IMPLEMENTASI ALGORITMA *ONE TIME PAD CIPHER* DAN TRANSFORMASI *MYSZKOWSKI CIPHER* PADA PESAN TEKS

SKRIPSI

OLEH
MUHAMMAD YUSRUL HANA
NIM. 14610090

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021

IMPLEMENTASI ALGORITMA *ONE TIME PAD CIPHER* DAN TRANSFORMASI *MYSZKOWSKI CIPHER* PADA PESAN TEKS

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
MUHAMMAD YUSRUL HANA
NIM. 14610090

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2021

IMPLEMENTASI ALGORITMA *ONE TIME PAD CIPHER* DAN TRANSFORMASI *MYSZKOWSKI CIPHER* PADA PESAN TEKS

SKRIPSI

Oleh
MUHAMMAD YUSRUL HANA
NIM. 14610090

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal 03 Juni 2021

Pembimbing I,	Pembimbing II,
Muhammad Khudzaifah, M.Si	Hisyam Fahmi, M.Kom.
NIP. 19901511 2016801 1 057	NIP. 19890727 201903 1 018

Mengetahui,
Ketua Program Studi Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

IMPLEMENTASI ALGORITMA *ONE TIME PAD CIPHER* DAN TRANSFORMASI *MYSZKOWSKI CIPHER* PADA PESAN TEKS

SKRIPSI

Oleh
MUHAMMAD YUSRUL HANA
NIM. 14610090

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Sains (S.Mat)
Tanggal 14 Juni 2021

Penguji Utama	:	Mohammad Nafie Jauhari, M.Si	.....
Ketua Penguji	:	Juhari, M.Si	.....
Sekretaris Penguji	:	Muhammad Khudzaifah, M.Si	.....
Anggota Penguji	:	Hisyam Fahmi, M.Kom	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Yusrul Hana

NIM : 14610090

Jurusan : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Algoritma *One Time Pad Cipher* dan Transformasi *Myszkowski Cipher* pada Pesan Teks.

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar rujukan. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 01 Juni 2021
Yang membuat pernyataan,

Muhammad Yusrul Hana
NIM. 14610090

MOTO

“Jadilah orang yang bermanfaat dimanapun anda berada ”

PERSEMBAHAN

Alhamdulillahi Robbil'alamin, dengan mengucap syukur kepada Allah Azza Wa Jalla, Penulis mempersembahkan skripsi ini untuk kedua orang tua saya tercinta, Almarhum Bapak Usman Ashofi, dan Ibu Lutiah yang selalu memberikan doa, dukungan, motivasi kepada penulis. Kakak Sirin Khumairoh dan adik-adik Khofifatus Shaqilah, Muhammad Nijam Masruri yang selalu membeikan dukungan semangat kepada penulis, serta Maslahatul Marwa, S.E yang selalu memberikan dukungan semangat untuk dapat menyelesaikan skripsi ini kepada penulis.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

السَّلاَمُ عَلَيْكُمْ وَرَحْمَةُ اللهِ وَبَرَكَاتُهُ

Segala puji bagi Allah Azza Wa Jalla Tuhan sekalian alam yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini sebagai syarat untuk memperoleh gelar sarjana di Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Nehgeri Maulana Malik Ibrahim Malang.

Dalam proses penyusunan skripsi ini, penulis banyak sekali mendapatkan pengarahan dan bimbingan dari berbagai pihak. Maka dari itu ucapan terima kasih yang sebesar-besarnya dari penulis sampaikan terutama kepada:

1. Prof. Dr. H. Abd. Haris, M.Ag, selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Usman Pagalay, M.Si, selaku ketua program studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Muhammad Khudzaifah, M.Si, selaku dosen pembimbing I yang telah banyak memberikan arahan, nasihat, motivasi, dan berbagai pengalaman yang sangat berharga kepada penulis.

5. Hisyam Fahmi, M.Kom, selaku dosen pembimbing II yang telah banyak memberikan arahan dan berbagai ilmunya kepada penulis.
6. Mohammad Nafie Jauhari, M.Si, selaku dosen penguji utama yang telah banyak memberikan arahan dan berbagai ilmunya kepada penulis.
7. Juhari, M.Si, selaku dosen ketua penguji yang telah banyak memberikan arahan dan berbagai ilmunya kepada penulis.

Semoga Allah Swt melimpahkan rahmat dan karunia-Nya kepada kita semua. Akhirnya penulis berharap semoga dengan rahmat dan izin-Nya mudah-mudahan skripsi ini bermanfaat bagi penulis dan bagi pembaca. *Amiin.*

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 10 Juni 2021

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Hana, Muhammad Yusrul. 2021. **Implementasi Algoritma *One Time Pad Cipher* Dan Transformasi *Myszkowski Cipher* Pada Pesan Teks.** Skripsi. Jurusan Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II) Hisyam Fahmi, M.Kom.

Kata kunci: Enkripsi, Dekripsi, Super Enkripsi, *One Time Pad Cipher*, *Myszkowski Cipher*

Enkripsi merupakan proses penyandiaan pesan teks menjadi pesan tak terbaca dan dekripsi merupakan proses kebalikannya. Penelitian ini bertujuan untuk meningkatkan tingkat keamanan pesan yang dienkripsi menggunakan metode Super Enkripsi, yaitu dengan menggunakan algoritma One Time Pad Cipher sebagai implementasi metode subtitusi dan algoritma Myszkowski Cipher sebagai implementasi metode transposisi.

Penggunaan super enkripsi dengan *One Time Pad Cipher* dan *Myszkowski Cipher* akan melipat gandakan keamanan dari pesan. Keamanan pertama terletak dari tingkat keamanan enkripsi pesan *One Time Pad Cipher* yang bergantung dari banyaknya variasi karakter yang bisa digunakan, untuk meningkatkan hal tersebut maka dalam penelitian ini digunakan ASCII untuk memperbanyak variasi karakter yang bisa digunakan. Selanjutnya keamanan kedua, pesan yang sudah tersandikan akan diproses menggunakan *Myszkowski Cipher* sehingga membuat pesan semakin sulit dipecahkan.

ABSTRACT

Hana, Muhammad Yusrul. 2021. **Implementasi Algoritma *One Time Pad Cipher* Dan Transformasi *Myszkowski Cipher* Pada PesanTeks.** Essay. Department of Mathematics, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Advisor: (I) Muhammad Khudzaifah, M.Si. (II) Hisyam Fahmi, M.Kom.

Keywords: Encryption, Decryption, Super Encryption, *One Time Pad Cipher*, *Myszkowski Cipher*

Encryption is the process of encoding text messages into unreadable messages and decryption is the reverse process. This study aims to increase the level of security of messages encrypted using the Super Encryption method, namely by using the One Time Pad Cipher algorithm as the implementation of the substitution method and the Myszkowski Cipher algorithm as the implementation of the transposition method.

The use of super encryption with One Time Pad Cipher and Myszkowski Cipher will double the security of the message. The first security lies in the security level of One Time Pad Cipher message encryption which depends on the number of character variations that can be used, to increase this, ASCII is used in this study to increase the variety of characters that can be used. Furthermore, the second security, messages that have been encrypted will be processed using Myszkowski Cipher, making the message more difficult to decipher.

ملخص

هنا, محمد يسرل.2021.تنف ذ خوارزم ة تشفير *One Time Pad* وتحويل تشفير *Myszkowski* في الرسائل النصية. . البحث العلمي. قسم الر ضيات, كلية العلوم والتكنولوخيا, جامعة مولا مالك إبراهيم الإسلامية الحكومية بمالانج. المشرف الأول (1) محمد حذيفة الماجستير، المشرف الثاني (2) هشام فهمي،،الماجستير.

االكمات الر ئيسية : تشفير, فك تشفير, تشفير فائق, تشفير *One Time Pad*, تشفير *Myszkowski*.

التشفير هو عملية تشفير الرسائل النصية إلى رسائل غير قابلة للقراءة وفك التشفير هو العملية العكسية. تهدف هذه الدراسة إلى ز دة مستوى أمان الرسائل المشفرة ستخدام طريقة التشفير الفائق ، أي ستخدام خوارزمية تشفير *One Time Pad* كتنفيذ لطريقة الاستبدال وخوارزمية تشفير*Myszkowski* كتنفيذ لطريقة النقل.

سيؤدي استخدام التشفير الفائق مع تشفير *One Time Pad* و تشفير *Myszkowski*إلى مضاعفة أمان الرسالة. يكمن الأمان الأول في مستوى أمان تشفير الرسائل تشفير *One Time Pad*الذي يعتمد على عدد متغيرات الأحرف التي يمكن استخدامها ، ولز دة ذلك ، يتم استخدام *ASCII* في هذه الدراسة لز دة تنوع الأحرف التي يمكن استخدامها. علاوة على ذلك ، الأمن الثاني ، الرسائل التي تم تشفيرها ستتم معالجتها ستخدام تشفير *Myszkowski* ، مما يجعل فك تشفير الرسالة أكثر صعوبة.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Pengiriman pesan khususnya yang berbentuk teks sangat banyak digunakan saat ini, teks tersebut ada yang bersifat rahasia dan ada yang tidak. Teks yang bersifat rahasia perlu mendapatkan pengaman agar kerahasiaan teks tidak diketahui oleh pihak yang tidak berwenang. Dalam menjamin sebuah kerahasiaan pesan maka sangatlah perlu untuk melakukan pengamanan pada pesan tersebut

Salah satu cara untuk memberikan pengamanan pada pesan teks adalah dengan kriptografi. Ilmu dan seni untuk menjaga kerahasiaan informasi bias juga disebut dengan pengertian kriptografi secara umum (Schneier, 1996). Dalam kriptografi banyak algoritma yang bisa diterapkan seperti *Hill Cipher, Vignere Cipher, Caesar Cipher Rail fence Cipher* dan *One Time Pad Cipher.*

Konsep dasar kriptografi berlandaskan pada teori-teori yang ada dalam ilmu matematika, seperti penguraian bilangan yang sangat besar, komputasi logaritma diskrit, teknik-teknik yang bersifat probabilistik dan lain sebagainya. Teori-teori inilah yang membuat kriptografi menjadi aman digunakan untuk mengirimkan pesan yang bersifat rahasia.

Menjaga amanah atau sebuah rahasia merupakan suatu langkah untuk mengamankan sebuah data atau pesan rahasia. Sebagaimana iman Allah SWT dalam An-Nisa ayat 58 :

إِنَّ ٱللَّهَ يَأۡمُرُكُمۡ أَن تُؤَدُّواْ ٱلۡأَمَٰنَٰتِ إِلَىٰٓ أَهۡلِهَا وَإِذَا حَكَمۡتُم بَيۡنَ ٱلنَّاسِ أَن تَحۡكُمُواْ بِٱلۡعَدۡلِۚ إِنَّ ٱللَّهَ نِعِمَّا يَعِظُكُم بِهِۦٓۗ إِنَّ ٱللَّهَ كَانَ سَمِيعَۢا بَصِيرٗا

Artinya: "*Sesungguhnya Allah menyuruh kamu menyampaikan amanat kepada yang berhak menerimanya, dan (menyuruh kamu) apabila menetapkan hukum diantara manusia supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberi pengajaran yang sebaik-baiknya kepadamu. Sesungguhnya Allah adalah Maha Mendengar lagi Maha Melihat."*

Seiring dengan perkembangan zaman, kriptografi sudah menjadi sebuah bahan objek penelitian yang dilakukan oleh banyak orang, dari berbagai cara dengan menggabungkan beberapa metode kriptografi hingga menciptakan metode kriptografi yang baru. Menggabungkan dua buah *cipher* itu merupakan salah satu cara membuat sebuah kriptografi yang lebih aman atau biasa juga cara tersebut dinamakan dengan super enkripsi. Hal itu dilakukan agar bisa mendapatkan *cipher* yang lebih kuat sehingga tidak mudah untuk dipecahkan, dan juga untuk mengatasi penggunaan *cipher* tunggal yang secara komparatif sangatlah lemah.

Kriptografi terbagi menjadi dua jenis yaitu kriptografi klasik dan modern. Algoritma kriptografi yang digunakan dalam penelitian ini adalah kriptografi klasik *cipher* transposisi yaitu algoritma *Myszkowski Cipher* dan kriptografi klasik *cipher* substitusi yaitu *One Time Pad Cipher.* Berdasarkan penelitian yang dilakukan oleh (Latifah, et al.,2017), kriptografi klasik cukup lemah jika diterapkan sendiri-sendiri, akan tetapi lebih kuat jika digabung dengan metode klasik lainnya.

Algoritma *One Time Pad Cipher* merupakan kriptografi klasik yang menggunakan satu buah kunci untuk melakukan sebuah enkripsi dan dekripsi yang sama, ditemukan oleh Major Joseph Mauborgne pada tahun 1917. Kunci kriptografi algoritma *One Time Pad Cipher* berisi barisan acak yang ketika disandikan akan menghasilkan *plantext* dengan barisan yang sepenuhnya acak. *One Time Pad*

Cipher harus menggunakan kunci yang *random* untuk meningkatkan keamanan dari algoritma *One Time Pad Cipher.*

Namun, masing-masing dari algoritma tersebut juga memiliki kelebihan dan kelemahan tersendiri. Jika suatu pesan dirahasiakan dengan salah satu algoritma saja misalkan *Myszkowski Cipher,* maka pesan tersebut masih belum bisa dikatakan aman. Karena algoritma *Myszkowski Cipher* sangatlah lemah jika diterapkan sendiri tanpa dikombinasikan dengan algoritma klasik lainnya. Maka, penelitian ini akan mengkombinasikan algoritma *One Time Pad Cipher* dengan algoritma *Myszkowski Cipher* dengan teknik super enkripsi yang menggabungkan dua buah *cipher*. Mengkombinasikan dua buah algoritma *cipher* substitusi dan *cipher* transposisi bertujuan untuk memberikan penyandian baru, sehingga pesan yang akan dikirim dalam bentuk pesan teks lebih sulit untuk kriptanalis dibandingkan dengan penyandian yang menggunakan satu algoritma.

Dari pemaparan di atas, penulis melakukan penelitian yang berjudul "Implementasi Algoritma *One Time Pad Cipher* dan *Transformasi Myszkowski Cipher* pada Pesan Teks*"*

1.2 Rumusan Masalah

Berdasarkan uraian dari latar belakang di atas, maka peneliti merumuskan sebagai berikut:

1. Bagaimana proses enkripsi dan dekripsi pada pesan teks dengan algoritma *One Time Pad Cipher*?
2. Bagaimana proses enkripsi dan dekripsi pada pesan teks dengan algoritma *Myszkowski Cipher*?

3. Bagaimana proses enkripsi dan dekripsi dengan algoritma super enkripsi pada pesan teks dengan algoritma *One Time Pad Cipher* dan algoritma *Myszkowski Cipher*?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan penelitian ini adalah sebagai beriikut:

1. Mengetahui proses enkripsi dan dekripsi pada pesan teks dengan algoritma *One Time Pad Cipher*.
2. Mengetahui proses enkripsi dan dekripsi pada pesan teks dengan algoritma *Myszkowski Cipher*.
3. Mengetahui proses enkripsi dan dekripsi dengan algoritma super enkripsi pada pesan teks dengan algoritma *One Time Pad Cipher* dan algoritma *Myszkowski Cipher*.

