PERBANDINGAN KINERJA ALGORITMA RSA DAN ALGORITMA RC4 DALAM MENGENKRIPSI DAN DEKRIPSI DATA FILE BERSASIS ANDROID

OLEH

MOH. REZALDY D. KASILI

T3120041

SKRIPSI

Untuk memenuhi salah satu syarat ujian

guna memperoleh gelara sarjana

PROGRAM SARJANA

TEKNIK INFORMATIKA

UNIVERSITAS ICHSAN GORONTALO

GORONTALO

2024

PERSETUJUAN SKRIPSI

PERBANDINGAN KINERJA ALGORITMA RSA DAN ALGORITMA RC4 DALAM MENGENKRIPSI DA DEKRIPSI DATA FILE BERBASIS ANDROID

Oleh

MOH. REZALDY D. KASILI

T3120041

SKRIPSI

Untuk memenuhisalah satu syarat ujian

Guna memperoleh gelar Sarjana

Program Studi Teknik Informatika

Ini telah disetujui oleh Tim Pembimbing

Gorontalo, 2024

Pembimbing Utama	Pembimbing Pendamping
Hasruti R. Dalai, M.Kom	**Warid Yunus, M.Kom**
NIDN : 0918038803	**NIDN : 0914059001**

PENGESAHAN SKRIPSI

PERBANDINGAN KINERJA ALGORITMA RSA DAN ALGORITMA RC4 DALAM MENGENKRIPSI DA DEKRIPSI DATA FILE BERBASIS ANDROID

Oleh

Moh. Rezaldy D. Kasili
T3120041

Diperiksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo

1. Ketua Penguji
   Sudirman Melangi, M.Kom .....
2. Anggota
   Sudirman S. Panna, M.Kom .....
3. Anggota
   Maryam Hasan, M.Kom .....
4. Anggota
   Hastuti Dalai, M.Kom .....
5. Anggota
   Warid Yunus, M.Kom .....

Mengetahui

Dekan Fakultas Ilmu Komputer	Ketua Program Studi
Irvan Abraham Salihi, M.Kom	**Sudirman S. Panna,M.Kom**
NIDN. 0918077302	**NIDN. 0924038205**

SURAT PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Univesitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan peneliian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Penyataan ini saya buat dengan sesungguhnya dan apabila kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, 6 / Juni / 2024
Yang Membuat Pernyataan

Moh. Rezaldy D. Kasili

ABSTRACT

MOH. REZALDY D. KASILI. T3120041. PERFORMANCE COMPARISON OF RSA AND RC4 ALGORITHM IN ENCRYPTING AND DECRYPTING ON ANDROID-BASED DATA FILES

This research compares the performance of RSA and RC4 algorithms in encrypting and decrypting data files on Android-based systems. The main objective is to evaluate the speed and security of the two algorithms to determine their suitability for various applications. The results show that RC4 tends to be faster in encryption and decryption than RSA. The difference is due to the algorithm complexity and the encryption method used. However, RSA is generally considered more secure than RC4, as RC4 has been subject to serious cryptanalysis attacks. RSA uses public keys and private keys, while RC4 uses symmetric keys. RSA is often used for applications requiring a high security level such as securing communications over a network. Although fast, RC4 has been widely abandoned in many applications due to its security weaknesses. Thus, the choice between RSA and RC4 depends on the specific needs of the user. The major consideration is about security and performance.

Keywords*: algorithm, RSA, RC4, data encryption, data decryption, Android*

ABSTRAK

MOH. REZALDY D. KASILI. T3120041. PERBANDINGAN KINERJA ALGORITMA RSA DAN ALGORITMA RC4 DALAM MENGENKRIPSI DAN DEKRIPSI DATA FILE BERBASIS ANDROID

Penelitian ini membandingkan kinerja algoritma RSA dan RC4 dalam mengenkripsi dan mendekripsi file data pada sistem berbasis Android. Tujuan utama adalah untuk mengevaluasi kecepatan dan keamanan kedua algoritma tersebut guna menentukan kesesuaiannya untuk berbagai aplikasi. Hasil penelitian menunjukkan bahwa RC4 cenderung lebih cepat dalam proses enkripsi dan dekripsi dibandingkan RSA. Perbedaan ini disebabkan oleh kompleksitas algoritma dan metode enkripsi yang digunakan. Namun, RSA umumnya dianggap lebih aman daripada RC4, karena RC4 telah terkena serangan kriptanalisis yang serius. RSA menggunakan kunci publik dan kunci pribadi, sedangkan RC4 menggunakan kunci simetris. RSA sering digunakan untuk aplikasi yang membutuhkan tingkat keamanan tinggi, seperti dalam pengamanan komunikasi melalui jaringan. Meskipun cepat, RC4 telah banyak ditinggalkan dalam banyak aplikasi karena kelemahan keamanannya. Dengan demikian, pemilihan antara RSA dan RC4 tergantung pada kebutuhan spesifik pengguna, dengan pertimbangan utama antara keamanan dan kinerja.

Kata kunci: algoritma, RSA, RC4, enkripsi data, dekripsi data, Android

KATA PENGANTAR

Alhamdulillah, penulis dapat menyelesaikan skripsi dengan judul ***"PERBANDINGAN KINERJA ALGORITMA RSA DAN RC4 DALAM MENGENKRIPSI DAN DEKRIPSI DATA FILE BERBASIS ANDROID",*** untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo.

Penulis menyadari sepenuhnya bahwa skripsi ini, tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu, dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan setinggi-tingginya kepada:

1. Ibu Dr. Yuriko Abdussamad, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Irvan Abrahman Salihi S.Kom.,M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan I Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
5. Ibu Irma Surya Kumala Idris, M.Kom, selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
6. Bapak Sudirman S Panna, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Ibu Hastuti R. Dalai S.Kom, M.Kom, Sebagai Pembimbing Utama dalam penelitian ini yang telah membimbing penulis selama penelitian;
8. Bapak Warid Yunus S.Kom.,M.Kom, Sebagai Pembimbing Pendamping dalam penelitian ini yang telah membimbing penulis selama Penelitian;

9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;
10. Kedua Orang Tua saya yang tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
11. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
12. Kepada semua pihak yang ikut membantu dalam penyelesaian Usulan Penelitian ini yang tak sempat penulis sebutkan satu-persatu.

Semoga Allah, SWT melimpahkan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kata sempurna dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif. Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin.

Gorontalo, … / … / 2024

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dengan perkembangan teknologi yang sangat pesat, kita telah mengalami perkembangan informasi apa pun dalam bidang pendidikan maupun bidang lainnya. Contoh perkembangan tersebut adalah internet, yang saat ini memungkinkan banyak orang bertukar data secara bebas melalui jaringan. Berkat kenyamanannya, internet telah menjadi salah satu media masa paling populer di dunia[1]. Keamanan data adalah salah satu hal yang patut mendapat perhatian lebih, terutama bagi pengguna yang selalu melakukan proses pertukaran data rahasia, sehingga perlu dilakukan pengamanan data agar beberapa pihak yang tidak memiliki kewenangan tidak dapat membuka informasi yang di kirimkan[2].

Android adalah sistem operasi yang cukup rentan terhadap serangan *malwere,* android sering terhubung dengan jaringan nirkabel publik atau hot-spot, yang sering kali tidak aman. Penyerangan dapat menggunakan jaringan ini untuk mencuri data file atau melakukan serangan yang membahayakan keamanan data file. Oleh karena itu dibutuhkan algoritma kriptografi dalam pengamanan data file berbasis android. Android adalah platform perangkat sumber terbuka, karena itu android dirancang untuk bekerja di semua jenis perangkat[3].

Kriptografi adalah ilmu sekaligus seni keamanan informasi. Kriptografi ada dua proses utama yaitu proses penyandian pesan yang dapat dibaca (*plaintext*) menjadi pesan yang tidak dapat dibaca atau telah disandikan (*chipertext*) disebut dengan proses enkripsi, dan proses pengembalian *chipertext* ke *plaintext* disebut dengan dekripsi[1]. Ada beberapa macam algoritma kriptografi dalam pengamanan data seperti algoritma Rivest Shamir Adler (RSA) dan algoritma Rivest *Cod*e 4 (RC4).

Algoritma Rivest Shamir Adler (RSA) adalah algoritma enkripsi kunci publik. RSA adalah algoritma pertama yang diketahui paling cocok

untuk menandai dan enkripsi, dan salah satu penemuan besar pertama dalam kriptografi kunci publik[4]. Algoritma ini memiliki keamanan yang terletak pada kompleksitas komputasi logaritma diskrit[1].

Algoritma kriptografi Rivest Code 4 (RC4) adalah salah satu algoritma kunci simetris yang dibuat oleh RSA *Dara Security Inc* (RSADSI) dalam bentuk *stream chiper*. Algoritma ini ditemukan pada tahun 1987 oleh Ronald dan Rivest dan menjadi simbol keamanan RSA. RC4 menggunakan kunci dengan panjang antara 1 dan 256 byte yang untuk menginisialisasi tabel sepanjang 256 byte[5].

Berdasarkan penelitian (Ulfa Indriani) dengan judul **"Penerapan Algoritma RSA dalam Keamanan File Ms Word"** Algoritma RSA dapat membantu meningkatkan data dengan menggunakan perhitungan pemfaktoran bilangan prima, dapat meningkatkan sistem keamanan data, sehingga data yang tersimpan akan lebih aktif dalam proses pembuatan aplikasi atau web baru. dapat diketahui bahwa langkah-langkah yang perlu dilakukan adalah baik untuk menjamin keamanan file (isi file). hal ini diperlukan untuk mempelajari sistem yang ada, kemudian merancang sistem dan melakukan perhitungan dengan mempertimbangkan kriteria pemilihan[4].

Sedangkan hasil dari penelitian (Fakhril Muhariza) dengan judul **"Implementasi Data Dokumen Menggunakan Kriptografi Dengan Algoritma Rivest Code 4 (RC4) Berbasis Web"** di dapat hasil enkripsi rata-rata ukuran 83481 byte dengan kecepatan waktu rata-rata 3150 milidetik dan hasil proses dekripsi didapat hasil rata-rata 83481 dengan kecepatan waktu 4005 milidetik[6].

Sesuai dengan uraian latar belakang di atas yang telah peneliti deskripsikan maka peneliti memberikan solusi untuk melakukan penelitian dengan judul, **"Perbandingan Kinerja Algoritma RSA dan Algoritma RC4 dalam mengenkripsi dan dekripsi data file berbasis android"** dengan harapan agar setiap kita dapat menjaga data agar tidak ada pihak ke tiga membobol atau memodifikasi data kita.

1.2 Identifikasi Masalah

Dari uraian latar belakang di atas maka yang menjadi masalah adalah setiap sistem operasi khususnya android memiliki kerentanan yang bisa dimanfaatkan oleh peretas. Android juga memiliki kerentanan yang meningkatkan risiko keamanan data pengguna. Penggunaan konektifitas maupun data yang tidak diatur dengan baik dapat mengakibatkan data pribadi pengguna disalahgunakan atau dibagikan tanpa izin.

1.3 Rumusan Masalah

Bagaimana kinerja algoritma RSA dan algoritma RC4 dalam megenkripsi dan dekripsi data file berbasis android?

1.4 Tujuan Penelitian

Untuk mengetahui hasil kinerja algoritma RSA dan algoritma RC4 dalam mengenkripsi dan dekripsi data file berbasis android.

1.5 Manfaat Penelitian

Penelitian ini dapat memberikan wawasan yang berharga dalam mengembangkan aplikasi keamanan untuk mengenkripsi dan dekripsi data file pada perangkat android. Data hasil penelitian ini dapat digunakan untuk memperbaiki dan meningkatkan keamanan aplikasi yang ada atau merancang aplikasi baru yang lebih aman, baik secara teoritis maupun praktis, diantaranya :

1.5.1 Manfaat Teoritis

Memberikan masukan bagi perkembangan ilmu tentang teknologi, khususnya dalam bidang ilmu komputer, yaitu berupa penerapan tentang kriptografi untuk mengenkripsi dan dekripsi.

1.5.2 Manfaat Praktis

Sumbangan pemikiran, karya, bahan pertimbangan, atau solusi bagi pihak yang ingin mengamankan data agar tidak terjadinya kebocoran data atau dokumen penting yang seharusanya kita.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Tinjauan Studi sangat berguna bagi peneliti dalam memberikan pedoman serta pegangan peneliti yang nantinya dengan adanya penelitian sebelumnya akan mempermudah peneliti dalam melakukan penelitian yang sesuai topik pembahasaan. Berikut beberapa jurnal yang relevan dengan penelitian ini yaitu:

Tabel 2. 1 Tabel Tinjauan Studi

NO	PENELITI	JUDUL	TAHUN	METODE	HASIL
1	Fakhril Muhariza, Noni Juliasari	Implementasi Pengamanan Dokumen Menggunakan Kriptografi Dengan Algoritma Rivest Code 4 (RC4) Berbasis Web	2023	Algoritma RC4	Aplikasi ini berhasil mengenkripsi dan dekripsi suatu file yang berformat (*.doc,*.xls,*.txt,*.pdf, dan *.ppt) dengan menggunakan algoritma RC4 yang sudah dimodifikasi
2	Rinaldi Maulana, R. Mahdalena Simanjorang	Implementasi Kriptografi Untuk Pengamanan Data Pribadi Siswa SMA Swasta Jaya Krama Beringin Dengan Algoritma RC4	2021	Algoritma RC4	Sistem mampu melakukan enkripsi pada data yang diimput sehingga dapat digunakan untuk meningkatkan keamanan data

No	PENELITI	JUDUL	TAHUN	METODE	HASIL
3	Ulfa Indriani, ommi alfina, Nita Syahputri	Penerapan Algoritma RSA dalam Keamanan File Ms Word	2022	Algoritma RSA	Algoritma RSA dapat Membantu meningkatkan data menggunakan perhitungan bilangan prima dapat meningkatkan sistem keamanan data, sehingga data yang tersimpan akan semakin terjaga

2.2 Tinjauan Pustaka

2.2.1 Perbandingan

Perbandingan adalah proses membandingkan atau mengevaluasi dua atau lebih hal untuk menentukan perbedaan, kesamaan, atau hubungan diantara mereka. Ini melibatkan memeriksa karakteristik, sifat, atau kualitas dari benda atau konsep yang dibandingkan. Perbandingan sering digunakan untuk memahami properti relatif dari objek dan membuat penilaian atau keputusan berdasarkan perbedaan atau kesamaan yang di temukan.

