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

SKRIPSI

OLEH
FIRDAUS ADJI S
NIM. 14610066

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2020

IMPLEMENTASI ALGORITMA *ONE TIME PAD CIPHER* DAN TRANSFORMASI *RAIL FENCE 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
Firdaus Adji S
NIM. 14610066

JURUSAN MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2020

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

SKRIPSI

Oleh
FIRDAUS ADJI S
NIM. 14610066

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal 23 November 2020

Pembimbing I,	Pembimbing II,
Muhammad Khudzaifah, M..Si	Ach. Nasichuddin, MA
NIP. 19901511 2016801 1 057	NIP. 19730705 200003 1002

Mengetahui,
Ketua Jurusan Matematika

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

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

SKRIPSI

Oleh
FIRDAUS ADJI S
NIM. 14610066

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Sains (S.Mat)
Tanggal 02 Desember 2020

Penguji Utama	:	Evawati Alisah, M.Pd	
Ketua Penguji	:	M. Nafie Jauhari, M.Si	
Sekretaris Penguji	:	Muhammad Khudzaifah, M..Si	
Anggota Penguji	:	Ach. Nasichuddin, MA	

Mengetahui,
Ketua Jurusan Matematika

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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Firdaus Adji S

NIM : 14610066

Jurusan : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Algoritma *One Time Pad Cipher* dan Transformasi *Rail Fence 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, 23 November 2020
Yang membuat pernyataan,

Firdaus Adji S
NIM. 14610066

MOTO

"Work in Silence"

PERSEMBAHAN

Alhamdulillahi Robbil'alamin, dengan mengucap syukur kepada Allah Azza Wa Jalla, Penulis mempersembahkan skripsi ini untuk kedua orang tua saya tercinta, Bapak Amat Subaweh, dan Ibu Emi Kristanti yang selalu memberikan doa, dukungan, motivasi dan lain sebagainya yang tidak mungkin bisa penulis balas dengan balasan yang setimpal.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

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 Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Muhammad Khudzaifa, M.Si, selaku dosen pembimbing I yang telah banyak memberikan arahan, nasihat, motivasi, dan berbagai pengalaman yang sangat berharga kepada penulis.
5. Ach. Nasichuddin, MA, selaku dosen pembimbing II yang telah banyak memberikan arahan dan berbagai ilmunya kepada penulis.

6. Segenap sivitas akademika Jurusan Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang terutama seluruh dosen, terima kasih atas segala ilmu dan bimbingannya.
7. Bapak Amat Subaweh dan ibu Emi Kristanti yang selalu memberikan doa, semangat, serta motivasi kepada penulis sampai saat ini.
8. Seluruh teman-teman di Jurusan Matematika angkatan 2014 (MATH EIGEN), seluruh jajaran Crew De'Rumah Rindu Alam Resort, seluruh teman-teman Biker N250R Batu Community, dan teman-teman komunitas Desain Batu terima kasih atas segala pengalaman berharga, kerjasama dan kebersamaan syahdu yang terukir indah selama ini. Kalian luar biasa.
9. Semua pihak yang tidak dapat disebutkan satu-persatu, yang telah membantu penulis dalam menyelesaikan skripsi ini baik moral maupun materi.

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. *Aamiin.*

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 23 November 2020

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Saputro, Firdaus Adji. 2020. **Implementasi Algoritma *One Time Pad Cipher* dan Transformasi *Rail Fence Cipher.*** Skripsi. Jurusan Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II). M. Nasichuddin, MA

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

Penggunaan suatu buah algoritma kriptografi klasik sangatlah beresiko jika tidak diperkuat dengan satu buah algoritma yang lain, sehingga sangatlah diperlukan dua kombinasi algoritma kriptografi klasik agar bisa memberikan jaminan bahwa pesan tersebut aman dan tidak diketahui oleh pihak lain. Sehingga dalam hal ini sangatlah perlu menggunakan sebuah pengamanan dengan super enkripsi yang diantara terdiri dari kombinasi cipher tranposisi yaitu algoritma *Rail Fence Cipher* dan kombinasi lainnya di cipher substitusi yaitu algoritma *One Time Pad Cipher*. Algoritma *Rail Fence Cipher* memiliki keunggulan dibanding algoritma lainnya karena dalam tahap proses penulisan *plaintext* menjadi *ciphertext* dapat dilakukan pada baris mana saja sehingga menambah kesulitannya dalam proses dekripsi dan dekripsi. *One Time Pad Cipher* merupakan sebuah algoritma kriptografi klasik yang sangat kuat dan memliki sifat yang *unbreakable,* 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.

Pengunaan super enkripsi dengan metode algorima *One Time Pad Cipher* dan *Rail Fence Cipher* akan sangat memperkuat keamanan dari pesan. Keamanan tersebut didapatkan dari yang pertama ialah dari metode algoritma *One Time Pad Cipher,* karena algorima tersebut mempunyai sifat yang *unbreakable* dan nilai yang digunakan seluruhnya sangat acak. Selanjutnya keamanan yang kedua didapatkan dari metode algoritma *Rail Fence Cipher,* dimana pesan tersebut ketika sudah disandikan akan membuat semakin sulit untuk dipecahkan.

ABSTRACT

Saputro, Firdaus Adji. 2020. **Implementation of the *One Time Pad Cipher Algorithm* and Transformation *Rail Fence Cipher.*** Thesis. Department of Mathematics, Faculty of Science and Technology, Maulana Malik Ibrahim State Islamic University of Malang. Advisor: (I) Muhammad Khudzaifah, M.Si. (II). M. Nasichuddin, MA

Keywords: Encryption, Decryption, Super Encryption, *One Time Pad Cipher*, *Rail Fence Cipher*

The use of one classical cryptographic algorithm is very risky if it is not strengthened by another algorithm, so it is necessary to combine two classical cryptographic algorithms in order to provide assurance that the message is safe and unknown by other parties. So in this case it is very necessary to use a security with super encryption which consists of a combination of transition ciphers, namely the algorithm *Rail Fence Cipher* and other combinations in substitution ciphers, namely the algorithm *One Time Pad Cipher*. The *Rail Fence Cipher* algorithm has advantages over other algorithms because in the process of writing *plaintext* into *ciphertext*, *it* can be done on any line, thus it increases the difficulty in the encryption and decryption process. *One Time Pad cipher* is a strong classic cryptographic algorithm that and possess properties that *unbreakable*, this is due to the nature of the algorithm the key must be a row of values completely randomized perfect *(truly random)* and also the length of the key to this algorithm must be equal to the length of *the plaintext* his.

The use of super encryption with the *One Time Pad Cipher algorithm* and *Rail Fence Cipher* will greatly strengthen the security of the message. This security is obtained the first one is algorithm *One Time Pad Cipher*, since it has an *unbreakable* property and the values used are all very random. Furthermore, the second security is obtained from method *Rail Fence Cipher*, where the message when it is encoded will make it more difficult to crack.

مستخلص البحث

سا بوترا ، فردوس آجي، ٢٠٢٠. تنفيذ خوارزمية وان تيم فد جفحر(*One Time pad Cipher*)
و ريل فنج جفحر(*Rail Fence Cipher*)، البحث العلمى، قسم الرياضيات ، كلية العلوم والتكنولوجيا ، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالا نج، المشرف : (١) محمد خديفة الماجستير، (٢) أ حمد ناصح الدين الماجستير.

كلمات مفتاحية : التشفير ، وفك تشفير ، سوبر تشفير ، تشفير، وان تيم فد جفحر، ريل فنج جفحر

استخدام الخوارزمية التشفير الكلاسيكية فيه المخاطر إذا لم تتم بالخوارزمية الأخرى، فلذلك من استخدامه محتاج بالخوارزمية الأخرى لتقويه لكيلا لتلك الرسالة آمنة وغير معروفة با لأخرين .أن هذه الحالة تحتاج الي استعمال الأمان بتشفير فائق فيه مزيج جفحر هو ريل فنج جفحر و أخر المزيج يعنى جفحر الاستبدال هو وان تيم فد جفحر الذان خيران من الاخرين لأنهما في كتابة الكلمة المقصود ة (*plaintext*) الي كلمة السر(*ciphertext*) يستطيع استعماله من أين نكون ر يل فنج جفحركان حتى يتصعّب فى التشفير و فك تشفير. وان تيم فد جفحر هو خوارزمية التشفير الكلاسيكية قويا غير قابل للكسر، لأنه يتصفه من مفاتح الخوارزمي بتطويل الي كلمة السر(*plaintext*).

استخدام سوبر تشفير بطريقة وان تيم فد جفحر و ريل فنج جفحر يقوّي سرّة الرسالة. و تنال سرية الرسالة من : الأول من طريقة خوارزمية وان تيم فد جفحر فيه غير قابل للكسر تسعمل قيمة من على نحو عشوائي، و الثانى من طريقة خوارزمية ريل فنج جفحر تستعمل بعد تكوين كلمه السر حتى تصعب فى تسهيله

.

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. Dalam surah Al-Mumtahanah ayat 1, Allah Azza Wa Jalla berfirman yang 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 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 kasih sayang. Aku lebih mengetahui apa yang kamu sembunyikan dana apa yang kamu tanyakan. Dan barangsiapa di antara kamu yang melakukannya, maka sesungguhnya dia telah tersesat dari jalanku yang lurus".

Dalam surah Al-Mumtahanah tersirat bahwa "janganlah kamu menyampaikan sesuatu kabar atau berita-berita", Agama Islam memerintahkan bahwa sebuah berita apapun merupakan hal penting yang perlu dijaga. Sehingga kalimat ini mengartikan bahwa sebuah pesan merupakan berita yang perlu dijaga kerahasiaannya. Maka agar terhindar dari ketidakbenaran dan kesalahpahaman berita-berita perlu adanya solusi agar berita-berita tersebut sampai kepada pihak yang dituju. Jadi agar data tersebut sampai sasaran dapat diterima oleh pihak yang dituju, maka dibuatkanlah suatu metode agar pesan tersebut benar-benar terjamin kerahasiaannya (Setyaningsih, 2009).

