PENERAPAN METODE AFFINE CIPHER UNTUK PENINGKATAN KEAMANAN DOKUMEN DENGAN TEKNIK KONGRUENSI LINEAR

Disusun dan Diajukan untuk Memenuhi Persyaratan Ujian Akhir Memperoleh
Gelar Sarjana Komputer pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH:

NAMA	:	**RIDHO PRABOWO**
NPM	:	**1514370254**
PROGRAM STUDI	:	**SISTEM KOMPUTER**

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
MEDAN
2022

Halaman Pengesahan

PENGESAHAN TUGAS AKHIR

JUDUL : PENERAPAN METODE AFFINE CIPHER UNTUK PENINGKATAN KEAMANAN DOKUMEN DENGAN TEKNIK KONGRUENSI LINEAR

NAMA : RIDHO PRABOWO
N.P.M : 1514370254
FAKULTAS : SAINS & TEKNOLOGI
PROGRAM STUDI : Sistem Komputer
TANGGAL KELULUSAN : 08 Agustus 2022

DIKETAHUI :

DEKAN

Hamdani, ST., MT.

KETUA PROGRAM STUDI

Eko Hariyanto, S.Kom., M.Kom

DISETUJUI
KOMISI PEMBIMBING

PEMBIMBING I

Dr Zulham Sitorus, S.Kom., M.Kom.

PEMBIMBING II

Randi Rian Putra, S.Kom., M.Kom.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : RIDHO PRABOWO
NPM : 1514370254
Fakultas/Program Studi : Sains dan Teknologi / Sistem Komputer
Judul Skripsi : Penerapan Metode Affine Cipher Untuk Peningkatan Keamanan Dokumen Dengan Teknik Kongruensi Linear

Dengan ini menyatakan bahwa:

1. Skripsi ini merupakan hasil karya tulis saya sendiri dan bukan merupakan hasil karua orang lain (plagiat);
2. Memberikan ijin hak bebas Royalti Non-Eksklusif kepada Universitas Pembangunan Panca Budi untuk menyimpan, mengalih-media/formatkan, mengelola, mendistribusikan dan mempublikasikan karya skripsinya melalui internet atau media lain bagi kepentingan akademis.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai dengan aturan yang berlaku apabila di kemudian hari diketahui bahwa pernyataan ini tidak benar.

Medan, November 2022

RIDHO PRABOWO

NPM: 1514370254

PERNYATAAN ORISINALITAS

Dengan ini menyatakan bahwa dalam skripsi ini tidak terdapat karya yang diajukan untuk memperoleh gelar kesarjanaan di dalam perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis di acu dalam skripsi ini dan disebutkan dalam daftar pustaka.

Medan, November 2022

RIDHO PRABOWO

1514370254

ABSTRAK

RIDHO PRABOWO

Penerapan Metode Affine Cipher Untuk Peningkatan Keamanan Dokumen dengan Teknik Kongruensi Linear

2022

Dokumen merupakan arsip yang harus dijaga. Kerahasiaan dokumen merupakan hal yang paling penting dalam menjaga keaslian dokumen termasuk kepada file teks. Hal ini dilakukan agar tidak terjadi penyelewengan terhadap isi dokumen kepada orang lain dengan cara menyebarluaskan melalui media sosial. Kerahasiaan dokumen dapat dijaga dengan melakukan enkripsi pada dokumen tersebut. Tetapi kunci yang digunakan sering kali mudah ditebak sehingga dokumen dapat terbuka. Algoritma Affine Cipher dapat digunakan menggunakan kunci yang akan dibangkitkan dengan teknik kongruensi linear. Teknik ini membangkitkan kunci dari bilangan acak pada Affine cipher sehingga komposisi atau susunan kunci tidak mudah ditebak. Panjang kunci dapat ditentukan sesuai keinginan pengguna. Penerapan teknik ini akan mengamankan dokumen elektronik dengan baik.

Kata Kunci: dokumen, keamanan, enkripsi, kongurensi, linear, *Affine*

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, karena dengan berkat dan rahmat-Nya penulis masih diberikan kesempatan untuk menyelesaikan skripsi ini sebagaimana mestinya. Skripsi ini berjudul "**Penerapan Metode Affine Cipher Untuk Peningkatan Keamanan Dokumen dengan Teknik Kongruensi Linear**". Dalam kesempatan ini, penulis mengucapkan rasa terima kasih yang tak terhingga kepada pihak-pihak yang telah membantu dalam penyelesaian skripsi ini. Penulis ingin mengucapkan terima kasih kepada :

1. Orang tua saya yang selalu memberikan semangat, dukungan dan motivasi dalam penyusunan skripsi ini.
2. Bapak Dr. H. Muhammad Isa Indrawan, S.E., M.M., selaku Rektor Universitas Pembangunanan Panca Budi Medan.
3. Bapak Hamdani, S.T., M.T., selaku Dekan Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
4. Bapak Eko Hariyanto, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer Universitas Pembangunan Panca Budi Medan.
5. Bapak Dr. Zulham Sitorus, S.Kom., M.Kom., selaku Dosen Pembimbing I yang telah memberikan arahan dan membimbing dalam penyelesaian skripsi ini.
6. Bapak Randi Rian Putra, S.Kom., M.Kom., selaku Dosen Pembimbing II yang telah memberikan ilmu pengetahuan, serta bimbingan dalam penyelesaian skripsi ini.
7. Dosen-dosen pada Program Studi Sistem Komputer Fakultas Sains dan Teknologi Universitas Pembangunan Panca Budi Medan.
8. Staff dan karyawan pada Universitas Pembangunan Panca Budi Medan.
9. Seluruh teman-teman penulis dari program studi Sistem Komputer, Fakultas Sains dan Teknologi, Universitas Pembangunan Panca Budi, Medan
10. Abang kandung dan adik serta pacar saya Inez Denia Salvira, S.Pd dan juga para sahabat saya yang telah memberikan semangat kepada saya sehingga dapat menyelesaikan skripsi ini.

Penulis juga menyadari bahwa penyusunan skripsi ini belum mendapatkan kesempurnaan dalam segi penulisan ataupun isi. Hal ini disebabkan pengetahuan penulis yang sangat terbatas. Penulis sangat mengharapkan adanya kritik dan saran dari pembaca untuk dapat memperbaiki isi skripsi.

Medan, 12 Desember 2021
Penulis

Ridho Prabowo
1514370254

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan sehari tidak terluput dari komunikasi antar manusia. Dalam melakukan komunikasi terkadang tidak dapat dilakukan secara langsung atau secara tatap muka. Komunikasi sering kali dilakukan dari jarak jauh dengan bantuan suatu media penghantar. Komunikasi sering dilakukan dengan bantuan sosial media pengirim pesan. Dokumen elektronik merupakan informasi yang dikemas dalam bentuk file yang memiliki fungsi dan tujuan masing-masing. Pengiriman dokumen kadang mengalami kendala, terutama masalah keamanan. Dokumen yang dikirim yang berupa informasi yang rahasia memerlukan teknik pengiriman yang baik dan aman agar pesan tersebut terhindar dari serangan orang yang tidak diinginkan. Pengiriman pesan memerlukan teknik kriptografi agar pesan terhindar dari kebocoran. Kebocoran informasi akan mengakibatkan kerugian yang sangat besar bagi pemilik pesan tersebut.

Teknik kriptografi akan melakukan pengubahan dokumen tersebut menjadi file yang tidak dapat terbuka. Dokumen yang sudah terenkripsi akan bebas dikirimkan ke mana saja tanpa khawatir adanya pencurian pesan pada proses pengiriman pesan tersebut.

Permasalahan yang sering terjadi adalah kebocoran kunci pada saat melakukan proses enkripsi dan dekripsi sehingga teknik kriptografi dengan menggunakan kunci yang sama pada proses enkripsi dan dekripsi sering dapat

dipecahkan atau dibobol oleh orang lain (Mokhtari & Naraghi, 2012). Hal ini karena kedua kunci yang digunakan adalah sama sehingga mudah ditebak.

Dengan bantuan kombinasi dari metode kongruensi linear agar kunci yg digunakan pada Affine Cipher tidak mudah ditebak karena sudah dikombinasikan dari kongruensi linear sehingga keamanan lebih kuat dari pada menggunakan metode tunggal dari Affine Cipher tersebut.

Bilangan acak adalah sebuah bilangan yang dihasilkan dari sebuah proses matematika yang angkanya tidak dapat diterka dan tidak akan menghasilkan bilangan yang sama pada proses berikutnya. Proses pembangkitan bilangan acak menggunakan komputer disebut *pseudorandom number generator*. Pengujian kerandoman dilakukan bertujuan untuk menentukan apakah bilangan dihasilkan oleh sebuah generator termasuk random atau bukan (Tang, 2017).

Kongruensi linear yang mempunyai bentuk ax b (mod m). Hasil perhitungan akan di umpan balik untuk mendapatkan nilai berikutnya. Hal ini akan dilakukan seterusnya sehingga mendapatkan bilangan deret yang teracak. Nilai inilah yang akan dimasukkan ke dalam persamaan enkripsi dan dekripsi sehingga meningkatkan keamanan kunci sehingga kunci tersebut sulit untuk ditebak.

Penelitian ini menggunakan algoritma Affine Cipher yang memiliki kunci yang berbeda pada saat proses enkripsi dan dekripsi. Kunci yang digunakan adalah bilangan ganjil atau bilangan prima. Kunci enkripsi berbeda dengan kunci dekripsi sehingga memberikan kesulitan dalam melakukan penebakan kunci. Selain itu algoritma Affine Cipher memiliki pergeseran yang meningkatkan keamanan kunci.

Teknik pembangkitan kunci pada penelitian ini dilakukan dengan teknik kongruensi linear yaitu teknik dalam membangkitkan bilangan acak yang dapat memberikan peluang besar bagi pembangkitan kunci pada algoritma Affine Cipher.

Teknik kongruensi linier bekerja dengan cara membangkitkan deretan bilangan integer yang akan menentukan karakter kunci yang akan digunakan dalam proses enkripsi dan dekripsi (Aulia et al., 2019). Penelitian ini akan menggunakan teknik kongruensi linier dalam membangkitkan kunci sehingga dapat digunakan pada algoritma Affine Cipher. Proses kombinasi ini akan lebih meningkatkan keamanan pada proses pengiriman dokumen. Hasil proses kriptografi diharapkan dapat mengamankan dokumen sebelum dokumen tersebut dikirimkan. Berdasarkan latar belakang di atas maka penulis mengambil judul "**PENERAPAN METODE AFFINE CIPHER UNTUK PENINGKATAN KEAMANAN DOKUMEN DENGAN TEKNIK KONGRUENSI LINEAR**".