Dalam kamus besar bahasa indonesia perbandingan berasal dari kata banding atau persamaan, kemudian mencocokan antar dua hal untuk mengetahui perbandingannya. Perbandingan diartikan sebagai selisih persamaan (Bambang Mahriyanto)[7].

Menurut (Sjachran Basah), perbandingan merupakan metode evaluasi atau penelitian dengan membuat perbandingan antara dua atau lebih objek penelitian dalam rangka menambah dan memperdalam pengetahuan tentang objek yang diteliti[7].

2.2.2 Kinerja Algoritma

Kinerja algoritma khususnya algoritma kriptografi adalah serangkaian aturan matematis yang digunakan untuk mengamankan komunikasi dan data. Kinerjanya dapat dijelaskan melalui beberapa aspek yaitu [5]:

1. Keamanan
    - Algoritma kriptografi harus mamppu melindungi informasi dari akses yang tidak sah.
    - Keamanan algoritma tergantung pada kompleksitas matematika yang melibatkannya dan ketahanannya terhadap berbagai jenis serangan.
2. Kecepatan
    - Kinerja algoritma kriptografi diukur dalam kecepatan pemrosesan data.
    - Algoritma yang cepat dapat memberikan keamanan tanpa mengorbankan efisiensi pengolahan data.
3. Kunci
    - Algoritma kriptografi menggunakan kunci untuk mengamankan dan mendekripsi informasi.
    - Keamanan bergantung pada kompleksitas kunci dan kemampuannya untuk tetap rahasia.
4. Efisiensi
    - Algoritma kriptografi harus efisien dalam hal penggunaan sumber daya seperti CPU dan memori.
    - Efisisensi penting dalam implementasi algoritma pada perangkat dengan keterbasan daya.
5. Tahan Terhadap Serangan
    - Kinerja Algoritma kriptografi diukur melalui ketahanannya terhadap berbagai serangan, seperti serangan *brute force*, serangan terhadap struktur matematis, atau serangan samping.

6. Skalabilitas

- Algoritma kriptografi harus dapat diimplementasikan dengan baik pada berbagai paltform dan perangkat.
- Skalabilitas memastikan bahwa algoritma dapat digunakan dalam berbagai konteks tanpa mengorbankan keamanan atau kinerja.

7. Ketahanan Terhadap Perkembangan Teknologi

- Kinerja algoritma kriptografi dinilai berdasarkan kemampuannya bertahan terhadap perkembangan teknologi, termasuk kemajuan dalam komputasi kuantum atau metode serangan baru.

2.2.3 Kriptografi

Kriptografi berasal dari bahasa yunani yaitu "*The Art Of Secreet Writing*" yang berarti seni menulis dengan rahasia, yang tujuannya adalah untuk membuat pesan menjadi tidak ada arti[6]. Kriptografi memiliki dua konsep dasar yaitu enkripsi dan dekripsi, enkripsi adalah proses melindungi pesan asli (*plaintext*) dengan mengubah bentuk data menjadi kode menggunakan kunci yang telah ditentukan oleh pengirim dan penerima pesan menjadi pesan rahasia (*chipertext*) sehingga data atau informasi yang dikirimkan tidak dapat dibaca sehingga dapat dirahasiakan. Sedangkan dekripsi adalah proses pembalikan pesan dalam bentuk (*chipertext*) menjadi pesan aslinya (*plaintext*)[6]. Kriptografi dapat dibagi menjadi dua kategori utama berdasarkan cara penggunaan kunci :

1. Kriptografi Simetris

Kriptografi simetris menggunakan kunci yang sama untuk enkripsi dan dekripsi. Contoh algoritma simetris termasuk DES, AES, dan RC4.

2. Kriptografi Asimetris (Kunci publik)

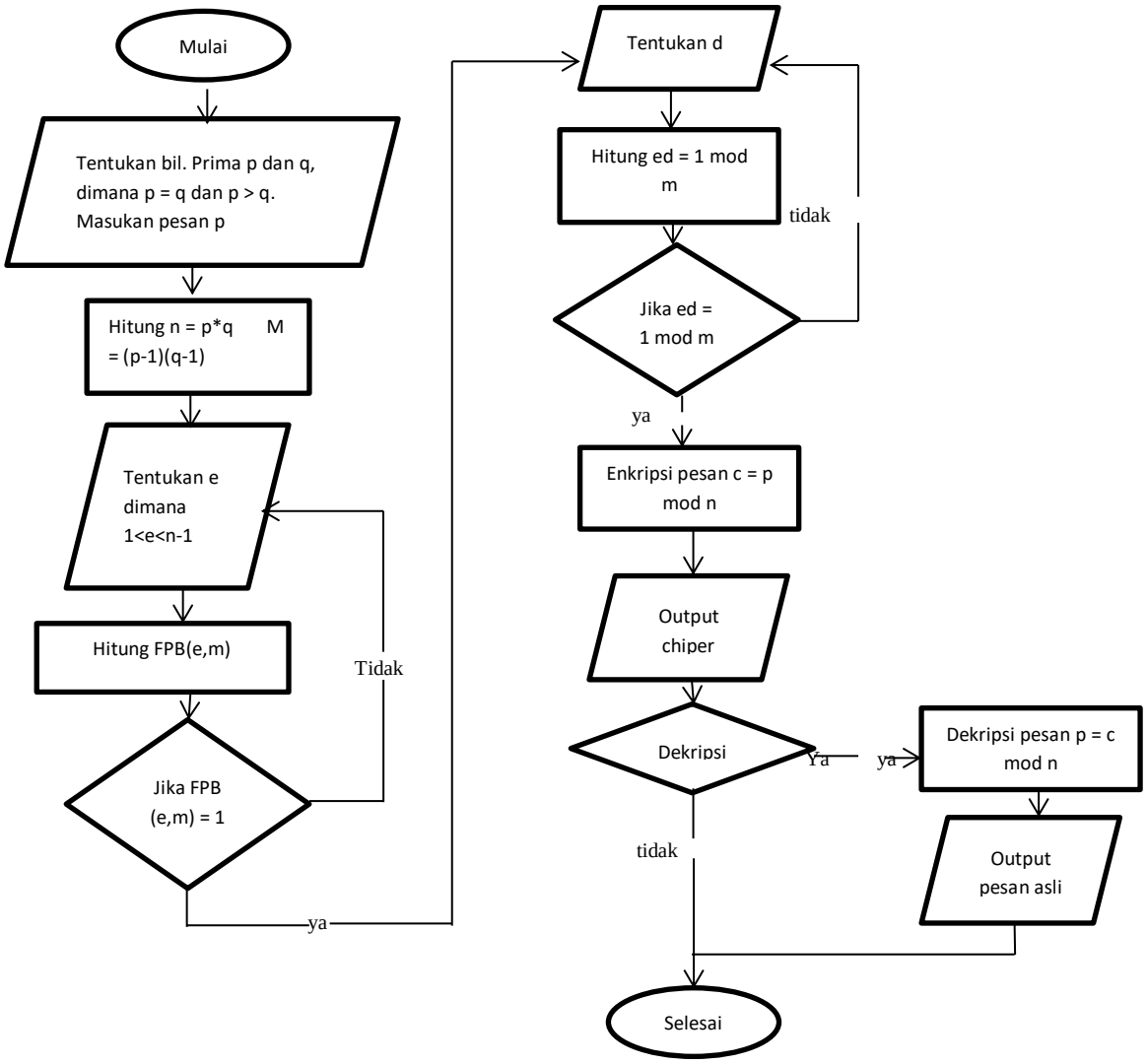
Kriptografi Asimetris menggunakan sepasang kunci, yaitu kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Contoh algoritma asimetris termasuk RSA, ECC, DSA.

2.2.4 Algortima RSA

Algoritma RSA pertama kali dikembangkan oleh Ron Rivest, Adi Shamir, dan Led Adlenan dari Massachusetts Institute of Thechnology pada tahun 1978. Nama RSA sendiri diambil dari tiga nama peneliti, yaitu: (R)ivest, (S)hamir, dan (A)dleman. RSA adalah algoritma kunci publik yang memiliki dua kunci, kunci publik (*public key*) dan kunci pribadi (*private key*). RSA dibagi menjadi tiga proses yaitu pembuatan kunci, enkripsi, dan dekripsi. Proses enkripsi dan dekripsi dilakukan dengan konsep bilangan prima dan modulo aritmatika. Kunci enkripsi tidak dirahasiakan dan diberikan kepada publik sedangkan kunci dekripsi bersifat rahasia, untuk menemukan kunci dekripsi dilakukan dengan menguraikan bilangan bulat menjadi pengganda keunggulan[1]. Besaran-besaran yang digunakan pada algoritma RSA antara lain:

1. p dan q bilangan prima (rahasia)
2. $n = pq$ (tidak rahasia)
3. $p(n) = (p\text{-}1) (p\text{-}1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)
5. d (kunci dekripsi) (rahasia)
6. m (plaintext) (rahasia)
7. c (chipertext) (tidak rahasia)

Berikut adalah *Flowchart* proses enkripsi dan dekripsi RSA :



Gambar 2. 1 *Flowchart* Enkripsi dan Dekripsi Algoritma RSA

Contoh penerapan Algoritma RSA [8]:

Proses pembangkit kunci, yaitu :

1. Pilih bilangan prima, misalnya p = 3 dan q = 641
2. Hitung n = p x q = 3 x 641 = 1923
3. Hitung n = (p - 1) (q – 1) = 2 x 640 = 1280
4. Pilih e yang relatif prima terhadap m, ged (e,n) = 1 e = 427 => gcd (427,1280) = 1 nilai e yang diambil adalah 427.

Bukti : (427,1280)

1280 mod 427 = 426

426 mod 427 = 1

1 mod 1 = 0

5. Sehingga cari nilai de = 1 (mod 1280) dan d < 1280

Mencari nilai = d x 427 = 1 mod 1280

d x 427 mod 1280 = 1

d = 427

Bukti : 427 x 3 mod 1280 = 1

Sehingga di dapatkan : Public key : e(427), n(1923) dan Private key : d(3), n(1923)

Contoh enkripsi RSA :

Public key dan *privat key* sudah diketahui, maka selanjutnya dilakukan enkripsi pada *plaintext*(p)= 10000000 = ASCII : 49 48 48 48 48 48 48 48 kemudian pecah menjadi sebuah blok yang berukuran 3 digit yaitu :

X1 = 494

X2 = 848

X3 = 484

X4 = 848

X5 = 484

X6 = 800 (diberi penambahan 00)

Maka akan dilakukan perhitungan dengan rumus Ci = Pie mod n untuk proses enkripsi yaitu :

C1 = 494427 mod 1923 = 533

C2 = 848427 mod 1923 = 209

C3 = 484427 mod 1923 = 874

C4 = 848427 mod 1923 = 209

C5 = 484427 mod 1923 = 874

C6 = 800427 mod 1923 = 1202

Hingga hasil yang didapatkan ciphertext (c) = 533.209.874.209.874.1202 dalam karakter ASCII

Contoh proses dekripsi :

Ciphertext (c) = 533.209.874.209.874.1202 dalam karakter ASCII. Kemudian, dekripsikan kembali dengan menggunakan rumus Pi = Cid mod n :

P1 = 5333 mod 1923 = 494

P2 = 2093 mod 1923 = 848

P3 = 8743 mod 1923 = 484

P4 = 2093 mod1923 = 848

P5 = 8743 mod 1923 = 484

P6 = 12023 mod 1923 = 800

Maka, hasil gabungan P1 sampai P6 = 533.209.874.209.874.1202, adalah 10000000

Contoh kasus penerapan algoritma RSA adalah dalam pengamanan komunikasi melalui internet, terutama dalam protokol HTTPS. Pada awalanya, server yang ingin membangun koneksi aman dengan klien menghasilkan sepasang kunci, yaitu kunci publik dan kunci privat RSA. Kunci publik digunakan untuk enkripsi data, sementara kunci privat digunakan untuk dekripsi data. Ketika klien terhubung ke server, server mengirim sertifikat digitalnya yang berisi kunci publiknya kepada klien. Klien menggunakan kunci publik server untuk mengenkripsi kunci sesi yang akan digunakan untuk enkripsi data selama sesi tersebut.

RSA (Rivest Shamir Adleman) adalah algoritma kriptografi yang umumnya efektif, tetapi kinerjanya dapat dipengaruhi oleh sejumlah faktor di lingkungan android. Perfoma RSA tergantung pada panjang kunci, prosesor perangkat, dan implementasi kripto yang digunakan. Menggunakan kunci RSA dengan panjang yang lebih besar dapat meningkatkan keamanan, tetapi juga memerlukan lebih banyak daya komputasi. Dengan menggunakan prosesor yang lebih kuat dapat mengeksekusi operasi RSA lebih cepat.

2.2.5 Algoritma RC4

RC4 adalah jenis *stream chiper* yang memproses unit pengukuran atau unit data dalam satu waktu. Dengan demikian, enkripsi dan dekripsi dapat di implementasikan dengan panjang yang variabel. Algoritma ini tidak perlu menunggu sejumlah input data sebelum memproses atau menambahkan byte tambahan untuk enkripsi[9].

RC4 menggunakan kunci dengan panjang 1 sampai 256 byte, yang digunakan untuk menginisialisasi tabel 256 byte. Tabel ini digunakan untuk pembuatan *pseudo random* yang menggunakan teks biasa XOR untuk menghasilkan teks sandi[6].