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 *Rail Fence Cipher* dan kriptografi klasik *cipher* substitusi yaitu *One Time Pad Cipher*. Di penelitian (Singh, Nandal dan Malik,

2015) menyatakan bahwa keamanan pesan tidak akan menjadi akurat apabila hanya dilakukan dengan menggunakan satu algoritma yaitu algoritma *Rail Fence Cipher* saja, hal ini dikarenakan pada teknik transposisi *Rail Fence Cipher* jika dilakukan dengan terpisah maka *cipher* akan mudah retak. Berdasarkan penelitian yang dilakukan oleh (Laifah, 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 *Rail Fence Cipher,* maka pesan tersebut masih belum bisa dikatakan aman. Karena algoritma *Rail Fence Cipher* sangatlah lemah jika diterapkan sendiri tanpa dikombinasikan dengan algoritma klasik lainnya. Maka, penelitian ini akan mengkombinasikan algoritma *One Time Pad Cipher* dengan algoritma *Rail Fence 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 Rail Fence Cipher"*

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 menggunakan teknik super enkripsi (algoritma *One Time Pad Cipher* dan transformasi *Rail Fence Cipher*)?
2. Bagaimana analisis keamanan algoritma *One Time Pad Cipher* dan transformasi *Rail Fence 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 menggunakan teknik super enkripsi (algoritma *One Time Pad Cipher* dan transformasi *Rail Fence Cipher*).
2. Mengetahui analisis keamanan algoritma *One Time Pad Cipher* dan transformasi *Rail Fence Cipher*.

1.4 Manfaat Penelitian

1. Untuk mendapatkan sebuah pemahaman tentang bagaimana proses enkripsi dan dekripsi pada pesan teks menggunakan teknik super enkripsi (algoritma *One Time Pad Cipher* dan transformasi *Rail Fence Cipher*)
2. Untuk pengetahuan sebuah analisis keamanan algoritma *One Time Pad Cipher* dan transformasi *Rail Fence Cipher.*

1.5 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 teknik super enkripsi menggunakan algoritma *Rail Fence Cipher* dan *Algoritma One Time Pad 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).*
3. Diimplementasikan dengan menggunakan aplikasi Matlab.

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 *rail fence cipher*

Bab III Pembahasan

Bab ini berisi tentang langkah-langkah pembentukan *ciphertext* yang melalui tahap super enkripsi substitusi dan transposisi yan dilakukan melalui metode transformasi *Rail Fence Cipher* dan algoritma *One Time Pad Cipher* sehingga didapatkan suatu *ciphertext* yang telah terenskripsi dan juga berisi implementasi algoritma ke dalam aplikasi Matlab

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. Merumuskan Masalah
2. Mencari data pendukung secara teoritis
3. Menyertakan Pesan teks
4. Menyusun enkripsi dengan Algoritma Super Enkripsi (Algoritma *Rail Fence* dan Algoritma *One Time Pad Cipher*) pada pesan teks.
    a. Menentukan 26 karakter huruf yang akan digunakan sebagai *plaintext*
    b. Mengoperasikan *plaintext* dengan algoritma *Rail Fence* yaitu dengan menuliskan dan membagi *plaintext* menjadi 4 baris secara zig-zag
    c. Menyusun hasil enkripsi dan dekripsi dari algoritma *Rail Fence* secara berbaris menyamping
5. Memaparkan proses penyandian *One Time Pad Cipher* dan *Rail Fence Cipher*
6. Menganalisa tingkat keamanan algoritma *One Time Pad Cipher* dan *Rail Fence 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 :

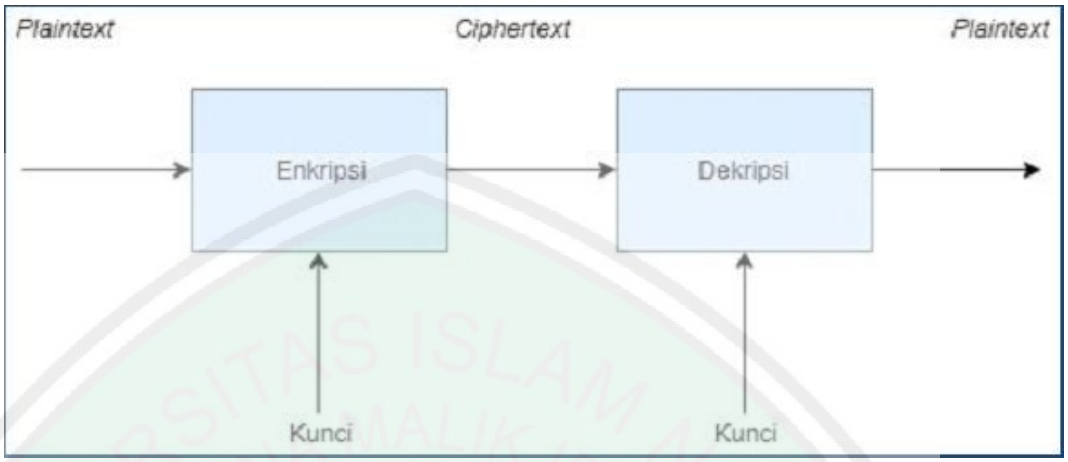
1. Enkripsi

   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.

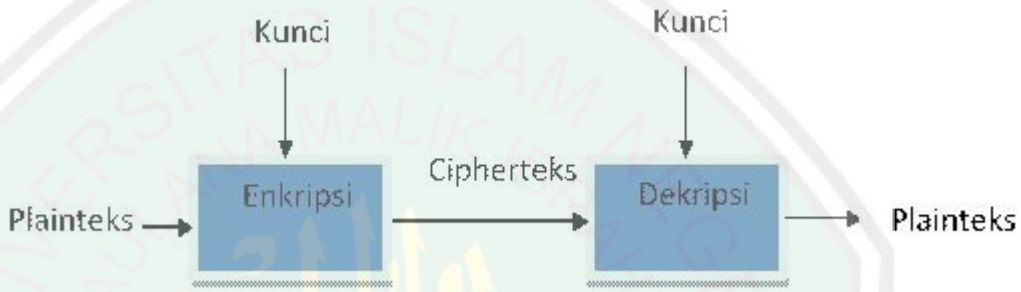
2.1.4 Algoritma Kriptografi Modern

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.

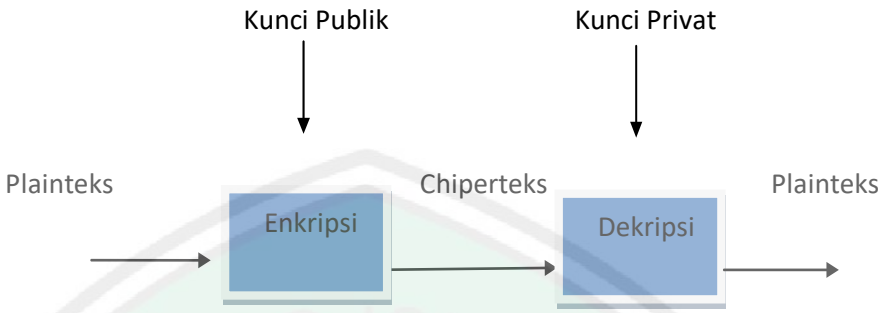


Gambar 2.2 Skema kriptografi simetri

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 Teori Bilangan

Teori bilangan (*number theory*) merupakan teori yang paling umum untuk memahami algoritma kriptografi. Bilangan yang dimaksudkan adalah bilangan bulat (*integer*).

2.2.1 Aritmatika Modulo

Aritmatika modulo menjadi dasar dan memainkan peran penting dalam komputasi bilangan bulat. Aritmatika digunakan pada operasi aritmatika dengan tujuan agar menghasilkan nilai *integer* pada ruang lingkup yang sama (Munir, Matematika Diskrit, 2010). 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. Notasi a mod $m = r$ sedemikian sehingga $a = mq + r$, dengan $0 \leq r < m$.

Bilangan *m* disebut **modulus** atau **modulo**, dan hasil aritmetika modulo *m* terletak di dalam himpunan {0, 1, 2, …, *m* – 1}.

Contoh 7. Beberapa hasil operasi dengan operator modulo:

(i) 23 mod 5 = 3	(23 = 5 · 4 + 3)
(ii) 27 mod 3 = 0	(27 = 3 · 9 + 0)
(iii) 6 mod 8 = 6	(6 = 8 · 0 + 6)
(iv) 0 mod 12 = 0	(0 = 12 · 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'* ≠ 0. Jadi |– 41| mod 9 = 5, sehingga –41 mod 9 = 9 – 5 = 4.

Kongruen

Ditentukan *p*, *q*, *m* adalah bilangan-bilangan bulat dan *m* ≠ 0, p disebut kongruen dengan *q* modulo *m*, ditulis *p* ≡ *q* (mod *m*), jika dan hanya jika *m* | *p*-*q*. Misalnya 38 mod 5 = 3 dan 13 mod 5 = 3, maka kita katakan 38 ≡ 13 (mod 5) (baca: 38 kongruen dengan 13 dalam modulo 5). Misalkan *a* dan *b* adalah bilangan bulat dan *m* adalah bilangan > 0, maka *a* ≡ *b* (mod *m*) jika *m* habis membagi *a* – *b*. Jika *a* tidak kongruen dengan *b* dalam modulus *m*, maka ditulis *a* ≡/ *b* (mod *m*).

Contoh 8.

17 ≡ 2 (mod 3)	(3 habis membagi 17 – 2 = 15)
–7 ≡ 15 (mod 11)	(11 habis membagi –7 – 15 = –22)
12 ≡/ 2 (mod 7)	(7 tidak habis membagi 12 – 2 = 10)

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

Contoh 9.

$17 \equiv 2 \pmod 3$ dapat ditulis sebagai $17 = 2 + 5 \cdot 3$

$-7 \equiv 15 \pmod{11}$ dapat ditulis sebagai $-7 = 15 + (-2)11$

- Berdasarkan definisi aritmetika modulo, kita dapat menuliskan a mod $m = r$ sebagai

$$a \equiv r \pmod m$$

Contoh 10.