1.2 Rumusan Masalah

Adapun rumusan masalah yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

1. Bagaimana menciptakan aplikasi pengamanan dokumen elektronik?
2. Bagaimana memahami proses dan cara kerja teknik kongruensi linear dalam membangkitkan kunci?
3. Bagaimana menentukan kunci pada algoritma Affine Cipher?
4. Bagaimana melakukan proses enkripsi dan dekripsi dengan algoritma Affine Cipher?

1.3 Batasan Masalah

Adapun batasan masalah yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

1. Dokumen yang digunakan adalah berjenis .txt.
2. Besar kapasitas dokumen maksimal 10 *megabyte*.
3. Teknik kongruensi linear diterapkan pada pembangkitan bilangan acak pada kunci menggunakan *Linear Congruential Generator*.
4. Program dibuat menggunakan bahasa pemrograman Microsoft Visual Studio.
5. Program aplikasi berbasis desktop dan lokal.

1.4 Tujuan Penelitian

Adapun tujuan penelitian yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

1. Untuk menciptakan aplikasi pengamanan dokumen elektronik.
2. Untuk memahami proses dan cara kerja teknik kongruensi linier dalam membangkitkan kunci.
3. Untuk menentukan kunci pada algoritma Affine Cipher.
4. Untuk melakukan proses enkripsi dan dekripsi dengan algoritma Affine Cipher.

1.5 Manfaat Penelitian

Adapun manfaat penelitian yang digunakan dalam penulisan skripsi ini adalah sebagai berikut:

1. Isi dokumen aman dari gangguan.
2. Memberi kenyamanan dalam melakukan pengiriman dokumen elektronik.
3. Menambah pengetahuan tentang teknik kongruensi linier dan algoritma Affine Cipher.

BAB II

LANDASAN TEORI

2.1 Data

Data adalah karakteristik atau informasi, biasanya numerik, yang dikumpulkan melalui observasi. Dalam pengertian yang lebih teknis, data adalah seperangkat nilai variabel kualitatif atau kuantitatif tentang satu atau lebih orang atau objek, sedangkan datum (data tunggal) adalah nilai tunggal dari variabel tunggal (Tuomi, 2019).

Meskipun istilah "data" dan "informasi" sering digunakan secara bergantian, istilah-istilah ini memiliki arti yang berbeda. Dalam beberapa publikasi populer, data kadang-kadang dikatakan diubah menjadi informasi ketika dilihat dalam konteks atau pasca analisis. Namun, dalam perawatan akademik subjek, data hanyalah unit informasi. Data digunakan dalam penelitian ilmiah, manajemen bisnis (misalnya, data penjualan, pendapatan, laba, harga saham), keuangan, tata kelola (misalnya, tingkat kejahatan, tingkat pengangguran, tingkat melek huruf), dan dalam hampir setiap bentuk aktivitas organisasi manusia lainnya (misalnya, sensus jumlah tunawisma oleh organisasi nirlaba).

Data diukur, dikumpulkan dan dilaporkan, dan dianalisis, yang kemudian dapat divisualisasikan menggunakan grafik, gambar atau alat analisis lainnya. Data sebagai konsep umum mengacu pada fakta bahwa beberapa informasi atau pengetahuan yang ada diwakili atau dikodekan dalam beberapa bentuk yang cocok untuk penggunaan atau pemrosesan yang lebih baik. Data mentah ("data yang tidak

diproses") adalah kumpulan angka atau karakter sebelum "dibersihkan" dan dikoreksi oleh para peneliti. Data mentah perlu diperbaiki untuk menghapus outlier atau kesalahan instrumen atau entri data yang jelas (mis., Pembacaan termometer dari lokasi luar Arktik yang merekam suhu tropis). Pemrosesan data biasanya terjadi secara bertahap, dan "data yang diproses" dari satu tahap dapat dianggap sebagai "data mentah" dari tahap berikutnya. Data lapangan adalah data mentah yang dikumpulkan di lingkungan "in situ" yang tidak terkendali. Data eksperimental adalah data yang dihasilkan dalam konteks penyelidikan ilmiah dengan observasi dan rekaman (Barone et al., 2017).

2.1.1 Teknik Pengumpulan Data

Pengumpulan data didefinisikan sebagai "proses mengumpulkan dan mengukur informasi tentang variabel yang diminati, dengan cara sistematis yang memungkinkan seseorang untuk menjawab pertanyaan, menyatakan pertanyaan penelitian, menguji hipotesis, dan mengevaluasi hasil."

Ada banyak alasan untuk pengumpulan data, tetapi di sini lebih berfokus terutama pada yang terkait dengan bisnis dan pemasaran:

1 Ini membantu seseorang mempelajari lebih lanjut tentang pelanggan seseorang

2 Ini memungkinkan seseorang untuk menemukan tren dalam cara orang mengubah pendapat dan perilaku mereka dari waktu ke waktu atau dalam situasi yang berbeda

3 Ini memungkinkan seseorang mengelompokkan audiens seseorang ke dalam kelompok pelanggan yang berbeda dan mengarahkan berbagai strategi pemasaran di masing-masing kelompok berdasarkan kebutuhan masing-masing

4 Ini memfasilitasi pengambilan keputusan dan meningkatkan kualitas keputusan yang dibuat

5 Ini membantu menyelesaikan masalah dan meningkatkan kualitas produk atau layanan seseorang berdasarkan umpan balik yang diperoleh

Sebelum menyelami lebih dalam berbagai teknik dan metode pengumpulan data, ada perbedaan singkat antara dua jenis data utama yaitu kuantitatif dan kualitatif. Beberapa metode yang dibahas di sini adalah kuantitatif, berurusan dengan sesuatu yang dapat dihitung. Lainnya kualitatif, artinya mereka mempertimbangkan faktor selain nilai numerik. Secara umum, kuesioner, survei, dan dokumen serta catatan bersifat kuantitatif, sedangkan wawancara, kelompok fokus, observasi, dan sejarah lisan bersifat kualitatif. Bisa juga ada crossover antara kedua metode.

2.1.2 Data kuantitatif

Jenis data ini berkaitan dengan hal-hal yang dapat diukur dan dapat dinyatakan dalam angka atau angka, atau menggunakan nilai-nilai lain yang menyatakan kuantitas. Karena itu, data kuantitatif biasanya dinyatakan dalam bentuk angka dan dapat mewakili ukuran, panjang, durasi, jumlah, harga, dan

sebagainya. Penelitian kuantitatif kemungkinan besar memberikan jawaban atas pertanyaan seperti siapa? kapan? dimana? apa? dan berapa banyak? Pertanyaan survei kuantitatif dalam banyak kasus tertutup dan dibuat sesuai dengan tujuan penelitian, sehingga membuat jawaban mudah ditransformasikan menjadi angka, grafik, grafik, dan tabel.

Data yang diperoleh melalui metode pengumpulan data kuantitatif dapat digunakan untuk menguji ide atau prediksi yang ada, mempelajari tentang pelanggan seseorang, mengukur tren umum, dan menjadikannya penting. Misalnya, seseorang dapat menggunakannya untuk mengukur kesuksesan produk seseorang dan aspek mana yang mungkin perlu ditingkatkan, tingkat kepuasan pelanggan seseorang, untuk mengetahui apakah dan mengapa pesaing seseorang mengalahkan seseorang, dan sebagainya.

Karena metode pengumpulan data kuantitatif sering didasarkan pada perhitungan matematis, data yang diperoleh dengan cara itu biasanya dipandang lebih objektif dan dapat diandalkan daripada kualitatif. Beberapa teknik pengumpulan data kuantitatif yang paling umum termasuk survei dan kuesioner (dengan pertanyaan tertutup).

Dibandingkan dengan teknik kualitatif, metode kuantitatif biasanya lebih murah dan membutuhkan waktu lebih sedikit untuk mengumpulkan data dengan cara ini. Plus, karena tingkat standarisasi yang cukup tinggi, jauh lebih mudah untuk membandingkan dan menganalisis temuan yang diperoleh dengan menggunakan metode pengumpulan data kuantitatif (Jovancic, 2019).

2.1.3 Data kualitatif

Tidak seperti data kuantitatif, yang berkaitan dengan angka dan angka, data kualitatif lebih bersifat deskriptif daripada numerik. Data kualitatif biasanya tidak mudah diukur secara kuantitatif dan dapat diperoleh melalui observasi atau survei terbuka atau pertanyaan wawancara. Penelitian kualitatif kemungkinan besar akan memberikan jawaban atas pertanyaan seperti "mengapa?" dan bagaimana?". Seperti disebutkan, metode pengumpulan data kualitatif kemungkinan besar terdiri dari pertanyaan terbuka dan jawaban deskriptif dan sedikit atau tidak ada nilai numerik. Data kualitatif adalah cara terbaik untuk mendapatkan wawasan tentang pemikiran dan perilaku audiens (mungkin yang seseorang identifikasi menggunakan penelitian kuantitatif, tetapi tidak dapat menganalisis secara lebih rinci).

Data yang diperoleh dengan menggunakan metode pengumpulan data kualitatif dapat digunakan untuk menemukan ide-ide baru, peluang, dan masalah, menguji nilai dan keakuratannya, merumuskan prediksi, mengeksplorasi bidang tertentu secara lebih rinci, dan menjelaskan angka-angka yang diperoleh dengan menggunakan teknik pengumpulan data kuantitatif. Karena metode pengumpulan data kuantitatif biasanya tidak melibatkan angka dan perhitungan matematis tetapi lebih mementingkan kata-kata, bunyi, pikiran, perasaan, dan data yang tidak terukur lainnya, data kualitatif sering dianggap lebih subyektif, tetapi pada saat yang sama memungkinkan pemahaman yang lebih dalam. Beberapa teknik pengumpulan data kualitatif yang paling umum termasuk survei terbuka dan kuesioner, wawancara, kelompok fokus, observasi, studi kasus, dan sebagainya (Jovancic, 2019).