Beberapa kerentanan telah ditemukan dalam RC4 yang membuat rentan terhadap serangan kunci dan mengurangi tingkat keamanannya. Pada tahun 2015, penelitian menunjukan serangan baru yang di sebut "*Bar Mitzvah*" yang memperlibatkan adanya kerentanan dalam implementasi RC4, tapi RC4 cepat dalam memproses Enkripsi dan dekripsi.

Penerapan RC4 secara manual melibatkan langkah-langkah untuk menghasilkan *keystream* dan melakukan XOR dengan teks terbuka (*plaintext*). Berikut adalah contoh langkah-langkah secara manual:

1. Inisialisasi S-box (*State Box*)
   - Buat array S-box berisi nilai 0 hingga 255.
   - Misalnya S = [0,1,2,….,255].
2. Inisialisasi KSA (*Key Scheduling Algorithm*)

- Gunakan kunci (*Key*) sebagai input.
- Permutasikan nilai dalam S-box berdasarkan kunci.
- Langkah ini melibatkan pertukaran elemen-elemen S-box sesuai dengan Algoritma RC4.

3. Inisialisasi PRGA (*Pseudo-Random Generation Algorithm*):
   - Gunakan S-box yang telah diinisialisasi.
   - Iterasi PRGA sebanyak byte pada teks terbuka (*plaintext*).
   - Dalam setiap iterasi, pertukaran elemen S-box dan hasilkan byte dari *keystream.*
4. XOR dengan *Plaintext*
   - Dapatkan *keystream* dari PRGA.
   - Lakukan operasi XOR antara byte keystream dan byte *plaintext* untuk menghasilkan byte *chipertext*

Salah satu contoh kasus penerapan algoritma RC4 adalah dalam protokol keamanan jaringan seperti SSL/TLS. RC4 pernah digunakan sebagai salah satu algoritma enkripsi dalam protokol SSL/TLS untuk mengamankan komunikasi antara server dan klien, meskipun sekarang tidak lagi direkomendasikan karena beberapa kerentanan keamanan yang ditemukan dalam implementasinya. Meskipun beigtu, RC4 masih digunakan dalam beberapa konteks lain seperti protokol WEB dalam jaringan nirkabel.

RC4 rentan terhadap serangan kunci lemah jika kunci lemah jika kunci yang digunakan tidak dipilih dengan baik. Ini dapat memungkinkan serangan pencarian kecil (*smaller key space attacks*) yang memungkinkan penyerang memecahkan pesan yang dienkripsi dengan menggunakan kuci yang lemah.

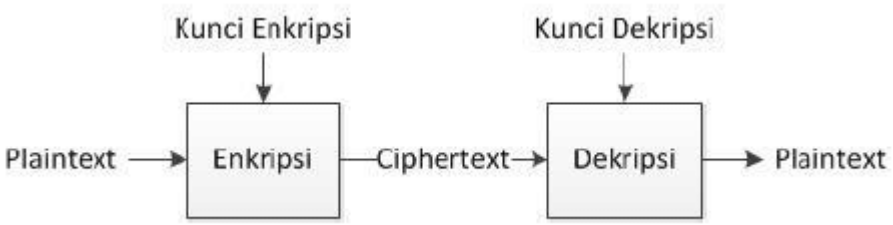
2.2.6 Enkripsi

Enkripsi adalah proses yang dilakukan untuk menyandikan teks biasa menjadi teks sandi sehingga pesan tidak dapat dibaca oleh orang yang tidak berwenang[9]. Enkripsi sangat penting dalam kriptografi yang menjamin keamanan data yang dikirimkan dan dirahasiakan. Pesan asli disebut *plaintext* yang diubah menjadi kode yang tidak bisa dipahami atau disebut dengan *chipertext*[10]. Terdapat tiga kategori enkripsi yaitu:

1. Kunci enkripsi rahasia, dalam hal ini adalah kunci yang digunakan untuk melakukan deposit, sekaligus untuk mendeskripsikan informasi.
2. Kunci enkripsi publik dalam hal ini digunakan dua kunci yang pertama digunakan untuk proses mengenkripsi dan kedua digunakan untuk proses dekripsi.
3. Fungsi satu arah diaman informasi dienkripsi untuk membuat "*signature*" dari informasi asli yang dapat digunakan untuk tujuan otentikasi.

2.2.7 Dekripsi

Dekripsi adalah proses mengekstraksi tek biasa dari teks terenkripsi[9]. Dekripsi merupakan kebalikan dari enkripsi, pesan yang telah terenkripsi dikembalikan ke bentuk aslinya (*plaintext*) di sebut dekripsi pesan[10]. Berikut adalah gambar proses enkripsi dan dekripsi :



Gambar 2. 2 Proses enkripsi dan dekripsi

Berdasarkan metodenya, dekripsi dapat dibagi menjadi beberapa kategori dekripsi umum :

1. Simetris (*Private Key*)

   Menggunakan kunci yang sama untuk enkripsi dan dekripsi contohnya : AES (*Advanced Encryption Standard*), DES *(Data Encryption Standard*), dan RC4.

2. Asimetris (*Public Key*)

   Menggunakan sepasang kunci, kunci publik untuk enkripsi dan kunci pribadi untuk sekripsi. Contoh algoritma asimetris yaitu : RSA (Rivest Shamir Adleman), ECC (*Eliptic Curve Cryptogrphy*).

3. Hash-Based

   Menggunakan Fungsi hash untuk menghasilkan nilai yang unik sebagai kunci dekripsi contohnya : HMAC (*Hash-Based message Authentication code*).

4. Steganografi

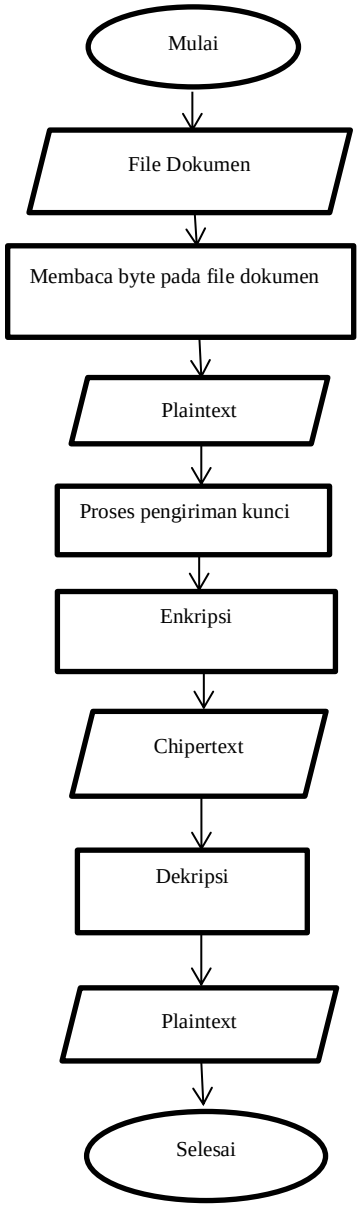
   Menyembunyikan data terenkripsi didalam data lainnya untuk menghindari deteksi contohnya menyembunyikan pesan terenkripsi di dalam ganbar.

2.2.8 Data File

File adalah kumpulan data yang disimpan dalam sebuah penyimpanan elektronik. File bisa berupa teks, gambar, video, atau jenis data lainnya yang dapat diakses dan digunkan oleh komputeratau perangkat elektronik lainnya. File biasanya memiliki ekstensi yang menunjukan jenis format datanya, seperti .txt untuk file teks atau .jpg untuk file gambar.

Keamanan data yang terkandung dalam file tersebut diperlukan selama proses penyimpanan maupun pengiriman data rahasia, hal ini dikarenakan kemungkinan terjadinya pencurian data file selama proses penyimpanan dan peniriman file. Kemudian file yang berisis data yang dilindungi harus dienkripsi sedemikian rupa sehingga struktur dan data dalam file tersebut tidak dapat dibaca atau dikenali tanpa proses dekripsi[11].

Berikut adalah proses enkripsi dan dekripsi data file[12] :



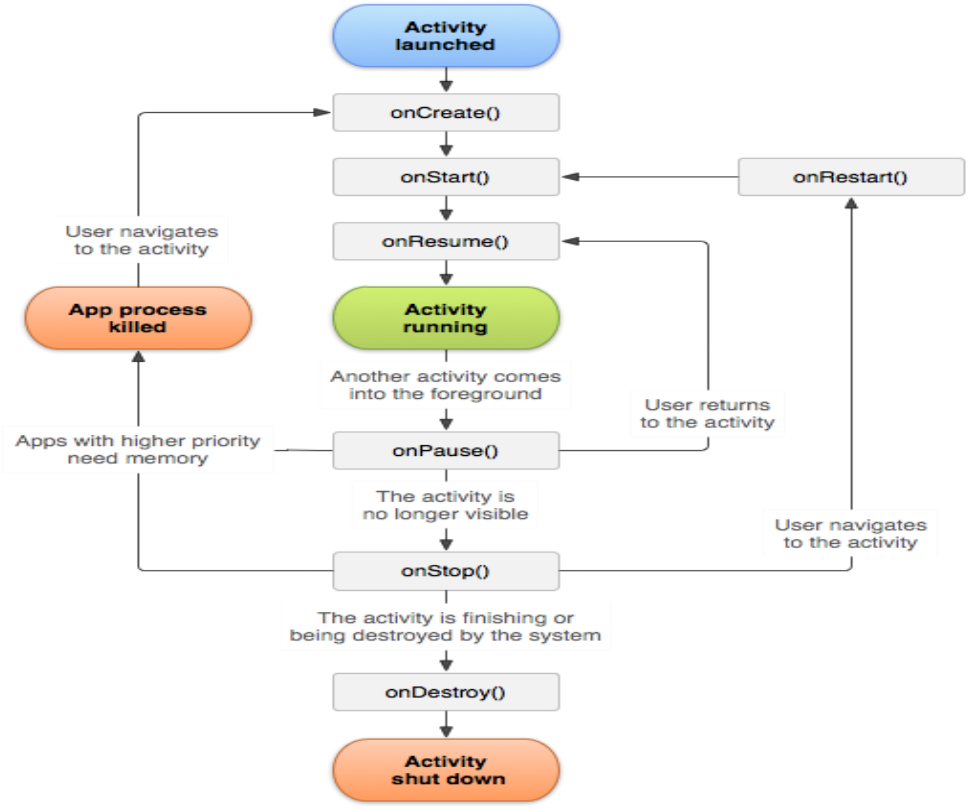
Gambar 2. 3 *Flowchart* enkripsi dan dekripsi data file

2.2.9 Android

Android adalah sistem operasi berbasis linux yang dirancang untuk perangkat mobile seperti smartphone dan tablet. Sistem operasi ini dikembangkan oleh google dan banyak dipakai oleh berbagai produsen perangkat mobile. Android memliki berbagai fitur, seperti kemampuan

untuk mengunduh aplikasi dari google play stor, integrasi dengan layanan google, tampilan antarmuka yang dapat disesuaikan, dan banyak lagi.

Aplikasi android memiliki siklus hidup, android memiliki beberapa *activity* atau aktifitas yang terkait. *Activity* adalah komponen aplikasi yang menyediakan antarmuka pengguna di layar sehingga pengguna dapat berinteraksi dengan aplikasi yang sedang diluncurkan[3]. Berikut adalah *Actifity* atau siklus hidup android pada gambar di bawa ini :



Gambar 2. 4 Siklus hidup android

Kinerja android terhadap kriptografi melibatkan sejumlah aspek, dan sebagian besar tergantung pada implementasi kriptografi yang diterapkan serta spesifikasi perangkat keras yang digunakan. Beberapa poin termasuk :

1. Prosesor dan Hardwere

Kinerja kriptografi dapat dipengaruhi oleh kemampuan perangkat keras, khususnya kemampuan prosesor untuk menangani operasi kriptografi secara efisien. Perangkat keras tambahan seperti modul keamanan (*secure element*) juga dapat memberikan kontribusi positif terhadap kinerja kriptografi.

2. Algoritma Kriptografi yang Digunakan

Pilihan algoritma Kriptografi dapat memengaruhi kinerja. Algoritma yang lebih kompleks biasanya membutuhkan lebih banyak daya komputasi dibandingkan algoritma yang lebih sederhana.

3. Optimasi Implementasi Software

Efisiensi implementasi perangkat lunak dari algoritma kriptografi juga memainkan peran penting. Optimasi pernagkat lunak yang baik dapat meningkatkan kinerja, dan android menyediakan API keamanan yang dioptimalkan untuk operasi kriptografi.

4. Keamanan dan Kepatuhan

Tingkat keamanan yang diterapkan juga dapat memengaruhi kinerja. Implementasi kriptografi yang lebih ketat untuk memengaruhi standar keamanan tertentu atau kepatuhan regulasi mungkin memerlukan lebih banyak sumber daya.

Dengan pemilihan yang tepat dari algoritma kriptografi, implementasi yang efisien, dan mempertimbangkan kebutuhan keamanan aplikasi, android dapat memberikan kinerja kriptografi yang memadai untuk menjaga data dan komunikasi pengguna dengan aman. Kriptografi memiliki beberapa manfaat penting pada android, khususnya dalam konteks keamanan dan privasi. Berikut adalah peran kriptografi pada android :

1. Privasi Pengguna

- Kriptografi digunakan untuk melindungi data sensitif pengguna seperti informasi login, detailkartu kredit, dan data pribadi.

- Enkripsi data di android mencegah akses tidak sah ke informasi pribadi pengguna jika perngkat dicuri atau hilang.