Beberapa hasil operasi dengan operator modulo berikut:

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

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

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

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

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

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

Teorema 2. Misalkan m adalah bilangan bulat positif.

a) Jika $a \equiv b \pmod m$ dan c adalah sembarang bilangan bulat maka

(i) $(a + c) \equiv (b + c) \pmod m$

(ii) $ac \equiv bc \pmod m$

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

b) Jika $a \equiv b \pmod m$ dan $c \equiv d \pmod m$, maka

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

(ii) $ac \equiv bd \pmod m$

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

1(ii) $a \equiv b \pmod m$ berarti:

$\Leftrightarrow a = b + km$

$\Leftrightarrow a - b = km$

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

$\Leftrightarrow ac = bc + Km$

$\Leftrightarrow ac \equiv bc \pmod m$

2(i) $a \equiv b \pmod m \Leftrightarrow a = b + k_1 m$

$c \equiv d \pmod m \Leftrightarrow c = d + k_2 m$

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

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

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

Contoh 11.

Misalkan $17 \equiv 2 \pmod 3$ dan $10 \equiv 4 \pmod 3$, maka menurut

Teorema 2:

$17 + 5 = 2 + 5 \pmod 3$	$\Leftrightarrow$	$22 = 7 \pmod 3$
$17 \cdot 5 = 5 \cdot 2 \pmod 3$	$\Leftrightarrow$	$85 = 10 \pmod 3$
$17 + 10 = 2 + 4 \pmod 3$	$\Leftrightarrow$	$27 = 6 \pmod 3$
$17 \cdot 10 = 2 \cdot 4 \pmod 3$	$\Leftrightarrow$	$170 = 8 \pmod 3$

- 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:

  (i) $10 \equiv 4 \pmod 3$ dapat dibagi dengan 2 karena 10/2 = 5 dan 4/2 = 2, dan $5 \equiv 2 \pmod 3$

  (ii) $14 \equiv 8 \pmod 6$ tidak dapat dibagi dengan 2, karena 14/2 = 7 dan 8/2 = 4, tetapi $7 \not\equiv 4 \pmod 6$.

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 *a* sedemikian sehingga

$$aa \equiv 1 \pmod{m}.$$

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

$$pa + qm \equiv 1 \pmod{m}$$

Karena $qm \equiv 0 \pmod{m}$, maka

$$pa \equiv 1 \pmod{m}$$

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*.

2.2.2 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 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 1.1 *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 :

$$Ci = (Pi + Ki) \bmod 26 \qquad (1)$$

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 :

$$Pi = (Ci - Ki) \bmod 26 \qquad (2)$$

Keterangan persamaan :

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

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

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

Dimana Pi adalah rangkaian *plaintext*, Ki adalah *key*, Ci adalah c*iphertext* 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 :

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

$= (6 + 18) \bmod 26$

$= (24) \bmod 26$

$C_1 = 24$

Maka $C_1 =$ **24**

Karakter yang diperoleh dengan nilai *ciphertext* **24** adalah **Y**

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

$= (20 + 8) \bmod 26$

$= (2) \bmod 26$

$C_2 = 2$

Maka $C_2 =$ **2**

Karakter yang diperoleh dengan nilai *ciphertext* **2** adalah **C**

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

$= (19 + 0) \bmod 26$

$= (19) \bmod 26$

$C_3 = 19$

Maka $C_3 =$ **19**

Karakter yang diperoleh dengan nilai *ciphertext* **19** adalah **T**

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

$= (4 + 13) \bmod 26$

$= (17) \bmod 26$

$C_4 = 17$

Maka $C_4 =$ **17**

Karakter yang diperoleh dengan nilai *ciphertext* **17** adalah **R**

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

$= (13 + 6) \bmod 26$

$= (19) \bmod 26$

$C_5 = 19$

Maka $C_5 =$ **19**

Karakter yang diperoleh dengan nilai *ciphertext* **19** adalah **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 :

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

$= (24 - 18) \bmod 26$

$= (6) \bmod 26$

$P_1 = 6$

Karakter yang diperoleh dengan nilai *plaintext* **6** adalah G

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

$= (2 - 8) \bmod 26$

$= (-6) \bmod 26$

$P_2 = 20$

Karakter yang diperoleh dengan nilai *plaintext* **20** adalah U

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

$= (19 - 0) \bmod 26$

$= (19) \bmod 26$

$P_3 = 19$

Karakter yang diperoleh dengan nilai *plaintext* **19** adalah T

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

$= (17 - 13) \bmod 26$

$= (4) \bmod 26$

$P_4 = 4$

Karakter yang diperoleh dengan nilai *plaintext* **4** adalah E

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

$= (19 - 6) \bmod 26$

$= (13) \bmod 26$

$P_5 = 13$

Karakter yang diperoleh dengan nilai *plaintext* **13** adalah N

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

2.4 Algoritma *Rail Fence Cipher*

Algoritma ini berasal dari sebuah *cipher* transposisi. Oleh karena itu *cipher* dapat disebut sebagai *cipher* transposisi karena sebenarnya metode *cipher* transposisi ini mempermutasikan karakter-karakter plainteks, yaitu dengan menyusun ulang urutan karakter dalam pesan teks. Contoh paling sederhana dalam penggunaan *cipher* transposisi adalah dengan membalikkan karakter-karakter dalam suatu kata. Misalkan kata MASBONO dienkripsi menjadi ONOBSAM, ini adalah contoh paling sederhana. Sedangkan contoh *cipher* transposisi yang lebih rumit sebagai berikut:

Misalkan kita mempunyai plainteks

AKU ADALAH SEORANG CAPT

Untuk mengenkripsi pesan, plainteks ditulis secara horizontal dengan lebar kolom tetap, misal sebesar 5 karakter (kunci k=5)

A K U A D

A L A H S

E O R A N

G C A P T

Maka chiperteksnya dibaca secara vertikal menjadi

AAEGKLOCUARAAHAPDSNT

Pada zaman Yunani dahulu, tentara sparta menggunakan sebuah alat yang dinamakan *scrytale*. Alat ini terdiri dari sebuah silinder dan pita panjang dari daun *papyrus*. Pesan dituliskan horizontal dan bila pita dilepaskan, maka huruf-huruf didalamnya telah tersusun membentuk sebuah pesan rahasia. *Scrytale* merupakan sebuah penerapan *cipher*r transposisi pada zaman dahulu.

Cipher transposisi mempunyai berbagai macam algoritma yang berbeda-beda seperti *Rail Fence Cipher, Route Cipher, Columnar Transposition,* dan *Myzkowski Transposition*. Setiap algoritma itu mempunyai sebuah kelebihan masing-masing. Oleh karena itu penulis akan mencoba membahas Algoritma *Rail Fence Cipher*.

Algoritma ini melibatkan penulisan *plainteks* sehingga mempunyai baris atas dan baris bawah yang terpisah. Urutan karakter pada baris atas akan diikuti oleh karakter berikutnya pada baris bawahnya, dan seterusnya sehingga n-rail. Apabila penulisan ke bawah sudah mencapai n, maka penulisan dilakukan ke baris atasnya dan seterusnya. Bila penulisan ke atas juga sudah mencapai n-rail, maka penulisan dilakukan seperti awal. *Ciphertext* dibaca secara horizontal. Untuk lebih jelasnya, berikut adalah contohnya:

Misalkan kita mempunyai *plainteks*

YUNIARTHIE

enkripsi dilakukan dengan kunci k = 3, offset = 0

Y . . . A . . . I .

. U . I . R . H . E

. . N . . . T . . .

maka *ciphertext*-nya menjadi

YAIUIRHENT

Namun enkripsi juga dapat dilakukan dengan memulainya bukan dari baris paling atas (offset = 0), namun bisa juga dari baris lainnya. Dengan menggunakan contoh plainteks di atas:

Enkripsi dilakukan dengan kunci k = 3, offset = 0

. . N . . . T . . .

. U . I . R . H . E

Y . . . A . . . I .

maka cipherteksnya menjadi NTUIRHEYAI

Biasanya penulisan cipherteks dilakukan menjadi blok-blok standar biasanya sepanjang 5 karakter. Bila hasil *cipherteks* tidak habis dibagi dengan panjang karakter, maka penambahan karakter *dummy* dilakukan pada saat pengenkripsian.

Rail Fence Cipher mempunyai kelebihan dibandingkan algoritma lainnya dalam proses penulisan *plainteks* menjadi cipherteks karena penulisan dapat dilakukan di baris mana saja. Hal ini akan menambah kerumitan dalam proses enkripsi maupun dekripsi.

Secara keseluruhan algoritma *cipher* Transposisi ini mempunyai kelemahan karena serumit apapun algoritma yang kita pakai untuk mengubah posisi atau permutasi suatu teks, kita hanya akan mengubah urutan teks (*plainteks*) tersebut tidak mengubahnya menjadi karakter lain. Kemunculan karakter *plainteks* akan sama dengan *cipherteksnya*, hal ini dapat memberikan pertunjuk bahwa proses enkripsi menggunakan salah satu algoritma *cipher* transposisi. Sehingga, usaha untuk memecahkan suatu *cipher* transposisi tidaklah sulit bila kita mencoba semua algoritma *cipher* transposisi.

2.5 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.

2.5.1 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 untyk mencari hasil dari proses enkripsi tersebut hanya perlu membaca karakter dari blok per blok di atas dan akan didapat *ciphertext* akhir sebagai berikut.

VDUDEGJDEDGDXXBHGLQQX

2.5.2 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.

2.6 Kajian Alqur'an tentang Pesan

Ditinjau dari Al-Qur'an, kita sebagai umat manusia harus hati-hati dalam menjaga sebuah pesan yang telah diberikan. Hal ini dikarenakan, sebagai orang beriman kita harus mempunyai sifat amanah (dapat dipercaya) jika kita diberi amanah berupa pesan agar kita wajib menjaga, memelihara dan tidak mengkhianati jika diberi amanah berupa sebuah pesan tersebut. Hal ini dijelaskan dalam Al-Qur'an Surat Al-Ma'arij ayat 32:

وَٱلَّذِينَ هُمْ لِأَمَٰنَٰتِهِمْ وَعَهْدِهِمْ رَٰعُونَ

Yang artinya: *Dan orang-orang yang memelihara amanah-amanah (yang dipikulnya) dan janjinya.* (QS Al-Ma'arij:32)

Ayat tersebut menerangkan tentang orang-orang yang memelihara dan tidak mengkhianati perkara agama yang diamanahkan kepada mereka dan yang orang lain janjikan kepadanya.