2.2 Keamanan Data

Teknologi digital sekarang hanya bagian dari kehidupan. Dari belanja online hingga perbankan bersih dan bisnis hingga infrastruktur pemerintah, teknologi digital memainkan peran penting. Terlepas dari berbagai keuntungan digitalisasi, serangan dunia maya adalah titik hitam. Dalam beberapa tahun terakhir, kami telah menyaksikan banyak serangan dunia maya tingkat tinggi. Bahkan, kita dapat mengatakan bahwa jumlah serangan siber telah meningkat pesat dalam beberapa tahun terakhir (Barot, 2018).

2.2.1 Pengertian Keamanan Data

Sederhananya, keamanan data adalah praktik pengamanan data seseorang. Ini juga dikenal sebagai keamanan informasi, Keamanan TI, atau keamanan informasi elektronik. Data dapat diamankan menggunakan berbagai teknologi perangkat keras dan perangkat lunak. Beberapa alat umum adalah antivirus, enkripsi, firewall, otentikasi dua faktor, tambalan perangkat lunak, pembaruan, dll.

2.2.2 Pentingnya Keamanan Data

Banyak orang memiliki kesalahpahaman umum bahwa hanya organisasi besar, pemerintah, dan bisnis yang menjadi target pelaku cyber. Ya, ini tidak benar. Keamanan data tidak hanya penting untuk bisnis atau pemerintah. Komputer, tablet, dan perangkat seluler seseorang bisa menjadi target selanjutnya. Biasanya, pengguna biasa menjadi sasaran penyerang karena informasi sensitif mereka, seperti detail kartu kredit, detail perbankan, kata sandi, dll (Rao & Selvamani, 2015).

Keamanan dunia maya harus menyeluruh dan mulus untuk semua orang - apakah seseorang seorang individu atau bisnis. Menurut perkiraan oleh Pusat Studi Strategis dan Internasional, kejahatan dunia maya merugikan ekonomi global lebih dari 400 miliar USD per tahun. Tidak perlu dikatakan, pelanggaran data dan serangan cyber akan meningkat pada waktunya karena jaringan komputer berkembang - serangan cyber semakin besar dan semakin baik setiap hari.

2.2.3 Dampak Penyerangan Terhadap Data

Kejadian penyalahgunaan data tidak ingin menakut-nakuti seseorang atau apa pun, tetapi ada banyak cara di mana seseorang dapat terpengaruh. Cara-cara ini termasuk serangan phishing, serangan malware, serangan ransomware, serangan man-in-the-middle, dll. Ingat, kesadaran seseorang adalah keamanan seseorang. Di sini, saya membagikan praktik penting yang perlu seseorang mulai hari ini untuk melindungi diri dari peretas:

1 Jangan pernah mengklik spam, phishing, atau email yang mencurigakan. Verifikasi atau periksa email atau tautan dengan cermat sebelum membuka lampiran apa pun.

2 Jika sesuatu tampak terlalu bagus untuk menjadi kenyataan, mungkin itu benar. Jangan menjadi korban penawaran, seperti "iPhone X hanya dengan $ 10" atau "Selamat! seseorang memenangkan mobil. Buka lampiran untuk mengklaim sekarang."

3 Jangan pernah mengunduh perangkat lunak atau aplikasi yang tidak tepercaya atau bajakan.

4 Jangan mengunduh perangkat lunak keamanan palsu.

5 Gunakan antivirus dan / atau firewall

6 Jangan melakukan transaksi online jika situs web tidak diamankan. Periksa HTTPS atau bilah alamat hijau sebelum melakukan pembayaran atau mengetikkan detail sensitif apa pun

7 Gunakan otentikasi dua faktor.

8 Jangan membagikan informasi pribadi atau sensitif seseorang kepada orang asing.

2.2.4 Hubungan Keamanan Data Dengan Bisnis

Informasi dan data dalam bisnis seseorang adalah aset bisnis yang berharga. Ini bisa menjadi kunci pertumbuhan dan kesuksesan. Keamanan data seseorang, oleh karena itu, harus menjadi prioritas dalam bisnis seseorang. Itu perlu dilindungi dari akses tidak sah untuk mencegahnya dirusak, dihancurkan atau diungkapkan kepada orang lain. Keamanan dapat dilanggar dalam sejumlah cara, misalnya oleh kegagalan sistem, pencurian, penggunaan yang tidak tepat, akses tidak sah atau virus komputer. Setiap kali seseorang terlibat dalam apa pun yang melibatkan Internet, keamanan data seseorang berisiko. Praktik kerja modern seperti kerja jarak jauh, perangkat IT portabel dan Wi-Fi semuanya meningkatkan ancaman terhadap keamanan data. Bahkan jika seseorang bekerja sendirian dari perangkat berbasis meja tunggal seseorang masih berisiko.

Efek dari pelanggaran keamanan data bisa menjadi bencana besar. Tidak hanya dalam hal gangguan pada operasi bisnis seseorang, tetapi juga potensi

kerusakan jangka panjang pada reputasi seseorang. seseorang mungkin telah menghabiskan beberapa tahun membangun merek dan reputasi seseorang untuk dihancurkan hanya dalam beberapa jam. Ada banyak cara untuk memastikan keamanan data - mulai dari pendidikan staf seseorang hingga solusi perangkat lunak dan perangkat keras. Tidak ada metode tunggal yang berdiri sendiri akan menawarkan solusi keamanan data yang lengkap sehingga penting untuk memahami di mana kerentanan seseorang dan melindungi diri seseorang sendiri.

Keamanan data adalah seperangkat alat dan teknologi yang melindungi data dari kehancuran, modifikasi, atau pengungkapan yang disengaja atau tidak disengaja. Keamanan data dapat diterapkan dengan menggunakan berbagai teknik dan teknologi, termasuk kontrol administratif, keamanan fisik, kontrol logis, organisasi, dan teknik perlindungan lainnya yang membatasi akses ke pengguna atau proses yang tidak sah atau berbahaya (Rao & Selvamani, 2015).

Keamanan data mengacu pada langkah-langkah privasi digital pelindung yang diterapkan untuk mencegah akses tidak sah ke komputer, database, dan situs web. Keamanan data juga melindungi data dari korupsi. Keamanan data adalah aspek penting dari TI untuk organisasi dari berbagai ukuran dan tipe. Keamanan data juga dikenal sebagai keamanan informasi atau keamanan komputer.

Contoh teknologi keamanan data termasuk backup, masking data dan penghapusan data. Ukuran teknologi keamanan data utama adalah enkripsi, di mana data digital, perangkat lunak / perangkat keras, dan hard drive dienkripsi dan karenanya tidak dapat dibaca oleh pengguna dan peretas yang tidak sah. Salah satu metode yang paling umum dijumpai dalam mempraktikkan keamanan data adalah

penggunaan otentikasi. Dengan otentikasi, pengguna harus memberikan kata sandi, kode, data biometrik, atau bentuk data lainnya untuk memverifikasi identitas sebelum akses ke sistem atau data diberikan. Keamanan data juga sangat penting untuk catatan perawatan kesehatan, sehingga pendukung kesehatan dan praktisi medis di AS dan negara-negara lain berupaya menerapkan privasi rekam medis elektronik dengan menciptakan kesadaran tentang hak-hak pasien terkait dengan pelepasan data ke laboratorium, dokter, rumah sakit dan fasilitas medis lainnya.

2.2.5 Solusi Keamanan Data

Data membutuhkan enkripsi dalam mengamankan informasi yang ada dalam data tersebut. Dengan enkripsi data canggih, tokenization, dan manajemen utama untuk melindungi data di seluruh aplikasi, transaksi, penyimpanan, dan platform big data, Teknik ini menyederhanakan perlindungan data sensitif bahkan dalam kasus penggunaan yang paling kompleks sekalipun. Beberapa model keamanan data antara lain:

1. Keamanan akses cloud - Platform perlindungan yang memungkinkan seseorang untuk pindah ke cloud dengan aman sambil melindungi data dalam aplikasi cloud.
2. Enkripsi data - Solusi keamanan data-sentris dan tokenisasi yang melindungi data di lingkungan perusahaan, cloud, seluler, dan data besar.
3. Modul keamanan perangkat keras - Modul keamanan perangkat keras yang menjaga data keuangan dan memenuhi persyaratan keamanan dan kepatuhan industri.

4. Manajemen kunci - Solusi yang melindungi data dan memungkinkan kepatuhan regulasi industri.
5. Enterprise Data Protection - Solusi yang menyediakan pendekatan data-centric end-to-end untuk perlindungan data perusahaan.
6. Keamanan Pembayaran - Solusi menyediakan enkripsi dan tokenisasi point-to-point lengkap untuk transaksi pembayaran ritel, memungkinkan pengurangan lingkup PCI.
7. Big Data, Hadoop, dan perlindungan data IofT - Solusi yang melindungi data sensitif di Danau Data - termasuk Hadoop, Teradata, Micro Focus Vertica, dan platform Big Data lainnya.
8. Keamanan Aplikasi Seluler - Melindungi data sensitif di aplikasi seluler asli sembari menjaga data dari ujung ke ujung.
9. Keamanan Peramban Web - Melindungi data sensitif yang diambil di peramban, dari titik pelanggan memasukkan pemegang kartu atau data pribadi dan menjaganya agar tetap terlindungi melalui ekosistem ke tujuan tuan rumah tepercaya.
10. eMail Security - Solusi yang menyediakan enkripsi ujung ke ujung untuk email dan olahpesan seluler, menjaga informasi pribadi dan informasi kesehatan pribadi tetap aman dan pribadi.

2.2.6 Kerahasiaan

Kerahasiaan mengacu pada melindungi informasi agar tidak diakses oleh pihak yang tidak berwenang. Dengan kata lain, hanya orang yang diberi wewenang

untuk melakukannya yang dapat memperoleh akses ke data sensitif. Bayangkan catatan bank harus dapat diakses, tentu saja, dan karyawan di bank yang membantu dalam menjalankan transaksi harus dapat mengaksesnya, tetapi tidak ada orang lain yang seharusnya. Kegagalan untuk menjaga kerahasiaan berarti bahwa seseorang yang seharusnya tidak memiliki akses telah berhasil mendapatkannya, melalui perilaku yang disengaja atau karena kecelakaan. Kegagalan kerahasiaan seperti itu, umumnya dikenal sebagai pelanggaran, biasanya tidak dapat diperbaiki. Setelah rahasia itu terungkap, tidak ada cara untuk mengetahuinya. Jika catatan bank diposting di situs web publik, semua orang dapat mengetahui nomor rekening bank, saldo, dll., Informasi itu tidak dapat dihapus dari pikiran, kertas, komputer, dan tempat lain mereka. Hampir semua insiden keamanan utama yang dilaporkan di media saat ini melibatkan kerugian besar kerahasiaan. Jadi, secara ringkas, pelanggaran kerahasiaan berarti bahwa seseorang memperoleh akses ke informasi yang seharusnya tidak memiliki akses ke sana.