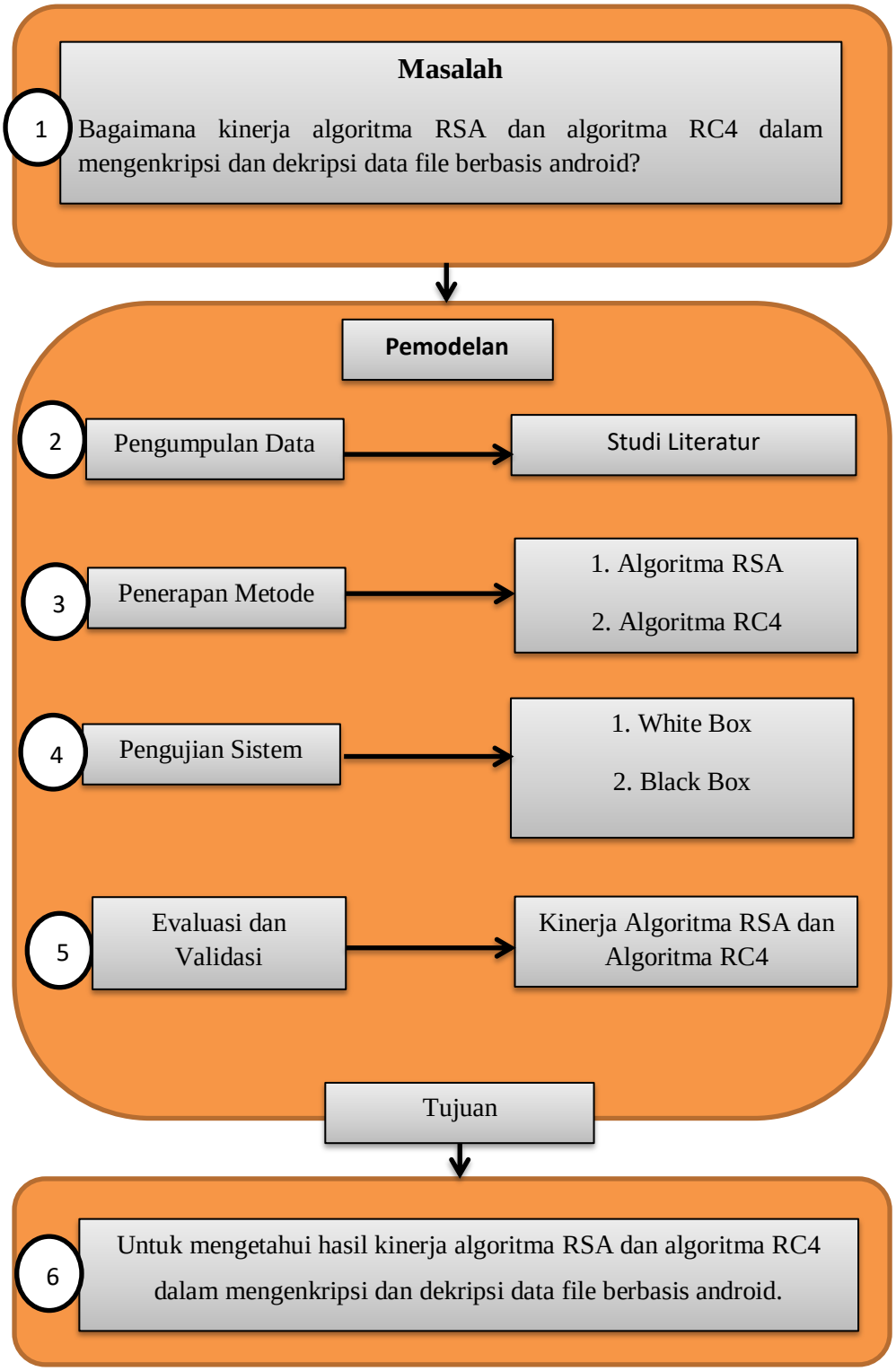
2. Pertukaran Data Aman
    - Protokol kriptografi digunakan dalam komunikasi jaringan, seperti HTTPS, untuk melindungi data yang dikirimkan dan diterima antara perangkat android dan server.
    - Mencegah serangan *Man-in-the-Middle* (MITM) yang dapat mengakses atau memanipulasi data selama transit.
3. Penyimpanan Aman
    - Kriptografi digunakan untuk mengamankan data yang disimpan diperangkat android, termasuk penyimpanan file dan database lokal.
    - Enkripsi penyimpanan dapat melindungi data pengguna dari akses tidak sah atau pencurian fisik perangkat.
4. Autentikasi Pengguna
    - Kriptografi digunakan dalam proses autentikasi pengguna untuk memastikan bahwa hanya pengguna yang sah yang dapat mengakses perangkat atau aplikasi tertentu.
    - Metode seperti HMAC (*Hash-bassed Message Authentication Code*) dan token kriptografi digunakan untuk mengonfirmasi identitas pengguna.
5. Keamanan Aplikasi
    - Menggunakan enkripsi untuk melindungi sumber daya aplikasi, seperti file konfigurasi, data cache, dan sumber daya lainnya.
    - Mencegah modifikasi atau pencurian data aplikasi yang dapat merugikan integritas dan keamanan.
6. Keamanan Pembayaran Digital
    - Kriptografi diterapkan dalam pembayaran digital dan transaksi keuangan melalui protokol keamanan seperti SSL/TLS.
    - Melindungi detail transaksi dan informasi keuangan pengguna dari ancaman keamanan.

7. Keamanan Platform

- Kriptografi membantu melindungi integritas dan keamanan sistem operasi android itu sendiri.
- Menjamin bahwa update perangkat lunak dan aplikasi didistribusikan secara aman dan tidak dimanipulasi.

Penerapan kriptografi pada android esensial untuk menciptakan lingkungan yang aman dan dapat dipercaya bagi pengguna, mencegah potensi risiko keamanan, dan melindungi data pribadi serta informasi sesnsitif lainnya. Berikut adalah manfaat kriptografi pada andorid:

2.3 Kerangka Pikir



BAB III
METODE PENELITIAN

3.1 Jenis Metode, Subjek, Waktu dan Lokasi Penelitian

Dipandang dari tingkat penerapannya, maka penelitian ini merupakan penelitian teoritis. Metode ini menggunakan studi literatur. Oleh karena itu data penelitian ini diperoleh dengan memperbanyak literatur tentang algoritma RSA dan RC4 dibuku maupun jurnal penelitian sebelumnya.

Berdasarkan latar belakang dan kerangka pikir seperti yang telah diuraikan di Bab I dan Bab II, maka yang menjadi objek penelitian yaitu **"Perbandingan Kinerja Algoritma RSA dan Algoritma RC4 dalam Mengenkripsi dan Dekripsi Data File Berbasis Android"**. Penelitian ini dimulai pada tanggal 27 september tahun 2023.

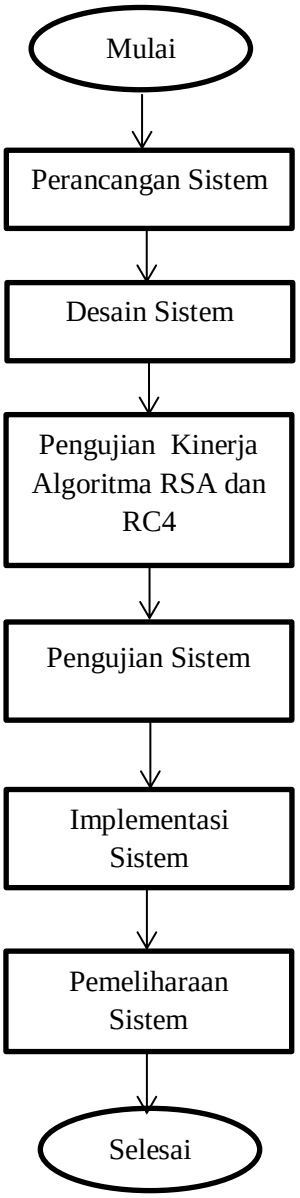
3.2 Pengumpulan Data

Metode pengumpulan data yang digunakan untuk mendapatkan data dan informasi digunakan hanya satu jenis data yaitu data sekunder sebagai berikut :

3.2.1 Data Sekunder

Data sekunder merupakan pengambilan informasi dengan melakukan pengkajian ke perpustakaan yang berisis dasar-dasar teori. Metode ini digunakan untuk mengambil contoh dokumen yang berhubungan dengan objek penelitian.

3.3 Pemodelan



Gambar 3. 1 Model Usulan

3.4 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa sistem atau perangkat lunak yang dikembangkan berfungsi sesuai dengan kebutuhan dan spesifikasi yang ditetapkan. Pengujian sistem bertujuan untuk mengidentifikasi dan memperbaiki bug atau masalah lainnya dalam sistem

sebelum diluncurkan secara resmi. Ada beberapa tahapan dalam pengujian sistem yaitu :

3.4.1 *White Box Testing*

White box testing adalah metode pengujian perangkat lunak yang dilakukan dengan memiliki pengetahuan dan akses ke internal struktur, kode, dan desain dari sistem yang diuji. Dalam *white box testing*, diuji dengan memeriksa logika program, alur kontrol, dan kondisi yang terjadi didalam kode dengan melakukan tes unit, tes integrasi, dan tes sistem.

Tujuan dari white box testing adalah untuk memastikan bahwa semua jalur eksekusi dalam kode telah diuji dengan baik dan semua statement telah dievaluasi dengan benar. *Tester* juga dapat mengidentifikasi kode yang tidak efisien, *dead code*, dan bug potensial.

Metode pengujian ini digunakan pada tahap pengembangan perangkat lunak karena memerlukan pemahaman yang mendalam tentang struktur internalnya. *White box testing* juga membantu dalam meningkatkan kualitas kode, mengurangi kesalahan logis, dan meningkatkan keamanan sistem.

3.4.2 *Black Box Testing*

Black box testing adalah metode pengujian perangkat lunak yang dilakukan tanpa pengetahuan internal tentang kode atau struktur internal sistem yang diuji. Pada pengujian ini, fokus diberikan pada *input* dan *output* yang dihasilkan oleh perangkat lunak tanpa memperhatikan proses internal yang terjadi didalamnya.

Metode ini bertujuan untuk menguji fungsionalitas perangkat lunak, dengan memeriksa apakah perangkat lunak menghasilkan *output* yang diharapkan berdasarkan *input* yang diberikan, tanpa memperdulikan bagaimana pernagkat lunak mencapai hasil tersebut.

Untuk melakukan *black cox testing,* biasanya digunakan teknik pengujian seperti pengujian kecacatan, pengujian batas, pengujian kasus uji, dan pengujian urutan. Dengan pengujian ini, pengembang dapat

mengidentifikasi kesalahan dan masalah fungsionalitas yang mungkin terjadi di dalam perangkat lunak tanpa harus mengetahui detil internalnya.

BAB IV
HASIL PENELITIAN

4.1 Proses Pengumpulan Data

Peneliti telah mengambil data melalui internet dan data yang dikumpulkan telah diproses menjadi file txt, yang dimana data file txt tersebut akan di uji pada dua algoritma kriptografi yaitu algoritma RSA dan algoritma RC4

4.2 Analisa dan implementasi sistem

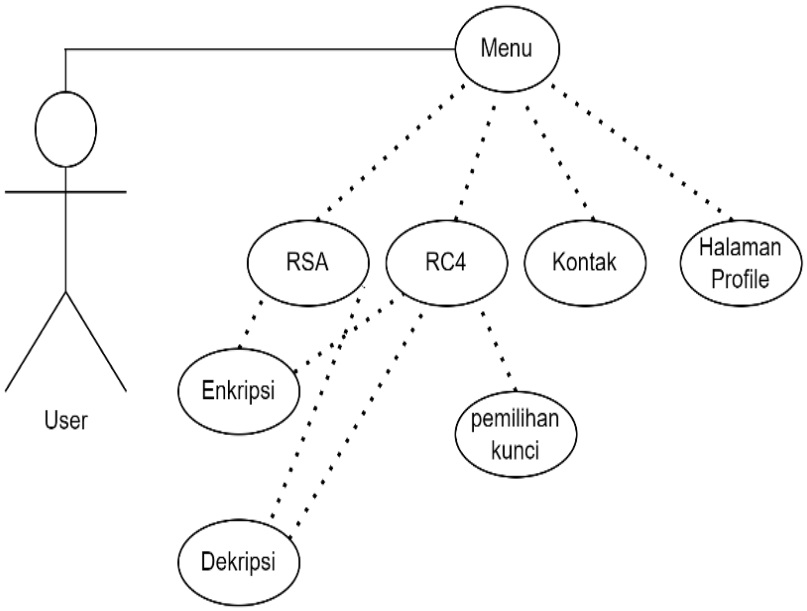
Analisa dan implementasi sistem adalah penguraian dari suatu system informasi yang utuh kedalam bagian-bagian komponen dengan tujuan mengidentifikasikan dan mengevaluasi permasalahan, hambtan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan dapat diusulkan perbaikan.

Analisa merupakan tahap pertama dalam pemngembangan perangkat lunak system, dimana ahli teknik system menganalisis hal-hal yang diperlukan dalam melaksanakan proyek pembuatan atau pengembangan perangkat lunak. Adapun diagram yang digunakan untuk UML dalam aplikasi ini.