Ayat ini Allah Azza Wa Jalla menerangkan bahwa salah satu sifat mukmin yang sangat beruntung adalah suka memelihara dan menjaga amanah-amanah yang dipikulnya, baik itu dari makhluk ataupun dari Allah Azza Wa Jalla. Bilamana ini terjadi pada sesama manusia ataupun makhluk, penerapannya ialah jika diantara mereka sesama manusia mereka mengadakan perjanjian, mereka memenuhinya dengan sempurna. Tidak pula berkhianat dan mereka benar-benar menyampaikan amanah itu sebagai semestinya. Mereka orang yang memelihara amana-amanah pasti sangat menjauhkan diri dari sifat kemunafikan. Seperti dalam sebuah hadist, yang menjelaskan bahwa tanda-tanda orang munafik itu ada tiga, yaitu kalau berbicara suka berdusta, jika menjanjikan sesuatu suka menyalahi janji dan jika diberi amanah akan berkhianat (M. Dawam Raharjo,1996)

Menurut Muhammad Nassib Ar-Rifa'I dalam Buku *Ringkasan Ibnu Katsir Jilid 4* menjelaskan orang-orang yang memelihara amanah-amanah dan janjinya, apabila mereka diberi amanah tidak mengkhianatinya dan bila berjanji tidak pernah melanggarnya. Inilah sifat yang dimiliki oleh orang-orang beriman sedangkan yang sebaliknya adalah sifat-sifat orang munafik. Apabila mereka diberi amanah, mereka tidak khianat dan apabila mereka berjanji mereka tidak ingkar. Orang-orang yang menjaga amanah Allah Azza Wa Jalla dan makhluk-Nya yang mereka emban dan menepati janji-janji tanpa membatakannya, apalagi melanggarnya (Ibnu-Katsir, 2007).

BAB III
PEMBAHASAN

3.1 Teknik 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* di mana terdapat sebuah *plaintext* "KHUMAIRA" dengan *key* "PASURUAN"

Plaintext = "KHUMAIRA"

Key = "PASURUAN"

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

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

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

$= (10 + 15) \bmod 26$

$= (25) \bmod 26$

$C_1 = 25$

Maka $C_1 =$ **25**

Karakter yang diperoleh dengan nilai *ciphertext* **25** adalah **Z**

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

$= (7 + 0) \bmod 26$

$C_2 = 7$

Maka C_2 = **7**

Karakter yang diperoleh dengan nilai *ciphertext* **7** adalah **H**

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

$= (20 + 18) \bmod 26$

$= (38) \bmod 26$

$C_3 = 12$

Maka C_3 = **12**

Karakter yang diperoleh dengan nilai *ciphertext* **12** adalah **M**

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

$= (12 + 20) \bmod 26$

$= (32) \bmod 26$

$C_4 = 6$

Maka C_4 = **6**

Karakter yang diperoleh dengan nilai *ciphertext* **6** adalah **G**

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

$= (0 + 17) \bmod 26$

$= (17) \bmod 26$

$= (17) \bmod 26$

$C_5 = 17$

Maka C_5 = **17**

Karakter yang diperoleh dengan nilai *ciphertext* **17** adalah **R**

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

$= (8 + 20) \bmod 26$

$= (28) \bmod 26$

$C_6 = 2$

Maka $C_6 = 2$

Karakter yang diperoleh dengan nilai *ciphertext* **2** adalah C

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

= (17 + 0) mod 26

= (17) mod 26

C_7 = 17

Maka C_7 = **17**

Karakter yang diperoleh dengan nilai *ciphertext* **17** adalah **R**

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

= (0 + 13) mod 26

= (13) mod 26

C_8 = 13

Maka C_8 = 13

Karakter yang diperoleh dengan nilai *ciphertext* **13** adalah **N**

Plainteks	K	H	U	M	A	I	R	A
Key	P	A	S	U	R	U	A	N
Cipherteks	Z	H	M	G	R	C	R	N

Gambar 3. 1 Hasil persamaan enkripsi *One Time Pad*

Jadi, hasil enkripsi plainteks "KHUMAIRA" adalah "ZHMGRCRN"

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

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

Dengan melakukan konsep yang sama maka didapatkan hasil sebagai berikut :

Plaintext = "KHUMAIRA"

Key = ”PASURUAN”

Ciphertext = ”ZHMGRCRN”

Proses dekripsi dapat dilihat pada perhitungan dibawah ini :

Ciphertext = ”ZHMGRCRN”

Key = ”PASURUAN”

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

Proses Dekripsi

$P_1 = (C_1 - K_1) \bmod 26$
$= (25 - 15) \bmod 26$
$= (10) \bmod 26$
$P_1 = 10$
Karakter yang diperoleh dengan nilai *plaintext* **10** adalah K

$P_2 = (C_2 - K_2) \bmod 26$
$= (7 - 0) \bmod 26$
$= (7) \bmod 26$
$P_2 = 7$
Karakter yang diperoleh dengan nilai *plaintext* **7** adalah **H**

$P_3 = (C_3 - K_3) \bmod 26$
$= (12 - 18) \bmod 26$
$= (-6) \bmod 26$
$P_3 = 20$
Karakter yang diperoleh dengan nilai *plaintext* **20** adalah **U**

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

= (6 - 20) mod 26

= (-14) mod 26

$P_4 = 12$

Karakter yang diperoleh dengan nilai *plaintext* **12** adalah **M**

$P_5 = (C_5 – K_5)$ mod 26

= (17 - 17) mod 26

= (0) mod 26

$P_5 = 0$

Karakter yang diperoleh dengan nilai *plaintext* **0** adalah A

$P_6 = (C_6 – K_6)$ mod 26

= (2- 20) mod 26

= (-18) mod 26

$P_6 = 8$

Karakter yang diperoleh dengan nilai *plaintext* **8** adalah **I**

$P_7 = (C_7 – K_7)$ mod 26

= (17 - 0) mod 26

= (17) mod 26

$P_7 = 17$

Karakter yang diperoleh dengan nilai *plaintext* **17** adalah **R**

$P_8 = (C_8 – K_8)$ mod 26

= (13 - 13) mod 26

= (0) mod 26

$P_8 = 0$

Karakter yang diperoleh dengan nilai *plaintext* **0** adalah **A**

sehingga dapat diperoleh hasil dekripsi sebagai berikut:

Cipherteks	Z	H	M	G	R	C	R	N
Key	P	A	S	U	R	U	A	N
Plainteks	K	H	U	M	A	I	R	A

Gambar 3. 2 Gambar dekripsi *One Time Pad*

Jadi hasil dekripsi dari plainteks "IVKVTVJO" adalah "SURABAIA"

3.1.2 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 dienkrepsikan. *One Time Pad* memiliki kelemahan panjang kunci yang terlalu panjang, tetapi kelemahan itu juga merupakan kelebihannya. Oleh karena itu, untuk mendistribusikankuncinya harus melalui jalur yang berbeda dari pengiriman pesan yang akan dienkripsi.

3.2 Teknik penyandian *Rail Fence Cipher*

Mempunyai plainteks TANGGUNGJAWAB

Enkripsi dilakukan dengan kunci k = 3, offset = 0 (artinya dalam 3 baris dimulai dari baris ke-0 atau awal atau paling atas)

T . . . G . . . J . . . B

. A . G . U . G . A . A .

. . N . . . N . . . W . .

Maka hasil enkripsi cipherteksnya adalah TGJBAGUGAANNW, sesuai dengan pola yang digunakan pada metode *rail fence cipher*.

Proses Dekripsi *Rail fence Cipher*

Ciphertext = **TGJBAGUGAANNW**

Kunci dekripsi = 3

Baris 1 = **TGJB**

Baris 2 = **AGUGAA**

Baris 3 = **NNW**

Maka *ciphertext* diatas disusun dengan jumlah baris 3 agar bisa didekripsi , seperti pada gambar berikut ini:

Baris 1	T		G	J	B		
Baris 2		A	G	U	G	A	a
Baris 3			N	N	W		

Gambar 3.3 Hasil dekripsi *Rail Fence*

Langkah selanjutnya menggambar kembali baris yang sudah ditentukan sesuai karakter pada masing-masing baris.

T				**G**				**J**				**B**
	A		**G**		**U**		**G**		**A**		**A**	
		N				**N**				**W**		

Gambar 3.4 Hasil akhir dekripsi *Rail Fence*

3.2.1 Analisa keamanan *Rail fence Cipher*

Teknik penyandian *Railfence Cipher* adalah teknik penyandian sederhana yang merupakan penyandian dengan teknik transposisi yang merubah posisi dari setiap karakter huruf berdasarkan nilai kunci, hal ini menjadi sangat efektif untuk memecahkan pesan yang tersandikan menggunakan teknik *Railfance Cipher* yaitu dengan mencoba semua kemungkinan kunci di mana kemungkinan kunci dari

teknik *Railfance Cipher* sangatlah terbatas yaitu sejumlah bilangan bulat yang kurang dari jumlah nilai panjang *plainteks* yang ada. Sehingga teknik ini sangat rentan untuk dipecahlan. Misalkan pesan disandikan dengan *railfance cipher* dengan kunci 3, maka hanya menggunakan 3 kali percobaan agar *plainteks* bisa didapatkan.

3.3 Penyandian Super Enkripsi *One Time Pad Cipher* dan *Rail Fence Cipher*

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

Contoh :

Bono sebagai sender akan mengirimkan pesan "**BONOSEORANGBIKERSEJATI"** kepada ozil sebagai receiver. Namun karena bono ingin agar pesan tersebut aman dan tidak diketahui oleh semua orang, maka bono akan menggunakan super enkripsi untuk menjadikan pesan. Teknik yang digunakan adalah *One Time Pad cipher* dengan kunci enkripsi "**AIUDHNSJTKNHDUIIWORHRT**" dan transformasi *Railfence Cipher* dengan kunci 4.