2.2.7 Integritas

Integritas mengacu pada memastikan keaslian informasi — bahwa informasi tidak diubah, dan bahwa sumber informasi itu asli. Bayangkan jika seseorang memiliki situs web dan seseorang menjual produk di situs itu. Sekarang bayangkan penyerang dapat berbelanja di situs web dan dengan jahat mengubah harga produk seseorang sehingga mereka dapat membeli apa pun dengan harga berapa pun yang mereka pilih. Itu akan menjadi kegagalan integritas karena informasi dalam hal ini, harga suatu produk telah diubah dan perubahan ini tidak

dapat digagalkan. Contoh lain dari kegagalan integritas adalah ketika sesorang mencoba terhubung ke situs web dan penyerang jahat antara seseorang dan situs web mengalihkan lalu lintas ke situs web yang berbeda. Dalam hal ini, situs yang dituju tidak asli.

2.2.8 Ketersediaan

Ketersediaan berarti informasi dapat diakses oleh pengguna yang berwenang. Jika penyerang tidak dapat mengkompromikan dua elemen pertama dari keamanan informasi (lihat di atas) mereka dapat mencoba melakukan serangan seperti penolakan layanan yang akan menurunkan server, membuat situs web tidak tersedia untuk pengguna yang sah karena kurangnya ketersediaan.

2.2.9 Kontrol Akses

Kesalahan terbesar yang bisa dilakukan oleh perancang aplikasi adalah mengabaikan kontrol akses sebagai bagian dari fungsionalitas yang diperlukan. Jarang bahwa setiap pengguna atau sistem yang berinteraksi dengan suatu aplikasi harus memiliki hak yang sama di seluruh aplikasi itu. Beberapa pengguna mungkin memerlukan akses ke data tertentu dan bukan yang lain; beberapa sistem harus atau tidak dapat mengakses aplikasi. Akses ke komponen, fungsi, atau modul tertentu dalam aplikasi juga harus dikontrol. Kontrol akses juga penting untuk kepatuhan audit dan peraturan. Beberapa cara umum mengelola kontrol akses adalah:

1. Baca, tulis, dan jalankan hak istimewa: File
2. Kontrol akses berbasis peran: administrator, pengguna

3. Alamat IP akses berbasis host, nama mesin
4. Objek kode kontrol akses tingkat objek, banyak pembaca / penulis tunggal

2.3 Algoritma

Pertimbangkan bagaimana seseorang menggunakan komputer pada hari-hari biasa. Misalnya, seseorang mulai mengerjakan laporan, dan setelah seseorang menyelesaikan paragraf, seseorang melakukan pemeriksaan ejaan. seseorang membuka aplikasi spreadsheet untuk melakukan beberapa proyeksi keuangan untuk melihat apakah seseorang dapat membeli pinjaman mobil baru. Seseorang menggunakan browser web untuk mencari secara online jenis mobil yang ingin seseorang beli (Gurevich, 2012).

seseorang mungkin tidak memikirkan hal ini dengan sangat sadar, tetapi semua operasi yang dilakukan oleh komputer seseorang terdiri dari algoritma. Algoritma adalah prosedur yang didefinisikan dengan baik yang memungkinkan komputer untuk memecahkan masalah. Cara lain untuk menggambarkan suatu algoritma adalah urutan instruksi yang tidak ambigu. Penggunaan istilah 'tidak ambigu' menunjukkan bahwa tidak ada ruang untuk interpretasi subyektif. Setiap kali seseorang meminta komputer seseorang untuk melakukan algoritma yang sama, ia akan melakukannya dengan cara yang persis sama dengan hasil yang sama persis.

Pertimbangkan contoh-contoh sebelumnya lagi. Pengecekan ejaan menggunakan algoritma. Perhitungan keuangan menggunakan algoritma. Mesin

pencari menggunakan algoritma. Bahkan, sulit untuk memikirkan tugas yang dilakukan oleh komputer seseorang yang tidak menggunakan algoritma.

Contoh algoritma yang sangat sederhana adalah menemukan angka terbesar dalam daftar angka yang tidak disortir. Jika Anda diberi daftar lima nomor yang berbeda, Anda akan dapat memecahkannya dalam waktu singkat, tidak perlu komputer. Sekarang, bagaimana dengan lima juta angka yang berbeda? Jelas, Anda akan membutuhkan komputer untuk melakukan ini, dan komputer membutuhkan algoritma.

Berikut ini adalah bagaimana algoritma itu terlihat. Katakanlah input terdiri dari daftar angka, dan daftar ini disebut L. Angka L1 akan menjadi angka pertama dalam daftar, L2 angka kedua, dll. Dan kita tahu daftar tidak diurutkan - jika tidak, jawabannya akan sangat mudah. Jadi, input ke algoritma adalah daftar angka, dan output harus menjadi angka terbesar dalam daftar.

Algoritma akan terlihat seperti ini:

Langkah 1: Biarkan Terbesar = L1

Ini berarti Anda mulai dengan mengasumsikan bahwa angka pertama adalah angka terbesar.

Langkah 2: Untuk setiap item dalam daftar:

Ini berarti Anda akan melalui daftar angka satu per satu.

Langkah 3: Jika item> Terbesar:

Jika Anda menemukan angka terbesar baru, lanjutkan ke langkah empat. Jika tidak, kembali ke langkah kedua, yang berarti Anda beralih ke nomor berikutnya dalam daftar.

Langkah 4: Kemudian Terbesar = item

Ini menggantikan angka terbesar lama dengan jumlah terbesar baru yang baru saja Anda temukan. Setelah ini selesai, kembali ke langkah dua hingga tidak ada lagi angka yang tersisa dalam daftar.

Langkah 5: Kembalikan Terbesar

Ini menghasilkan hasil yang diinginkan.

Perhatikan bahwa algoritma dijelaskan sebagai serangkaian langkah logis dalam bahasa yang mudah dipahami. Agar komputer dapat benar-benar menggunakan instruksi ini, mereka harus ditulis dalam bahasa yang dapat dimengerti oleh komputer, yang dikenal sebagai bahasa pemrograman (Zandbergen, 2019).

2.3.1 Desain Konseptual

Algoritma adalah serangkaian instruksi, sering disebut sebagai "proses," yang harus diikuti ketika memecahkan masalah tertentu. Meskipun secara teknis tidak dibatasi oleh definisi, kata itu hampir selalu terkait dengan komputer, karena algoritma yang diproses komputer dapat mengatasi masalah yang jauh lebih besar daripada manusia, jauh lebih cepat. Karena komputasi modern menggunakan algoritma jauh lebih sering daripada pada titik lain dalam sejarah manusia, bidang telah tumbuh di sekitar desain, analisis, dan penyempurnaan. Bidang desain algoritma membutuhkan latar belakang matematika yang kuat, dengan gelar ilmu komputer yang sangat dicari kualifikasi. Ini menawarkan semakin banyak pilihan karir yang sangat dikompensasi, karena kebutuhan akan lebih banyak (dan juga lebih canggih) algoritma terus meningkat.

Pada tingkat yang paling sederhana, algoritma pada dasarnya hanya seperangkat instruksi yang diperlukan untuk menyelesaikan tugas. Pengembangan algoritma, meskipun umumnya tidak disebut demikian, telah menjadi kebiasaan yang populer dan pengejaran profesional untuk semua catatan sejarah. Jauh sebelum fajar era komputer modern, orang menetapkan rutinitas yang telah ditentukan untuk bagaimana mereka akan melakukan tugas sehari-hari, sering menuliskan daftar langkah-langkah yang harus diambil untuk mencapai tujuan penting, mengurangi risiko melupakan sesuatu yang penting. Ini, pada dasarnya, adalah apa itu algoritma. Desainer mengambil pendekatan yang mirip dengan pengembangan algoritma untuk tujuan komputasi: pertama, mereka melihat masalah. Kemudian, mereka menguraikan langkah-langkah yang akan diperlukan

untuk menyelesaikannya. Akhirnya, mereka mengembangkan serangkaian operasi matematika untuk mencapai langkah-langkah tersebut.

2.3.2 Tugas Algoritma

Tugas sederhana dapat diselesaikan dengan algoritma yang dihasilkan dengan beberapa menit, atau paling banyak pekerjaan pagi. Tingkat kompleksitas menjalankan tantangan yang panjang, namun, sampai pada masalah yang sangat rumit sehingga mereka telah menghalangi matematikawan yang tak terhitung jumlahnya selama bertahun-tahun - atau bahkan berabad-abad. Komputer modern menghadapi masalah pada tingkat ini di bidang-bidang seperti keamanan dunia maya, serta penanganan data besar - penyortiran set data yang efisien dan menyeluruh sedemikian besar sehingga bahkan komputer tidak dapat memprosesnya secara tepat waktu. Contoh data besar mungkin termasuk "setiap artikel di Wikipedia," "setiap halaman web yang diindeks dan diarsipkan akan kembali ke tahun 1998," atau "enam bulan terakhir pembelian online yang dilakukan di Amerika."

2.3.3 Rekayasa Algoritma

Ketika desain algoritma baru diterapkan dalam istilah praktis, disiplin terkait dikenal sebagai rekayasa algoritma. Kedua fungsi tersebut sering dilakukan oleh orang yang sama, meskipun organisasi yang lebih besar (seperti Amazon dan Google) mempekerjakan desainer dan insinyur khusus, mengingat tingkat kebutuhan mereka akan algoritma baru dan khusus. Seperti proses desain, rekayasa

algoritma sering kali melibatkan akreditasi sains komputer, dengan latar belakang yang kuat dalam matematika: di mana mereka ada sebagai profesi yang terpisah dan terspesialisasi, insinyur algoritma mengambil ide-ide konseptual dari desainer dan proses kreatif dari mereka yang akan dipahami oleh komputer. Dengan kemajuan teknologi digital yang mantap, para insinyur yang berdedikasi akan terus menjadi semakin umum.