1.4 Manfaat Penelitian

1. Untuk mendapatkan sebuah pemahaman tentang bagaimana proses enkripsi dan dekripsi pada pesan teks dengan algoritma *One Time Pad Cipher*.
2. Untuk mendapatkan sebuah pemahaman tentang bagaimana proses enkripsi dan dekripsi pada pesan teks dengan algoritma *Myszkowski Cipher*.
3. Untuk mendapatkan sebuah pemahaman tentang bagaimana proses enkripsi dan dekripsi dengan algoritma super enkripsi pada pesan teks dengan algoritma *One Time Pad Cipher* dan algoritma *Myszkowski Cipher*.

1.5 Batasan masalah

Batasan masalah Batasan masalah ini digunakan agar pembahasan dalam skripsi ini tidak meluas dan tidak menimbulkan permasalahan yang baru, maka ruang lingkup penulis dalam melakukan peneliian ini memberi batasan sebagai berikut.

1. Peneletian ini hanya membahas super enkripsi menggunakan algoritma *One Time Pad Cipher* dan algoritma *Myszkowski Cipher* dengan menjelaskan enkripsi dan dekripsi dari masing-masing algoritma terebut.
2. Pada penelitian ini hanya berlaku untuk pesan berbentuk teks dengan 26 variable huruf dalam standart yang digunakan di ASCII (*American Code for Information Interchange).*

1.6 Sistematika Penulisan

Sistematika penulisan dalam skripsi ini dibagi menjadi empat bab dan setiap bab memiliki beberapa subbab sebagai berikut:

Bab I Pendahuluan

Bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

Bab II Kajian Pustaka

Bab ini penulis menjelaskan konsep-konsep yang berkaitan dengan pembahasan penelitian, yaitu kriptografi, simetris, asimetris, enkripsi, dekripsi, algoritma *One Time Pad Cipher* dan algoritma *Myszkowski Cipher*

Bab III Pembahasan

Bab ini berisi tentang langkah-langkah pembentukan *ciphertext* yang melalui tahap super enkripsi substitusi dan transposisi yang dilakukan melalui metode transformasi *Myszkowski Cipher* dan algoritma *One Time Pad Cipher* sehingga didapatkan suatu *ciphertext* yang telah terenskripsi.

Bab IV Penutup

Bab ini berisi kesimpulan dari pembahsan yang sesuai dengan hasil peneletian dan uji coba, yang selanjutnya dapat digunakan sebagai saran bagi pembaca dan peneliti selanjutnya.

1.7 Metode Penelitian

Metode penelitian yang dilakukan dalam penelitian ini adalah studi literature dan melakukan uji coba terhadap algoritma-algoritma yang digunakan oleh penulis. Penulisan dimulai dengan mempelajari jurnal-jurnal, tugas akhir, artikel dan buku-buku tentang enkripsi serta deskripsi pada pesan teks beserta algoritma-algoritmanya. Adapun langkah-langkah penyelesaian penelitian ini, sebagai berikut:

1. Menyusun enkripsi dan dekripsi dengan algoritma *One Time Pad Cipher* pada pesan teks.
    a. Menentukan 26 karakter huruf yang akan digunakan sebagai *plaintext.*
    b. *Plaintext* dan kunci yang diberikan harus sama panjang, hal ini merupakan sifat dari algoritma *One Time Pad Cipher*.
    c. Proses dekripsi pada algoritma *One Time Pad Cipher* adalah kebalikan atau mengembalikan *plaintext* menjadi data semula dengan persamaan $P_1 = (C_1 - K_1) \bmod 26$.
2. Menyusun enkripsi dan dekripsi dengan *Myszkowski Cipher* pada pesan teks.

a. Langkah pertama terlebih dahulu dilakukan pembentukan kunci, beberapa huruf yang dibentuk secara manual ataupun acak dapat menambah variasi pembentukan kunci.

b. Proses enkripsi dimulai dengan membentuk sejumlah baris dan kolom untuk menampung *plaintext.*

c. Terdapat 14 huruf pada *plaintext* yang akan menjadi acuan dalam membentuk baris, dan 3 huruf pada kunci akan menjadi acuan untuk membentuk kolom, sehingga jumlah kolom dan baris yang dibutuhkan : Kunci = 3 huruf => 3 kolom

Plaintext = 14 huruf => 14/3 = 4,6 =>5 baris

d. Setelah membentuk baris dan kolom yang memungkinkan, *plaintext* dapat ditulis secara berurutan dan horizontal.

e. Proses dekripsi dapat dilakukan dengan menuliskan *ciphertext* secara vertical dari atas kebawah secara beruntun sesuai dengan penomoran kunci, pada penomoran kunci yang sama, *ciphertext* ditulis secara horizontal.

f. Untuk mendapatkan hasil dekripsi *Myszkowski Cipher* dengan cara membaca huruf dari kolom paling kiri tanpa memperdulikan penomoran kunci, sehingga diperoleh hasil dekripsi *Myszkowski Cipher.*

3. Menyusun super enkripsi dengan algoitma *One Time Pad Cipher* dan algoritma *Myszkowski Cipher* pada pesan teks.

a. Proses super enkripsi menggunakan algoritma *One Time Pad Cipher*, kemudian pesan hasil enkripsi tersebut dienkripsi lagi menggunakan *Myszkowski Cipher* sehingga akan terbentuk keamanan dua lapis.
b. Mengembalikan pesan agar terbaca kembali maka dilakukan dekripsi menggunakan *Myszkowski Cipher* kemudian pesan didekripsi menggunakan *One Time Pad Cipher*.

BAB II
KAJIAN PUSTAKA

2.1 Kriptografi

Kriptografi (*cryptography*) yaitu gabungan dari kata "*crypt*" yang artinya "*hidden*" (tersembunyi/rahasia) dan "*graphy*" yang mengacu pada "*writing*" (tulisan), sehingga kriptografi merupakan tulisan yang terahasiakan dan umumnya mengacu pada bagian enkripsi untuk membangun sebuah sistem untuk mengirimkan rahasia.

Kriptografi merupakan ilmu dan seni yang mempelajari bagaimana memproteksi pesan yang akan disampaikan menjadi lebih aman dengan sistem mengubah pesan menjadi bentuk yang tidak dapat diketahui. *Plaintext* merupakan teks yang asli dan dapat dibaca serta dapat diketahui maknanya. *Ciphertext* merupakan teks yang tidak dapat dibaca dan tidak dapat diketahui maknanya.

Terdapat dua proses utama pada kriptografi yaitu sebagai berikut :

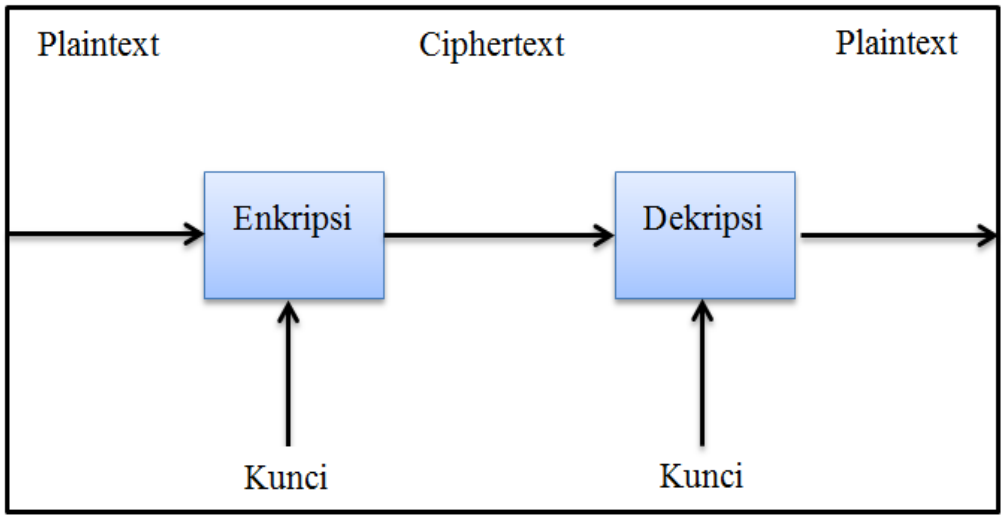
1. Enkripsi

   Enkripsi merupakan proses pengubahan *plaintext* dengan menggunakan kunci yang telah ditentukan menjadi *ciphertext*. Proses diubahnya data atau pesan yang hendak dikirim menjadi bentuk yang hampir tidak dikenali oleh pihak ketiga.

2. Dekripsi

   Dekripsi merupakan proses pengembalian *ciphertext* dengan menggunakan kunci yang sama pada enkripsi menjadi *plaintext*. Proses mengubah data atau pesan kembali kebentuk semula sehingga data atau pesan dapat tersampaikan

dan dimengerti oleh penerima. Pada Gambar 2.1 merupakan skema proses enkripsi dan dekripsi yang digunakan dalam pengiriman pesan.



Gambar 2.1 skema enkripsi dan dekripsi

2.1.1 Tujuan Kriptografi

Menurut (Setyaningsih, 2015) beberapa dari tujuan kriptografi adalah sebagai berikut:

1. Kerahasiaan (*confidentiality*) yaitu layanan perlindungan agar pesan yang dikirim tidak dapat dibaca oleh pihak-pihak yang tidak bertanggungjawab. Secara umum *confidentiality* dilakukan dengan aturan membuat suatu algoritma matematis tertentu yang dapat mengubah data hingga sulit ntuk dimengerti.
2. Integritas data (*data integrity*) merupakan layanan yang dapat mendeteksi adanya pesan masih dikatakan asli atau belum pernah dimanipulasi selama masa pengiriman.

3. Otentikasi (*authentication*) adalah layanan penerima pesan yang dapat memastikan keaslian pengirimannya. Penyerang tidak dapat berpura-pura sebagai orang lain.
4. Penyangkalan (*Non-repudiation*) adalah layanan yang dapat mencegah pembuktian bahwa pengirim tidak dapat menyangkal bahwa pengirim telah mengirim pesan, dan penerima juga tidak dapat menyangkal bahwa penerima telah menerima pesan.

2.1.2 Komponen Kriptografi

Di dalam kriptografi, akan sering ditemukan berbagai istilah atau terminologi. Berikut adalah beberapa istilah yang penting untuk diketahui.

1. *Plaintext* dan *Ciphertext*.

   Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks (*plaintext*) atau teks-jelas (*cleartext*). Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut cipherteks (*ciphertext*) atau kriptogram (*cryptogram*). Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca.

2. Pengirim dan Penerima

   Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Entitas di sini dapat berupa orang, mesin (komputer), kartu kredit, dan sebagainya.

3. Enkripsi dan dekripsi

   Enkripsi (*encryption*) merupakan proses menyandikan plainteks menjadi cipherteks. Sedangkan, proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi (*decryption*). (Anusha, et al., 2016).

4. Kriptanalis dan Kriptologi

   Kriptanalis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan cipherteks menjadi plainteks, tanpa memerlukan kunci yang digunakan. Pelakunya disebut dengan *cryptanalyst*. Kriptanalis berusaha memecahkan cipherteks tersebut untuk menemukan *plaintext* atau *key*. Kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalis.

2.1.3 Jenis Algoritma Kriptografi

Berdasarkan perkembangannya, kriptografi terbagi atas dua jeis yaitu kriptografi modern dan kriptografi klasik. Kriptografi modern terbagi menjadi dua jenis yaitu simetris dan asimetris. Kriptografi klasik terbagi menjadi dua jenis yaitu substitusi dan transposisi.

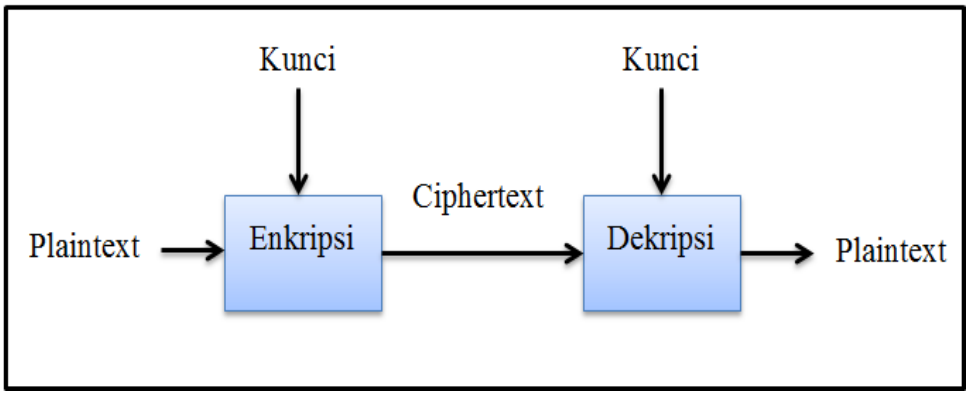
2.1.4 Algoritma Kriptografi Modern

Secara umum ada dua jenis kriptografi berdasarkan kuncinya, yaitu : algoritma simetris dan algoritma asimetris.

1. Algoritma Simetris

   Algoritma simetris adalah algoritma yang mempergunakan kunci yang sama dalam proses enkripsi dan dekripsinya. Aplikasi kriptografi simetri yang utama

adalah melindungi kerahasiaan data yang dikirim melalui saluran tidak aman dan melindungi kerahasiaan data yang disimpan pada media yang tidak aman. Kelemahan dari sistem ini adalah baik pengirim maupun penerima pesan harus memiliki kunci yang sama, sehingga pengirim pesan harus mencari cara yang aman untuk memberitahukan kunci kepada penerima pesan (Munir, 2006). Contoh algoritma kriptografi simetris adalah DES, Beaufort Cipher, Twofish, AES (Rijndael), Blowfish, GOST, dan lain-lain. Skema kriptografi simetri dapat dilihat pada Gambar 2.1.

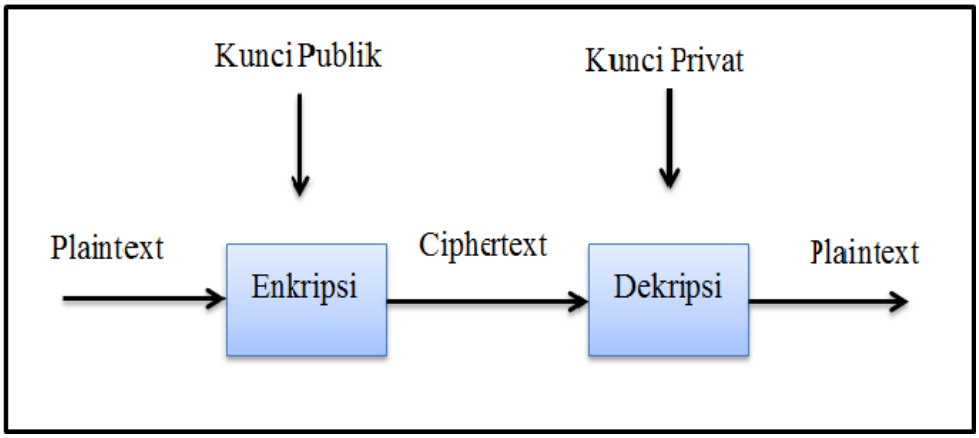


Gambar 2.2 Skema kriptografi simetri

2. Algoritma Asimetris

Algoritma Asimetris adalah algoritma kriptografi yang mempergunakan kunci yang berbeda pada enkripsi dan dekripsinya. Pada kriptografi asimetris kunci untuk enkripsi tidak rahasia dan dapat diketahui siapapun (diumumkan ke publik), sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan (karena itu rahasia). Pada kriptografi jenis ini, setiap orang yang berkomunikasi mempunyai sepasang kunci, yaitu kunci privat dan kunci publik. Pengirim mengenkripsi pesan dengan menggunakan kunci publik si penerima pesan (*receiver*). Hanya penerima pesan yang dapat mendekripsi pesan karena hanya ia yang mengetahui kunci

privatnya sendiri (Munir, 2006). Algoritma yang termasuk dalam algoritma asimetri adalah RSA, RSA-CRT, *Elgamal*, DSA, dsb. Skema kriptografi asimetri dapat dilihat pada gambar di bawah ini.