3.3.1 Proses Enkripsi Pesan

One Time Pad Cipher

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

Plaintext = "**BONOSEORANGBIKERSEJATI"**

Key = "**AIUDHNSJTKNHDUIIWORHRT"**

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

$C_1 = (P_1 + K_1) \bmod 26$
$= (1 + 0) \bmod 26$
$= (1) \bmod 26$
$C_1 = 1$
Maka $C_1 =$ **1**
Karakter yang diperoleh dengan nilai *ciphertext* **1** adalah **B**

$C_2 = (P_2 + K_2) \bmod 26$
$= (14 + 8) \bmod 26$
$= (22) \bmod 26$
$C_2 = 22$
Maka $C_2 =$ **22**
Karakter yang diperoleh dengan nilai *ciphertext* **22** adalah **W**

$C_3 = (P_3 + K_3) \bmod 26$
$= (13 + 20) \bmod 26$
$= (33) \bmod 26$
$C_3 = 7$
Maka $C_3 =$ **7**
Karakter yang diperoleh dengan nilai *ciphertext* **7** adalah **H**

$C_4 = (P_4 + K_4) \bmod 26$
$= (14 + 3) \bmod 26$
$= (17) \bmod 26$
$C_4 = 17$
Maka $C_4 =$ **17**
Karakter yang diperoleh dengan nilai *ciphertext* **17** adalah **R**

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

$= (18 + 7) \bmod 26$

$= (25) \bmod 26$

$C_5 = 25$

Maka $C_5 =$ **25**

Karakter yang diperoleh dengan nilai *ciphertext* **25** adalah **Z**

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

$= (4 + 13) \bmod 26$

$= (17) \bmod 26$

$C_6 = 17$

Maka $C_6 =$ **17**

Karakter yang diperoleh dengan nilai *ciphertext* **17** adalah **R**

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

$= (14 + 18) \bmod 26$

$= (32) \bmod 26$

$C_7 = 6$

Maka $C_7 =$ **6**

Karakter yang diperoleh dengan nilai *ciphertext* **6** adalah **G**

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

$= (17 + 9) \bmod 26$

$= (26) \bmod 26$

$C_8 = 0$

Maka $C_8 =$ **0**

Karakter yang diperoleh dengan nilai *ciphertext* **0** adalah **A**

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

$= (0 + 19) \bmod 26$

$= (19) \bmod 26$

$C_9 = 19$

Maka C_9 = **19**

Karakter yang diperoleh dengan nilai *ciphertext* **19** adalah **T**

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

$= (13 + 10) \bmod 26$

$= (23) \bmod 26$

$C_{10} = 23$

Maka C_{10} = **23**

Karakter yang diperoleh dengan nilai *ciphertext* **23** adalah **X**

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

$= (6 + 13) \bmod 26$

$= (19) \bmod 26$

$C_{11} = 19$

Maka C_{11} = **19**

Karakter yang diperoleh dengan nilai *ciphertext* adalah **T**

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

$= (1 + 7) \bmod 26$

$= (8) \bmod 26$

$C_{12} = 8$

Maka C_{12} = **8**

Karakter yang diperoleh dengan nilai *ciphertext* **8** adalah **I**

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

$= (8 + 3) \bmod 26$

$= (11) \bmod 26$

$C_{13} = 11$

Maka C_{13} = **11**

Karakter yang diperoleh dengan nilai *ciphertext* **11** adalah **L**

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

= (10 + 20) mod 26

= (30) mod 26

C_{14} = 4

Maka C_{14} = **4**

Karakter yang diperoleh dengan nilai *ciphertext* **4** adalah **E**

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

= (4 + 8) mod 26

= (12) mod 26

C_{15} = 12

Maka C_{15} = **12**

Karakter yang diperoleh dengan nilai *ciphertext* **12** adalah **M**

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

= (17 + 8) mod 26

= (25) mod 26

C_{16} = 25

Maka C_{16} = **25**

Karakter yang diperoleh dengan nilai *ciphertext* **25** adalah **Z**

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

= (18 + 22) mod 26

= (40) mod 26

C_{17} = 14

Maka C_{17} = **14**

Karakter yang diperoleh dengan nilai *ciphertext* **14** adalah **O**

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

= (4 + 14) mod 26

= (18) mod 26

C_{18} = 18

Maka C_{18} = **18**

Karakter yang diperoleh dengan nilai *ciphertext* **18** adalah **S**

$C_{19} = (P_{19} + K_{19}) \bmod 26$

$= (9 + 17) \bmod 26$

$= (26) \bmod 26$

$C_{19} = 0$

Maka $C_{19} =$ **0**

Karakter yang diperoleh dengan nilai *ciphertext* **0** adalah **A**

$C_{20} = (P_{20} + K_{20}) \bmod 26$

$= (0 + 7) \bmod 26$

$= (7) \bmod 26$

$C_{20} = 7$

Maka $C_{20} =$ **7**

Karakter yang diperoleh dengan nilai *ciphertext* **7** adalah **H**

$C_{21} = (P_{21} + K_{21}) \bmod 26$

$= (19 + 17) \bmod 26$

$= (36) \bmod 26$

$C_{21} = 10$

Maka $C_{21} =$ **10**

Karakter yang diperoleh dengan nilai *ciphertext* **10** adalah **K**

$C_{22} = (P_{22} + K_{22}) \bmod 26$

$= (8 + 19) \bmod 26$

$= (27) \bmod 26$

$C_{22} = 1$

Maka $C_{22} =$ **1**

Karakter yang diperoleh dengan nilai *ciphertext* **1** adalah B

Sehingga *ciphertext* yang didapat adalah **BWHRZRGATXTILEMBOSAHKB,** *ciphertext* tersebut kemudian akan di enkripsi kembali menggunakan *railfence*

cipher.

Railfence Cipher

Langkah pertama yaitu membuat matriks dengan jumlah baris sebanyak nilai dari kunci yang ada di dalam kasus ini bernilai 4 dan jumlah kolom sebanyak jumlah dari karakter pesan yang akan disandikan yang ada di dalam kasus ini sebanyak 22 karakter.

Pesan **BWHRZRGATXTILEMZOSAHKB,** diinputkan ke dalam matriks 4x22 diatas dengan format penginputan zigzag ke kanan

B	.	.	.	.	.	G	.		.	.	.	L	.	.	.		.	A	.	.	.
.	W	.		.	R	.	A	.		.	I	.	E	.	.	.	S	.	H	.	.
.	.	H	.	Z	.		.	T	.	T	.	.	.	M	.	O	.	.	.	K	.
.	.	.	R	.	.	.	.	.	X	.	.	.	.	.	Z	.	.	.	.	.	B

Gambar 3.5 Hasil enkripsi *Rail Fence*

Kemudian mengelompokkan pesan menjadi seperti berikut

- Baris pertama diperoleh BGLA
- Baris kedua diperoleh WRAIESH
- Baris ketiga diperoleh HZTTMOK
- Baris keempat diperoleh RXZB

Untuk mendapatkan hasil enkripsi dari *railfence cipher* denga cara membaca karakternya secara kolom sehingga akan didapatkan hasil enkripsi dari *railfence cipher* yaitu **BGLAWRAIESHHZTTMOKRXZB**

3.3.2 Proses Dekripsi Pesan

Setelah ozil (sebagai pihak receiver) menerima pesan dari bono (sebagai pihak sender) berupa *ciphertext* **BGLAWRAIESHHZTTMOKRXZB,** maka diperlukan teknik dekripsi pesan agar *cphertext* kembali menjadi plainteks yang bisa dibaca dan dipahami. Pada proses dekripsi dilakukan teknik yang berlawanan

dengan proses enkripsi yaitu dilakukan teknik dekripsi *Railfence Cipher* kemudian dilanjutkan dengan *One Time Pad Cipher.*

Railfence Cipher

Seperti pada proses dekripsi pesan *Railfence Cipher*, pertama adalah membuat sebuah matriks dengan ukuran nilai kunci enkripsi x jumlah karakter pesan, kemudian dari kolom 1 dan baris 1 di beri suatu tanda yang mana ini menjadi patokan dalam penginputan pesan tanda ini dibuat bergerak secara zig-zag ke arah kanan. Cara penginputan pada proses dekripsi *Rail Fence Cipher* ini sangatlah berbeda dengan proses sebelumnya di proses enkripsi *Rail Fence Cipher.* Penginputan teks dilakukan secara horizontal sesuai dengan hasil *ciphertext* nya. Sehingga *ciphertext* dari **BGLAWRAIESHHZTTMOKRXZB** akan menjadi seperti berikut ini.

B	.	.	.	.	.	G	.		.	.	.	L	.	.	.		.	A	.	.	.
.	W	.		.	R	.	A	.		.	I	.	E	.	.	.	S	.	H	.	.
.	.	H	.	Z	.		.	T	.	T	.	.	.	M	.	O	.	.	.	K	.
.	.	.	R	.	.	.	.	.	X	.	.	.	.	.	Z	.	.	.	.	.	B

Gambar 3.6 Hasil dekripsi *Rail Fence*

Untuk mendapatkan hasil dekripsi dari *railfence cipher* dengan cara membaca karakternya secara baris ke baris atau kearah kanan. Sehingga didapatkan hasil dekripsi *Rail fence Cipher* **BWHRZRGATXTILEMZOSAHKB,**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/mengembalikan plainteks menjadi data semula, dapat diperlihatkan dengan menggunakan persamaan sebagai berikut:

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

Dengan *ciphertext* **BWHRZRGATXTILEMZOSAHKB,**dan *key*

AIUDHNSJTKNHDUIIWORHRT

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

3.3.2 Proses Dekripsi

$P_1 = (C_1 - K_1) \bmod 26$
$= (1 - 0) \bmod 26$
$= (1) \bmod 26$
$P_1 = 1$
Karakter yang diperoleh dengan nilai *plaintext* **1** adalah **B**

$P_2 = (C_2 - K_2) \bmod 26$
$= (22 - 8) \bmod 26$
$= (14) \bmod 26$
$P_2 = 14$
Karakter yang diperoleh dengan nilai *plaintext* **14** adalah **O**

$P_3 = (C_3 - K_3) \bmod 26$
$= (7 - 20) \bmod 26$
$= (-13) \bmod 26$
$P_3 = 13$
Karakter yang diperoleh dengan nilai *plaintext* **13** adalah **N**