2.4 Kriptografi

Kriptografi adalah teknik mengubah dan mentransmisikan data rahasia dengan cara disandikan sehingga hanya pengguna yang berwenang dan dimaksudkan dapat memperoleh atau bekerja di dalamnya. Ini adalah kata asal Yunani di mana "crypto" berarti tersembunyi dan "graphy" berarti menulis, jadi kriptografi berarti tulisan tersembunyi atau rahasia. Ini memperkenalkan triad seperti kerahasiaan, non-penolakan, integritas dan keaslian dalam komunikasi data yang sedang berlangsung.

Kriptografi adalah disiplin atau teknik yang digunakan dalam melindungi integritas atau kerahasiaan pesan elektronik dengan mengubahnya menjadi bentuk (ciphertext) yang tidak dapat dibaca. Hanya penggunaan kunci rahasia yang dapat mengubah teks sandi menjadi bentuk yang dapat dibaca manusia (teks jelas). Perangkat lunak kriptografi dan / atau perangkat keras menggunakan rumus matematika (algoritma) untuk mengubah teks dari satu bentuk ke bentuk lainnya.

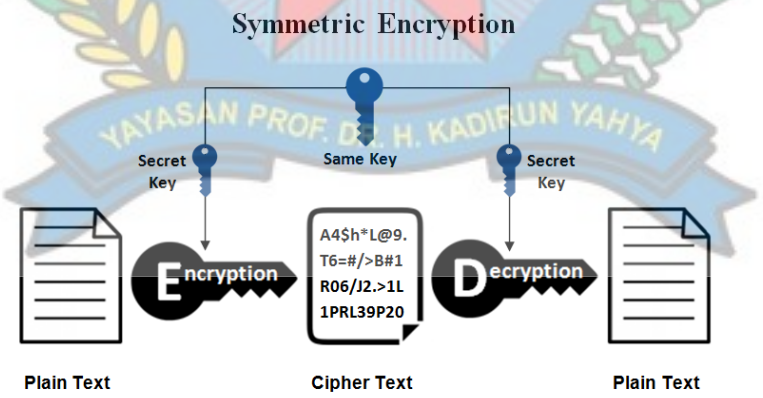
Komunikasi yang aman dapat disediakan menggunakan teknik, di hadapan konten pihak ketiga berbahaya yang disebut musuh. Teknik-teknik ini dapat disebut

sebagai Kriptografi. Pesan pribadi apa pun dapat disembunyikan dari publik atau pihak ketiga, menggunakan seperangkat protokol. Protokol-protokol ini perlu dianalisis dan dibangun dengan cara yang efisien untuk menjaga kerahasiaan pesan yang dikirim. Kriptografi modern memiliki aspek tertentu yang merupakan pusatnya seperti integritas data, otentikasi, kerahasiaan dll. Di dunia modern, kriptografi sangat bergantung pada mata pelajaran seperti matematika dan ilmu komputer. Algoritma untuk Kriptografi dirancang sedemikian rupa sehingga sulit untuk dipecahkan dalam praktik oleh pihak ketiga jahat yang juga dikenal sebagai musuh. Pendekatan praktis terhadap pemecahan algoritma semacam itu akan gagal, namun, pendekatan teoritis mungkin memecahkan sistem tersebut. Dengan demikian, algoritma apa pun dapat disebut sebagai aman, jika sifat kuncinya tidak dapat disimpulkan, dengan ciphertext yang diberikan. Kriptografi dapat dikategorikan menjadi dua cabang: Symmetric dan Asymmetric. Dengan pendekatan simetris, satu kunci digunakan untuk proses enkripsi dan dekripsi yaitu pengirim dan penerima harus memiliki kunci bersama. Namun, dengan pendekatan ini, distribusi kunci adalah tautan yang lemah, yang memunculkan pendekatan baru.

2.4.1 Kriptografi Simetris

Kriptografi kunci simetris adalah setiap algoritma kriptografi yang didasarkan pada kunci bersama yang digunakan untuk mengenkripsi atau mendekripsi teks / cyphertext, dalam kontrak dengan kriptografi kunci asimetris, di mana kunci enkripsi dan dekripsi dihubungkan oleh berbeda. Enkripsi simetris umumnya lebih efisien daripada enkripsi asimetris dan karenanya lebih disukai

ketika sejumlah besar data perlu dipertukarkan. Membuat kunci bersama sulit menggunakan hanya algoritma enkripsi simetris, sehingga dalam banyak kasus, enkripsi asimetris digunakan untuk membuat kunci bersama antara dua pihak. Contoh untuk kriptografi kunci simetris termasuk AES, DES, dan 3DES. Protokol pertukaran kunci yang digunakan untuk membangun kunci enkripsi bersama termasuk Diffie-Hellman (DH), Eliptic Curve (EC) dan RSA. Berikut ini skema dari kriptografi simetris (Ayushi, 2010).

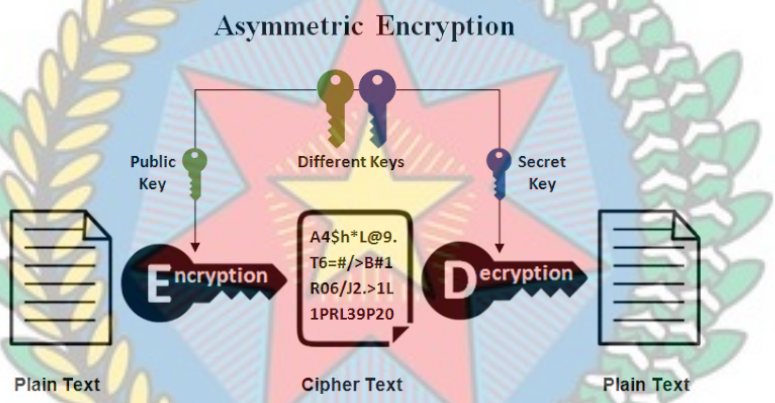


Gambar 2.1 Skema kriptografi simetris
Sumber: (Ayushi, 2010)

2.4.2 Kriptografi Asimetris

Dalam versi kriptografi asimetris, pengirim dan penerima memiliki dua kunci, publik dan pribadi. Kunci pribadi dirahasiakan sedangkan kunci publik terbuka ke dunia luar. Set data apa pun, yang dienkripsi dengan kunci publik hanya dapat didekripsi menggunakan kunci pribadi yang sesuai. Ketika datang ke perbandingan, pendekatan simetris lebih cepat daripada yang asimetris. Contoh - tseseorang tangan digital menggunakan kriptografi asimetris untuk mengenkripsi

pesan dalam hash alih-alih pesan lengkap. Berikut ini skema kriptografi asimetris (S. et al., 2012).



Gambar 2.2 Skema kriptografi asimetris
Sumber: (Ayushi, 2010)

2.5 *Affine Cipher*

Algoritma Affine Cipher merupakan bagian dari kriptografi klasik yang merupakan metode substitusi dalam pengamanan data, yaitu melakukan pergeseran plaintext dengan mengalikan nilai plaintext dengan kunci bernilai bilangan prima. Kelemahan Algoritma Affine Cipher adalah kunci yang mudah digunakan. Affine Cipher ialah sebuah bagian cipher monoalphabetic yang mana huruf-huruf dalam alphabetic dibuat ke dalam bentuk numerik, kemudian dienkripsi dengan fungsi suatu matematika sangat sederhana, dan diubah kembali dalam bentuk karakter kata. Algoritma Affine Cipher ialah algoritma yang file atau teks awal yang dikirim berubah dimana awalnya dimengerti maknanya oleh manusia menjadi file atau teks yang terenkripsi kemudian dimasukkan ke dalam media penyimpanan. Affine cipher dalam algoritma affine merupakan pengembangan algoritma Caesar Cipher

dimana plaintext (P) dikalikan nilai (b) kemudian ditambahkan ke pergeseran (k) (Nasution, 2020).

2.6 Metode Kongruensi Linear

Metode kongruensi linear (*linear congruent method*), dapat disingkat dengan LCM) merupakan algoritma yang menghasilkan barisan bilangan acak semu lewat persamaan linear bagian-demi-bagian. Metode ini juga dikenal dengan metode kongruen linear, pembangkit kongruensial linear dan generator kongruensial linear (*linear congruential generator, LCG*). Metode ini termasuk algoritma yang tertua dan terkenal untuk membangkitkan bilangan acak semu. Konsep metode ini relatif mudah dipahami, mudah diimplementasikan, dan memiliki waktu eksekusi yang cepat, khususnya untuk perangkat keras komputer yang mendukung aritmetika modular dengan pemotongan pada bit-bit penyimpanan. LCM memanfaatkan relasi rekursif linear:

$$X_{n+1} = (aX_n + c) \mod m$$

Terdapat 50 soal pada sebuah sistem database ujian. Untuk setiap ujian, soal akan dipilih secara acak dari database. Untuk mengusahakan tidak terjadi repetisi soal-soal yang telah dikerjakan, sistem memilih soal "baru" dengan menggunakan LCM; dengan konstanta a=11, c=7, m=50 dan X_0=1. Hasil deretnya adalah sebagai berikut:

X1 = (11 (1) + 7) mod 50 = 18

X2 = (11(18) + 7) mod 50 = 5

X3 = (11 (15) + 7) mod 50 = 12

2.7 *Unified Modeling Language (UML)*

Unified Modeling Language (UML) adalah bahasa pemodelan yang memungkinkan pengembang menentukan, memvisualisasikan, membuat, dan mendokumentasikan artefak sistem perangkat lunak (Technopedia, 2019). Dengan demikian, UML membuat artefak ini dapat diskalakan, aman, dan kuat dalam eksekusi. UML adalah aspek penting yang terlibat dalam pengembangan perangkat lunak berorientasi objek. Ini menggunakan notasi grafis untuk membuat model visual dari sistem perangkat lunak. Arsitektur UML didasarkan pada fasilitas meta-objek, yang mendefinisikan dasar untuk membuat bahasa pemodelan. Mereka cukup tepat untuk menghasilkan seluruh aplikasi. UML yang sepenuhnya dapat dieksekusi dapat digunakan untuk berbagai platform menggunakan teknologi yang berbeda dan dapat digunakan dengan semua proses sepanjang siklus pengembangan perangkat lunak. UML dirancang untuk memungkinkan pengguna mengembangkan bahasa pemodelan visual yang ekspresif, siap pakai. Selain itu, mendukung konsep pengembangan tingkat tinggi seperti kerangka kerja, pola, dan kolaborasi (Wasserkrug et al., 2019).