4.2.1 Perancangan Aplikasi

Untuk perancangan aplikasi peneliti menggunakan *use case diagram* yang bekerja dengan cara mendekripsikan tipe interaksi user dengan sebuah system melalui diagram. *Use case diagram* merupakan sebuah pemodelan untuk menggambarkan perilaku (tingkah laku) system aplikasi yang dibuat. Sebuah *use case* menggambarkan sebuah interaksi antara pengguna (*user*) dengan system yang ada. Berikut ini merupakan fungsi yang diterapkan pada *use case diagram* system aplikasi android dalam mengenkripsi dan dekripsi data file txt menggunakan algoritma RSA dan RC4

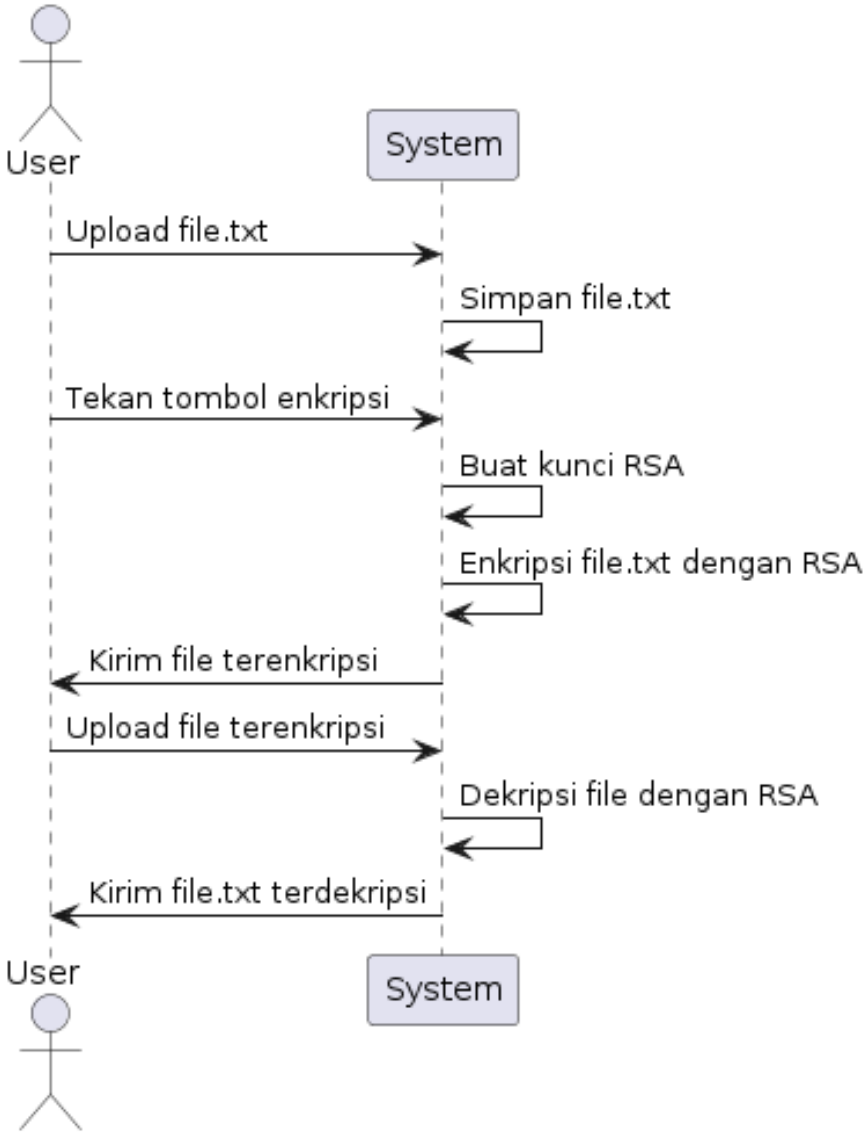
4.2.2 *Use Case Diagram*



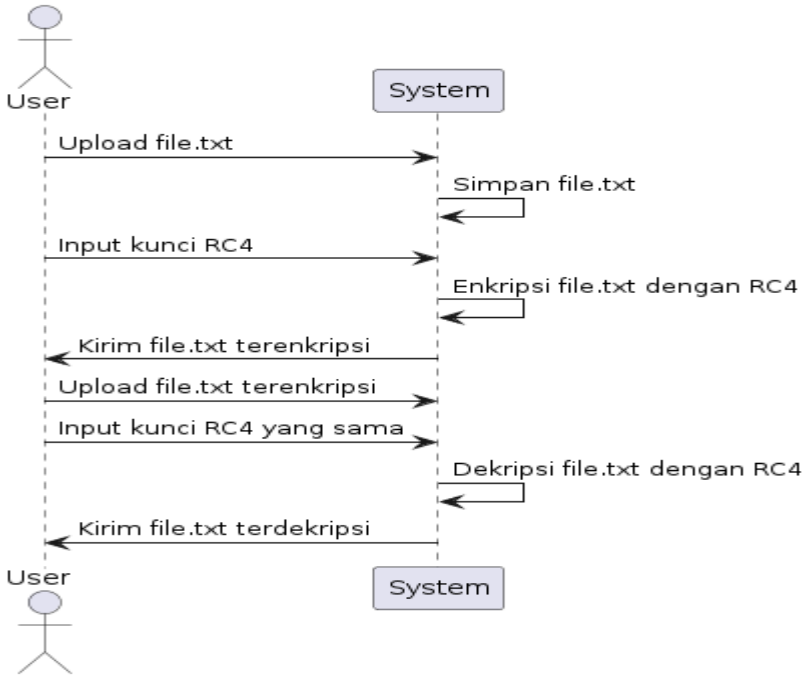
Gambar 4. 1 *Use Case Diagram*

Use Case, fungsi ini disebut fungsionakitas yang ada pada system sebagai unit-unit yang saling bertukar pesan antara unit atau *user*. Biasanya komponen ini dinyatakan dengan menggunakan sebuah karja diawal frase nama *use case*

4.2.3 *Sequance Diagram*



Gambar 4. 2 *Sequance Diagram* RSA



Gambar 4. 3 *Sequance Diagram* RC4

4.3 *Interface design*

4.3.1 *Mekanisme User*

Tabel 4. 1 Mekanisme User

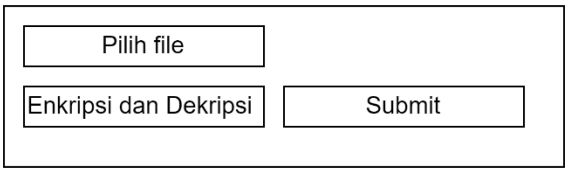
User	Kategori	Akses input	Akses Output
user	user	All	All

4.3.2 *Mekanisme Navigasi Home*

Tabel 4. 2 Mekanisme Navugasi Hone

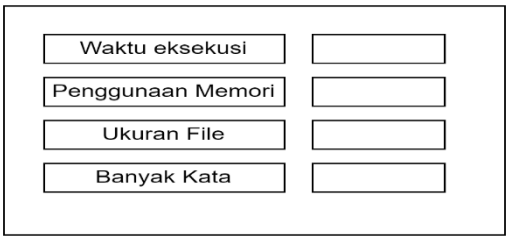
	Tampilan home				
	Tampilan Utama	Menu RSA	Menu RC4	Tampilan Kontak	Tampilan Profil

4.3.3 Mekanisme Input Data



Gambar 4. 4 Mekanisme input data

4.3.4 Mekanisme Output Data

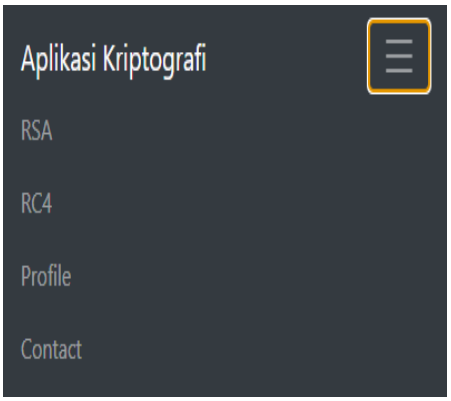


Gambar 4. 5 Mekanisme output data

4.4 Proses Pengujian

Prose pengujian dilakukan menggunakan aplikasi yang telah dibuat oleh peneliti yang dimana aplikasi ini berbasis android. Berikut adalah proses cara pengujian Algoritma RSA dan RC4 dalam mengenkripsi dan dekripsi data file txt menggunakan aplikasi berbasis android :

1. Pilih menu RSA/RC4



Gambar 4. 6 Pilihan menu

2. Pilih file txt yang akan di enkripsi atau dekrpsi

Gambar 4. 7 Menu RSA

3. Pilih enkripsi Atau dekripsi

Gambar 4. 8 Menu RSA

Terus tekan submit nanti akan timbul hasilnya dari waktu eksekusi, penggunaan memori, ukuran file, dan banyak kata.

4.4.2 Pengujian Kinerja Enkripsi RSA dan RC4

Tabel 4. 3 Hasil Pengujian Enkripsi RSA dan RC4

UKURAN FILE	JUMLAH KATA	WAKTU EKSEKUSI	PENGGUNAAN MEMORI	UKURAN FILE
RSA				
File 1	1 kata	1.0459 detik	0.0549 MB	5 bytes
File 2	2 kata	1.0265 detik	0.0117 MB	13 bytes
File 3	5 kata	1.0414 detik	0.0156 MB	26 bytes
File 4	7 kata	1.0255 detik	0.0117 MB	38 bytes
File 5	9 kata	1.0564 detik	0.0039 MB	51 bytes
File 6	11 kata	1.0738 detik	0.0078 MB	62 bytes
File 7	13 kata	1.0429 detik	0.0039 MB	78 bytes
File 8	16 kata	1.0524 detik	0.0078 MB	94 bytes
File 9	18 kata	1.0733 detik	0.0039 MB	108 bytes
File 10	21 kata	1.0298 detik	0.0039 MB	121 bytes
File 11	13 kata	1.0359 detik	0.0078 MB	512 bytes
File 12	9 kata	1.0441 detik	0 MB	768 bytes
File 13	22 kata	1.0428 detik	0 MB	1024 bytes
File 14	38 kata	1.0622 detik	0 MB	1538 bytes
File 15	50 kata	1.0711 detik	0 MB	2048 bytes
RC4				
UKURAN FILE	JUMLAH KATA	WAKTU EKSEKUSI	PENGGUNAAN MEMORI	UKURAN FILE
File 1	1 kata	0.1132 detik	0.0039 MB	5 bytes
File 2	2 kata	0.1105 detik	0.0039 MB	13 bytes
File 3	5 kata	0.1063 detik	0.0273 MB	26 bytes
File 4	7 kata	0.1095 detik	0 MB	38 bytes
File 5	9 kata	0.1098 detik	0 MB	51 bytes
File 6	11 kata	0.1139 detik	0 MB	62 bytes
File 7	13 kata	0.1117 detik	0 MB	78 bytes
File 8	16 kata	0.1053 detik	0 MB	94 bytes
File 9	18 kata	0.1106 detik	0 MB	108 bytes
File 10	21 kata	0.11 detik	0.0039 MB	121 bytes
File 11	33 kata	0.1259 detik	0.0117 MB	234 bytes
File 12	67 kata	0.1027 detik	0.00117 MB	487 bytes
File 13	93 kata	0.1748 detik	0.0469 MB	682 bytes
File 14	146 kata	0.142 detik	0.1133 MB	1116 bytes
File 15	150 kata	0.11416 detik	0.0547 MB	1479 bytes

4.4.3 Pengujian Kinerja Dekripsi RSA dan RC4

Tabel 4. 4 Hasil Pengujian Dekripsi RSA dan RC4

UKURAN FILE	JUMLAH KATA	WAKTU EKSEKUSI	PENGGUNAAN MEMORI	UKURAN FILE
RSA				
File 1	4 kata	1.0257 detik	0 MB	256 bytes
File 2	8 kata	1.0566 detik	0.0078 MB	256 bytes
File 3	6 kata	1.057 detik	0.0078 MB	256 bytes
File 4	6 kata	1.0265 detik	0 MB	256 bytes
File 5	5 kata	1.0428 detik	0 MB	256 bytes
File 6	8 kata	1.0402 detik	O MB	256 bytes
File 7	4 kata	1.0887 detik	0.0039 MB	256 bytes
File 8	11 kata	1.0428 detik	0 MB	256 bytes
File 9	6 kata	1.0415 detik	0.0078 MB	256 bytes
File 10	4 kata	1.0559 detik	0.0117 MB	256 bytes
File 11	33 kata	1.0885 detik	0.0117 MB	234 bytes
File 12	67 kata	1.0711 detik	0.0039 MB	487 bytes
File 13	93 kata	1.0724 detik	0.0039 MB	682 bytes
File 14	146 kata	1.1198 detik	0.00078 MB	1116 bytes
File 15	190 kata	1.1059 detik	0 MB	1479 bytes
RC4				
UKURAN FILE	JUMLAH KATA	WAKTU EKSEKUSI	PENGGUNAAN MEMORI	UKURAN FILE
File 1	2 kata	0.1101 detik	0 MB	5 bytes
File 2	3 kata	0.1123 detik	0 MB	13 bytes
File 3	3 kata	0.1117 detik	0 MB	26 bytes
File 4	3 kata	0.1114 detik	0 MB	38 bytes
File 5	3 kata	0.1164 detik	0 MB	51 bytes
File 6	4 kata	0.1257 detik	0 MB	62 bytes
File 7	4 kata	0.1038 detik	0 MB	78 bytes
File 8	5 kata	0.1257 detik	0 MB	94 bytes
File 9	5 kata	0.1105 detik	0 MB	108 bytes
File 10	5 kata	0.1262 detik	0 MB	121 bytes
File 11	4 kata	0.1113 detik	0 MB	234 bytes
File 12	12 kata	0.1258 detik	0.0039 MB	487 bytes
File 13	13 kata	0.1257 detik	0 MB	682 bytes
File 14	19 kata	0.1107 detik	0.0273 MB	1116 bytes
File 15	29 kata	0.1348 detik	0.0469 MB	1479 bytes

4.4.4 Penerapan Algoritma RSA

Dik : P = 25 q = 7

n = 25x7 = 175

Mencari nilai Ø(n) = 25-1 x 7-1 = 24 x 6 = 144

Pilih kunci e = 5

Menghitung kunci privat d dengan ke kongruenan

e.d = 1(mod(Ø(n))/e

penyelesaian : d = 1 + (k*(Ø(n))/e

= 1 + (1*144)/5

= 1 + 145 / 5

= 29

Maka didapat nilai d yang bulat adalah 29

Mengkonversi plainteks “VITO” dengan table ASCII sehingga diperoleh :

V = 86 plainteks : 86738479

I = 73

T = 84

O = 79

Buat M menjadi blok-blok yang lebih kecil

m1 : 86 m3 : 84

m2 : 73 m4 : 79

Proses enkripsi :

Rumus : Ci = mi^e mod n

$C1 = 86^5 \bmod 175 = 151$

$C2 = 73^5 \bmod 175 = 68$

$C3 = 84^5 \bmod 175 = 49$

$C4 = 79^5 \bmod 175 = 74$

Chipertext = 151684974

Proses dekripsi :

Rumus : mi = Ci^d mod n

$m1 = 151^{29} \bmod 175 = 86$ $m3 = 49^{29} \bmod 175 = 84$

$m2 = 69^{29} \bmod 175 = 73$ $m4 = 74^{29} \bmod 175 = 79$

plaintext M = 86738479 yang Dimana dalam table ASCII karakter hurufnya adalah VITO.