Gambar 2. 3 Skema kriptografis asimetris

Algoritma simetris dan asimetris memiliki keunggulan tersendiri dari masing-masing konsep kerjanya. Pada algoritma simetris, kecepatan operasi enkripsi dan dekripsi lebih tinggi dan ukuran kuncinya juga relatif pendek bila dibandingkan dengan algoritma asimetris. Namun algoritma asimetris memiliki manajemen kunci yang lebih baik. Tidak seperti algoritma simetris yang harus sering mengubah kunci setiap kali melaksanakan komunikasi, pasangan kunci privat dan kunci publik pada algoritma asimetris tidak perlu diubah dalam jangka waktu yang sangat lama.

2.1.5 Algoritma Kriptografi Klasik

Kriptografi klasik adalah algoritma yang sudah digunakan pada sejak zaman dahulu sebelum ditemukannya komputer. Kriptografi klasik dilakukan dengan cara mengacak huruf pada *plaintext*. Pada dasarnya kriptografi klasik dapat dikelompokkan menjadi dua macam *cipher*, yaitu sebagai berikut:

1. *Cipher* Substitusi

Cipher Substitusi adalah algoritma kriptografi yang mengubah sebuah karakter pada *plaintext* dengan sebuah karakter *ciphertext* (Setyaningsih, 2015). *Cipher* substitusi mempunyai berbagai macam algoritma yang berbeda-beda seperti *Vigenere Cipher, Caesar Cipher, dan Playfair Cipher.*

2. *Cipher* Transposisi

   Cipher Transposisi adalah mengubah urutan huruf *plaintext* atau melakukan *transpose* terhadap rangkaian karakter (Setyaningsih, 2015). *Cipher* transposisi mempunyai berbagai macam algoritma yang berbeda-beda seperti *Rail Fence Cipher, Myszkowski Transposition, Route Cipher, Columnar Transposition*.

2.2 File Teks

File Teks yaitu *file* yang di dalamnya berisi informasi-informasi dalam bentuk teks. Masukan dari data teks terdiri dari karakter, angka, huruf dan tanda baca. *Input* dan *output* data teks direpresentasikan sebagai sistem kode atau set karakter yang dikenal oleh sistem komputer. Pada sistem komputer terdapat beberapa macam set karakter yang biasa digunakan seperti *ASCII, Unicode, EBCDIC*. Salah satu standar internasional dalam kode huruf dan simbol yang bersifat *universal* yaitu *ASCII (American Code for Information Interchange)*. Berikut ini merupakan beberapa ekstensi dari *file* teks:

1. Teks Biasa (*.txt)

   Teks Biasa (*.txt) merupakan jenis berkas yang di dalamnya mengandung *editor* teks yang diformat menggunakan sistem kode pada *ASCII*. Berkas

ini hanya terdiri dari angka, karakter, tanda baca, tabulasi, dan pemisah baris untuk sistem *input* dan *output*.

2. *File* (*.doc)

   File (*.doc) pertama kali muncul pada tahun 1980 yang merupakan singkatan dari dokumen. Ukuran file (*.doc) lebih besar dibandingkan file (*.docx). File (*.doc) tidak mudah dikonversi tanpa bantuan *software* atau *external converter*.

3. *File* (*.docx)

   File (*.docx) merupakan file yang dikembangkan setelah versi (*.doc). Ukuran dokumen dari file (*.docx) lebih kecil dibandingkan dengan file (*.doc) sehingga lebih cepat dalam proses pengiriman. File (*.docx) dapat dengan mudah dikonversi ke format doc, html, rtf dan format lainnya.

4. *File* (*.pdf)

   File (*.pdf) atau *Portable Document Format* yaitu format berkas yang dibuat pada tahun 1993 untuk keperluan pertukaran dokumen digital. File (*.pdf) dapat digunakan untuk mempresentasikan dokumen dua dimensi yang meliputi teks, huruf, citra dan grafik vector dua dimensi.

2.3 Teori Bilangan

Teori bilangan (number theory) adalah teori yang mendasar dalam memahami algoritma kriptografi. Bilangan yang dimaksudkan adalah bilangan bulat (integer).

2.3.1 Bilangan Bulat

1. Bilangan bulat adalah bilangan yang tidak mempunyai pecahan desimal, misalnya $8, 21, 8765, -34, 0$
2. Berlawanan dengan bilangan bulat adalah bilangan riil yang mempunyai titik desimal, seperti $8.0, 34.25, 0.02$.

Sifat Pembagian pada Bilangan Bulat

1. Misalkan *a* dan *b* adalah dua buah bilangan bulat dengan syarat $a \neq 0$. Kita menyatakan bahwa *a* habis membagi *b* (*a divides b*) jika terdapat bilangan bulat *c* sedemikian sehingga *b* = *ac*.
2. Notasi: $a \mid b \; jika \; b = ac, c \in \boldsymbol{Z}$ dan $a \neq 0$. (**Z** = himpunan bilangan bulat).
3. Kadang-kadang pernyataan "*a* habis membagi *b*" ditulis juga "*b* kelipatan *a*".
4. Contoh 1: $4 \mid 12 \; karena \; 12 \div 4 = 3$ (bilangan bulat) atau $12 = 4 \times 3$. Tetapi $4 \mid 13$ karena $13 \div 4 = 3.25$ (bukan bilangan bulat).

Teorema 1 (Teorema Euclidean). Misalkan *m* dan *n* adalah dua buah bilangan bulat dengan syarat $n > 0$. Jika m dibagi dengan *n* maka terdapat dua buah bilangan bulat unik $q \; (quotient) \; dan \; r$ (*remainder*), sedemikian sehingga

$$m = nq + r \qquad (1)$$

dengan $0 \leq r < n$.

Contoh 2.

1. $1987 \; dibagi \; dengan \; 97$ memberikan hasil bagi $20 \; dan \; sisa \; 47$:

$$1987 = 97 \cdot 20 + 47$$

2. $-22\ dibagi\ dengan\ 3\ memberikan\ hasil\ bagi\ -8\ dan\ sisa\ 2$:

   $-22 = 3(-8) + 2$

   tetapi $-22 = 3(-7) - 1$ salah karena $r = -1$ tidak memenuhi syarat $0 \leq r < n$.

2.3.2 Pembagi Bersama Terbesar (PBB)

Misalkan *a* dan *b* adalah dua buah bilangan bulat tidak nol. Pembagi bersama terbesar (PBB – *greatest common divisor* atau *gcd)* dari *a* dan *b* adalah bilangan bulat terbesar d sedemikian sehingga $d \mid a\ dan\ d \mid b$. Dalam hal ini kita nyatakan bahwa PBB (*a*, *b*) = *d*.

Contoh 3.

Faktor pembagi 45: $1, 3, 5, 9, 15, 45$; Faktor pembagi 36: $1, 2, 3, 4, 9, 12, 18, 36$; Faktor pembagi bersama dari 45 dan 36 adalah $1, 3, 9$ PBB $(45, 36) = 9$.

Algoritma Euclidean

1. Algoritma Euclidean adalah algoritma untuk mencari PBB dari dua buah bilangan bulat.
2. Euclid, penemu algoritma Euclidean, adalah seorang matematikawan Yunani yang menuliskan algoritmanya tersebut dalam bukunya yang terkenal, *Element*.
3. Diberikan dua buah bilangan bulat tak-negatif $m\ dan\ n\ (m \geq n)$. Algoritma Euclidean berikut mencari pembagi bersama terbesar dari *m* dan *n*.

Algoritma Euclidean

1. Jika $n = 0$ maka

m adalah PBB (m, n);

stop.

tetapi jika $n \neq 0$,

lanjutkan ke langkah 2.

2. Bagilah *m* dengan *n* dan misalkan *r* adalah sisanya.
3. Ganti nilai *m* dengan nilai *n* dan nilai *n* dengan nilai *r*, lalu ulang kembali ke langkah 1.

2.3.3 Relatif Prima

1. Dua buah bilangan bulat *a* dan *b* dikatakan *relatif prima* jika PBB $(a, b) = 1$.
2. Contoh 5**.** 20 dan 3 relatif prima sebab PBB $(20, 3) = 1$. Begitu juga 7 dan 11 relatif prima karena PBB $(7, 11) = 1$. Tetapi 20 dan 5 tidak relatif prima sebab PBB $(20, 5) = 5 \neq 1$.
3. Jika *a* dan *b* relatif prima, maka terdapat bilangan bulat *m* dan *n* sedemikian sehingga

$$ma + nb = 1 \qquad (2)$$

Contoh 6**.** Bilangan 20 dan 3 adalah relatif prima karena PBB $(20, 3) = 1$, atau dapat ditulis

$$2 . 20 + (-13) . 3 = 1$$

dengan $m = 2$ dan $n = -13$. Tetapi 20 dan 5 tidak relatif prima karena PBB $(20, 5) = 5 \neq 1$ sehingga 20 dan 5 tidak dapat dinyatakan dalam $m . 20 + n . 5 = 1$.

2.3.3 Aritmatika Modulo

1. Misalkan *a* adalah bilangan bulat dan *m* adalah bilangan bulat > 0. Operasi *a* mod *m* (dibaca "*a* modulo *m*") memberikan sisa jika *a* dibagi dengan *m*.
2. Notasi: *a* mod *m* = *r* sedemikian sehingga $a = mq + r$, dengan $0 \leq r < m$.
3. Bilangan *m* disebut modulus atau modulo, dan hasil aritmetika modulo *m* terletak di dalam himpunan {0, 1, 2, …, *m* – 1} (mengapa?).

Contoh 7. Beberapa hasil operasi dengan operator modulo:

(i) $23 \; mod \; 5 = 3$ $(23 = 5 \cdot 4 + 3)$

(ii) $27 \; mod \; 3 = 0$ $(27 = 3 \cdot 9 + 0)$

(iii) $6 \; mod \; 8 = 6$ $(6 = 8 \cdot 0 + 6)$

(iv) $0 \; mod \; 12 = 0$ $(0 = 12 \cdot 0 + 0)$

(v) $-41 \; mod \; 9 = 4$ $(-41 = 9(-5) + 4)$

(vi) $-39 \; mod \; 13 = 0$ $(-39 = 13(-3) + 0)$

Penjelasan (*v*): Karena *a* negatif, bagi |*a*| dengan *m* mendapatkan sisa *r'*. Maka $a \; mod \; m = m - r' \; bila \; r' \neq 0$. Jadi $|-41| \; mod \; 9 = 5$, sehingga $-41 \; mod \; 9 = 9 - 5 = 4$.

Kongruen

1. Misalnya $38 \; mod \; 5 = 3$ dan $13 \; mod \; 5 = 3$, maka kita katakan $38 \equiv 13 \; (mod \; 5)$ (baca: 38 kongruen dengan 13 dalam modulo 5).
2. Misalkan *a* dan *b* adalah bilangan bulat dan *m* adalah bilangan > 0, maka $a \equiv b \; (mod \; m)$ jika *m* habis membagi $a - b$.
3. Jika a tidak kongruen dengan b dalam modulus *m*, maka ditulis $a \not\equiv b \; (mod \; m)$.

Contoh 8.

$17 \equiv 2 \ (mod\ 3)$ ($3\ habis\ membagi\ 17 - 2 = 15$)

$-7 \equiv 15 \ (mod\ 11)$ ($11\ habis\ membagi - 7 - 15 = -22$)

$12 \not\equiv 2 \ (mod\ 7)$ ($7\ tidak\ habis\ membagi\ 12 - 2 = 10$)

$-7 \not\equiv 15 \ (mod\ 3)$ ($3\ \ tidak\ habis\ membagi - 7 - 15 = -22$)

Kekongruenan a ≡ b (mod m) dapat pula dituliskan dalam hubungan

$$a = b + km \qquad (3)$$

yang dalam hal ini k adalah bilangan bulat.

Contoh 9.

$17 \equiv 2 \ (mod\ 3)$ dapat ditulis sebagai $17 = 2 + 5 \cdot 3$

$-7 \equiv 15 \ (mod\ 11)$ dapat ditulis sebagai $-7 = 15 + (-2)11$

Berdasarkan definisi aritmetika modulo, kita dapat menuliskan $a\ mod\ m = r$ sebagai $a \equiv r \ (mod\ m)$

Contoh 10**.**

Beberapa hasil operasi dengan operator modulo berikut:

(i) $23\ mod\ 5 = 3$ dapat ditulis sebagai $23 \equiv 3 \ (mod\ 5)$

(ii) $27\ mod\ 3 = 0$ dapat ditulis sebagai $27 \equiv 0 \ (mod\ 3)$

(iii) $6\ mod\ 8 = 6$ dapat ditulis sebagai $6 \equiv 6 \ (mod\ 8)$

(iv) $0\ mod\ 12 = 0$ dapat ditulis sebagai $0 \equiv 0 \ (mod\ 12)$

(v) $-41\ mod\ 9 = 4$ dapat ditulis sebagai $-41 \equiv 4 \ (mod\ 9)$

(vi) $39\ mod\ 13 = 0$ dapat ditulis sebagai $-39 \equiv 0 \ (mod\ 13)$

Teorema 2. Misalkan *m* adalah bilangan bulat positif.

1. Jika $a \equiv b \ (mod\ m)\ dan\ c$ adalah sembarang bilangan bulat maka

(i) $(a + c) \equiv (b + c) \, (mod \, m)$

(ii) $ac \equiv bc \, (mod \, m)$

(iii) $ap \equiv bp \, (mod \, m)$ untuk suatu bilangan bulat tak negatif *p*.