Penggunaan model ini bertujuan untuk mengidentifikasikan bagian-bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain diluarnya (Sukmawati & Priyadi, 2019).

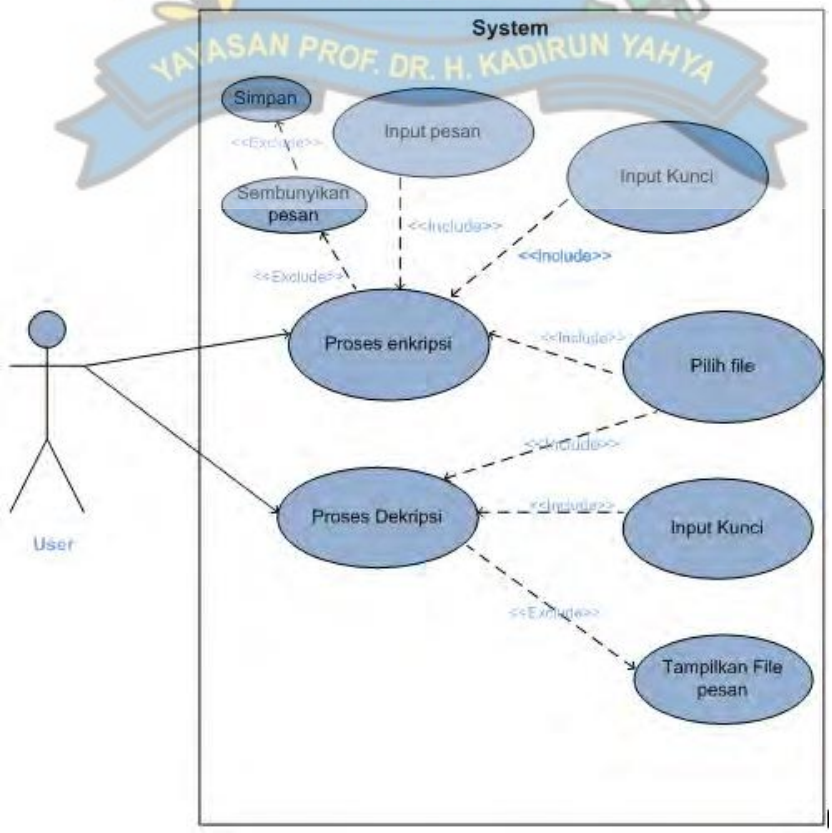
2.7.1 *Use Case Diagram*

Use Case Diagram adalah model tentang bagaimana berbagai jenis pengguna berinteraksi dengan sistem untuk memecahkan masalah. Dengan demikian, ini menggambarkan tujuan pengguna, interaksi antara pengguna dan sistem, dan perilaku sistem yang diperlukan dalam memenuhi tujuan-tujuan ini. *Use Case Diagram* terdiri dari sejumlah elemen model. Elemen model yang paling penting adalah kasus penggunaan, aktor dan hubungan di antara mereka. *Use Case Diagram* digunakan untuk menggambarkan secara grafis subset dari model untuk menyederhanakan komunikasi. Biasanya akan ada beberapa diagram kasus penggunaan yang terkait dengan model yang diberikan, masing-masing menunjukkan subset elemen model yang relevan untuk tujuan tertentu. Elemen model yang sama dapat ditampilkan pada beberapa *Use Case Diagram*, tetapi setiap instance harus konsisten. Jika alat digunakan untuk mempertahankan *Use Case Diagram*, kendala konsistensi ini otomatis sehingga setiap perubahan pada elemen model (mengubah nama misalnya) akan secara otomatis tercermin dalam setiap *Use Case Diagram* yang menunjukkan elemen itu (UTM, 2019).

Use Case Diagram dapat berisi paket yang digunakan untuk menyusun model untuk menyederhanakan analisis, komunikasi, navigasi, pengembangan, pemeliharaan, dan perencanaan. Faktanya, sebagian besar *Use Case Diagram* adalah tekstual, dengan teks yang ditangkap dalam Spesifikasi Use-Case yang terkait dengan setiap elemen *Use Case Diagram*. Spesifikasi ini menjelaskan alur peristiwa *Use Case*. *Use Case Diagram* berfungsi sebagai utas pemersatu sepanjang pengembangan sistem. Ini digunakan sebagai spesifikasi utama dari persyaratan

fungsional untuk sistem, sebagai dasar untuk analisis dan desain, sebagai input untuk perencanaan iterasi, sebagai dasar mendefinisikan kasus uji dan sebagai dasar untuk dokumentasi pengguna. (Kurniawan, 2018).

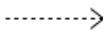
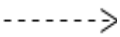
Use Case Diagram merupakan suatu diagram yang berisi *Use Case*, *actor*, serta *relationship* diantaranya. *Use Case Diagram* dapat digunakan untuk kebutuhan apa saja yang diperlukan dalam suatu sistem, sehingga sistem dapat digambarkan dengan jelas bagaimana proses dari sistem tersebut, bagaimana cara aktor menggunakan sistem, serta apa saja yang dapat dilakukan pada suatu sistem.

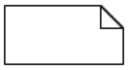


Gambar 2.3 Use Case Diagram Enkripsi dan Dekripsi
Sumber: (Nurgoho, 2019)

Gambar 2.4 adalah contoh dari diagram pada proses enkripsi dan dekripsi. Use-case memiliki beberapa simbol untuk menyatakan kegiatan dari use-case tersebut. Adapun simbol dari *Use Case* adalah sebagai berikut:

Tabel 2.1 Simbol *Use Case Diagram*

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *Use Case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
3		*Generalization*	Hubungan dimana objek anak berbagi perilaku dan struktur data dari objek yang ada diatasnya.
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 actor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

Sumber: (Kurniawan, 2018)

2.7.2 *Activity Diagram*

Activity Diagram (Diagram Aktifitas) menggambarkan berbagai alir aktifitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir

berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir (Ladjamudin, 2017).

Activity Diagram menurut adalah salah satu cara untuk memodelkan *event-event* yang terjadi dalam suatu *Use Case*. Diagram ini juga dapat digantikan dengan sejumlah teks.

Tabel 2.2 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 /diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

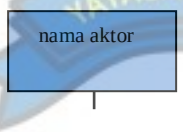
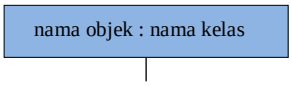
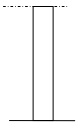
Sumber: (Kurniawan, 2018)

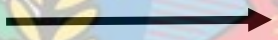
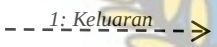
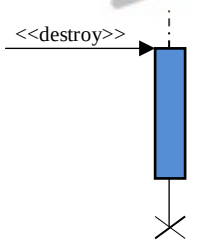
2.7.3 *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima

antar objek. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Tabel berikut adalah simbol-simbol yang ada pada diagram sekuen.

Tabel 2.3 Simbol *Sequence Diagram*

Simbol-simbol	Deskripsi
Aktor  Atau 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal *frase* nama aktor
Garis hidup / *Lifeline* 	Menyatakan kehidpan suatu objek
Objek 	Menyatakan objek yang berinteraksi
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.
Pesan tipe *create*  <<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

Pesan tipe *call* 	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki operasi / metode, karena ini memanggil operasi / metode maka operasi / metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe *send* 	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
Pesan tipe *return* 	Menyatakan suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe *destroy* 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada *destroy*

Sumber: (Kurniawan, 2018)

2.8 Flowchart

Flowchart digunakan dalam mendesain dan mendokumentasikan proses atau program sederhana. Seperti jenis diagram lainnya, diagram membantu memvisualisasikan apa yang sedang terjadi dan dengan demikian membantu memahami suatu proses, dan mungkin juga menemukan fitur-fitur yang kurang jelas dalam proses tersebut, seperti kekurangan dan hambatan. Ada berbagai jenis

diagram alur: masing-masing jenis memiliki set kotak dan notasi sendiri. Dua jenis kotak yang paling umum dalam diagram alur adalah:

1 langkah pemrosesan, biasanya disebut aktivitas dan dilambangkan sebagai kotak persegi panjang.

2 keputusan biasanya dilambangkan sebagai berlian.

Diagram alir digambarkan sebagai "lintas fungsional" ketika bagan dibagi menjadi bagian vertikal atau horizontal yang berbeda, untuk menggambarkan kontrol unit organisasi yang berbeda. Simbol yang muncul di bagian tertentu berada dalam kendali unit organisasi itu. Flowchart lintas fungsional memungkinkan penulis untuk menemukan tanggung jawab untuk melakukan suatu tindakan atau membuat keputusan dengan benar, dan untuk menunjukkan tanggung jawab masing-masing unit organisasi untuk bagian berbeda dari satu proses tunggal.

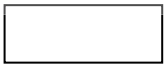
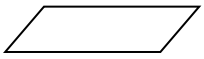
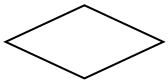
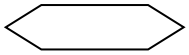
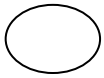
Diagram alir menggambarkan aspek-aspek tertentu dari proses dan biasanya dilengkapi dengan jenis diagram lainnya. Misalnya, Kaoru Ishikawa, mendefinisikan diagram alir sebagai salah satu dari tujuh alat dasar kendali mutu, di sebelah histogram, diagram Pareto, lembar periksa, diagram kontrol, diagram sebab-akibat, dan diagram sebaran. Demikian pula, di UML, notasi pemodelan konsep yang digunakan dalam pengembangan perangkat lunak, diagram aktivitas, yang merupakan jenis diagram alur, hanyalah salah satu dari banyak jenis diagram yang berbeda.

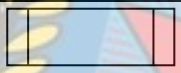
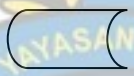
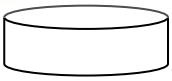
Diagram Nassi-Shneiderman dan Drakon-chart adalah notasi alternatif untuk aliran proses. Nama alternatif umum termasuk diagram alir, diagram alur

proses, diagram alur fungsional, peta proses, diagram proses, diagram proses fungsional, model proses bisnis, model proses, diagram alir proses, diagram alur kerja, diagram alir bisnis. Istilah "diagram alur" dan "diagram alir" digunakan secara bergantian (Nakatsu, 2019).