4.4.5 Penerapan Algoritma RC4

Diketahui :

1. Inisiasi array S-Box

Tabel 4. 5 *array S-Box*

INDEX	0	1	2	3
S	0	1	2	3

2. Inisiasi array kunci S-Box

   Plainteks : VITO

   Kunci : 0540

   Penyelesaisan :

3. Permutasi

   Iterasi 1

Tabel 4. 6 Permutasi Iterasi 1

S[1]	0	1	2	3
K[1]	0	5	4	0

S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, melakukan KSA

i = 0

j = (j + s[i] + k[0]) mod 4

j = (0 + 0 + 0) mod 4

j = 0

swap = (S[0] + S[0]) mod 4

swap = (0 + 0) mod 4 = 0

Perpindahan

Tabel 4. 7 Perpindahan Iterasi 1

S[i]	2	1	0	3
Key	0	5	4	0

Iterasi 2

Tabel 4. 8 Permutasi Iterasi 2

S[i]	2	1	0	3
Key	0	5	4	0

S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, melakukan KSA

i = 1

j = (j + s[i] + k[1]) mod 4

j = (0 + 1 + 5) mod 4

j = 2

Swap = (S[1] + S[2]) mod 4

Swap = (1 + 0) mod 4 = 1

Perpindahan

Tabel 4. 9 Perpindahan Iterasi 2

S[i]	1	2	0	3
Key	0	5	4	0

Iterasi 3

Tabel 4. 10 Permutasi Iterasi 3

S[i]	1	2	0	3
Key	0	5	4	0

S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, melakukan KSA

i = 2

j = (j + s[i] + k[2]) mod 4

j = (2 +0 + 4) mod 4

j = 2

Swap = (S[2] + S[2]) mod 4

Swap (0 + 0) mod 4 = 0

Perpindahan :

Tabel 4. 11 Perpindahan Iterasi 3

S[i]	1	2	3	0
Key	0	5	4	0

Iterasi 4

Tabel 4. 12 Permutasi Iterasi 4

S[i]	1	2	3	0
Key	0	5	4	0

S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, melakukan KSA

i = 3

j = (j +s[1] + k[3]) mod 4

j = (2 + 0 + 0) mod 4

j = 2

Swap = (S[3] + S[2]) mod 4

Swap = (0 +3) mod 4 =3

Perpindahan :

Tabel 4. 13 Perpindahan Iterasi 4

S[i]	1	2	3	0
Key	0	5	4	0

4. Aliran kunci

   Iterasi 1

   i = 0

   j = 0

   i = (I +1) mod 4

   I = (0 + 1) mod 4

   = 1

   j = (0 + s[i]) mod 4

   j = (0 + s[1]) mod 4

j = (0 + 3) mod 4

j = 3

Swap (s[i], s[j])

Swap (s[1], s[3])

Tabel 4. 14 Aliran Kunci Iterasi 1

Array Awal	1	3	2	0
Hasil setelah Swap	1	0	2	3

t ⟵ (s[1] + s[3]) mod 4

(0 + 3) mod 4

= 3

K ⟵ s[t] = 00000011

Iterasi 2

i = 1

j = 3

i = (i +1) mod 4

I = (1 + 1) mod 4

= 2

j = (3 + s[i]) mod 4

j = (3 + s[2]) mod 4

j = (3 + 2) mod 4

j = 3

Swap (s[i], s[j])

Swap (s[2], s[1])

Tabel 4. 15 Aliran Kunci Iterasi 2

Array Awal	1	0	2	3
Hasil Setelah Swap	1	2	0	3

t ⟵ (s[2] + s[1]) mod 4

(0 + 2) mod 4

= 2

K ◄——— s[t] = 00000010

Itreasi 3

i = 2

j = 1

i = (i +1) mod 4

I = (2 + 1) mod 4

= 3

j = (1 + s[i]) mod 4

j = (1 + s[3]) mod 4

j = (1 + 3) mod 4

j = 0

Swap (s[i], s[j])

Swap (s[3], s[0])

Tabel 4. 16 Aliran Kunci Iterasi 3

Array Awal	1	2	0	3
Hasil Setelah Swap	3	2	0	1

t ◄——— (s[3] + s[0]) mod 4

(1 + 3) mod 4

= 0

K ◄——— s[t] = 00000000

Iterasi 4

i = 3

j = 0

i = (i +1) mod 4

I = (3 + 1) mod 4

= 0

j = (0 + s[i]) mod 4

j = (0 + s[3]) mod 4

j = (0 + 3) mod 4

j = 0, Swap (s[i], s[j]) Swap (s[0], s[3])

Tabel 4. 17 Aliran Kunci Iterasi 4

Array Awal	3	2	0	1
Hasil Setelah Swap	1	2	0	3

t ⟵ (s[0] + s[3]) mod 4

(1 + 3) mod 4

= 0

K ⟵ s[t] = 00000000

5. Enkripsi

Plaintext :

V	I	T	O
01010110	01001001	01010100	01001111

Kunci :

00000011	00000010	00000000	00000000

XOR

Cipherteks

01010101	01001011	01010100	01001111

6. Dekripsi

Cipherteks :

01010101	01001011	01010100	01001111

XOR

Byte Acak Semu :

00000011	00000010	00000000	00000000

Plainteks :

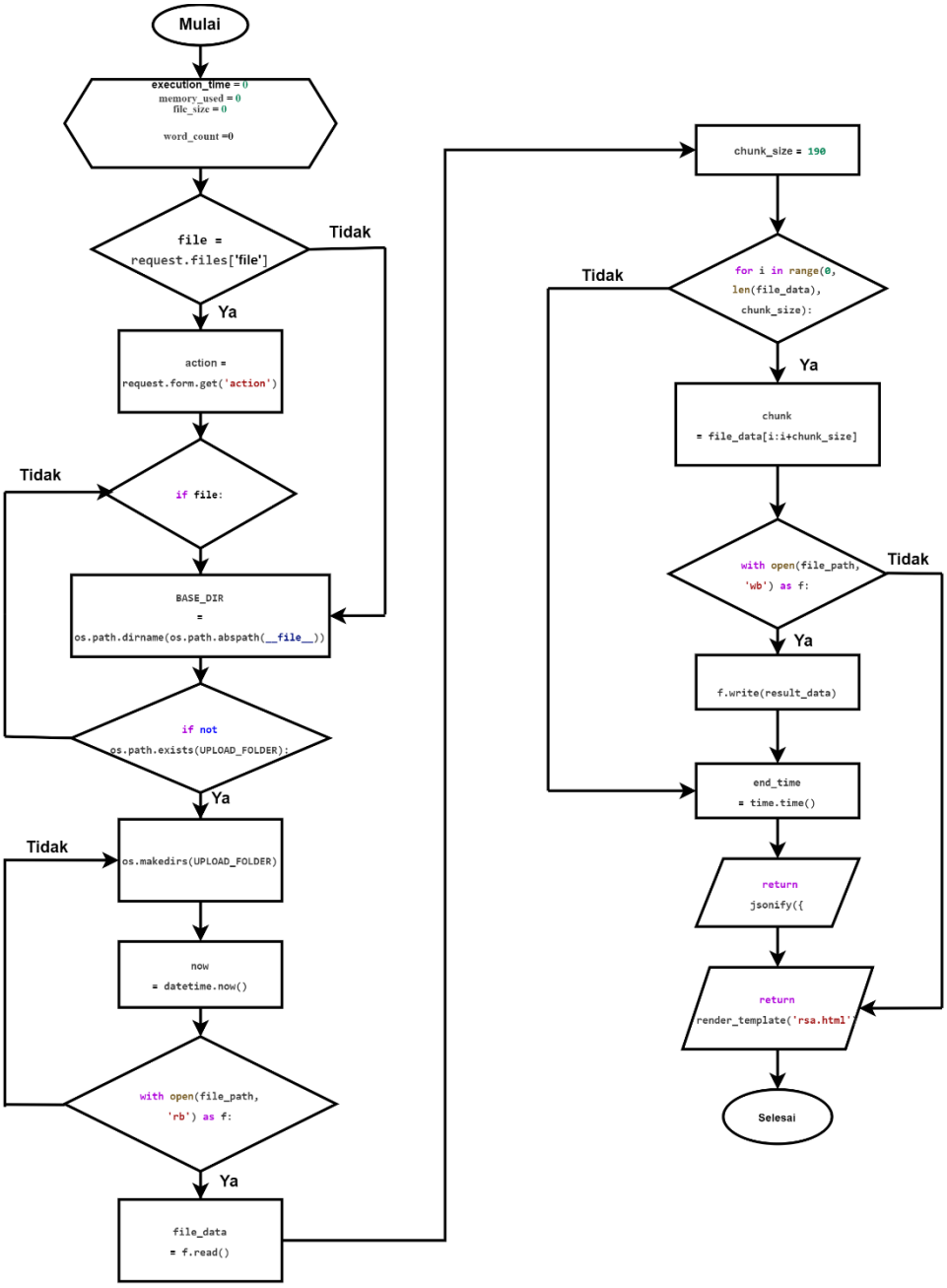
01010110	01001001	01010100	01001111
V	I	T	O

4.5 Pengujian Sistem

4.5.1 Pengujian *White Box*

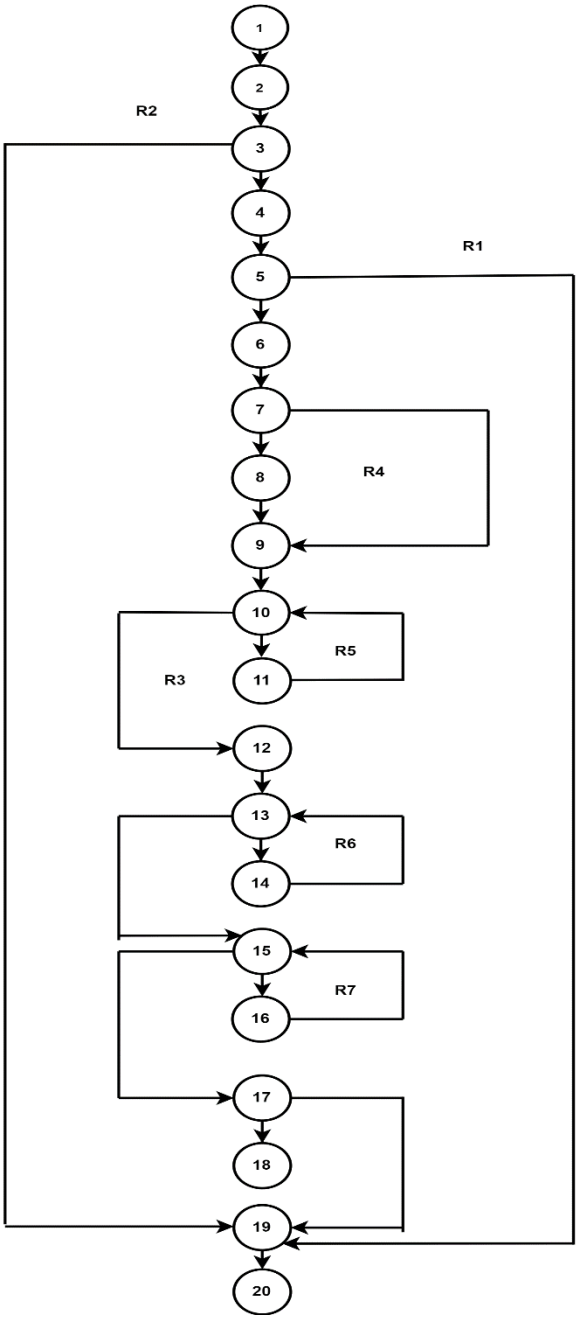
```
execution_time = 0……...1
memory_used = 0……..…….....1
file_size = 0……..1
word_count = 0……1
……..2
   file = request.files['file'] ……...3
   action = request.form.get('action') ……...3
   if file: ……...4
      BASE_DIR = os.path.dirname(os.path.abspath(__file__))……..5
      UPLOAD_FOLDER = os.path.join(BASE_DIR, 'uploads') ……...5
      if not os.path.exists(UPLOAD_FOLDER): …….....6
         os.makedirs(UPLOAD_FOLDER) ……..7

      now = datetime.now()……8
      formatted_time = now.strftime("%H_%M_%S-%d_%m_%Y").....8
      start_time = time.time()……8
      initial_memory_usage = memory_usage(-1, interval=0.1, timeout=1).8
      new_filename = f"rsa_{action}-{formatted_time}.txt"……8
      file_path = os.path.join(UPLOAD_FOLDER, new_filename) …….8
      file.save(file_path) ……8

      with open(file_path, 'rb') as f: ……9
         file_data = f.read()……10

      chunk_size = 190……..11
      encrypted_data = bytearray()……11

      for i in range(0, len(file_data), chunk_size): ……...12
         chunk = file_data[i:i+chunk_size] ……...13
         encrypted_chunk = cipher_encrypt.encrypt(chunk) ……13
         encrypted_data.extend(encrypted_chunk) ……...13

      with open(file_path, 'wb') as f: ……14
         f.write(result_data) ……...15

      end_time = time.time()……16
      final_memory_usage = memory_usage(-1, interval=0.1, timeout=1)..16
      execution_time = round(end_time - start_time, 4)……...16
      memory_used=round(max(final_memory_usage)-
min(initial_memory_usage),4)……16
      file_size = os.path.getsize(file_path)……16
      word_count = len(open(file_path, 'rb').read().split())……...16

      return jsonify({
```