2. Jika $a \equiv b \, (mod \, m) \; dan \; c \equiv d \, (mod \, m)$, maka

(i) $(a + c) \equiv (b + d) \, (mod \, m)$

(ii) $ac \equiv bd \, (mod \, m)$

Bukti (hanya untuk 1(ii) dan 2(i) saja):

1. (ii) $a \equiv b \, (mod \, m)$ berarti:

$\Leftrightarrow a = b + km$

$\Leftrightarrow a - b = km$

$\Leftrightarrow (a - b)c = ckm$

$\Leftrightarrow ac = bc + Km$

$\Leftrightarrow ac \equiv bc \, (mod \, m)$

2. (i) $a \equiv b \, (mod \, m) \Leftrightarrow a = b + k_1 m$

$c \equiv d \, (mod \, m) \Leftrightarrow c = d + k_2 m$

$\Leftrightarrow (a + c) = (b + d) + (k_1 + k_2)m$

$\Leftrightarrow (a + c) = (b + d) + km \quad (k = k_1 + k_2)$

$\Leftrightarrow (a + c) = (b + d) \, (mod \, m)$

Contoh 11**.**

Misalkan 17 ≡ 2 (mod 3) dan 10 ≡ 4 (mod 3), maka menurut

Teorema 2,

$17 + 5 = 2 + 5 \, (mod \, 3) \qquad \Leftrightarrow \qquad 22 = 7 \, (mod \, 3)$

$17 \,.\, 5 \;=\; 5 \cdot 2 \,(mod\ 3) \qquad \Leftrightarrow \qquad 85 \;=\; 10 \,(mod\ 3)$

$17 \;+\; 10 \;=\; 2 \;+\; 4 \,(mod\ 3) \qquad \Leftrightarrow \qquad 27 \;=\; 6 \,(mod\ 3)$

$17 \,.\, 10 \;=\; 2 \cdot 4 \,(mod\ 3) \qquad \Leftrightarrow \qquad 170 \;=\; 8 \,(mod\ 3)$

Perhatikanlah bahwa Teorema 2 tidak memasukkan operasi pembagian pada aritmetika modulo karena jika kedua ruas dibagi dengan bilangan bulat, maka kekongruenan tidak selalu dipenuhi. Misalnya:

1. $10 \;\equiv\; 4 \,(mod\ 3)$ dapat dibagi dengan 2 karena $10/2 \;=\; 5$ dan $4/2 \;=\; 2$, dan $5 \;\equiv\; 2 \,(mod\ 3)$
2. $14 \;\equiv\; 8 \,(mod\ 6)$ tidak dapat dibagi dengan 2, karena $14/2 \;=\; 7$ dan $8/2 \;=\; 4$, tetapi $7 \;\not\equiv\; 4 \,(mod\ 6)$.

Balikan Modulo (modulo invers)

Jika *a* dan *m* relatif prima dan $m \;>\; 1$, maka kita dapat menemukan balikan (invers) dari a modulo *m*. Balikan dari *a* modulo *m* adalah bilangan bulat $\bar{a}$ sedemikian sehingga

$$\bar{a}a \;\equiv\; 1 \,(mod\ m)$$

Bukti: Dari definisi relatif prima diketahui bahwa PBB $(a, m) \;=\; 1$, dan menurut persamaan (2) terdapat bilangan bulat *p* dan *q* sedemikian sehingga

$$pa \;+\; qm \;=\; 1$$

yang mengimplikasikan bahwa

$$pa \;+\; qm \;\equiv\; 1 \,(mod\ m)$$

Karena $qm \;\equiv\; 0 \,(mod\ m)$, maka

$pa \;\equiv\; 1 \,(mod\ m)$

Kekongruenan yang terakhir ini berarti bahwa *p* adalah balikan dari *a* modulo *m*.

Pembuktian di atas juga menceritakan bahwa untuk mencari balikan dari a modulo m, kita harus membuat kombinasi lanjar dari a dan m sama dengan 1. Koefisien a dari kombinasi lanjar tersebut merupakan balikan dari a modulo m.

Contoh 12**.**

Tentukan balikan dari $4\ (mod\ 9), 17\ (mod\ 7), dan\ 18\ (mod\ 10)$.

Penyelesaian:

1. Karena PBB $(4, 9) = 1$, maka balikan dari $4\ (mod\ 9)$ ada. Dari algoritma Euclidean diperoleh bahwa

   $$9 = 2 \cdot 4 + 1$$

   Susun persamaan di atas menjadi

   $$-2 \cdot 4 + 1 \cdot 9 = 1$$

   Dari persamaan terakhir ini kita peroleh –2 adalah balikan dari 4 modulo 9. Periksalah bahwa

   $-2 \cdot 4 \equiv 1\ (mod\ 9)$ $(9\ habis\ membagi\ -2 \cdot 4 - 1 = -9)$

2. Karena PBB(17, 7) = 1, maka balikan dari 17 (mod 7) ada. Dari algoritma Euclidean diperoleh rangkaian pembagian berikut:

   $17 = 2 \cdot 7 + 3$ (i)

   $7 = 2 \cdot 3 + 1$ (ii)

   $3 = 3 \cdot 1 + 0$ (iii) (yang berarti: PBB $(17, 7) = 1$))

   Susun (ii) menjadi:

   $1 = 7 - 2 \cdot 3$ (iv)

   Susun (i) menjadi

   $3 = 17 - 2 \cdot 7$ (v)

   Sulihkan (v) ke dalam (iv):

$1 = 7 - 2 \cdot (17 - 2 \cdot 7) = 1 \cdot 7 - 2 \cdot 17 + 4 \cdot 7 = 5 \cdot 7 - 2 \cdot 17$ atau $-2 \cdot 17 + 5 \cdot 7 = 1$

Dari persamaan terakhir ini kita peroleh –2 adalah balikan dari 17 modulo 7.

$-2 \cdot 17 \equiv 1 \ (mod\ 7)$ $\quad (7\ habis\ membagi -2 \cdot 17 - 1 = -35)$

3. Karena PBB $(18, 10) = 2 \neq 1$, maka balikan dari 18 $(mod\ 10)$ tidak ada.

2.3.5 Aritmetika Modulo dan Kriptografi

Aritmetika modulo cocok digunakan untuk kriptografi karena dua alasan:

1. Oleh karena nilai-nilai aritmetika modulo berada dalam himpunan berhingga (0 sampai modulus m – 1), maka kita tidak perlu khawatir hasil perhitungan berada di luar himpunan.
2. Karena kita bekerja dengan bilangan bulat, maka kita tidak khawatir kehilangan informasi akibat pembulatan (round off) sebagaimana pada operasi bilangan riil.

2.3.6 Bilangan Prima

1. Bilangan bulat positif p $(p > 1)$ disebut bilangan prima jika pembaginya hanya 1 $dan\ p$.

   Contoh: 23 adalah bilangan prima karena ia hanya habis dibagi oleh 1 dan 23.

2. Karena bilangan prima harus lebih besar dari 1, maka barisan bilangan prima dimulai dari 2, yaitu $2, 3, 5, 7, 11, 13$, seluruh bilangan prima adalah bilangan ganjil, kecuali 2 yang merupakan bilangan genap.

3. Bilangan selain prima disebut bilangan komposit (*composite*). Misalnya 20 adalah bilangan komposit karena 20 dapat dibagi oleh 2, 4, 5, dan 10, selain 1 dan 20 sendiri.

Teorema 3. ***(The Fundamental Theorem of Arithmetic).*** Setiap bilangan bulat positif yang lebih besar atau sama dengan 2 dapat dinyatakan sebagai perkalian satu atau lebih bilangan prima.

Contoh 15**.**

$9 = 3 \times 3$ (2 buah faktor prima)

$100 = 2 \times 2 \times 5 \times 5$ (4 buah faktor prima)

$13 = 13 \quad (atau\ 1 \times 13)$ (1 buah faktor prima)

Untuk menguji apakah *n* merupakan bilangan prima atau komposit, kita cukup membagi *n* dengan sejumlah bilangan prima, mulai dari 2, 3, …, bilangan prima $\leq \sqrt{n}$. Jika *n* habis dibagi dengan salah satu dari bilangan prima tersebut, maka *n* adalah bilangan komposit, tetapi jika *n* tidak habis dibagi oleh semua bilangan prima tersebut, maka *n* adalah bilangan prima.

Contoh 16**.**

Tunjukkan apakah (i) 171 dan (ii) 199 merupakan bilangan prima atau komposit.

Penyelesaian :

1. $\sqrt{171} = 13.077$. Bilangan prima yang $\leq \sqrt{171}$ adalah $2, 3, 5, 7, 11, 13$. Karena 171 habis dibagi 3, maka 171 adalah bilangan komposit.
2. $\sqrt{199} = 14.107$. Bilangan prima yang $\leq \sqrt{199}$ adalah $2, 3, 5, 7, 11, 13$. Karena 199 tidak habis dibagi $2, 3, 5, 7, 11, dan\ 13$, maka 199 adalah bilangan prima.

Terdapat metode lain yang dapat digunakan untuk menguji keprimaan suatu bilangan bulat, yang terkenal dengan **Teorema Fermat**. Fermat (dibaca "Fair-ma") adalah seorang matematikawan Perancis pada tahun 1640.

Teorema 4 (Teorema Fermat). Jika *p* adalah bilangan prima dan *a* adalah bilangan bulat yang tidak habis dibagi dengan *p*, yaitu

PBB $(a, p) = 1$, maka

$$a^{p-1} \equiv 1 \ (mod\ p)$$

Contoh 17**.**

Kita akan menguji apakah 17 dan 21 bilangan prima atau bukan.

Di sini kita mengambil nilai $a = 2$ karena PBB $(17, 2) = 1$ dan

PBB $(21, 2) = 1$. Untuk 17,

$$217-1 = 65536 \equiv 1 \ (mod\ 17)$$

karena 17 tidak membagi $65536 - 1 = 65535 \quad (65535 \div 17 = 3855)$.

Untuk 21,

$$221-1 = 1048576 \equiv\backslash\ 1 \ (mod\ 21)$$

karena 21 tidak habis membagi $1048576 - 1 = 1048575$.

Kelemahan Teorema Fermat: terdapat bilangan komposit *n* sedemikian sehingga $2^{n=1} \equiv 1 \ (mod\ n)$. Bilangan bulat seperti itu disebut bilangan prima semu *(pseudoprimes).* Misalnya komposit 341 $(yaitu\ 341 = 11 \cdot 31)$ adalah bilangan prima semu karena menurut teorema Fermat,

$$2340 \equiv 1 \ (mod\ 341)$$

Untunglah bilangan prima semu relatif jarang terdapat.

Fungsi Euler f

Fungsi Euler φ medefinisikan $\varphi(n)$ untuk $n \geq 1$ yang menyatakan jumlah bilangan bulat positif <*n* yang relatif prima dengan *n*.

Contoh 18

Tentukan $\varphi(20)$.

Penyelesaian:

Bilangan bulat positif yang lebih kecil dari 20 adalah 1 sampai 19. Di antara bilangan-bilangan tersebut, terdapat $\varphi(20) = 8$ buah yang relatif prima dengan 20, yaitu 1, 3, 7, 9, 11, 13, 17, 19.

Untuk *n* = 1, 2, …, 10, fungsi Euler adalah

$\varphi(1) = 0$ $\varphi(6) = 2$

$\varphi(2) = 1$ $\varphi(7) = 6$

$\varphi(3) = 2$ $\varphi(8) = 4$

$\varphi(4) = 2$ $\varphi(9) = 6$

$\varphi(5) = 4$ $\varphi(10) = 4$

Jika *n* prima, maka setiap bilangan bulat yang lebih kecil dari *n* relatif prima terhadap *n*. Dengan kata lain, $\varphi(n) = n - 1$ hanya jika *n* prima.

Contoh 19.

$\varphi(3) = 2, \varphi(5) = 4, \varphi(7) = 6, \varphi(11) = 10, \varphi(13) = 12$, dst.

Teorema 5. Jika *n* = *pq* adalah bilangan komposit dengan *p* dan *q* prima, maka $\varphi(n) = \varphi(p)\,\varphi(q) = (p - 1)(q - 1)$.

Contoh 20.

Tentukan $\varphi(21)$.

Penyelesaian :

Karena $21 = 7 \cdot 3, \varphi(21) = \varphi(7)\,\varphi(3) = 6 \cdot 2 = 12$ buah bilangan

bulat yang relatif prima terhadap 21, yaitu $1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20$.

Teorema 6. Jika p bilangan prima dan $k > 0$, maka $\varphi(pk) = pk - pk - 1 = pk - 1(p - 1)$.

Contoh 21.

Tentukan $\varphi(16)$.

Penyelesaian:

Karena $\varphi(16) = \varphi(24) = 24 - 23 = 16 - 8 = 8$, maka ada delapan buah bilangan bulat yang relatif prima terhadap 16, yaitu $1, 3, 5, 7, 9, 11, 13, 15$.

Teorema 7 (*Euler's generalization of Fermat theorem*). Jika

PBB $(a, n) = 1$, maka $a\varphi(n) \bmod n = 1$ $(atau\ a\varphi(n) \equiv 1\ (mod\ n))$

2.4 Algoritma One Time Pad Cipher

Algoritma *One Time Pad Cipher* adalah sebuah metode yang menerapkan algoritma kunci simetris atau proses enkripsi dan dekripsi menggunakan kunci yang acak. Kerahasiaan kunci merupakan faktor utama dalam penentuan keamanan atau pesan yang dikirimkan. Algoritma *One Time Pad Cipher* diciptakan oleh Mayor J. Maugboune dan Gilbert Vernam pada tahun 1917.

Algoritma *One Time Pad Cipher* juga bisa disebut dengan algoritma *Unbreakable Cipher*. Hal ini dikarenakan sifat dari algoritma ini kunci harus berupa barisan nilai yang seluruhnya acak sempurna (*truly random)* dan juga panjang dari kunci algoritma ini haruslah sama dengan panjang *plainteks* nya. Dari sifat-sifat yang ada pada algoritma *One Time Pad* ini menyebabkan beberapa *plainteks* yang sama belum tentu bisa dienkripsi menjadi *cipherteks* yang sama pula. Maksudnya adalah kriptanalis akan mendapatkan hasil bahwa sebuah *cipherteks* yang

didekripsinya mungkin menghasilkan beberapa *plainteks* berbeda namun memiliki makna. Hal ini akan membingungkannya dalam menentukan *plainteks* mana yang benar. *Unbreakable Cipher* dikatakan memiliki tingkat kerahasiaan yang sempurna (*perfect secrecy).* Satu-satunya algoritma kriptografi sempurna aman dan tidak dapat dipecahkan adalah *One Time Pad Cipher.*

Algoritma *One Time Pad* adalah stream *cipher* yang melakukan enkripsi dan dekripsi satu karakter setiap kali. Algoritma ini merupakan perbaikan dari *Vernamcipher* untuk menghasilkan keamanan yang sempurna. *Cipher* ini termasuk ke dalam kelompok algoritma kriptografi simetri. *One Time Pad* (*pad* = kertas bloknot) berisi barisan karakter-karakter kunci yang dibangkitkan secara acak. Satu buah *One Time Pad* adalah sebuah pita (tape) yang berisi barisan karakter-karakter kunci. Satu pad hanya digunakan sekali (*one time*) saja untuk mengenkripsi pesan, setelah itu pad yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain. Prinsip enkripsi pada algoritma ini adalah dengan mengkombinasikan masing-masing karakter pada *plaintext* dengan satu karakter pada kunci. Oleh karena itu, panjang kunci harus sama dengan panjang *plaintext*. Enkripsi dapat dinyatakan sebagai penjumlahan modulo 26 (menggunakan kode ASCII) dari satu bit *ciphertext* dengan satu bit kunci.

Contoh penerapan:

1. Karakter pembentuk *plaintext* dan *ciphertext* yang digunakan adalah seluruh abjad romawi yaitu 26 huruf(A-Z) dengan nomor index karakter 0-25, maka nilai modulus yang digunakan modulo 26.

2. Dilakukan proses enkripsi dengan operasi matematika penjumlahan, sementara untuk dekripsi menggunakan operasi matematika pengurangan.Penggunaan *One Time Pad* berikut yang digunakan penulis menggunakan tabel ASCII, berikut merupakan tabel ASCII.