Struktur grafik yang mendasari diagram alur adalah grafik aliran, yang mengabstraksi jenis simpul, isinya, dan informasi tambahan lainnya. Adapun simbol-simbol flowchart lihat pada tabel sebagai berikut :

Tabel 2.4 Simbol *Flowchart*

NO	SIMBOL	FUNGSI
1.		**Terminal**, untuk memulai atau mengakhiri suatu program
2.		**Proses**, suatu simbol yang menunjukkan setiap pengolahan yang dilakukan.
3.		**Input-Output**, untuk memasukkan menunjukkan hasil dari suatu proses
4.		**Decision**, suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan
5.		**Preparation,** suatu symbol yang menyediakan tempat pengolahan
6.		**Connector**, suatu prosedur penghubung yang akan masuk atau keluar melalui symbol ini dalam lembar yang sama
7.		**Off-Page Connector**, merupakan symbol masuk atau keluarannya

		suatu prosedur pada lembaran kertas lainnya
8.		**Arus/Flow**, dari pada prosedur yang dapat dilakukan atas ke bawah dari bawah ke atas, ke atas dari kiri ke kanan ataupun dari kanan ke kiri
9.		**Predefined Process**, untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur
10.		Simbol untuk output, yang ditunjukkan ke suatu device, seperti printer, dan sebagainya
11		Penyimpanan file secara sementara
12		Menunjukkan input / Output Hardisk (media penyimpanan)

Sumber: (Kurniawan, 2018)

BAB IV

HASIL DAN PEMBAHASAN

Program aplikasi merupakan hasil wujud dari perancangan sistem yang telah dibuat sebelumnya. Dalam program aplikasi, pengguna dapat mengetahui proses pengamanan dokumen menggunakan algoritma *Affine* dan teknik kongruensi linear. Program aplikasi ini dikembangkan dengan bahasa pemrograman *Visual Basic.NET 2010* yang dikemas dalam *Microsoft Visual Studio 2010.* Dalam membuat program, ada beberapa kebutuhan perangkat yang harus dipenuhi agar sistem pengamanan dokumen berjalan dengan baik.

4.1 Kebutuhan Sistem

Sistem merupakan suatu alat yang dapat menjalankan program aplikasi dengan baik. Sistem yang digunakan harus sesuai dengan kebutuhan minimal dari program aplikasi yang akan dibangun. Kebutuhan sistem melibatkan dua buah perangkat yaitu perangkat keras dan lunak. Kedua perangkat ini harus saling bekerja sama dalam mencapai membangun sistem tersebut.

4.1.1 Kebutuhan *Hardware*

Penerapan algoritma *Affine Cipher* dalam meningkatkan keamanan dokumen harus memiliki perangkat keras yang sesuai standar. Tabel 4.1 adalah penggunaan perangkat keras pada penelitian ini.

Tabel 4.1 Spesifikasi perangkat keras

No.	Komponen	Spesifikasi
1	Processor	Intel Core i3 2.4 GHz
2	RAM	4096 MB
3	SSD	128 GB
4	Display	13 inch

4.1.2 Kebutuhan *Software*

Perangkat lunak juga harus saling mendukung dengan perangkat keras dimana perangkat ini dibutuhkan dalam membuat program aplikasi. Tabel 4.2 adalah penggunaan perangkat lunak pada penelitian ini.

Tabel 4.2 Spesifikasi perangkat lunak

No.	Komponen	Spesifikasi
1	Sistem Operasi	Windows 10 64 Bit
2	IDE Pemrograman	Microsoft Visual Basic.NET 2010
3	Pengolah Kata	Microsoft Word
4	Data Editor	Microsoft Excel

4.2 Hasil Sistem Yang Dibangun

Hasil sistem yang dibangun pada penelitian ini merupakan sebuah perangkat lunak atau *software* yang dapat melakukan proses enkripsi pada dokumen atau file lainnya. Hasil enkripsi dapat disimpan sehingga dapat dikembalikan kembali menjadi file aslinya pada saat dibutuhkan. Dalam melakukan proses enkripsi dan dekripsi, penulis membuat program aplikasi dengan beberapa tampilan yang akan dijelaskan sebagai berikut.

4.2.1 Halaman Menu Utama

Halaman *Menu Utama* merupakan halaman program aplikasi yang pertama muncul pada saat program *running*. Gambar 4.1 adalah hasil tampilan *Menu Utama*.

Gambar 4.1 Halaman Menu Utama

4.2.2 Halaman Info

Halaman *Info* adalah menu halaman singkat yang menjelaskan gambaran penelitian yang terdiri dari latar belakang, rumusan masalah, tujuan penelitian dan hasil yang dicapai setelah sistem pengamanan dokumen ini berhasil dibangun. Gambar 4.2 adalah hasil tampilan dari halaman *Info*.

Gambar 4.2 Halaman Info

4.2.3 Halaman About

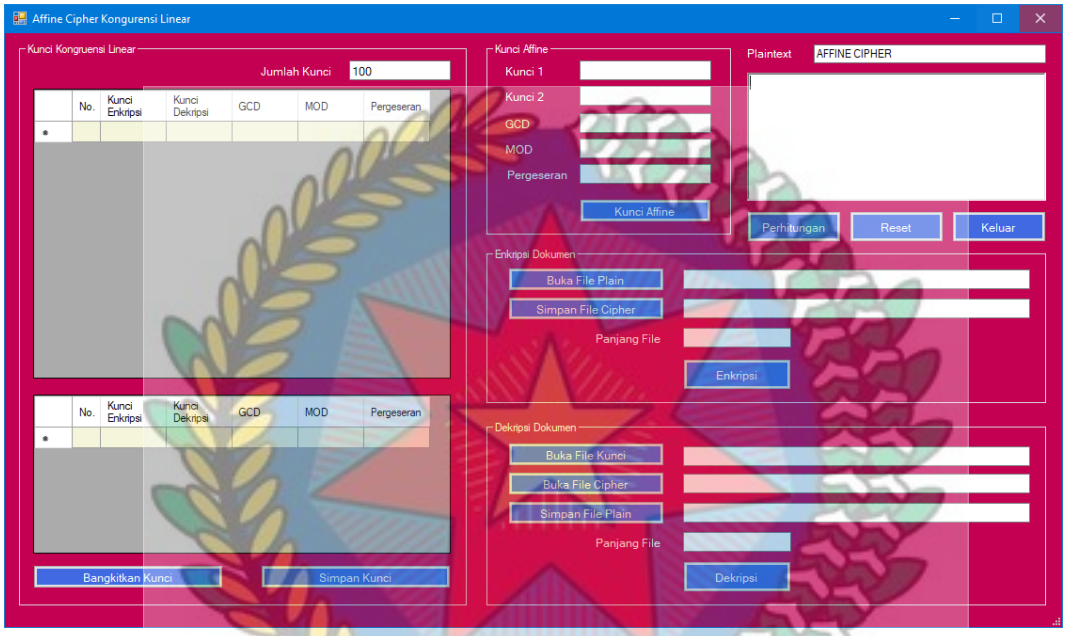
Halaman *About* menampilkan informasi penulis yang dapat memberikan informasi kepada pembaca atau pengguna program. Gambar 4.3 adalah tampilan dari halaman About. Adapun informasi penulis tersebut, antara lain:

1. Judul
2. Nama
3. NPM
4. Program Studi
5. Universitas
6. Tahun Ajaran

Gambar 4.3 Halaman About

4.2.4 Halaman *Affine Cipher*

Halaman *Affine Cipher* adalah halaman dimana proses enkripsi dan dekripsi dapat dilakukan terhadap file dokumen yang dianggap penting untuk diamankan. Proses enkripsi dan dekripsi harus melibatkan kunci sehingga kunci yang dibangkitkan dengan teknik kongruensi linear harus disimpan dalam bentuk file teks agar dapat digunakan kembali dalam melakukan dekripsi file tersebut. Kehilangan kunci akan mengakibatkan file hasil enkripsi tidak dapat dibuka kembali. Gambar 4.4 adalah hasil tampilan dari halaman *Affine Cipher*.



Gambar 4.4 Halaman *Affine Cipher*

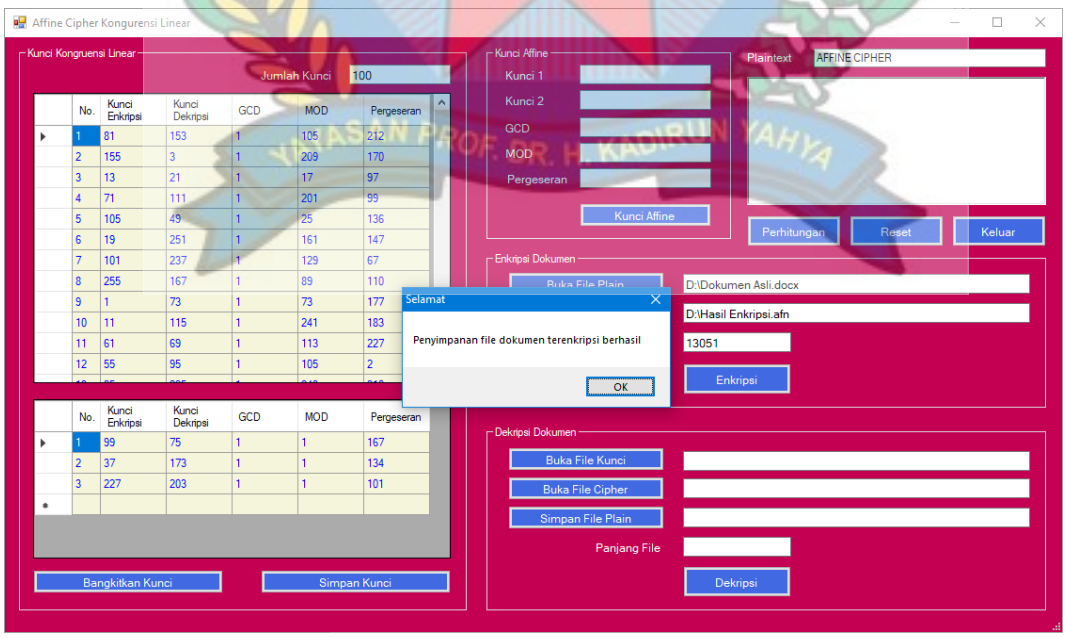
4.3 Proses Enkripsi dan Dekripsi

Bagian ini adalah bagian yang akan melakukan pengujian terhadap proses enkripsi dan dekripsi dari program aplikasi yang berhasil dibangun. Pengujian ini melibatkan sebuah data yang berisi abstrak dari penelitian ini. Data uji merupakan file yang bertipe *.docx* yang dibuat menggunakan aplikasi *Microsoft Word*. Hasil dekripsi harus dapat mengembalikan file tersebut ke bentuk semula sehingga isi dari file tersebut tetap dapat dibaca menggunakan program aplikasi *Microsoft Word*. Kunci yang digunakan dalam proses enkripsi dan dekripsi tidak sama sehingga lebih menghindari dari pencurian kunci.

