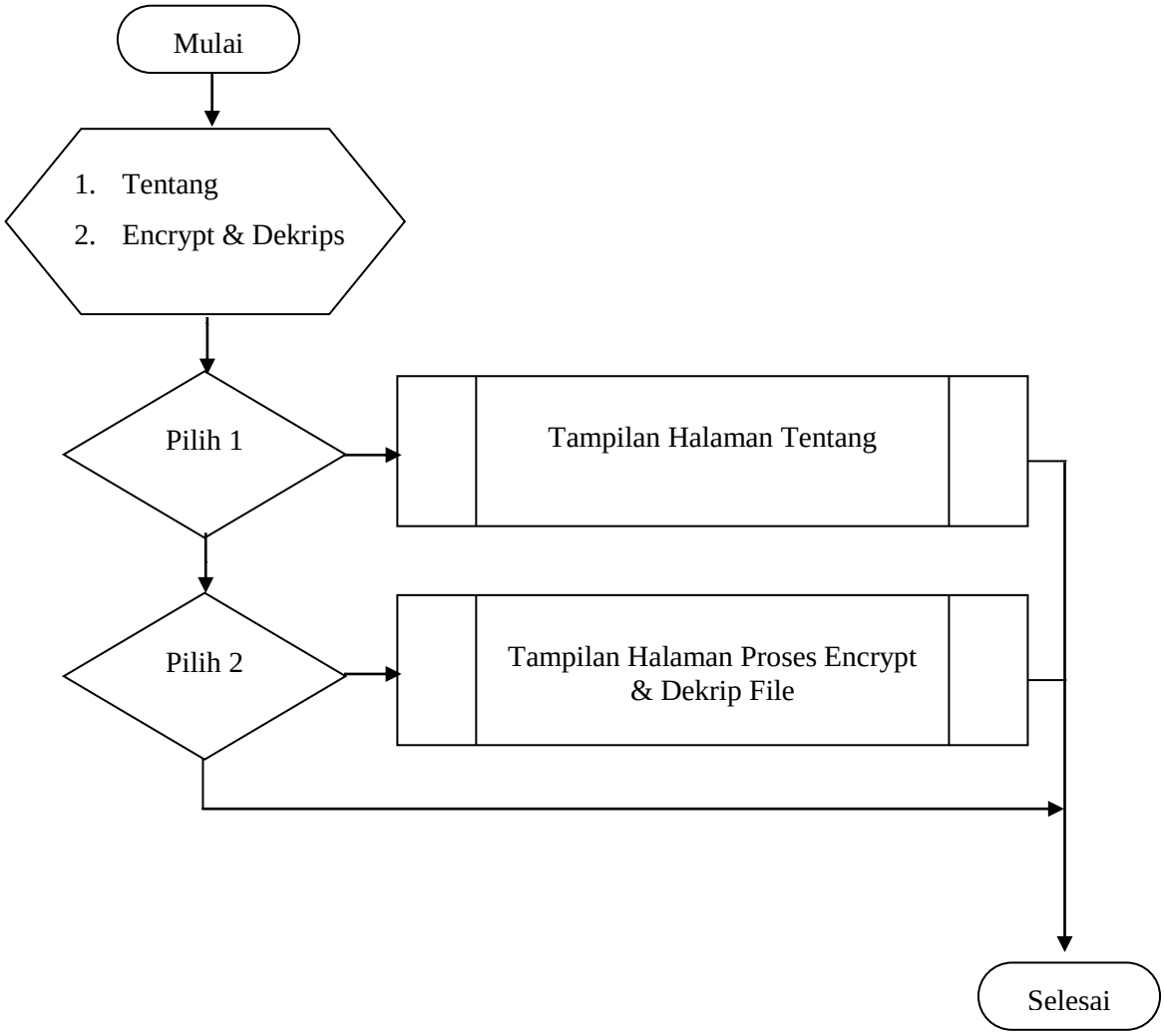
Persentase Validasi Keamanan Data = (950 / 1000)×100% = 95%

Jadi, persentase validasi keamanan data adalah 95%.

2. *WhiteBox*

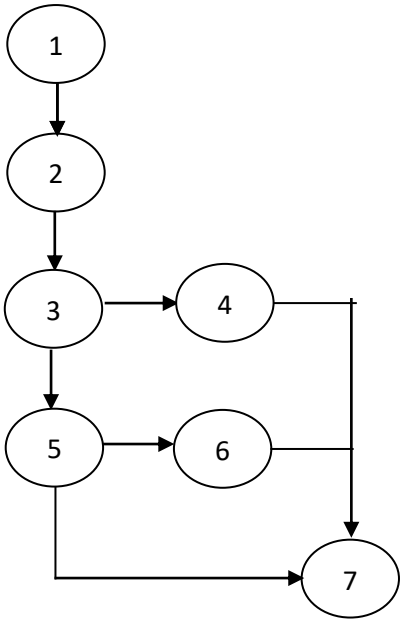
Pengujian aplikasi dilakukan dengan cara pengujian *WhiteBox*. *WhiteBox* adalah metode pengujian perangkat lunak yang menguji struktur internal atau cara kerja aplikasi, yang bertentangan dengan fungsinya. Dalam pengujian kotak putih, perspektif internal sistem digunakan untuk merancang kasus uji.