KARAKTER	ASCII CODE
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7
I	8
J	9
K	10
L	11
M	12
N	13
O	14
P	15
Q	16
R	17
S	18
T	19
U	20
V	21
W	22
X	23
Y	24
Z	25

Tabel 2.1 *One Time Pad* Sesuai Tabel ASCII

Aturan enkripsi yang digunakan pada algoritma *One Time Pad Cipher* sangatlah persis dengan aturan enkripsi pada algoritma *Vignere Cipher*. Pengirim pesan menggunakan setiap karakter kunci untuk mengenkripsikan satu karakter

plainteks. Enkripsi dapat dinyatakan sebagai penjumlahan modulo 26 dari satu karakter *plainteks* dengan satu karakter kunci *One Time Pad.*

Berikut persamaan dari enkripsi *One Time Pad* yaitu :

Ci = (Pi + Ki) mod 26 (1)

Yang dalam hal ini, p_i adalah *plainteks* ke-*i, Ki* adalah huruf kunci ke-I dan *Ci* adalah huruf *cipherteks* ke- *i*. Perhatikan bahwa panjang kunci sama dengan panjang *plainteks,* sehingga tidak ada kebutuhan mengulang penggunaan kunci selama enkripsi.

Setelah pengirim mengenkripsi pesan dengan kunci, ia menghancurkan kunci tersebut (makanya disebut satu kali pakai atau *One Time Pad*). Penerima pesan menggunakan kunci yang sama untuk mendeskripsikan karakter-karakter *cipherteks* menjadi karakter-karakter dan persamaan dekripsi dari *One Time Pad* yaitu :

$P_i = (C_i - K_i) \; mod \; 26$ (2)

Keterangan persamaan :

C_i = Pergeseran karakter pada *cipherteks(Ciphertext),*

P_i = Pergeseran karakter pada *plainteks(Plaintext),*

K_i = Kunci dalam bentuk decimal yang dihasilkan dari table konversi.

Dimana P_i adalah rangkaian *plaintext*, K_i adalah *key*, C_i adalah *ciphertext* yang diperoleh dan *n* adalah jumlah karakter yang digunakan. *Key* yang digunakan pada algoritma *One Time Pad* diambil secara acak dan harus memiliki panjang karakter yang sama dengan *plaintext* (Mollin, 2007).

Contoh 1. Berdasarkan persamaan (1), proses enkripsi di algoritma *One Time Pad* sebagai berikut :

Plaintext : "GUTEN"

Key : "SIANG"

Karena sifat dari *One Time Pad,* jumlah *key* harus sama panjangnya dengan *plainteks.*

Langkah selanjutnya yaitu *plainteks* dan *kunci* diubah menjadi angka sesuai dengan telah yang diberikan, berikut ini adalah proses enkripsinya :

$C_1 = (P_1 + K_1) mod\ 26$

$= (6 + 18) mod\ 26$

$= (24) mod\ 26$

$C_1 = 24$

$maka\ C_1 = 24$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 24\ adalah\ \boldsymbol{Y}$

$C_2 = (P_2 + K_2) mod\ 26$

$= (20 + 8) mod\ 26$

$= (2) mod\ 26$

$C_2 = 2$

$maka\ C_2 = 2$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 2\ adalah\ \boldsymbol{C}$

$C_3 = (P_3 + K_3) mod\ 26$

$= (19 + 0) mod\ 26$

$= (19) mod\ 26$

$C_3 = 19$

$maka\ C_3 = 19$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 19\ adalah\ \boldsymbol{T}$

$C_4 = (P_4 + K_4) mod\ 26$

$= (4 + 13) mod\ 26$

$= (17) mod\ 26$

$C_4 = 17$

$maka\ C_4 = 17$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 17\ adalah\ \boldsymbol{R}$

$C_5 = (P_5 + K_5) mod\ 26$

$= (13 + 6) mod\ 26$

$= (19) mod\ 26$

$C_5 = 19$

$maka\ C_5 = 19$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 19\ adalah\ \boldsymbol{T}$

Setelah melakukan proses enkripsi *One Time Pad* seperti diatas, maka *cipherteks* hasil dari proses enkripsi tersebut adalah

Cipherteks = "YCTRT"

Lalu hasil dari *cipherteks* tersebut akan digunakan untuk menghitung persamaan dari dekripsi (2) *One Time Pad.* Berikut merupakan proses menghitung persamaan dekripsinya:

Cipherteks = "YCTRT"

Key = "SIANG"

Langkah selanjutnya yaitu *cipherteks* dan *key* diubah menjadi angka sesuai dengan telah yang diberikan, berikut ini adalah proses dekripsinya :

$P_1 = (C_1 - K_1) mod\ 26$

$= (24 - 18) mod\ 26$

$= (6) mod\ 26$

$P_1 = 6$

$maka\ P_1 = 6$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 6\ adalah\ \boldsymbol{G}$

$P_2 = (C_2 - K_2) mod\ 26$

$= (2 - 8) mod\ 26$

$= (-6) mod\ 26$

$P_2 = 20$

$maka\ P_2 = 20$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 20\ adalah\ \boldsymbol{U}$

$P_3 = (C_3 - K_3) mod\ 26$

$= (19 - 0) mod\ 26$

$= (19) mod\ 26$

$P_3 = 19$

$maka\ P_3 = 19$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 19\ adalah\ \boldsymbol{T}$

$P_4 = (C_4 - K_4) mod\ 26$

$= (17 - 13) mod\ 26$

$= (4) mod\ 26$

$P_4 = 4$

$maka\ P_4 = 4$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 4\ adalah\ \boldsymbol{E}$

$P_5 = (C_5 - K_5) mod\ 26$

$= (19 - 6) mod\ 26$

$= (13) mod\ 26$

$P_5 = 13$

$maka\ P_5 = 13$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 13\ adalah\ \boldsymbol{N}$

Dengan demikian hasil proses dekripsi dengan menggunakan persamaan (2) adalah "GUTEN" atau sesuai dengan *plainteks* diberikan di awal sebelum melakukan proses enkripsi.

2.5 Algoritma *Myszkowski Cipher*

Myszkowski Transposisi Cipher merupakan salah satu jenis algoritma transposisi cipher yang memiliki keunikan tersendiri. Pada proses enkripsi, *plaintext* ditulis secara horizontal dari kiri kekanan, kemudian *ciphertext* dibaca secara vertikal sesuai dengan urutan kunci. Adapun algoritma *Myszkowski Transposisi Cipher* adalah sebagai berikut :

1. Enkripsi

Sebelum melakukan proses enkripsi, terlebih dahulu dilakukan pembentukan kunci. Beberapa huruf yang dibentuk secara manual ataupun acak dapat menambah variasi pembentukan kunci. Misalkan terdapat *plaintext* dan kunci sebagai berikut :

Plaintext : NURUL KHAIRINA

Kunci : UMA

Proses enkripsi dimulai dengan membentuk sejumlah baris dan kolom untuk menampung *plaintext.* Terdapat 13 huruf pada *plaintext* yang akan menjadi acuan

dalam membentuk baris,dan 3 huruf pada kunci akan menjadi acuan untuk membentuk kolom. Sehingga jumlah kolom dan baris yang dibutuhkan adalah :

Kunci = 3 huruf => 3 kolom

Plaintext = 13 huruf => 13/3 = 4.3 => 5 baris

Kunci pada contoh diatas terdiri dari 3 huruf,sehingga dapat kita beri penomoran sesuai urutan abjad,yaitu U = 3; M = 2; A = 1.

Setelah membentuk baris dan kolom yang memungkinkan,*plaintext* dapat ditulis secara berurutan dan horizontal. Secara rinci dapat dilihat pada tabel berikut :

U	**M**	**A**
3	**2**	**1**
N		

(a)

U	**M**	**A**
3	**2**	**1**
N	U	R

(b)

U	**M**	**A**
3	**2**	**1**
N	**U**	**R**
U		

(c)

U	**M**	**A**
3	**2**	**1**
N	**U**	**R**
U	**L**	**K**
H	**A**	**I**
R	**I**	**N**
A	**X**	**X**

(d)

Tabel 2.2 Proses Enkripsi (a), (b), (c). Hasil Enkripsi (d).

Pada Tabel 2.2 dua baris teratas merupakan kunci dan penomoran kunci yang digunakan dalam proses enkripsi, kemudian 2 huruf X terakhir merupakan *dummy* yang digunakan untuk memenuhi kotak kosong yang tidak terisi oleh *plaintext.*

Dari proses enkripsi tersebut, *ciphertext* dapat diperoleh dengan membaca huruf secara vertikal dari atas kebawah sesuai dengan penomoran kunci.

Kunci	**1**	**2**	**3**
Ciphertext	**RAKINX**	**ULAIX**	**NUHRA**

Sehingga *Ciphertext* menjadi :

Ciphertext : RAKINX ULAIX NUHRA

Keunikan algoritma Myszkowski terlihat apabila terdapat kunci yang memiliki huruf yang berulang. Pada kasus ini, *Ciphertext* akan dibaca secara horizontal. Misalkan :

Plaintext : NURUL KHAIRINA

Kunci : OKTOBER

Proses enkripsi :

O	**K**	**T**	**O**	**B**	**E**	**R**
4	**3**	**6**	**4**	**1**	**2**	**5**
N	**U**	**R**	**U**	**L**	**K**	**H**
A	**I**	**R**	**I**	**N**	**A**	**X**

Tabel 2.3 Variasi Enkripsi

Ciphertext : LN KA UI NU UI HX RR

Dapat dilihat pada tabel 2.3, terdapat dua huruf kunci yang sama yaitu huruf O, dengan penomoran kunci = 4. Pembacaan *ciphertext* tidak dilakukan secara vertikal, melainkan secara horizontal dan menghasilakan NU AI.

O	**O**
4	**4**
N	**U**
A	**I**

Tabel 2.4 Pembacaan *Ciphertext* secara horizontal terhadap huruf kunci yang sama.

2. Dekripsi :

Proses dekripsi dapat dilakukan apabila pihak penerima pesan rahasia (*ciphertext*) mengetahui pola kunci dan ukuran baris dan kolom yang digunakan pihak pengirim pesan. Proses dekripsi dapat dilakukan dengan menuliskan

ciphertext secara vertikal dari atas kebawah secara berurutan sesuai dengan penomoran kunci. Pada penomoran kunci yang sama, *ciphertext* ditulis secara horizontal.

Ciphertext : LN KA UI NU AI HX RR

Kunci : OKTOBER

Proses Dekripsi :

O	K	T	O	B	E	R
4	3	6	4	1	2	5
				L		
				N		

O	K	T	O	B	E	R
4	3	6	4	1	2	5
				L	K	
				N	A	

O	K	T	O	B	E	R
4	3	6	4	1	2	5
	U			L	K	
	I			N	A	

O	K	T	O	B	E	R
4	3	6	4	1	2	5
N	U		U	L	K	
	I			N	A	

O	K	T	O	B	E	R
4	3	6	4	1	2	5
N	U		U	L	K	
A	I			N	A	

O	K	T	O	B	E	R
4	3	6	4	1	2	5
N	U	R	U	L	K	H
A	I	R	I	N	A	X

Tabel 2.5 Proses Dekripsi

Dari tabel 2.5, *plaintext* dapat diperoleh dengan membaca huruf dari kolom paling kiri tanpa memperdulikan penomoran kunci, sehingga diperoleh :

Plaintext : NURUL KHAIRINA

2.6 Super Enkripsi

Algoritma kriptografi memberikan keamanan namun tidak menjamin keamanan 100 persen, sehingga diajukan solusi yang dirancang untuk meningkatkan keamanan tersebut, melalui kombinasi penggunaan algoritma kriptografi yang berbeda untuk mengenkripsi pesan.

Multiple encryption, dimana salah satu contohnya adalah *double* enkripsi (super enkripsi) adalah proses enkripsi yang dilakukan sebanyak dua kali atau lebih. Pertama enkripsi *plaintext* menjadi *ciphertext*, kemudian enkripsi *ciphertext* itu, mungkin menggunakan *cipher* lain dan kunci.

Super enkripsi adalah salah satu kriptografi berbasis karakter yang menggabungkan *cipher* substitusi dan *cipher* transposisi. Hal tersebut bertujuan utnuk mendapatkan *cipher* yang lebih kuat dari hanya menggunakan satu *cipher* saja, sehingga tidak mudah untuk dipecahkan. enkripsi dan dekripsi dapat dilakukan dengan urutan *cipher* substitusi kemudian *cipher* transposisi, atau sebaliknya. Super enkripsi dapat dilakukan dengan menggunakan kedua *cipher* tersebut secara berulang-ulang, namun pada makalah ini hanya akan dibahas mengenai proses enkripsi dan dekripsi satu kali dengan menggunakan *cipher* substitusi dan satu kali dengan menggunakan *cipher* transposisi.

2.6.1 Enkripsi

Super enkripsi dalam melakukan proses enkripsi dapat menggunakan kedua *cipher* tersebut secara berurutan. Misalnya ada sebuah *plainteks* sebagai berikut.

SAYA BERADA DI BANDUNG

Plainteks tersebut akan dienkripsi dengan menggunakan kunci k = 3. Mula-mula lakukan enkripsi dengan menggunakan *cipher* substitusi sehingga akan didapatkan *ciphertext* sebagai berikut.

VDBDEHUDGDGLEDQGXQJXX

Selanjutnya enkripsi kembali *ciphertext* tersebut dengan menggunakan *cipher* transposisi dengan panjang kunci yang sama, yaitu 3 sehingga akan didapatkan hasil sebagai berikut.

V D B	D E H	U D G
DG L	E D Q	G X Q J X X

Di akhir dari hasil *ciphertext* tersebut ditambahkan dua buah karakter tambahan, yaitu 2 buah huruf X. Huruf X dipilih karena jumlahnya hanya 1 buah saja. Karena cipherteks tersebut didapatkan juga dengan menggunakan *cipher* substitusi, pemilihan huruf X dapat menyulitkan kriptanalisis untuk memecahkan cipherteks tersebut dengan menggunakan metode analisis frekuensi karena adanya perubahan jumlah untuk jumlah karakter X. Selanjutnya untuk mencari hasil dari proses enkripsi tersebut hanya perlu membaca karakter dari blok per blok di atas dan akan didapat *ciphertext* akhir sebagai berikut.

VDUDEGJDEDGDXXBHGLQQX

2.6.2 Dekripsi

Untuk mengembalikan cipherteks tersebut menjadi plainteks yang memiliki makna,kita hanya perlu melakukan dekripsi secara berurutan dengan menggunakan *cipher* substitusi dan *cipher* transposisi namun urutannya dekripsinya ditukar.

Mula-mula lakukan dekrispi dengan menggunakan *cipher* transposisi dengan jumlah kolom yang ada adalah 21 dibagi 3, yaitu 7 sehingga akan didapatkan blok-blok sebagai berikut:

V D U D E G J

D E D G D X X

B H G L Q Q X

Berdasarkan blok yang ada di atas, akan didapatkan *ciphertext* baru sebagai berikut:

VDBDEHUDGDGLEDQGXQJXX

Karena kita tidak tahu apakah dua karakter di akhir merupakan karakter tambahan atau bukan, maka kita tidak bisa langsung menghilangkan karakter tersebut. Selanjutnya *ciphertext* tersebut didekripsi sekali lagi dengan menggunakan *cipher* substitusi dengan panjang kunci $k = 3$ sehingga akan kita dapatkan *plainteks* sebagai berikut:

SAYABERADADIBANDUNGUU

Dengan cepat dapat kita pisah plainteks tersebut menjadi susunan kata yang memiliki makna sebagai berikut.