4.3.1 Hasil Enkripsi

Hasil enkripsi diperoleh dengan cara melakukan perhitungan kunci yang dibangkitkan dengan teknik kongruensi linear. Kunci yang dibangkitkan

merupakan kunci yang digunakan untuk enkripsi dan dekripsi. Kunci tersebut akan dihitung menggunakan formula *Affine Cipher* dan menentukan nilai GCD dan MODULO yang masing-masing harus bernilai 1. Kunci yang dibangkitkan akan difilter sehingga mendapatkan kunci yang hanya memenuhi syarat untuk proses enkripsi dan dekripsi *Affine Cipher*. Gambar 4.5 adalah tampilan dari hasil proses enkripsi algoritma *Affine Cipher* dalam melakukan pengamanan dokumen berbentuk file *.docx*.

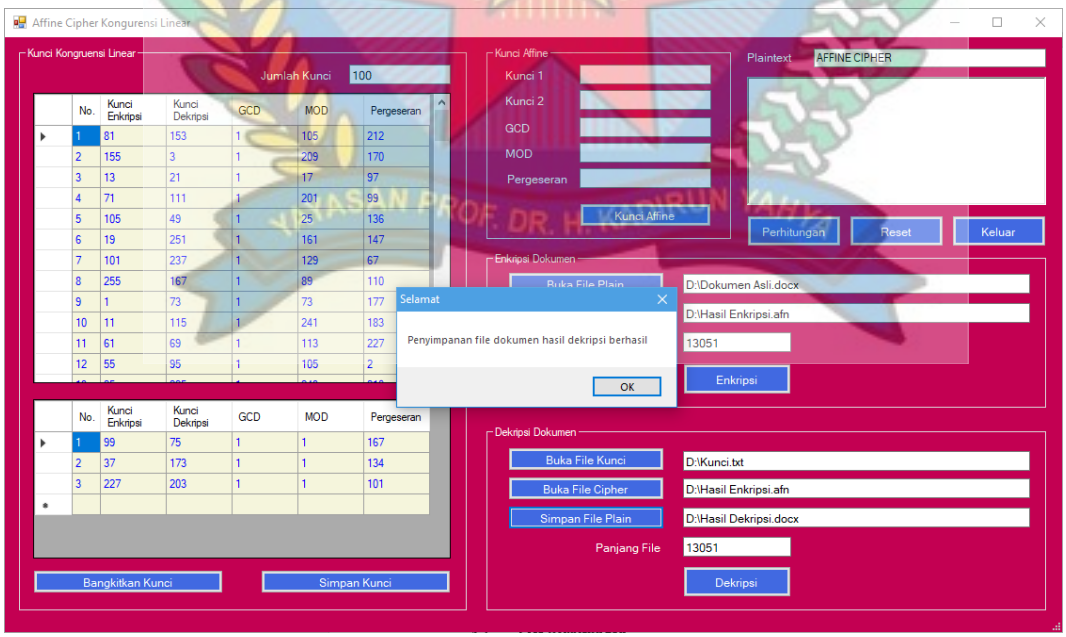


Gambar 4.5 Hasil enkripsi algoritma *Affine Cipher*

4.3.2 Hasil Dekripsi

Hasil dekripsi diperoleh dengan cara melakukan pemeriksaan kunci yang dimasukkan pada bagian input kunci. Kunci tersebut berfungsi untuk melakukan dekripsi deretan byte pada data file yang sudah terenkripsi. Pengguna harus memasukkan file terenkripsi untuk dapat melakukan proses dekripsi. Apabila kunci

yang dimasukkan bernilai benar, maka file tersebut akan dapat dibaca kembali menggunakan program aplikasi pembuka dokumen tersebut. Apabila kunci yang dimasukkan salah, proses dekripsi tetap berjalan, tetapi hasil dekripsi dari dokumen tersebut tidak dapat dibaca atau dibuka oleh program aplikasi pembuka dokumen tersebut. Gambar 4.6 adalah tampilan dari hasil proses dekripsi algoritma *Affine Cipher*.



Gambar 4.6 Hasil dekripsi algoritma *Affine Cipher*

BAB V

PENUTUP

5.1 Kesimpulan

Penulis menarik beberapa kesimpulan berdasarkan hasil pengujian sistem pengamanan dokumen, antara lain:

1. Pengamanan dokumen dengan algoritma *Affine Cipher* dan teknik kongruensi linear bekerja dengan baik.
2. Teknik kongruensi linear akan melakukan pembangkitan kunci acak yang akan dimasukkan ke dalam proses enkripsi dan dekripsi.
3. *Affine Cipher* menggunakan deretan kunci yang bernilai GCD=1 dan MODULO=1 dalam melakukan proses enkripsi dan dekripsi.

5.2 Saran

Penelitian ini diharapkan dapat dikembangkan menjadi lebih sempurna. Adapun saran yang dapat dikembangkan, antara lain:

1. Sebaiknya program aplikasi dapat digunakan secara *online*.
2. Hendaknya jumlah karakter yang dapat diproses melebihi dari 10 *megabyte*.

DAFTAR PUSTAKA

Aulia, R., Zakir, A., & Zulhafiz, M. (2019). Penerapan Algoritma One Time Pad & Linear Congruential Generator Untuk Keamanan Pesan Teks. *Jurnal Nasional Informatika Dan Teknologi Jaringan*, *4*(1), 37–41.

Ayushi, M. (2010). A Symmetric Key Cryptographic Algorithm. *International Journal of Computer Applications*, *1*(15), 1–6. https://doi.org/10.5120/331-502

Barone, L., Williams, J., & Micklos, D. (2017). Unmet needs for analyzing biological big data: A survey of 704 NSF principal investigators. *PLOS Computational Biology*, *13*(10), e1005755. https://doi.org/10.1371/journal.pcbi.1005755

Barot, S. (2018). *Why Is Data Security Important for Everyone?* Security Zone. https://dzone.com/articles/why-is-data-security-important-for-everyone

Gurevich, Y. (2012). *What Is an Algorithm?* (pp. 31–42). https://doi.org/10.1007/978-3-642-27660-6_3

Jovancic, N. (2019). *5 Data Collection Methods for Obtaining Quantitative and Qualitative Data*. LeadQuizzes. https://www.leadquizzes.com/blog/data-collection-methods/

Kurniawan, T. A. (2018). Pemodelan Use Case (UML): Evaluasi Terhadap beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *5*(1), 77. https://doi.org/10.25126/jtiik.201851610

Ladjamudin, A.-B. bin. (2017). *Analisis dan Desain Sistem Informasi*. Graha Ilmu.

Mokhtari, M., & Naraghi, H. (2012). Analysis and Design of Affine and Hill Cipher. *Journal of Mathematics Research*, *4*(1). https://doi.org/10.5539/jmr.v4n1p67

Nakatsu, R. T. (2019). *Reasoning with Diagrams : Decision-Making and Problem-Solving with Diagrams.* John Wiley & Sons.

Nasution, A. B. (2020). Modifikasi Algoritma Affine Cipher Untuk Mengamankan Data. *Jurnal Teknologi Informasi*, *4*(2), 377–382.

Nurgoho, A. (2019). *Rekayasa Perangkat Lunak Menggunakan UML dan JAVA*. Andi Offset.

Rao, R. V., & Selvamani, K. (2015). Data Security Challenges and Its Solutions in

Cloud Computing. *Procedia Computer Science*, *48*, 204–209. https://doi.org/10.1016/j.procs.2015.04.171

S., G., L. Ribeiro, A. R., & David, E. (2012). Asymmetric Encryption in Wireless Sensor Networks. In *Wireless Sensor Networks - Technology and Protocols*. InTech. https://doi.org/10.5772/48464

Sukmawati, R., & Priyadi, Y. (2019). Perancangan Proses Bisnis Menggunakan UML Berdasarkan Fit/Gap Analysis Pada Modul Inventory Odoo. *INTENSIF: Jurnal Ilmiah Penelitian Dan Penerapan Teknologi Sistem Informasi*, *3*(2), 104. https://doi.org/10.29407/intensif.v3i2.12697

Tang, H.-C. (2017). An analysis of linear congruential random number generators when multiplier restrictions exist. *European Journal of Operational Research*, *182*(2), 820–828. https://doi.org/10.1016/j.ejor.2006.08.055

Technopedia. (2019). *Unified Modeling Language (UML)*. Technopedia. https://www.techopedia.com/definition/3243/unified-modeling-language-uml

Tuomi, I. (2019). Data Is More Than Knowledge: Implications of the Reversed Knowledge Hierarchy for Knowledge Management and Organizational Memory. *Journal of Management Information Systems*, *16*(3), 103–117. https://doi.org/10.1080/07421222.1999.11518258

UTM. (2019). *Concept: Use-Case Model*. Univesidad Technologica de La Mixteca. http://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/use_case_model_CD178AF9.html

Wasserkrug, S., Dalvi, N., Munson, E. V., Gogolla, M., Sirangelo, C., Fischer-Hübner, S., Ives, Z., Velegrakis, Y., Bevan, N., Jensen, C. S., & Snodgrass, R. T. (2019). Unified Modeling Language. In *Encyclopedia of Database Systems* (pp. 3232–3239). Springer US. https://doi.org/10.1007/978-0-387-39940-9_440

Zandbergen, P. (2019). *What is a Computer Algorithm?* Study.Com. https://study.com/academy/lesson/what-is-a-computer-algorithm-design-examples-optimization.html