$P_4 = (C_4 - K_4 + 26) \bmod 26$
$= (17 - 3) \bmod 26$
$= (14) \bmod 26$

$P_4 = 14$

Karakter yang diperoleh dengan nilai *plaintext* **14** adalah **O**

$P_5 = (C_5 – K_5) \bmod 26$

$= (25 - 7) \bmod 26$

$= (18) \bmod 26$

$P_5 = 18$

Karakter yang diperoleh dengan nilai *plaintext* **18** adalah **S**

$P_6 = (C_6 – K_6) \bmod 26$

$= (17 - 13) \bmod 26$

$= (4) \bmod 26$

$P_6 = 4$

Karakter yang diperoleh dengan nilai *plaintext* **4** adalah **E**

$P_7 = (C_7 – K_7) \bmod 26$

$= (6 - 18) \bmod 26$

$= (-12) \bmod 26$

$P_7 = 14$

Karakter yang diperoleh dengan nilai *plaintext* **14** adalah **O**

$P_8 = (C_8 – K_8) \bmod 26$

$= (0 - 9) \bmod 26$

$= (-9) \bmod 26$

$P_8 = 17$

Karakter yang diperoleh dengan nilai *plaintext* **17** adalah **R**

$P_9 = (C_9 – K_9) \bmod 26$

$= (19 - 19) \bmod 26$

$= (0) \bmod 26$

$P_9 = 0$

Karakter yang diperoleh dengan nilai *plaintext* **0** adalah **A**

$P_{10} = (C_{10} – K_{10}) \bmod 26$

$= (23 - 10) \bmod 26$

$= (13) \bmod 26$

$P_{10} = 13$

Karakter yang diperoleh dengan nilai *plaintext* **13** adalah **N**

$P_{11} = (C_{11} – K_{11}) \bmod 26$

$= (19 - 13) \bmod 26$

$= (6) \bmod 26$

$P_{11} = 6$

Karakter yang diperoleh dengan nilai *plaintext* **6** adalah **G**

$P_{12} = (C_{12} – K_{12}) \bmod 26$

$= (8 - 7) \bmod 26$

$= (1) \bmod 26$

$P_{12} = 1$

Karakter yang diperoleh dengan nilai *plaintext* **1** adalah **B**

$P_{13} = (C_{13} – K_{13}) \bmod 26$

$= (11 - 3) \bmod 26$

$= (8) \bmod 26$

$P_{13} = 8$

Karakter yang diperoleh dengan nilai *plaintext* **8** adalah **I**

$P_{14} = (C_{14} – K_{14}) \bmod 26$

$= (4 - 20) \bmod 26$

$= (-16) \bmod 26$

$P_{14} = 10$

Karakter yang diperoleh dengan nilai *plaintext* 10 adalah **K**

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

$= (12 - 8) \text{ mod } 26$

$= (4) \text{ mod } 26$

$P_{15} = 4$

Karakter yang diperoleh dengan nilai *plaintext* **4** adalah **E**

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

$= (25 - 8) \text{ mod } 26$

$= (17) \text{ mod } 26$

$P_{16} = 17$

Karakter yang diperoleh dengan nilai *plaintext* **17** adalah **R**

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

$= (14 - 22) \text{ mod } 26$

$= (-8) \text{ mod } 26$

$P_{17} = 18$

Karakter yang diperoleh dengan nilai *plaintext* **18** adalah **S**

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

$= (18 - 14) \text{ mod } 26$

$= (4) \text{ mod } 26$

$P_{18} = 4$

Karakter yang diperoleh dengan nilai *plaintext* **4** adalah **E**

$P_{19} = (C_{19} – K_{19}) \text{ mod } 26$

$= (0 - 17) \text{ mod } 26$

$= (-17) \text{ mod } 26$

$P_{19} = 9$

Karakter yang diperoleh dengan nilai *plaintext* **9** adalah **J**

$P_{20} = (C_{20} – K_{20}) \text{ mod } 26$

$= (7 - 7) \text{ mod } 26$

= (0) mod 26

P_{20} = 0

Karakter yang diperoleh dengan nilai *plaintext* **0** adalah **A**

$P_{21} = (C_{21} – K_{21})$ mod 26

= (10 - 17) mod 26

= (-7) mod 26

P_{21}= 19

Karakter yang diperoleh dengan nilai *plaintext* **19** adalah **T**

$P_{22} = (C_{22} – K_{22})$ mod 26

= (1 - 19) mod 26

= (-18) mod 26

P_{22} = 8

Karakter yang diperoleh dengan nilai *plaintext* **8** adalah **I**

Setelah pesan terdekripsi maka ozil sebagai pihak receiver dapat membaca isi pesan asli yang dikirimkan bono sebagai pihak sender yaitu

BONOSEORANGBIKERSEJATI

3.3.3 Analisa Keamanan Penyandian *One Time Pad Cipher* dan *Rail fence Cipher*

Teknik penyandian *One Time Pad Cipher dan Railfence Cipher* jika digabungkan dalam penerapannya akan menghasilkan keamanan yang sangat sempurma untuk sebuah pesan teks. Kombinasi dari algoritma-algoritma tersebut membuat pesan teks yang terenkripsi maupun yang didekripsi sangatlah aman jika menggunakan teknik penyandian-penyandian tersebut.

3.4 Implementasi Super Enkripsi Dengan Matlab

Untuk merancang sebuah program aplikasi penyandian maka langkah-langkah penyandian super enkripsi pada pembahasan sebelumnya kemudian dapat disajikan dalam bentuk *flowchart* sehingga memudahkan proses pembuatan aplikasi di Matlab.

Proses Super Enkripsi dua algoritma *One Time Pad Cipher* dan algoritma *Rail Fence Cipher* menggunakan jenis tahapan yang pertama dienkripsi lalu yang kedua didekripsi agar hasil dari teks tersebut kembali seperti semula.

1. Implementasi Super Enkripsi *One Time Pad Cipher* dan *Railfence Cipher* proses enkripsi

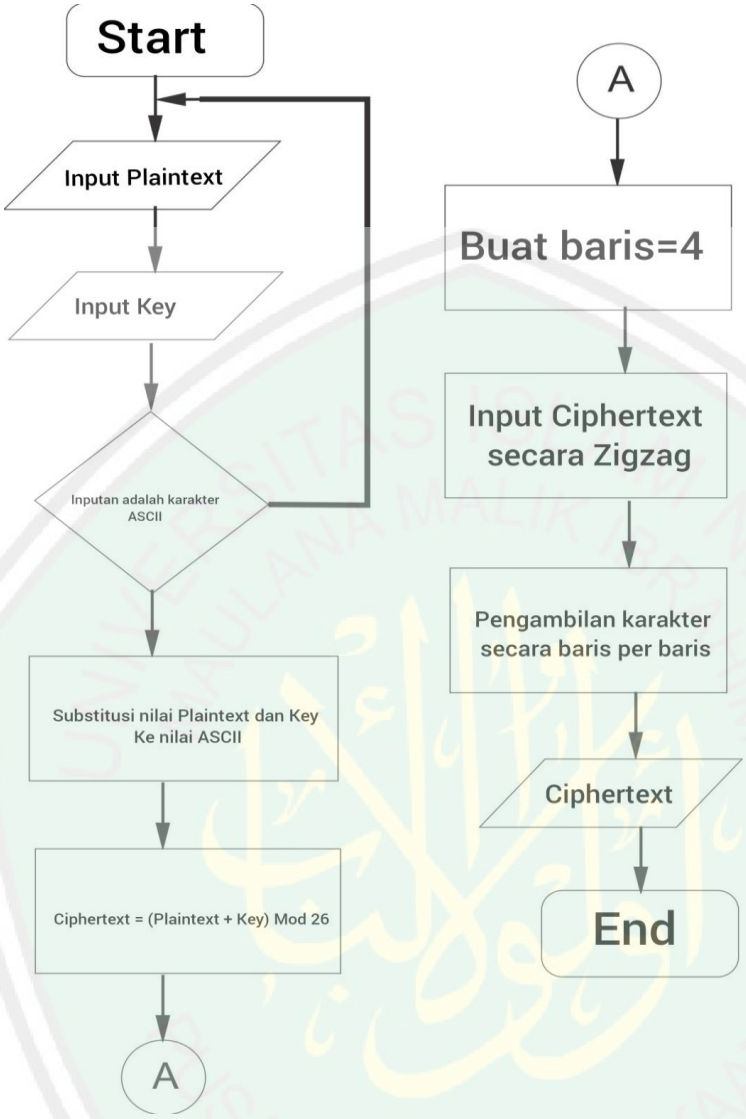
```
function x=otp(a)
c=enkripsi(otp(a),baris)
```

2. Implementasi Super Enkripsi *One Time Pad Cipher* dan *Railfence Cipher* proses dekripsi

```
function x=dotp(c)
p=dotp(dekrip(c,4))
```

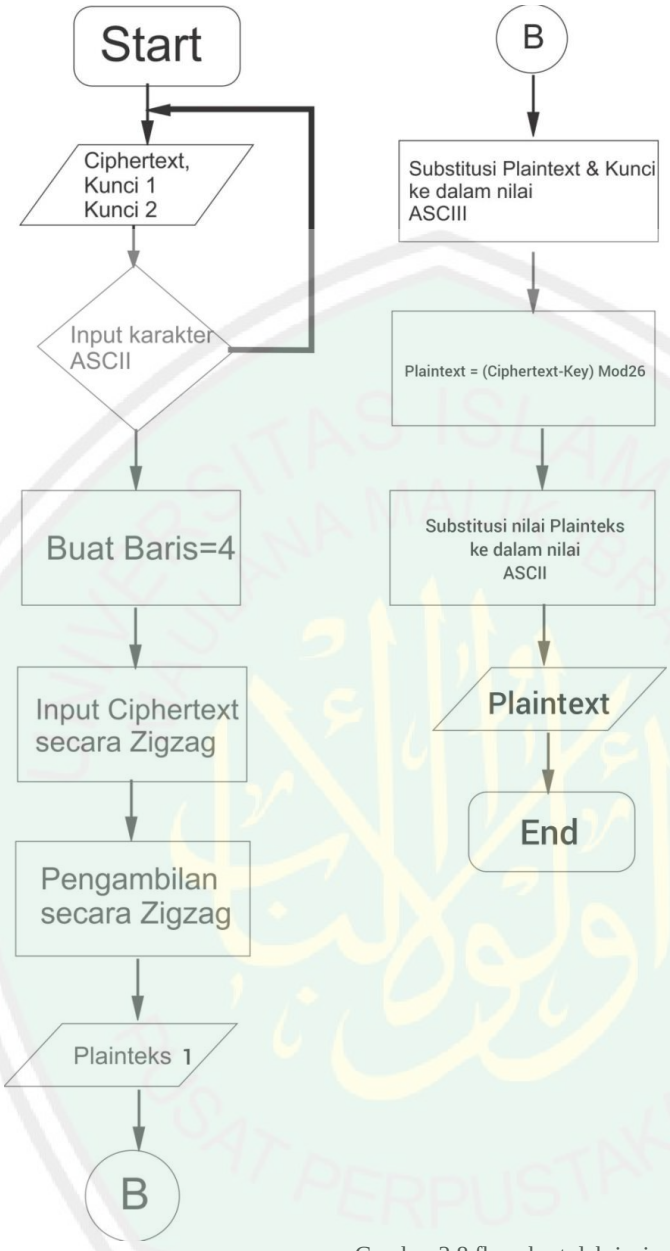
Berikut ini ialah rancangan flowchart penyandian super enkripsi *One Time Pad Cipher* dan *Railfence Cipher* dengan langkah-langkah yang telah dijelaskan sebelumnya.