SAYA BERADA DI BANDUNG UU

Saat ini dapat dipastikan bahwa dua huruf di belakang plainteks adalah karakter tambahan karena kata tersebut tidak memiliki makna yang bersesuaian dengan isi *plainteks* yang lain sehingga kita bisa menghilangkannya.

Bila kita tidak menambahkan karakter tambahan di ujung plainteks, maka akan didapat *ciphertext* yang jumlah karakternya sama dengan jumlah karakter pada plainteks awal sehingga cipherteks skhir yang didapat adalah sebagai berikut:

VDUDEGJDEDGDXBHGLQQ

Dan bila dilakukan dekripsi terhadap *ciphertext* tersebut dengan menggunakan *cipher* transposisi, maka akan didapatkan *ciphertext* baru, yaitu sebagai berikut.

VDBDEHUDGDGLEDQGXQJ

Dan dapat dipastikan bahwa tidak ada karakter tambahan pada *ciphertext* tersebut sehingga kita hanya perlu untuk melakukan dekripsi dengan menggunakan *cipher* substitusi sehingga akan didapatkan plainteks mula-mula tanpa adanya karakter tambahan.

SAYABERADADIBANDUNG

Plainteks tersebut kemudian dipisahkan menjadi katakata dalam Bahasa Indonesia yang dapat diketahui, yaitu:

SAYA BERADA DI BANDUNG

Untuk melakukan enkripsi dan dekripsi dengan urutan yang sebaliknya, proses yang dilakukan sama, namun urutannya terbalik.

BAB III
PEMBAHASAN

3.1 Penyandian Algoritma *One Time Pad Cipher*

Bentuk umum penyandian algoritma *One Time Pad Cipher* adalah menggunakan karakter pembentuk *plaintext* dan *ciphertext* yang menggunakan seluruh abjad romawi yaitu 26 huruf (A-Z) dengan nomor index karakter 0-25, maka nilai modulus yang digunakan modulo 26. Berikut ini proses enkripsi algoritma *One Time Pad Cipher* di mana terdapat sebuah *plaintext* "KHOFIFAH" dengan key " WLEIQHTO"

Plaintext = "KHOFIFAH"

Key = "WLEIQHTO"

Plaintext dan *key* yang diberikan di atas memang harus sama panjang, hal ini merupakan sifat dari algoritma *One Time Pad Cipher* tersebut.

Langkah selanjutnya yaitu *plaintext* dan key di atas disandikan sesuai dengan barisan huruf telah yang diberikan, berikut ini adalah proses enkripsinya :

$C_1 = (P_1 + K_1) mod\ 26$

$= (10 + 22) mod\ 26$

$= (32) mod\ 26$

$C_1 = 6$

$maka\ C_1 = 6$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 6\ adalah\ \boldsymbol{G}$

$C_2 = (P_2 + K_2) mod\ 26$

$= (7 + 11) mod\ 26$

$= (18) mod\ 26$

$C_2 = 18$

$maka\ C_2 = 18$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 18\ adalah\ \boldsymbol{S}$

$C_3 = (P_3 + K_3) mod\ 26$

$= (14 + 4) mod\ 26$

$= (18) mod\ 26$

$C_3 = 18$

$maka\ C_3 = 18$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 18\ adalah\ \boldsymbol{S}$

$C_4 = (P_4 + K_4) mod\ 26$

$= (5 + 8) mod\ 26$

$= (13) mod\ 26$

$C_4 = 13$

$maka\ C_4 = 13$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 13\ adalah\ \boldsymbol{N}$

$C_5 = (P_5 + K_5) mod\ 26$

$= (8 + 16) mod\ 26$

$= (24) mod\ 26$

$C_5 = 24$

$maka\ C_5 = 24$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 24\ adalah\ \boldsymbol{Y}$

$C_6 = (P_6 + K_6) mod\ 26$

$= (5 + 7) mod\ 26$

$= (12) mod\ 26$

$C_6 = 12$

$maka\ C_6 = 12$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 12\ adalah\ \boldsymbol{M}$

$C_7 = (P_7 + K_7) mod\ 26$

$= (0 + 19) mod\ 26$

$= (19) mod\ 26$

$C_7 = 19$

$maka\ C_7 = 19$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 19\ adalah\ \boldsymbol{T}$

$C_8 = (P_8 + K_8) mod\ 26$

$= (7 + 14) mod\ 26$

$= (21) mod\ 26$

$C_8 = 21$

$maka\ C_8 = 21$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 21\ adalah\ \boldsymbol{V}$

Plainteks	K	H	O	F	I	F	A	H
Key	W	L	E	I	Q	H	T	O
Cipherteks	G	S	S	N	Y	M	T	V

Tabel 3.1 Hasil persamaan enkripsi *One Time Pad*

Jadi, hasil enkripsi *plaintext* "KHOFIFAH" adalah "GSSNYMTV"

Proses dekripsi pada metode algoritma *One Time Pad Cipher* adalah kebalikan atau mengembalikan plainteks menjadi data semula, dapat diperlihatkan dengan menggunakan persamaan sebagai berikut:

$P_1 = (C_1 - K_1) mod\ 26$

Dengan melakukan konsep yang sama maka didapatkan hasil sebagai berikut :

Plaintext = "KHOFIFAH"

Key = "GSSNYMTV"

Ciphertext = "GSSNYMTV"

Proses dekripsi dapat dilihat pada perhitungan dibawah ini :

Ciphertext = ”GSSNYMTV”

Key = ”WLEIQHTO”

Langkah selanjutnya yaitu *ciphertext* dan key diubah menjadi angka sesuai dengan telah yang diberikan, berikut ini adalah proses dekripsinya :

Proses Dekripsi

$P_1 = (C_1 - K_1) mod\ 26$

$= (6 - 22) mod\ 26$

$= (-16) mod\ 26$

$P_1 = 10$

$maka\ P_1 = 10$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 10\ adalah\ \boldsymbol{K}$

$P_2 = (C_2 - K_2) mod\ 26$

$= (18 - 11) mod\ 26$

$= (7) mod\ 26$

$P_2 = 7$

$maka\ P_2 = 7$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 7\ adalah\ \boldsymbol{H}$

$P_3 = (C_3 - K_3) mod\ 26$

$= (18 - 4) mod\ 26$

$= (14) mod\ 26$

$P_3 = 14$

$maka\ P_3 = 14$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 14\ adalah\ \boldsymbol{O}$

$P_4 = (C_4 - K_4) mod\ 26$

$= (13 - 8) mod\ 26$

$= (5) mod\ 26$

$P_4 = 5$

$maka\ P_4 = 5$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 5\ adalah\ \boldsymbol{F}$

$P_5 = (C_5 - K_5) mod\ 26$

$= (24 - 16) mod\ 26$

$= (8) mod\ 26$

$P_5 = 8$

$maka\ P_5 = 8$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 8\ adalah\ \boldsymbol{I}$

$P_6 = (C_6 - K_6) mod\ 26$

$= (12 - 7) mod\ 26$

$= (5) mod\ 26$

$P_6 = 5$

$maka\ P_6 = 5$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 5\ adalah\ \boldsymbol{F}$

$P_7 = (C_7 - K_7) mod\ 26$

$= (19 - 19) mod\ 26$

$= (0) mod\ 26$

$P_7 = 0$

$maka\ P_7 = 0$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 0\ adalah\ \boldsymbol{A}$

$P_8 = (C_8 - K_8) mod\ 26$

$= (21 - 14) mod\ 26$

$= (7) mod\ 26$

$P_8 = 7$

$maka\ P_8 = 7$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 7\ adalah\ \boldsymbol{H}$

sehingga dapat diperoleh hasil dekripsi sebagai berikut:

Cipherteks	G	S	S	N	Y	M	T	V
Key	W	L	E	I	Q	H	T	O
Plainteks	K	H	O	F	I	F	A	H

Tabel 3.2 pesamaan dekripsi *One Time Pad*

Jadi hasil dekripsi dari *ciphertext* “GSSNYMTV” adalah “KHOFIFAH”

3.1.1 Analisa Algoritma *One Time Pad Cipher*

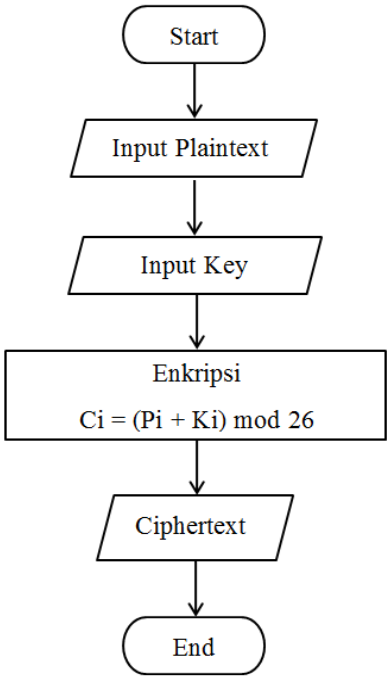
One Time Pad (OTP) merupakan algoritma klasik yang tidak dapat dipecahkan. Hal itu dikarenakan panjang kunci enkripsi memiliki panjang yang sama dengan jumlah yang akan dienkripsikan. *One Time Pad Cipher* memiliki kelemahan panjang kunci yang terlalu panjang, tetapi kelemahan itu juga merupakan kelebihannya. Oleh karena itu, untuk mendistribusikan kuncinya harus melalui jalur yang berbeda dari pengiriman pesan yang akan dienkripsi.

3.1.2 Algoritma dan Flowchart Proses Enkripsi dan Dekripsi dari File Teks

Prosedur ini digunakan untuk melakukan proses enkripsi dan dekripsi. Pada tahap ini juga akan dipanggil beberapa prosedur pendukung yang telah dijelaskan sebelumnya.

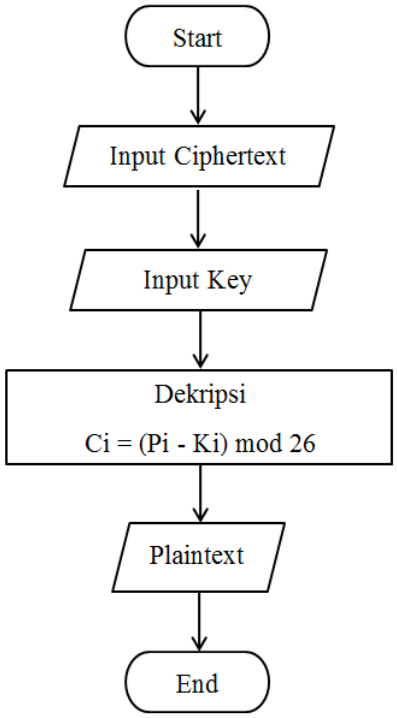
Di bawah ini akan dijelaskan prosesnya secara lebih rinci :

Flowchart enkripsi Algoritma *One Time Pad Cipher*



Gambar 3.1 Flowchart Enkripsi

Flowchart dekripsi Algoritma *One Time Pad*



Gambar 3.2 Flowchart Dekripsi

3.2 Penyandian *Myszkowski Cipher*

Myszkowski Transposisi Cipher merupakan salah satu jenis algoritma transposisi cipher yang memiliki keunikan tersendiri. Pada proses enkripsi, *plaintext* ditulis secara horizontal dari kiri kekanan, kemudian *ciphertext* dibaca secara vertikal sesuai dengan urutan kunci. Adapun algoritma *Myszkowski Transposisi Cipher* adalah sebagai berikut :

a. Enkripsi

Sebelum melakukan proses enkripsi, terlebih dahulu dilakukan pembentukan kunci. Beberapa huruf yang dibentuk secara manual ataupun acak dapat menambah variasi pembentukan kunci. Misalkan terdapat *plaintext* dan kunci sebagai berikut :

Plaintext : SIRIN KHUMAIROH

Kunci : INA

Proses enkripsi dimulai dengan membentuk sejumlah baris dan kolom untuk menampung *plaintext*. Terdapat 14 huruf pada *plaintext* yang akan menjadi acuan dalam membentuk baris, dan 3 huruf pada kunci akan menjadi acuan untuk membentuk kolom. Sehingga jumlah kolom dan baris yang dibutuhkan adalah :

Kunci = 3 huruf => 3 kolom

Plaintext = 14 huruf => 14/3 = 4,6 => 5 baris

Kunci pada contoh diatas terdiri dari 3 huruf, sehingga dapat kita beri penomoran sesaui urutan abjad, yaitu I = 3; N = 2; A = 1.

Setelah membentuk baris dan kolom yang memungkinkan, *plaintext* dapat ditulis secara berurutan dan horizontal. Secara rinci dapat dilihat pada tabel berikut :

I	N	A
3	2	1
S		

(a)

I	N	A
3	2	1
S	I	R

(b)

I	N	A
3	2	1
S	I	R
I		

(c)

I	N	A
3	2	1
S	I	R
I	N	K
H	U	M
A	I	R
O	H	X

(d)

Tabel 3.3 Proses Enkripsi (a), (b), (c). Hasil Enkripsi (d).

Pada Tabel 3.3 dua baris teratas merupakan kunci dan penomoran kunci yang digunakan dalam proses enkripsi, kemudian 1 huruf terakhir merupakan *dummy* yang digunakan untuk memenuhi kotak kosong yang tidak terisi oleh *plaintext.*

Dari proses enkripsi tersebut, *ciphertext* dapat diperoleh dengan membaca huruf secara vertikal dari atas kebawah sesuai dengan penomoran kunci.

Kunci	**1**	**2**	**3**
Ciphertext	**RKMRX**	**INUIH**	**SIHAO**

Sehingga *ciphertext* menjadi : RKMRX INUIH SIHAO

Keunikan algoritma *Myszkowski Cipher* terlihat apabila terdapat kunci yang memiliki huuf yang berulang. Pada kasus ini, *ciphertext* akan dibaca secara horizontal. Misalkan :

Plaintext : SIRIN KHUMAIROH

Kunci : GERIMIS

Proses enkripsi :

G	E	R	I	M	I	S
2	1	5	3	4	3	6
S	I	R	I	N	K	H
U	M	A	I	R	O	H

Tabel 3.4 Variasi Enkripsi

Ciphertext : IM SU II KO NR RA HH

Dapat dilihat pada tabel 3.4, terdapat dua huruf kunci yang sama yaitu huruf I, dengan penomoran kunci = 3. Pembacaan *ciphertext* tidak dilakukan secara vertikal, melainkan secara horizontal dan menghasilkan II KO

I	I
3	3
I	K
I	O

Tabel 3.5 Pembacaan *Ciphertext* secara horizontal

Tabel 3.5 Pembacaan *ciphertext* secara horizontal terdapat huruf kunci yang sama.

2. Dekripsi

Proses dekripsi dapat dilakukan apabila pihak penerima pesan rahasia (*ciphertext*) mengetahui pola kunci dan ukuran baris dan kolom yang digunakan pihak pengirim pesan. Proses dekripsi dapat dilakukan dengan menuliskan *ciphertext* secara vertikal dari atas kebawah secara berurutan sesuai dengan penomoran kunci. Pada penomoran kunci yang sama, *ciphertext* ditulis secara horizontal.