a. *Flowchart* dan *Flowgraph* Aktivitas



Gambar 4. 9. Flowchart Aktivitas

Dari *Flowchart* yang digunakan *untuk* pengujian perangkat lunak, maka ditentukan *Flowgraph* sebagai berikut:



Gambar 4.10. *Flowgraph* Aktivitas

Dari *Flowgraph* aktivitas diatas dapat dilakukan proses perhitungan sebagai berikut:

1. Menghitung *Cyclomatic Complexcity* V(G) dari *Egde* dan *Node*:

   Dengan rumus : V(G) = E – N + 2

   E (*edge*) = 8

   N (*Node*) = 7

   P (Predikat *Node*) = 2

   Penyelesaian : V(G) = E – N + 2

   = 8 – 7 + 2

   = 3

   Predikat (P) = P + 1

= 2 + 1

= 3

2. Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 3

3. *Independent path* pada *Flowgraph* diatas adalah:

   Path 1 = 1 – 2 – 3 – 4 – 7

   Path 2 = 1 – 2 – 3 – 5 – 6 – 7

   Path 3 = 1 – 2 – 3 – 5 – 7

4. Grafik Matriks Aktivitas

Tabel 4.10. Grafik Matriks Aktivitas

	1	2	3	4	5	6	7	E – 1
1		1						1 – 1 = 0
2			1					1 – 1 = 0
3				1	1			2 – 1 = 1
4							1	1 – 1 = 0
5						1	1	2 – 1 = 1
6							1	1 – 1 = 0
7								0
SUM (E+1)								2 + 1 = 3

BAB V

PENUTUP

A. Kesimpulan

1. Setelah dilakukan uji coba enkripsi sebanyak 10 kali dikatakan aplikasi berjalan karena menghasilkan chipertext.
2. Setelah dilakukan uji coba sebanyak dekripsi file (plaintext) sebanyak 10 kali didapatkan hasil yang sama dengan file teks sebelumnya.
3. Hasil dari enkripsi dan dekripsi file tersimpan didalam media penyimpanan internal handphone.

B. Saran

Saran untuk pengembangan sistem lebih lanjut sebagai bahan masukan agar perancangan aplikasi ini dapat berkembang dan bermanfaat sesuai dengan perkembangan teknologi, dengan menambah fitur - fitur yang dapat membuat aplikasi jadi lebih memudahkan *user* dan menggunakan algoritma lainnya untuk perbandingan hasil kriptografi.

DAFTAR PUSTAKA

Andika, D. (2018). Pengertian dan sejarah kriptografi. Diperoleh dari: https://www.it-jurnal.com/pengertian-dan-sejarah-kriptografi/.(Diakses 1Desember 2022).

Arifianto, R. 2020. Materi Pembahasan Flowchart. From https://rahmatarifianto. wordpress.com/2014/11/20/pengertian-Flowchart-dan-jenis-jenisnya.html. (01 Januari 2023).

Dharma.A.K. (2016). Kolaborasi dahsyat android dengan php dan msql.

Eko Juliansyah. (2017). Implementasi algoritma kriptografi rc6 menggunakan data teks. Medan.

Kurniawan. B. (2020). Penegertian, sejarah, fungsi, karakter dan contoh ASCII Diperoleh dari ilmuelektro.id/ascii-adalah/. (Diakses 1 Desembar 2022).

Laurentinus. (2017). Implementasi dan kompresi SMS menggunakan algoritma RC6 dan algoritma huffman berbasis android.

Medana A.P. (2017). Penerapan Algoritma Huffman Dan Shannon - Fano Dalam Pemampatan File Teks.

Mohammad. K.A.H. (2021). Rancang bangun aplikasi enkripsi-dekripsi sms pada android dengan metode RC6.

Nur W.R. (2015). Semarang Charity Map, Penyajian Peta Donasi Sosial Kota Semarang Berbasis Blogger Javascript.

Safaat. N. H. (2012). Pemrograman Aplikasi Mobile Smartphone Dan Tablet Pc Berbasis Android. Pekan Baru,Riau.

Santi. R. (2019). Analisa dan pemodelan framework cordova berbasisi android. Sora, N. 2018. Pengertian UML dan Jenis - jenisnya. From http://www. pengertianku.net/2018/09/pengertian-uml-dan-jenis-jenisnya-serta-contoh-diagramnya.html. (01 Januari 2023).