- flowchart enkripsi



Gambar 3. 7 flowchart enkripsi

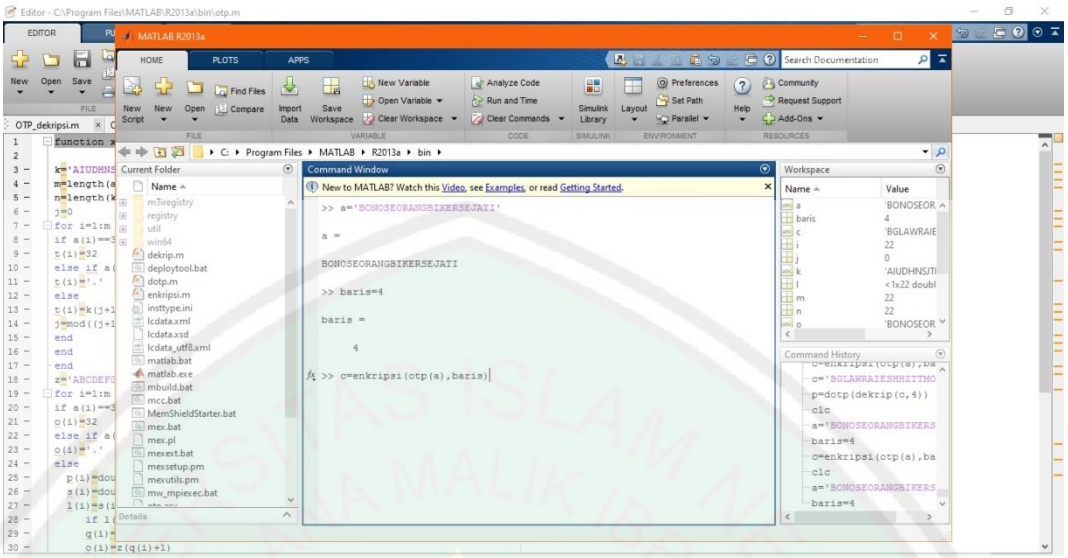
- flowchart dekripsi



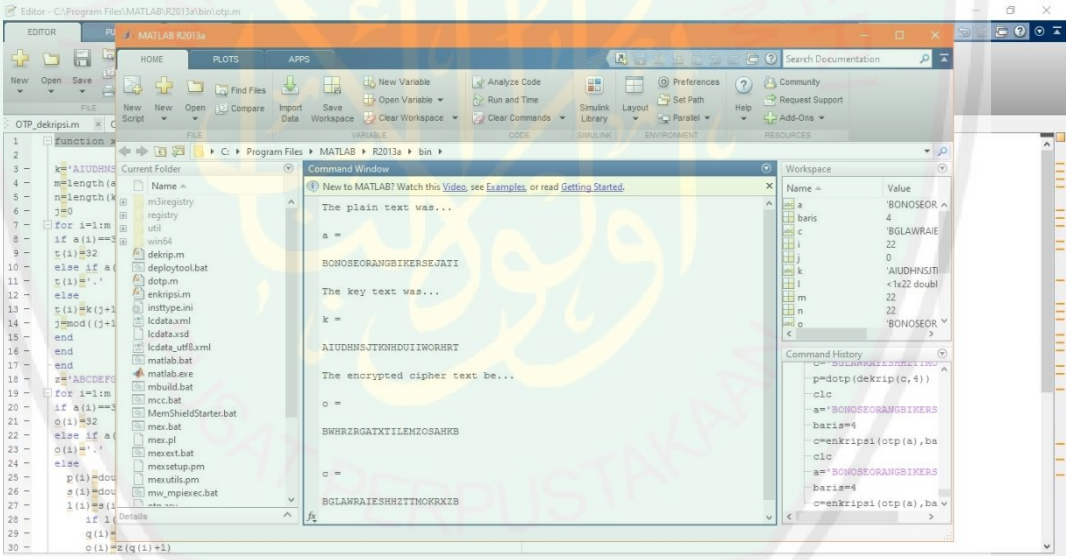
Gambar 3.8 flowchart dekripsi

Langkah berikutnya adalah mengimplementasikan rangkaian flowchart tersebut kedalam skrip bahasa pemrograman Matlab. Implementasi tersebut akan melakukan proses enkripsi dan dekripsi sebagai berikut

a) Implementasi Enkripsi

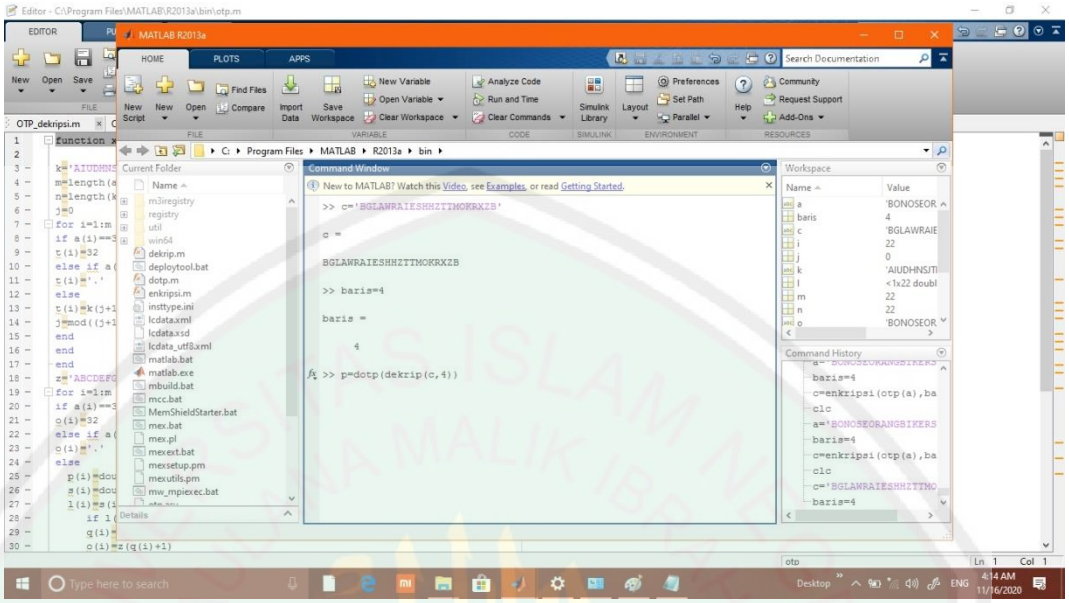


Gambar 3.9 Input *Plaintext* dan *key* (Enkripsi)

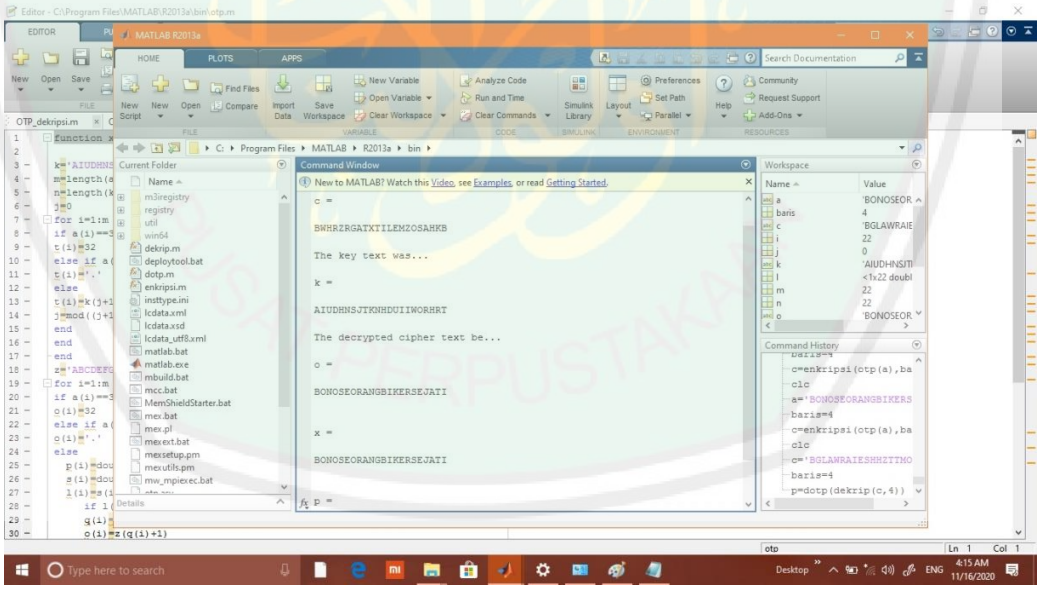


Gambar 3. 10 Hasil output Enkripsi

b) Implementasi Dekripsi



Gambar 3.11 Input *Plaintext* dan *key* (Dekripsi)



Gambar 3. 12 Output Hasil Dekripsi

Output teks pada aplikasi tersebut menunjukkan hasil yang sama dengan hasil penyandian secara manual. jadi di proses dekripsi di program aplikasi Matlab berhasil mengembalikan sesuai dengan *plaintext.*

3.5 Kajian Agama Islam

3.5.1 Perspektif Islam terhadap pentingnya sifat amanah

Agama Islam mewajibkan setiap pemeluknya untuk memliki hati dan jiwa yang kuat, dengan hati yang kuat semua hak-hak Allah Azza Wa Jalla dan hak-hak manusia dapat dipelihara dengan baik, semua amal perbuatan dapa dijauhkan dari sikap hati yang lalai. Karena itulah agama Islam ini mewajibkan setiap muslim memiliki sifat dapat dipercaya (amanah). Amanah dalam perspetif Islam memiliki makna dan kandungan yang sangat luas, dimana seluruh makna dan kandunga tersebut tertuju pada satu pengertian yaitu setiap orang merasakan bahwa Allah Azza Wa Jalla senantiasa menyertainya dalam setiap urusan yang dibebankan kepadanya, dan setiap orang memahami dengan penuh keyakinan bahwa kelak ia akan diminta pertanggung jawaban terhadap apa yang mereka kerjakan.