Ciphertext : IM SU II KO NR RA HH

Kunci : GERIMIS

Proses Dekripsi :

G	E	R	I	M	I	S
2	1	5	3	4	3	6
	I					
	M					

G	E	R	I	M	I	S
2	1	5	3	4	3	6
S	I					
U	M					

G	E	R	I	M	I	S
2	1	5	3	4	3	6
S	I		I		K	
U	M					

G	E	R	I	M	I	S
2	1	5	3	4	3	6
S	I		I	N	K	
U	M		I	R		

G	E	R	I	M	I	S
2	1	5	3	4	3	6
S	I	R	I	N	K	
U	M	A	I	R		

G	E	R	I	M	I	S
2	1	5	3	4	3	6
S	I	R	I	N	K	H
U	M	A	I	R	O	H

Tabel 3.6 Proses Dekripsi

Dari tabel 3.6, *plaintext* dapat diperoleh dengan membaca huruf dari kolom paling kiri tanpa memperdulikan penomoran kunci, sehingga diperoleh :

Plaintext : SIRIN KHUMAIROH

3.3 Penyandian Super Enkripsi One Time Pad Cipher dan Myszkkowski Cipher

Teknik penyandian pesan dimulai dengan proses enkripsi menggunakan *One Time Pad Cipher*, kemudian pesan hasil enkripsi tersebut dienkripsi lagi menggunakan transformasi *Myszkowski Cipher* sehingga akan terbentuk keamanan dua lapis, untuk mengembalikan pesan agar terbaca kembali maka dilakukan dekripsi menggunakan *Myszkowski Cipher* kemudian pesan didekripsi menggunakan *One Time Pad Cipher*. Proses enkripsi dan dekripsi pesan dilakukan menggunakan key dan *plaintext* yang sama.

Contoh :

Khofi sebagai sender akan mengirim pesan "**KHOFIFATUSSHAQILAH"** kepada sirin sebagai receiver. Namun karena khofi ingin agar pesan tersebut aman dan tidak diketahui oleh semua orang, maka khofi akan menggunakan super enkripsi untuk menjadikan pesan. Teknik yang digunakan adalah *One Time Pad cipher* dengan kunci enkripsi "**JFKDFHGUTOGKBFFNJD**" dan transformasi *Myszkowaski Cipher* dengan kunci 3.

3.3.1 Proses Enkripsi Pesan

One Time Pad Cipher

Langkah pertama adalah mengenkripsi pesan tersebut dengan menggunakan persamaan dari enkripsi *One Time Pad Cipher* dimana terdapat sebuah

Plaintext = "**KHOFIFATUSSHAQILAH"**

Key = "**JFKDFHGUTOGKBFFNJD"**

Langkah selanjutnya yaitu plainteks dan key diubah menjadi angka sesuai dengan telah yang diberikan, berikut ini adalah proses enkripsinya :

$C_1 = (P_1 + K_1) mod\ 26$

$= (10 + 9) mod\ 26$

$= (19) mod\ 26$

$C_1 = 19$

$maka\ C_1 = 19$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 19\ adalah\ \boldsymbol{T}$

$C_2 = (P_2 + K_2) mod\ 26$

$= (7 + 5) mod\ 26$

$= (12) mod\ 26$

$C_2 = 12$

$maka\ C_2 = 12$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 12\ adalah\ \boldsymbol{M}$

$C_3 = (P_3 + K_3) mod\ 26$

$= (14 + 10) mod\ 26$

$= (24) mod\ 26$

$C_3 = 24$

$maka\ C_3 = 24$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 24\ adalah\ \boldsymbol{Y}$

$C_4 = (P_4 + K_4) mod\ 26$

$= (5 + 3) mod\ 26$

$= (8) mod\ 26$

$C_4 = 8$

$maka\ C_4 = 8$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 8\ adalah\ \boldsymbol{I}$

$C_5 = (P_5 + K_5) mod\ 26$

$= (8 + 5) mod\ 26$

$= (13) mod\ 26$

$C_5 = 13$

$maka\ C_5 = 13$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 13\ adalah\ \boldsymbol{N}$

$C_6 = (P_6 + K_6) mod\ 26$

$= (5 + 7) mod\ 26$

$= (12) mod\ 26$

$C_6 = 12$

$maka\ C_6 = 12$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 12\ adalah\ \boldsymbol{M}$

$C_7 = (P_7 + K_7) mod\ 26$

$= (6 + 0) mod\ 26$

$= (6) mod\ 26$

$C_7 = 6$

$maka\ C_7 = 6$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 6\ adalah\ \boldsymbol{G}$

$C_8 = (P_8 + K_8) mod\ 26$

$= (19 + 20) mod\ 26$

$= (13) mod\ 26$

$C_8 = 13$

$maka\ C_8 = 13$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 13\ adalah\ \boldsymbol{N}$

$C_9 = (P_9 + K_9) mod\ 26$

$= (20 + 19) mod\ 26$

$= (13) mod\ 26$

$C_9 = 13$

$maka\ C_9 = 13$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 13\ adalah\ \boldsymbol{N}$

$C_{10} = (P_{10} + K_{10}) mod\ 26$

$= (18 + 14) mod\ 26$

$= (6) mod\ 26$

$C_{10} = 6$

$maka\ C_{10} = 6$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 6\ adalah\ \boldsymbol{G}$

$C_{11} = (P_{11} + K_{11}) mod\ 26$

$= (18 + 6) mod\ 26$

$= (24) mod\ 26$

$C_{11} = 24$

$maka\ C_{11} = 24$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 24\ adalah\ \boldsymbol{Y}$

$C_{12} = (P_{12} + K_{12}) mod\ 26$

$= (7 + 0) mod\ 26$

$= (17) mod\ 26$

$C_{12} = 17$

$maka\ C_{12} = 17$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 17\ adalah\ \boldsymbol{R}$

$C_{13} = (P_{13} + K_{13}) mod\ 26$

$= (1 + 0) mod\ 26$

$= (1) mod\ 26$

$C_{13} = 1$

$maka\ C_{13} = 1$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 1\ adalah\ \boldsymbol{B}$

$C_{14} = (P_{14} + K_{14}) mod\ 26$

$= (16 + 5) mod\ 26$

$= (21) mod\ 26$

$C_{14} = 21$

$maka\ C_{14} = 21$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 21\ adalah\ \boldsymbol{V}$

$C_{15} = (P_{15} + K_{15}) mod\ 26$

$= (8 + 5) mod\ 26$

$= (13) mod\ 26$

$C_{15} = 13$

$maka\ C_{15} = 13$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 7\ adalah\ \boldsymbol{N}$

$C_{16} = (P_{16} + K_{16}) mod\ 26$

$= (11 + 13) mod\ 26$

$= (24) mod\ 26$

$C_{16} = 24$

$maka\ C_{16} = 24$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 24\ adalah\ \boldsymbol{Y}$

$C_{17} = (P_{17} + K_{17}) mod\ 26$

$= (9 + 0) mod\ 26$

$= (9) mod\ 26$

$C_{17} = 9$

$maka\ C_{17} = 9$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 7\ adalah\ \boldsymbol{J}$

$C_{18} = (P_{18} + K_{18}) mod\ 26$

$= (7 + 3) mod\ 26$

$= (10) mod\ 26$

$C_{18} = 10$

$maka\ C_{18} = 10$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ ciphertext\ 10\ adalah\ \boldsymbol{K}$

Plaintext	K	H	O	F	I	F	A	T	U	S	S	H	A	Q	I	L	A	H
Key	J	F	K	D	F	H	G	U	T	O	G	K	B	E	F	N	J	D
Cipher Text	T	M	Y	I	N	M	G	N	N	G	Y	R	B	V	N	Y	J	K

Tabel 3.7 Hasil persamaan Enkripsi

Sehingga *ciphertext* yang didapat adalah **TMYINMGNNGYRBVNYJK,** *ciphertext* tersebut kemudian akan di enkripsi kembali menggunakan *Myszkowski Cipher.*

Myszkowski Cipher

Langkah pertama yaitu terlebih dahulu dilakukan pembentukan kunci. Beberapa huruf yang dibentuk secara manual ataupun acak dapat menambah variasi pembentukan kunci. Terdapat *plaintext* dan kunci sebagai berikut :

Plaintext : **TMYINMGNNGYRBVNYJK**

Kunci : **MEI**

Proses enkripsi dimulai dengan membentuk sejumlah baris dan kolom untuk menampung *plaintext.* Terdapat 18 huruf pada *plaintext* yang akan menjadi acuan dalam membentuk baris,dan 3 huruf pada kunci akan menjadi acuan untuk membentuk kolom. Sehingga jumlah kolom dan baris yang dibutuhkan adalah :

Kunci = 3 huruf => 3 kolom

Plaintext = 18 huruf => 18/3 = 6 baris

Kunci pada contoh diatas terdiri dari 3 huruf,sehingga dapat kita beri penomoran sesuai urutan abjad, yaitu M = 3; E = 2; I = 1.

Setelah membentuk baris dan kolom yang memungkinkan,*plaintext* dapat ditulis secara berurutan dan horizontal. Secara rinci dapat dilihat pada tabel berikut :

M	E	I
3	2	1
T		

(a)

M	E	I
3	2	1
T	M	Y

(b)

M	E	I
3	2	1
T	M	Y
I		

(c)

M	E	I
3	2	1
T	Y	M
I	N	M
G	N	N
G	Y	R
B	V	N
Y	J	K

(d)

Tabel 3.8 Proses Enkripsi (a),(b),(c). Hasil Enkripsi (d).

Dari proses enkripsi tersebut, *ciphertext* dapat diperoleh dengan membaca huruf secara vertikal dari atas kebawah sesuai dengan penomoran kunci.

Kunci	**1**	**2**	**3**
Ciphertext	**MMNRNK**	**YNNYVJ**	**TIGGBY**

Ciphertext : **MMNRNK YNNYVJ TIGGBY**

Keunikan algoritma Myszkowski terlihat apabila terdapat kunci yang memiliki huruf yang berulang. Pada kasus ini *ciphertext* akan dibaca secara horizontal.

Plaintext : **TMYINMGNNGYRBVNYJK**

Kunci : **OPTIMIS**

Proses enkripsi :

O	P	T	I	M	I	S
3	4	6	1	2	1	5
T	M	Y	I	N	M	G
N	N	G	Y	R	B	V
N	Y	J	K	X	X	X

Tabel 3.9 Variasi Enkripsi

Ciphertext : IYK MBX NRX TNN MNY GVX YGJ

Dapat dilihat pada tabel 3.9, terdapat dua huruf kunci yang sama yaitu huruf I, dengan penomoran kunci = 1. kemudian 3 huruf X terakhir merupakan *dummy* yang digunakan untuk memenuhi kotak kosong yang tidak terisi oleh *plaintext.* Pembacaan *ciphertext* tidak dilakukan secara vertikal, melainkan secara horizontal dan menghasilkan IYK MBX

I	I
1	1
I	M
Y	B
K	X

Tabel 3.10 Pembacaan *Ciphertext* secara horizontal

Tabel 3.10 Pembacaan *Ciphertext* secara horizontal terdapat huruf kunci yang sama

3.3.2 Proses Dekripsi Pesan

Proses dekripsi dapat dilakukan apabila pihak penerima pesan rahasia (*ciphertext*) mengetahui pola kunci dan ukuran baris dan kolom yang digunakan pihak pengirim pesan. Proses dekripsi dapat dilakukan dengan menuliskan *ciphertext* secara vertikal dari atas kebawah secara berurutan sesuai dengan penomoran kunci. Pada penomoran kunci yang sama, *ciphertext* ditulis secara horizontal.

Ciphertext : IYK MBX NRX TNN MNY GVX YGJ

Kunci : OPTIMIS

Proses Dekripsi :

O	P	T	I	M	I	S
3	4	6	1	2	1	5
			I			
			Y			
			K			

O	P	T	I	M	I	S
3	4	6	1	2	1	5
			I	N	M	
			Y	R		
			K			

O	P	T	I	M	I	S
3	4	6	1	2	1	5
T			I	N	M	
N			Y	R		
N			K			

O	P	T	I	M	I	S
3	4	6	1	2	1	5
T	M		I	N	M	
N	N		Y	R		
N	Y		K			

O	P	T	I	M	I	S
3	4	6	1	2	1	5
T	M		I	N	M	G
N	N		Y	R	B	V
N	Y		K			

O	P	T	I	M	I	S
3	4	6	1	2	1	5
T	M	Y	I	N	M	G
N	N	G	Y	R	B	V
N	Y	J	K	X	X	X

Tabel 3.11 Proses Dekripsi

Untuk mendapatkan hasil dekripsi dari *Myszkowski Cipher* dengan cara membaca huruf dari kolom paling kiri tanpa memperdulikan penomoran kunci, sehingga diperoleh hasil dekripsi *Myszkowski Cipher* seperti berikut :

TMYINMGNNGYRBVNYJK, kemudian teks hasil dekripsi tersebut akan didekripsikan lagi dengan menggunakan teknik *One Time Pad Cipher*

One Time Pad Cipher

Proses dekripsi pada metode algoritma *One Time Pad Cipher* adalah kebalikan atau mengembalikan *plaintext* menjadi data semula, dapat diperlihatkan dengan menggunakan persamaan sebagai berikut :

$P_1 = (C_1 - K_1) mod\ 26$, dengan

Ciphertext : **TMYINMGNNGYRBVNYJK**, dan

Key : **JFKDFHGUTOGKBFFNJD**

Langkah selanjutnya yaitu *ciphertext* dan key diubah menjadi angka sesuai dengan yang telah diberikan, berikut ini adalah proses dekripsinya :

$P_1 = (C_1 - K_1) mod\ 26$

$= (19 - 9) mod\ 26$

$= (10) mod\ 26$

$P_1 = 10$

$maka\ P_1 = 10$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 10\ adalah\ \boldsymbol{K}$

$P_2 = (C_2 - K_2) mod\ 26$

$= (12 - 5) mod\ 26$

$= (7) mod\ 26$

$P_2 = 7$

$maka\ P_2 = 7$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 7\ adalah\ \boldsymbol{H}$

$P_3 = (C_3 - K_3) mod\ 26$

$= (24 - 10) mod\ 26$

$= (14) mod\ 26$

$P_3 = 14$

$maka\ P_3 = 14$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 14\ adalah\ \boldsymbol{O}$

$P_4 = (C_4 - K_4) mod\ 26$

$= (8 - 3) mod\ 26$

$= (5) mod\ 26$

$P_4 = 5$

$maka\ P_4 = 5$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 5\ adalah\ \boldsymbol{F}$

$P_5 = (C_5 - K_5) mod\ 26$

$= (13 - 5) mod\ 26$

$= (10) mod\ 26$

$P_5 = 8$

$maka\ P_5 = 8$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 8\ adalah\ \boldsymbol{I}$

$P_6 = (C_6 - K_6) mod\ 26$

$= (12 - 7) mod\ 26$

$= (5) mod\ 26$

$P_6 = 5$

$maka\ P_6 = 5$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 5\ adalah\ \boldsymbol{F}$

$P_7 = (C_7 - K_7) mod\ 26$

$= (6 - 6) mod\ 26$

$= (0) mod\ 26$

$P_7 = 0$

$maka\ P_7 = 0$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 0\ adalah\ \boldsymbol{A}$

$P_8 = (C_8 - K_8) mod\ 26$

$= (13 - 20) mod\ 26$

$= (-7) mod\ 26$

$P_8 = 19$

$maka\ P_8 = 19$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 19\ adalah\ \boldsymbol{T}$

$P_9 = (C_9 - K_9) mod\ 26$

$= (13 - 19) mod\ 26$

$= (-6) mod\ 26$

$P_9 = 20$

$maka\ P_9 = 20$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 20\ adalah\ \boldsymbol{U}$