4.5.2 Flowchart Pengujian *white Box*



Gambar 4. 9 *Flowchart* Pengujian white box

4.5.3 *Flowgraph White Box*



Gambar 4. 10 Flowgraph Pengujian White Box

4.5.4 Perhitungan CC Pada Pengujian *White Box*

Dari *Flowgraph* tersebut didapatkan :

Diketahui : Edge(E) = 25

Node(N)= 20

Predikat Node (P)= 6

Region (R) = 7

Penyelesaian V(G)1 = E-N+2

= 25-20+2

= 5 +2

= 7

V(G)2 = P+1

= 6+1

= 7

4.5.5 Path Pada Pengujian *White Box*

No	Path	Ket
1	1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20	Ok
2	1-2-19	Ok
3	1-2-3-4-19	Ok
4	1-2-3-4-5-6-8…19	Ok
5	1-2-3-4-5-6-7-8-9-10-9…19	Ok
6	1-2-3-4-5-6-7-8-9-11-12-13…19	Ok
7	1-2-3…19	Ok

4.5.6 Pengujian *Black Box*

Pengujian black box ini dilakukan untuk memastikan bahwa suatu event atau masukan akan dijalankan dengan proses yang tepat dan menghasilkan output/keluaran yand sesuai dengan rancangan. Untuk contoh pengujian terhadap beberapa proses yang dapat memberikan hasil sebagai :

Tabel 4. 18 Hasil pengujian Black Box

Input/Event	Fungsi	Hasi yang Diharapkan	Hasil Uji
Klik Menu Utama	Menampilkan Halaman Judul	Menu Home Tampil	Sesuai
Klik Menu RSA	Menampilkan pengujian enkripsi dan dekripsi data file	Hasil pengujian kinerja enkripsi dan dekripsi RSA	Sesuai
Klik menu RC4	Menampilkan pengujian enkripsi dan Dekripsi data file	Hasil pengujian Kinerja enkripsi dan dekripsi RC4	Sesuai
Klik Contact	Menampilkan kontak dari perancang/peneliti	Tampil Halaman Kontak	Sesuai
Klik Profile	Menampilkan profile perancang/peneliti	Tampil Halaman Profile	Sesuai

BAB V
HASIL DAN PEMBAHASAAN

5.1 Pembahsaan Sistem

Hasil tampilan system perbandingan algoritma RSA dan algoritmaRC4 dalam mengenkripsi dan dekripsi data file bersasis android.

5.1.1 Kebutuhan *Hardwere/Softwere*

Penulis dalam mengembangkan program ini menggunakan bahasa pemrograman python, html, dan java script.

Pada dasarnya, untuk implementasi system ini membutuhkan konfigurasi dasar, diantaranya.

1. *Hardwhere* dan *softwhere*

   Spesifikasi yang disarankan untuk computer

   a. Prosesor Intel atau AMD x86-64.

   b. RAM (Memory) minimal 8 GB atau lebih

   c. Windows 9,10, atau windows 11

5.1.2 Langkah-langkah Menjalankan Program

1. Tampilan Menu Utama

Gambar 5. 1 Tampilan Menu Utama

Halaman ini akan muncul saat program baru pertama kali dibuka, pada halaman ini hanya terdapat penjelasan tentang kriptografi.

2. Tampilan Menu RSA

Gambar 5. 2 Tampilan Menu RSA

Pada tampilan halaman RSA, pengguna akan menguji kinerja enkripsi dan dekripsi algoritma RSA dari file txt untuk mengukur waktu eksekusi, penggunaan memori, ukuran file, banyak kata.

3. Tampilan Menu RC4

Gambar 5. 3 Tampilan Menu RC4

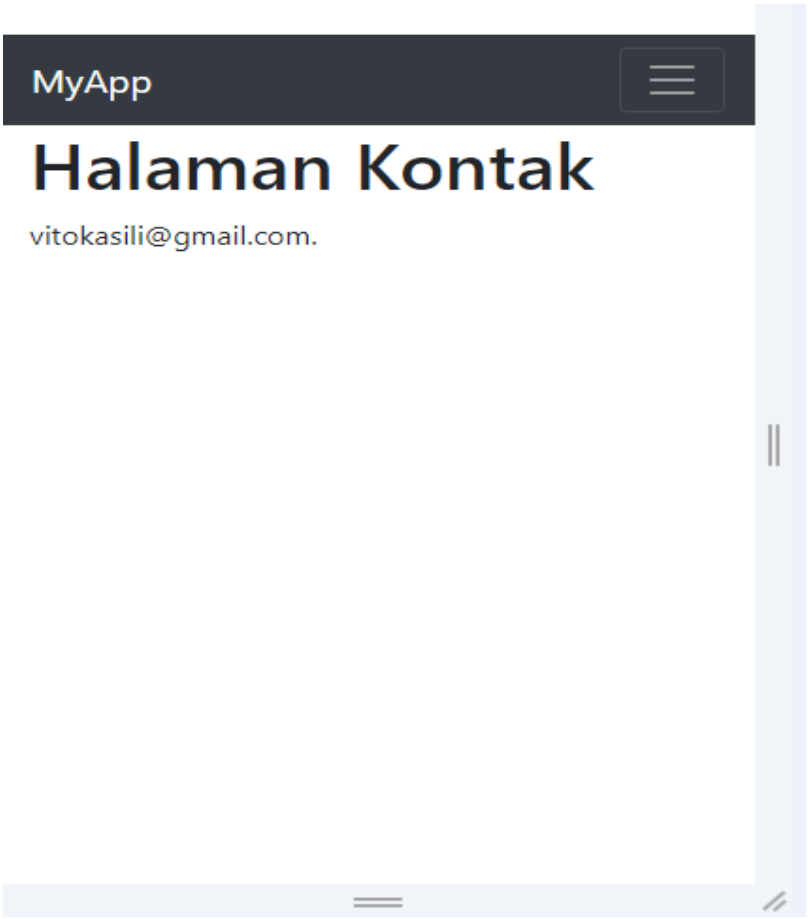
Pada tampilan halaman RC4, pengguna akan menguji kinerja enkripsi dan dekripsi algoritma RC4 dari file txt untuk mengukur waktu eksekusi, penggunaan memori, ukuran file, dan banyak kata.

4. Halaman Menu Profil

Gambar 5. 4 Tampilan Menu Profil

Halaman ini menjelaskan profile perancang.

5. Halaman Menu Kontak



Gambar 5. 5 Tampilan Menu Kontak

Halaman ini menjelaskan informasi kontak dari perancang aplikasi berupa email atau nomor yang masih bisa dihubungi.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdaarkan hasil penellitian yang sudah di lakukan dan hasil pengujian sistem,maka dapat disimpulkan bahwa :

1. Dapat diketahui RC4 cenderung lebih cepat dalam proses enkripsi dan dekripsi dibandingkan RSA. Hal ini disebabkan oleh perbedaan dalam kompleksitas algoritma dan metode enkripsi yang digunakan. Untuk keamanan RSA cenderung lebih aman dibandingkan RC4 karena RC4 telah terkena serangan kriptanalisis yang serius. RSA menggunakan kunci publik dan kunci pribadi , sementara RC4 hanya menggunakan kunci simetris. Penggunaan RSA sering digunakan untuk keperluan yang membutuhkan tingkat keamanan yang tinggi, seperti dalam pengamanan komunikasi melalui jaringan. RC4 meskipun cepat, telah ditinggalkan dalam banyak aplikasi karena kelemahan keamanannya. Dengan demikian, pemilihan antara RSA dan RC4 tergantung pada kebutuhan spesifik pengguna, dengan pertimbangan utama antara keamanan dan kinerja.

6.2 Saran

1. Dalam proses pembuatan aplikasi android, anda dapat melihat bahwa dari sudut pandang keamanan file semuanya bagus, Langkah-langkah yang perlu diselesaikan adalah mempelajari system yang ada, kemudian merancang sistem dan melakukan perhitungan berdasarkan kriteria seleksi.
2. Untuk penelitian selanjutnya peneliti menyarankan melakukan modifikasi program algoritma dalam melakukan perancangan aplikasi android.
3. Peneliti menyarankan untuk dilakukan kombinasi antara RC4 dan RSA untuk penelitian selanjutnya

DAFTAR PUSTAKA

[1] M. Rido Hasibuan, "Implementasi Algoritma Quicksort Untuk Pembangkitan Kunci Algoitma RSA Pada Pengamanan Data Audio," *J. Informatics, Electr. Electron. Eng.*, vol. 2, no. 1, pp. 18–25, 2022, doi: 10.47065/jieee.v2i1.392.

[2] Azlin, F. Musadat, and J. Nur, "Aplikasi Kriptografi Keamanan Data Menggunakan Algoritma Base64," *J. Inform.*, vol. 7, no. 2, pp. 1–5, 2018.

[3] R. R. Rohmansyah and H. Nurwasito, "Pengembangan Aplikasi Mobile untuk Sistem Keamanan Kantor Menggunakan NFC (Near Field Communication) dan Wi-Fi (Studi Kasus : PT. Rahmi Ida Nusantara)," *J. Pengemb. Teknlogi Inf. dan Ilmu Komput.*, vol. 2, no. 1, pp. 81–90, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/696

[4] N. S. Ulfah Indriani, Ommi Alfina, "Penerapan Algoritma Rsa Dan Affine Cipher Dalam Keamanan File Ms Word," vol. 01, no. 02, pp. 95–100, 2021, [Online]. Available: http://repository.potensi-utama.ac.id/jspui/handle/123456789/5074

[5] K. Fahmi, "RESOLUSI : Rekayasa Teknik Informatika dan Informasi Pengamanan Data Arsip Pada Balai Desa Sidodadi Menggunakan Kriptografi Modern RC4," *Media Online*, vol. 2, no. 2, pp. 58–66, 2021, [Online]. Available: https://djournals.com/resolusi

[6] F. Muhariza and N. Juliasari, "Implementasi Pengamanan Dokumen Menggunakan Kriptografi Dengan Algoritme Rivest Code 4 (Rc4) Berbasis Web Implementation of Document Security Using Cryptography With Web-Based Rivest Code 4 (Rc4)," vol. 2, no. September, pp. 131–139, 2023.

[7] S. B. H. Gmbh, "Definisi Perbandingan," pp. 1–23, 2016.

[8] M. S. Dairi, M. Setiani Asih, and corespondent author, "Implementasi Algoritma Kriptografi RSA Dalam Aplikasi Sistem Informasi Perpustakaaan Implementation Of RSA Cryptographic Algorithms in Library Information System Applications," *Januari*, vol. 2023, no. 2, pp. 214–223, 2022, [Online]. Available: https://jurnal.unity-academy.sch.id/index.php/jirsi/index%0Ahttp://creativecommons.org/licenses/by-sa/4.0/

[9] R. Maulana and R. M. Simanjorang, "Implementasi Kriptografi Pengamanan Data Pribadi Siswa SMA Swasta Jaya Krama Beringin Dengan Algoritma RC4," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 4,

no. 6, pp. 377–383, 2021, doi: 10.32672/jnkti.v4i6.3533.

[10] L. Oktasari, "Perbandingan Algoritma RC4 + dan RC4a dalam Pengamanan File Teks," *Bull. Comput. Sci. Res.*, vol. 1, no. 3, pp. 83–92, 2021, [Online]. Available: https://hostjournals.com/bulletincsr

[11] P. Studi and I. Komputer, "PENGAMANAN DATA FILE DENGAN MENGGUNAKAN ALGORITMA ENKRIPSI RIVEST CODE 5 1) Sayekti Harits Suryawan, 2) Hamdani," *J. Inform. Mulawarman Ed. Juli*, vol. 8, no. 2, pp. 44–49, 2013.

[12] S. Vivi Wahdini, D. Hartama, and I. Okta Kirana, "Pengamanan Data Pelanggan dan Penjualan Menggunakan Implementasi Algoritma Kriptografi," *J. Informatics Manag. Inf. Technol.*, vol. 1, no. 3, pp. 101–107, 2021.

LAMPIRAN

LAMPIRAN 1 : LISTING PROGRAM

```
{% extends "main_layout.html" %}

{% block title %}Contact{% endblock %}

{% block content %}
<h1>Halaman Kontak</h1>
<p>vitokasili@gmail.com.</p>
{% endblock %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bo
otstrap.min.css">
    <link rel="stylesheet" href="{{ url_for('static',
filename='css/style.css') }}">
    <title>{% block title %}Welcome{% endblock %}</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="/">Aplikasi
Kriptografi</a>
        <button class="navbar-toggler" type="button" data-
toggle="collapse" data-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="{{ url_for('rsa')
}}">RSA</a>
                </li>
                <li class="nav-item">
```

```
                    <a class="nav-link" href="{{ url_for('rc4')
}}">RC4</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{{
url_for('profile') }}">Profile</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="{{
url_for('contact') }}">Contact</a>
                </li>
            </ul>
        </div>
    </nav>

    <div class="container">
        {% block content %}
        {% endblock %}
    </div>


    <script src="https://code.jquery.com/jquery-
3.5.1.slim.min.js"></script>
    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.2/dist/umd
/popper.min.js"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/boot
strap.min.js"></script>
    <script src="{{ url_for('static', filename='js/script.js')
}}"></script>

    {% block js %}
    {% endblock %}
</body>
</html>

{% extends "main_layout.html" %}

{% block title %}RC4 Encryption{% endblock %}

{% block content %}
<h1>RC4 Encryption</h1>
<p>Halaman ini menyediakan alat enkripsi dan dekripsi Algoritma
RC4.</p>
```

```
<h1>File Encryptor/Decryptor</h1>
<form action="/rc4" method="post" enctype="multipart/form-
data">
    <input type="file" name="file" required>
    <input type="text" name="key" placeholder="Enter key"
required>
    <select name="action">
        <option value="encrypt">Encrypt</option>
        <option value="decrypt">Decrypt</option>
    </select>
    <button type="submit">Submit</button>
</form>
<div class="loader" id="loader"></div>
<hr>
<h3>Execution Time: <span id="executionTime">0
seconds</span></h3>
<h3>Memory Usage: <span id="memoryUsage">0 MB</span></h3>
<h3>File Size: <span id="fileSize">0 bytes</span></h3>
<h3>Word Count: <span id="wordCount">0 words</span></h3>
{% endblock %}

{% block js %}
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.
min.js"></script>
<script>
    $(document).ready(function(){
        $("form").on("submit", function(event){
            event.preventDefault(); // Mencegah submit form
secara tradisional
            var formData = new FormData(this);

            // Tampilkan loader
            $('#loader').show();

            $.ajax({
                url: '/rc4',
                type: 'POST',
                data: formData,
                contentType: false,
                processData: false,
                success: function(data) {
                    // Update UI dengan data baru
```

```
                    $('#executionTime').text(data.execution_tim
e + ' seconds');
                    $('#memoryUsage').text(data.memory_used + '
MB');
                    $('#fileSize').text(data.file_size + '
bytes');
                    $('#wordCount').text(data.word_count + '
words');

                    // Sembunyikan loader
                    $('#loader').hide();
                },
                error: function() {
                    alert('Error processing your request');
                    // Sembunyikan loader
                    $('#loader').hide();
                }
            });
        });
    });
</script>
{% endblock %}
```

LAMPIRAN 2 : SURAT KETERANGAN BEBAS PUSTAKA

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
UPT. PERPUSTAKAAN FAKULTAS
SK. MENDIKNAS RI NO. 84/D/0/2001
Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No : 005/Perpustakaan-Fikom/VI/2024

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota : Moh. Rezaldy D. Kasili
No. Induk : T3120041
No. Anggota : M202425

Terhitung mulai hari, tanggal : Rabu, 05 Juni 2024, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 05 Juni 2024
Mengetahui,
Kepala Perpustakaan

Apriyanto Alhamad, M.Kom
NIDN : 0924048601

LAMPIRAN 3 : SURAT REKOMENDASI PENELITIAN

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
LEMBAGA PENELITIAN
Kampus Unisan Gorontalo Lt.3 - Jln. Achmad Nadjamuddin No. 17 Kota Gorontalo
Telp: (0435) 8724466, 829975 E-Mail: lembagapenelitian@unisan.ac.id

SURAT KETERANGAN

Nomor : 4921/SK/LEMLIT-UNISAN/GTO/XII/2023

Yang bertanda tangan di bawah ini :

Nama	: Dr. Rahmisyari, ST.,SE.,MM
NIDN	: 0929117202
Jabatan	: Ketua Lembaga Penelitian

Menerangkan bahwa :

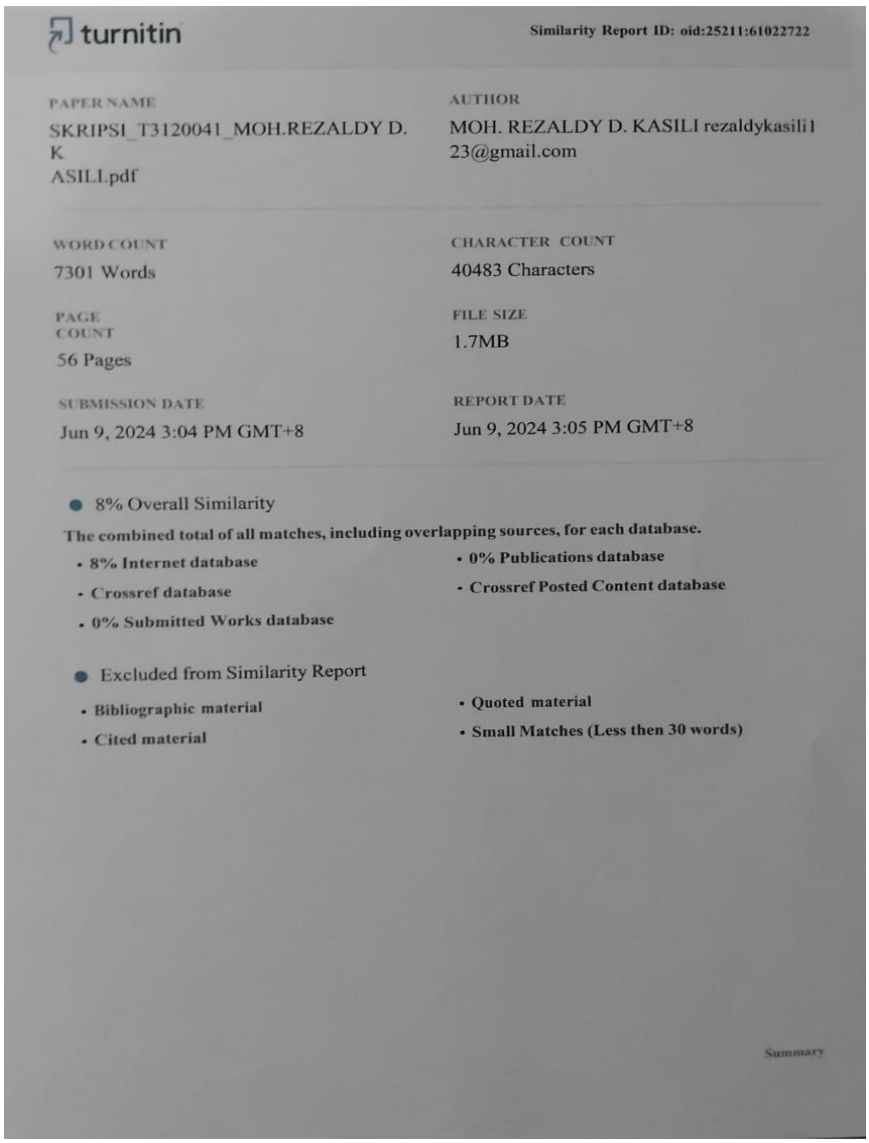
Nama Mahasiswa	: Moh. Rezaldy D. Kasili
NIM	: T3120041
Fakultas	: Fakultas Ilmu Komputer
Program Studi	: Teknik Informatika
Judul Penelitian	: PERBANDI8NGAN KINERJA ALGORITMA RSA DAN RC4 DALAM MENGENKRIPSI DAN DEKRIPSI DATA FILE BERBASIS ANDROID

Adalah benar telah melakukan pengambilan data penelitian dalam rangka Penyusunan Proposal/Skripsi pada UNIVERSITAS ICHSAN GORONTALO.

Gorontalo, 22 Desember 2023
Ketua

Dr. Rahmisyari, ST.,SE.,MM
NIDN 0929117202

LAMPIRAN 4 : TURNITIN



turnitin

Similarity Report ID: oid:25211:61022722

PAPER NAME
SKRIPSI_T3120041_MOH.REZALDY D. K ASILI.pdf

AUTHOR
MOH. REZALDY D. KASILI rezaldykasili123@gmail.com

WORD COUNT
7301 Words

CHARACTER COUNT
40483 Characters

PAGE COUNT
56 Pages

FILE SIZE
1.7MB

SUBMISSION DATE
Jun 9, 2024 3:04 PM GMT+8

REPORT DATE
Jun 9, 2024 3:05 PM GMT+8

● 8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

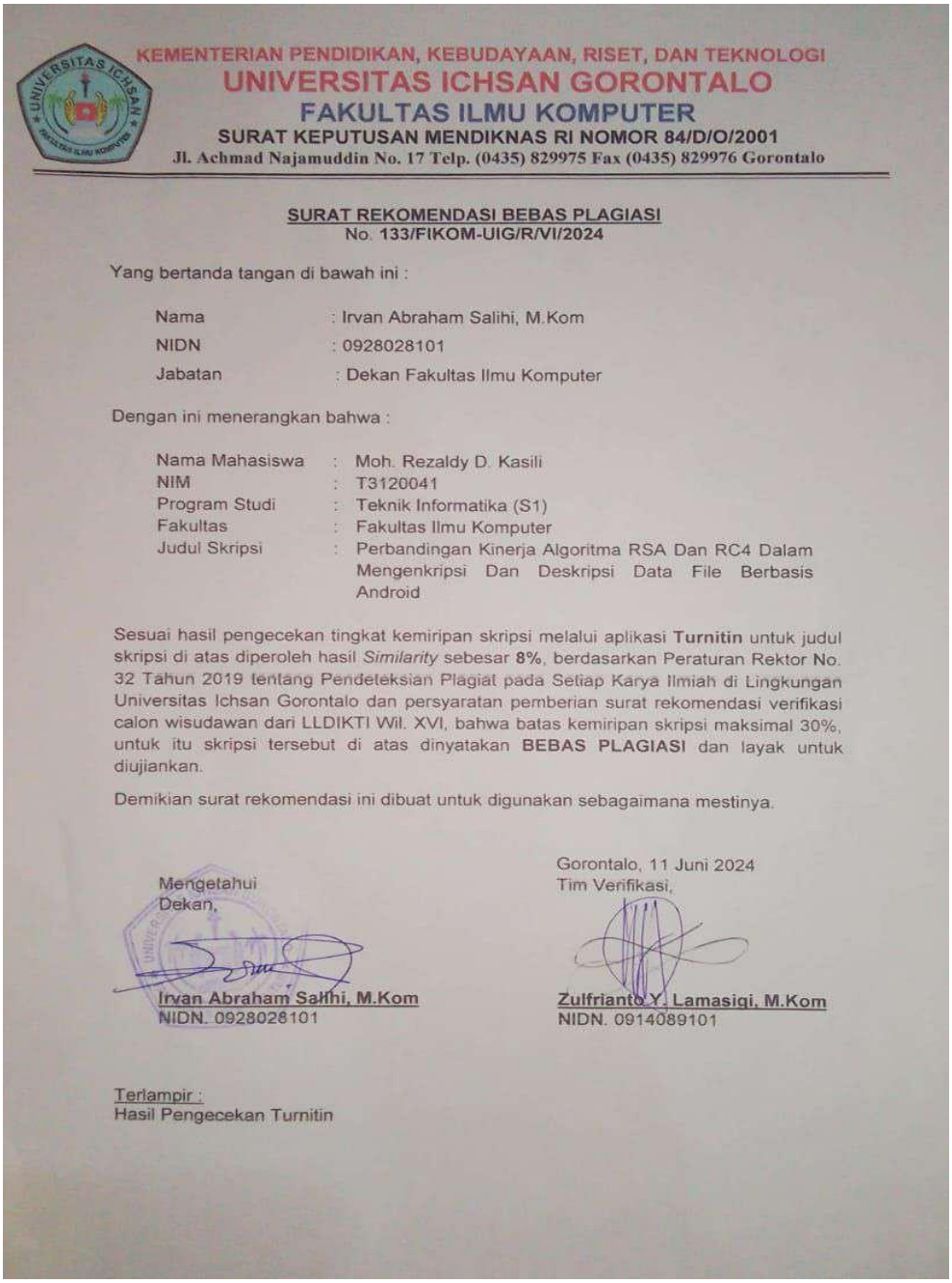
- 8% Internet database
- 0% Publications database
- Crossref database
- Crossref Posted Content database
- 0% Submitted Works database

● Excluded from Similarity Report

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less then 30 words)

Summary

LAMPIRAN 5 : BEBAS PLAGIASI



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Najamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI
No. 133/FIKOM-UIG/R/VI/2024

Yang bertanda tangan di bawah ini :

Nama	: Irvan Abraham Salihi, M.Kom
NIDN	: 0928028101
Jabatan	: Dekan Fakultas Ilmu Komputer

Dengan ini menerangkan bahwa :

Nama Mahasiswa	:	Moh. Rezaldy D. Kasili
NIM	:	T3120041
Program Studi	:	Teknik Informatika (S1)
Fakultas	:	Fakultas Ilmu Komputer
Judul Skripsi	:	Perbandingan Kinerja Algoritma RSA Dan RC4 Dalam Mengenkripsi Dan Deskripsi Data File Berbasis Android

Sesuai hasil pengecekan tingkat kemiripan skripsi melalui aplikasi **Turnitin** untuk judul skripsi di atas diperoleh hasil *Similarity* sebesar **8%**, berdasarkan Peraturan Rektor No. 32 Tahun 2019 tentang Pendeteksian Plagiat pada Setiap Karya Ilmiah di Lingkungan Universitas Ichsan Gorontalo dan persyaratan pemberian surat rekomendasi verifikasi calon wisudawan dari LLDIKTI Wil. XVI, bahwa batas kemiripan skripsi maksimal 30%, untuk itu skripsi tersebut di atas dinyatakan **BEBAS PLAGIASI** dan layak untuk diujiankan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 11 Juni 2024
Tim Verifikasi,

Zulfrianto Y. Lamasigi, M.Kom
NIDN. 0914089101

Mengetahui
Dekan,

Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

Terlampir :
Hasil Pengecekan Turnitin

LAMPIRAN 6 : DAFTAR RIWAYAT HIDUP

Nama : Moh. Rezaldy D. Kasili

Tempat Tanggal Lahir : Atinggola 12 Juli 2002

Pekerjaan : Mahasiswa

Email : vitokasili@gmail.com

Daftar Riwayat Pendidikan

1. Tahun 2014, menyelesaikan pendidikan sekolah dasar di SDN 7 Atinggola,kecamatan Atinggola, Kabupaten Gorontalo Utara
2. Tahun 2017, menyelesaikan pendidikan Sekolah Menengah Pertama Negeri1 Atinggola
3. Tahun 2020, meyelesaikan pendidikan Sekolah Menegah Atas Negeri 3Gorontalo Utara
4. Tahun 2020, Telah diterima menjadi Mahasiswa di perguruan tinggi swastaUnivesitas Ichsan Gorontalo