$P_{10} = (C_{10} - K_{10}) mod\ 26$

$= (6 - 14) mod\ 26$

$= (-8) mod\ 26$

$P_{10} = 18$

$maka\ P_{10} = 18$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 18\ adalah\ \boldsymbol{S}$

$P_{11} = (C_{11} - K_{11}) mod\ 26$

$= (24 - 6) mod\ 26$

$= (18) mod\ 26$

$P_{11} = 18$

$maka\ P_{11} = 18$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 18\ adalah\ \boldsymbol{S}$

$P_{12} = (C_{12} - K_{12}) mod\ 26$

$= (17 - 10) mod\ 26$

$= (7) mod\ 26$

$P_{12} = 7$

$maka\ P_{12} = 7$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 7\ adalah\ \boldsymbol{H}$

$P_{13} = (C_{13} - K_{13}) mod\ 26$

$= (1 - 1) mod\ 26$

$= (0) mod\ 26$

$P_{13} = 0$

$maka\ P_{13} = 0$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 0\ adalah\ \boldsymbol{A}$

$P_{14} = (C_{14} - K_{14}) mod\ 26$

$= (21 - 5) mod\ 26$

$= (16) mod\ 26$

$P_{14} = 16$

$maka\ P_{14} = 16$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 16\ adalah\ \boldsymbol{Q}$

$P_{15} = (C_{15} - K_{15}) mod\ 26$

$= (13 - 5) mod\ 26$

$= (8) mod\ 26$

$P_{15} = 8$

$maka\ P_{15} = 8$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 8\ adalah\ \boldsymbol{I}$

$P_{16} = (C_{16} - K_{16}) mod\ 26$

$= (24 - 13) mod\ 26$

$= (11) mod\ 26$

$P_{16} = 11$

$maka\ P_{16} = 11$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 11\ adalah\ \boldsymbol{L}$

$P_{17} = (C_{17} - K_{17}) mod\ 26$

$= (9 - 9) mod\ 26$

$= (0) mod\ 26$

$P_{17} = 0$

$maka\ P_{17} = 0$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 0\ adalah\ \boldsymbol{A}$

$P_{18} = (C_{18} - K_{18}) mod\ 26$

$= (10 - 3) mod\ 26$

$= (7) mod\ 26$

$P_{18} = 7$

$maka\ P_{18} = 7$

$Karakter\ yang\ diperoleh\ dengan\ nilai\ plaintext\ 7\ adalah\ \boldsymbol{H}$

Cipher Text	T	M	Y	I	N	M	G	N	N	G	Y	R	B	V	N	Y	J	K
Key	J	F	K	D	F	H	G	U	T	O	G	K	B	E	F	N	J	D
Plaintext	K	H	O	F	I	F	A	T	U	S	S	H	A	Q	I	L	A	H

Tabel 3.12 Hasil Persamaan Dekripsi

Setelah pesan terdekripsi maka Sirin sebagai pihak receiver dapat membaca isi pesan asli yang dikirimkan Khofi sebagai pihak sender yaitu :

KHOFIFATUSSHAQILAH

3.4 Kajian Agama Islam

3.4.1 Penyampaian Pesan dan Pengamanannya

Al-Quran adalah pedoman yang tidak hanya diperuntukkan kepada manusia, dan yang menjelaskan tentang pentingnya menyampaikan pesan kepada orang yang berhak menerimanya serta menjaga keamanan dari pesan itu sebagaimana firmannya dalam QS Al-Mumtahanan ayat 1:

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُوا لَا تَتَّخِذُوا۟ عَدُوِّى وَعَدُوَّكُمْ أَوْلِيَآءَ تُلْقُونَ إِلَيْهِم بِٱلْمَوَدَّةِ وَقَدْ كَفَرُوا۟ بِمَا جَآءَكُم مِّنَ ٱلْحَقِّ يُخْرِجُونَ ٱلرَّسُولَ وَإِيَّاكُمْ أَن تُؤْمِنُوا۟ بِٱللَّهِ رَبِّكُمْ إِن كُنتُمْ خَرَجْتُمْ جِهَٰدًا فِى سَبِيلِى وَٱبْتِغَآءَ مَرْضَاتِى تُسِرُّونَ إِلَيْهِم بِٱلْمَوَدَّةِ وَأَنَا۠ أَعْلَمُ بِمَآ أَخْفَيْتُمْ وَمَآ أَعْلَنتُمْ وَمَن يَفْعَلْهُ مِنكُمْ فَقَدْ ضَلَّ سَوَآءَ ٱلسَّبِيلِ

Artinya: *"Hai orang-orang yang beriman, janganlah kamu mengambil musuh-ku dan musuhmu menjadi teman-teman setia yang kamu sampaikan kepada mereka (berita-berita Muhammad), karena rasa kasih sayang; padahal sesungguhnya mereka telah ingkar kepada kebenaran yang datang kepadamu, mereka mengusir Rasul dan (mengusir) kamu karena kamu beriman kepada Allah, Tuhanmu. Jika kamu benar-benar keluar untuk berjihad di jalan-ku dan mencari keridhaan-Ku (janganlah kamu berbuat demikian). Kamu memberitahukan secara rahasia (berita-berita Muhammad) kepada mereka, karena rasa kasih sayang. Aku lebih mengetahui apa yang kamu sembunyikan dan apa yang kamu nyatakan. Dan barangsiapa di antara kamu yang melakukannya, maka sesungguhnya dia telah tersesat dari jalan yang lurus."*

Berdasarkan ayat ini Allah memberikan peringatan kepada kaum muslimin untuk tidak menyampaikan informasi rahasia kepada musuh Islam. Hal senada juga disabdakan oleh rasulullah dari hadits al-hasan dari samurah bahwa rasulullah saw bersabda bahwa

أَدِّ الْأَمَانَةَ إِلَى مَنِ ائْتَمَنَكَ، وَلاَ تَخُنْ مَنْ خَانَكَ

Artinya: "*Tunaikanlah amanah pada orang yang memberikan amanah itu kepadamu, dan jangan kau khianati orang yang pernah mengkhianatimu." (HR. Al-Imam Ahmad dan Ahlus Sunan)*

Hadits tersebut juga menjelaskan bahwa sudah seharusnya kita menyampaikan informasi kepada orang yang sudah kita percayai. Sehingga dalam hal ini merupakan orang yang memiliki protokol kunci yang sepakati oleh kedua belah pihak sehingga pesan tersebut hanya bisa dipahami oleh pengirim dan penerima pesan tersebut.

Ayat lain yang menjelaskan tentang pentingnya menjaga pesan terdapat dalam QS Al-Anfal ayat 27 yang berbunyi:

يَٰٓأَيُّهَا ٱلَّذِينَ ءَامَنُواْ لَا تَخُونُواْ ٱللَّهَ وَٱلرَّسُولَ وَتَخُونُوٓاْ أَمَٰنَٰتِكُمۡ وَأَنتُمۡ تَعۡلَمُونَ

Artinya: *"Hai orang-orang yang beriman, janganlah kamu menghianati Allah dan Rasul (Muhammad) dan juga janganlah kamu menghianati amanat-amanat yang dipercayakan kepadamu, sedang kamu mengetahuinya."*

Sepeti ayat-ayat sebelumnya, ayat ini juga menerangkan bahwa suatu amanat yang dalam konteks masa kini adalah suatu informasi yang sudah diamanatkan kepada kita, maka kita harus sebisa mungkin menjaga amanat itu agar tidak disadap, atau disalahgunakan oleh pihak yang tidak berkepentingan. Oleh karena itu, setidaknya kita melakukan ikhtiar untuk mengamankan pesan itu dengan salah satu caranya adalah dengan menyandikannya menggunakan teknik penyandian berlapis lapis.

BAB IV
PENUTUP

4.1 Kesimpulan

Berdasarkan hasil analisis dan implentasi yang telah dilakukan diatas, maka dapat disimpulkan bahwa:

1. Proses enkripsi menggunakan algoritma *One Time Pad Cipher* dengan menggunakan kunci acak dan panjang kunci sama dengan panjang *plaintext* membuat lebih aman. Proses dekripsi menggunakan algoritma *One Time Pad Cipher* dengan mengembalikan *ciphertext* menjadi *plaintext* menggunakan kunci yang telah diberikan.
2. Proses enkripsi menggunakan algoritma *Myszkowski Cipher* diawali dengan pembentukan kunci 3 huruf yang dibentuk secara manual ataupun acak untuk membentuk sejumlah kolom yang diberi penomoran sesuai urutan abjad dan 18 huruf hasil enkripsi dibagi 3 untuk membetuk sejumlah 6 baris, dengan membaca huruf secara vertikal dari atas kebawah sesuai dengan penomoran kunci dapat diperoleh *ciphertext* sehingga aman. Proses dekripsi menggunakan algoritma *Myszkowski Cipher* dengan menuliskan *ciphertext* secara vertikal dari atas kebawah secara berurutan sesuai dengan penomoran kunci, pada penomoran kunci yang sama *ciphertext* ditulis secara horizontal, untuk memperoleh hasil dekripsi dari Myszkowski Cipher dengan membaca huruf kolom paling kiri tanpa memperdulikan penomoran kunci.
3. Super enkripsi yang diimplementasikan dengan menggunakan algoritma *One Time Pad Cipher* dan algoritma *Myszkowski Cipher* membuat pesan tersebut sangatlah aman. Hal ini dikarenakan penggunaan dua buah jenis Cipher ini

sangatlah mendukung satu sama lain agar proses enkripsi dan dekripsi pesan dapat meningkat keamanannya.

4.2 Saran

Pada penelitian ini, terdapat beberapa saran yang bisa dipertimbangkan untuk pengembangan pada peneltian yang berikutnya, yaitu:

1. Untuk penelitian kedepannya diharapkan dapat membangun sebuah aplikasi yang bisa diterapkan pada perangkat lunak di Android, iOS, Windows dan lain sebagainya
2. Untuk pengembangan selanjutnya diharapkan untuk menambahkan suatu proses atau algoritma lain di kriptografi yang bisa mendukung algoritma-algoritma yang digunakan di penelitian ini.
3. Untuk pengembangan yang menggunakan algoritma pada penelitian ini diharapkan dapat menggunakan objek lain seperti gambar, audio, video dan lain sebagainya.

DAFTAR RUJUKAN

Ariyus, Doni. 2008. *Pengantar Ilmu Kriptografi*, Bandung: Andi.

A.Menezes, P. Van Oorschot,S. Vanstone. 1996. Handbook of Applied Cryptography, CRC Press Inc.

Cucu Tri Eka Yuliana, "*Implementasi Algoritma Kriptografi Blowfish dan Metode Steganografi End Of File (EOF) untuk Keamanan Data*", Skripsi Teknik Informatika Universitas Dian Nuswantoro, Semarang, 2014.

Effendy, Onong Uchjana. 1989. *KAMUS KOMUNIKASI*. Bandung : PT. Mandar Maju.

Khairina Nurul,dkk. 2019. *Modifikasi Myszkowski Transposition Cipher dengan Chess Board Pattern*

Latifah R, dkk. 2017. *Modifikasi Algoritma Caesar Cipher dan Rail Fence Untuk Peningkatan Keamanan Teks Alfanumerik dan Karakter Khusus.* Seminar Nasional Sains Dan Teknologi 2017.

Munir, R. (2006). *Kriptografi.* Bandung: Informatika.

Munir, R. (2010). *Matematika Diskrit.* Bandung: Informatika.

Munir, Rinaldi. 2019. *Kriptografi,* Bandung: Informatika Bandung.

Mollin, R. A. 2007. An Introduction to Cryptography. 2nd Edition. Chapman & Hall/CRC : Boca Raton, Florida.

Narender T. Dan Anita G.,"Comparative Analysis of Symmetric Key Encryption Algorithms". International Journal of Advanced Research in Computer Science and Software Engineering Vol. 4(8), pp. 348-354.

Nishika dan R.K. Yadav, "A Lookup Table Based Secure Cryptographic SMS Communication on Android Environment". International Journal of Computer Science and Mobile Computing Vol. 2(6), pp. 53-59.

Prayudi, Yudi, Idham Halik. 2005. *Studi Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/Dekripsi Data*. Seminar Nasional Aplikasi Teknologi Informasi 2005 (SNATI 2005), Yogyakarta.

Ramkesh, N. (2016). ADVANCED RAIL FENCE CIPHER ALGORITHM. *International Journal of Pharmacy and Technology*, 16541.

Schneier, Bruce. 1996. *Applied Cryptography, Second Edition: Protocol, Algorithms, and Source Code in* C. Jonh Wiley & Sons, Inc.

Sadikin, R. (2012). *Kriptografi untuk Keamanan Jaringan.* Yogyakarta: ANDI

Wirdasari, Dian. 2008. *Prinsip Kerja Kriptografi dalam Mengamankan Informasi*, Jurnal SAINTIKOM Vol.5 No.2.

Whitman, M.E., & Mattord, H.J, Management of Information Security, Third Edition, Boston: Course Technology, 2010.

RIWAYAT HIDUP

Muhammad Yusrul Hana dilahirkan di Banyuwangi pada tanggal 02 Maret 1996, merupakan anak kedua dari empat bersaudara, pasangan dari Bapak Usman Ashofi dan Ibu Lutfiah. Pendidikan dasarnya ditempuh di SDN 2 Karang bendo yang ditamatkan pada tahun 2008.

Pada tahun yang sama melanjutkan pendidikan menengah pertama di SMPN 1 Rogojampi dan lulus pada tahun 2011. Kemudian melanjutkan pendidikan menengah atas di SMKN Ihya' Ulumudin dan menamatkan pendidikan tersebut pada tahun 2014. Pendidikan berikutnya ditempuh di Universitas Islam Negeri Maulana Malik Ibrahim Malang melalui jalur MANDIRI dengan mengambil Jurusan Matematika di Fakultas Sains dan Teknologi.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAUALANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No. 50 Dinoyo Malang Telp./Fax.(0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Muhammad Yusrul Hana
NIM : 14610090
Fakultas/Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Algoritma One Time Pad Cipher dan Transformasi Myszkowski Cipher pada Pesan Teks.
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Hisyam Fahmi, M.Kom.

No	Tanggal	Hal	Tanda Tangan
1	03 Mei 2021	Konsultasi Bab I, Bab II, Bab III	1.
2	06 Mei 2021	Konsultasi Kajian Keagamaan pada Bab I dan Bab III	2.
3	30 Mei 2021	Revisi Bab I, Bab II, Bab III dan Bab IV	3.
4	22 Mei 2021	Revisi Kajian Keagamaan pada Bab I dan Bab III	4.
5	03 Juni 2020	Konsultasi Bab III, Bab IV, Bab V	5.
6	04 Juni 2021	Konsultasi Kajian Keagamaan & Kepenulisan pada Bab I dan III	6.
7	03 Juni 2021	ACC Bab I, Bab II, Bab III, Bab IV, Bab V	7.
8	18 Juni 2021	ACC Kajian Keagamaan pada Bab I dan Bab III	8.
9	18 Juni 2021	Revisi Keseluruhan	9.
10	18 Juni 2021	ACC Keseluruhan	10.

Malang, 18 Juni 2021
Mengetahui,
Ketua Jurusan Matematika

Dr. Usman Pagalay, M.Si
NIP. 19650414 200312 1 001