Sementara itu pengertian amanah secara awam seringkali diletakkan pada pehaman yang sempit, yaitu sebatas memelihara barang titipan, padahal maknanya jauh lebih besar dan lebih berat dari yang diduga. Amanah sebuah kewajiban, dimana sudah seharsunya semua orang islam saling mewasiati dan memohon kepada Allah Azza Wa Jalla dalam mejaganya.

Sedangkan makna amanah yang penulis maksudkan di sini adalah amanah dalam pengertian yang luas, yaitu mengenai tanggung jawab umat manusia, baik kepada Allah Azza Wa Jalla yang menciptakannya maupun terhadap sesama

makhluk. Amanah merupakan segala sesuatu yang diemban manusia, baik sesuatu terkait dengan urusan agama maupun urusan dunia, baik terkait dengan perkataan maupun dengan perbuatan dimana puncak amana adalah penjagaan dan pelaksanaan.

3.5.2 Super Enkripsi dengan sifat amanah

Konsep super enkripsi sangat mengedepankan penggunaan gabungan dari dua jenis metode yang berbeda dengan maksud untuk meningkatkan keamanan dari pesan tersebut agar metode tersebut tidak mudah dipecahkan. Dengan konsep tersebut, hal ini dapat mendukung pengamalan pada sifat amanah yaitu tentang penjagaan dan memliharanya yang Allah Azza Wa Jalla berfirman dalam surat Al-Hijr 9:

اِنَّا نَحْنُ نَزَّلْنَا الذِّكْرَ وَ اِنَّا لَهٗ لَحٰفِظُوْنَ

Artinya: *Sesungguhnya Kami-lah yang menurunkan Al-Qur'an, dan sesungguhnya Kami benar-benar memeliharanya.*

Dari Sayid Qutb yang telah menulis kitab Tafsir fenomenal di abad ini, dalam kitab Tafsirnya Fi Dhilalil Quran menyebutkan bahwa al-Quran sejak kemunculannya telah mengalami banyak usaha perubahan talbis dan tahrif dan juga fitnah-fitnah dari kelompok-kelompok yang menyimpang, sebagian mereka seperti Yahudi dan penyeru Qaumiyah bisa melakukan penakwilan terhadap hadits dan ayat al-Quran atau untuk mendukung pendapat mereka, tetapi satu yang tidak dapat mereka lakukan yaitu mendatangkan satu ayat seperti dalam al-Quran, sehingga al-Quran tetap terjaga sebagaimana ia diturunkan oleh Allah.

Hal ini juga berlaku pada konsep penyandian super enkripsi, karena hasil penyandian yang menggunakan teknik gabungan akan menghasilkan suatu *output* yang lebih rumit dan sulit dipecahkan.

BAB IV
PENUTUP

4.1 Kesimpulan

Berdasarkan hasil analisis dan implentasi program yang telah dilakukan dengan menggunakan aplikasi Matlab diatas, maka dapat disimpulkan bahwa:

1. Teknik super enkripsi dengan menggunakan metode algoritma *One Time Pad Cipher* dengan menggunakan persamaan $C_1 = (P_1 + K_1)$ mod 26 di proses enkripsinya, lalu hasil dari enkripsi tersebut disandikan kembali dengan menggunakan algoritma *Rail Fence Cipher* dengan mengurutkan karakter pada baris atas yang kemudian diikuti oleh karakter selanjutnya pada baris bawah, dan seterusnya sebanyak *plaintext* dan sifatnya algoritma tersebut yang zig-zag. Proses pengembalian pesan tersebut lalu menggunakan dekripsi dari *Algoritma Rail Fence Cipher* kemudian didekripsi lagi dengan menggunakan algoritma *One Time Pad Cipher* yang persamaan dekripsi tersebut menggunakan $Pi = (Ci - Ki)$ mod 26

2. Keamanan teknik super enkripsi terletak pada panjang karakter kunci *One Time Pad Cipher* dan besar nilai kunci *Rail Fence Cipher* yang digunakan. Penggunaan dua buah jenis *cipher* ini sangatlah memungkinkan agar tingkat keamanan pesan yang disandikan menjadi dua kali lipat.

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

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.

Setyaningsih, E 2015. *Kriptografi & Implementasinya Menggunakan Matlab*. Yogyakarta, ANDI

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

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

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

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

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.

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

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

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

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

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

LAMPIRAN

Enkripsi One Time Pad Cipher

```
function x=otp(a)

k='AIUDHNSJTKNHDUIIWORHRT'
m=length(a)
n=length(k)
j=0
for i=1:m
if a(i)==32
t(i)=32
else if a(i)=='.'
t(i)='.'
else
t(i)=k(j+1)
j=mod((j+1),n)
end
end
end
z='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
for i=1:m
if a(i)==32
o(i)=32
else if a(i)=='.'
o(i)='.'
else
  p(i)=double(a(i))-65
  s(i)=double(t(i))-65
  l(i)=s(i)+p(i)
    if l(i)>25
    q(i)= mod(l(i),26)
    o(i)=z(q(i)+1)
    else
    o(i)=z(l(i)+1)
    end
end
```

Dekripsi One Time Pad Cipher

```
function x=dotp(c)

k='AIUDHNSJTKNHDUIIWORHRT'
m=length(c)
n=length(k)
j=0
for i=1:m
if c(i)==32
t(i)=32
else if c(i)=='.'
t(i)='.'
else
t(i)=k(j+1)
```

```
j=mod((j+1),n)
end
end
end
z='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
for i=1:m
if c(i)==32
o(i)=32
else if c(i)=='.'
o(i)='.'
else
  p(i)=double(c(i))-65
  s(i)=double(t(i))-65
  l(i)=p(i)-s(i)
    if (l(i)>25) | (l(i)<25)
    q(i)= mod(l(i),26)
    o(i)=z(q(i)+1)
    else
    o(i)=z(l(i)+1)
    end
end
end
end
clc
disp('The cipher text was...')
c
disp('The key text was...')
k
disp('The decrypted cipher text be...')
o
x=o
end
```

Enkripsi Rail Fence Cipher

```
function c=enkripsi(x,baris)
n=size(x,2);
jeda1=2*baris-2;
jeda2=0;
c=[];
for i=1:baris
   k=i;
   c=[c x(k)];
   while k<=n
      k=k+jeda1;
      if k<=n & jeda1~=0
         c=[c x(k)];
      end
      k=k+jeda2;
      if k<=n & jeda2~=0
         c=[c x(k)];
      end
   end
   jeda1=jeda1-2;
   jeda2=jeda2+2;
```

```
end

Dekripsi Rail Fence Cipher

function p=dekrip(x,baris)
n=size(x,2);
jeda1=2*baris-2;
jeda2=0;
pos=0;
for i=1:baris
   k=i;
   pos=pos+1;
   deciper(k)=[x(pos)];
   while k<=n
      k=k+jeda1;
      if k<=n & jeda1~=0
      pos=pos+1;
      deciper(k)=[x(pos)];
   end
   k=k+jeda2;
   if k<=n & jeda2~=0
      pos=pos+1;
      deciper(k)=[x(pos)];
   end
end
jeda1=jeda1-2;
jeda2=jeda2+2;
end

c=[];
for i=1:n
   c=[c deciper(i)];
end
p=c
end


end
end
clc
disp('The plain text was...')
a
disp('The key text was...')
k
disp('The encrypted cipher text be...')
o
x=o;
end
```

RIWAYAT HIDUP

Firdaus Adji S, biasa dipanggil Adji, lahir di Blitar pada tanggal 7 Juli 1995. Bertempat tinggal di Kelurahan Lesanpuro RT 03 RW 03 Kecamatan Kedungkandang Kota Malang. Anak tunggal dari bapak Amat Subaweh dan Ibu Emi Kristanti

Mulai menempuh pendidikan dasar di SDN Lesanpuro 3 Malang pada tahun 2001 hingga 2007, menempuh pendidikan menengah pertama di SMPN 6 Malang pada tahun 2007 hingga 2010, dan menempuh pendidikan menengah atasnya di SMKN 4 Grafika Malang pada tahun 2010 hingga 2013. Selanjutnya pada tahun 2013, melanjutkan pendidikan di Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang mengambil Jurusan Matematika.

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

BUKTI KONSULTASI SKRIPSI

Nama : Firdaus Adji S
NIM : 14610066
Fakultas/Jurusan : Sains dan Teknologi/Matematika
Judul Skripsi : Implementasi Algoritma *One Time Pad Cipher* dan *Transformasi Rail Fence* pada pesan teks
Pembimbing I : Muhammad Khudzaifa, M.Si
Pembimbing II : Ach. Nasichuddin, MA

No	Tanggal	Hal	Tanda Tangan
1	13 Januari 2020	Konsultasi BAB I & II	1
2	03 Februari 2020	Revisi BAB I & II	2
3	17 April 2020	ACC BAB I & II	3
4	26 Agustus 2020	Konsultasi BAB I, II & III	4
5	30 Agustus 2020	Revisi BAB I, II & III	5
6	11 September 2020	ACC BAB I, II & III	6
7	20 September 2020	Konsultasi BAB IV	7
8	5 Oktober 2020	Konsultasi BAB IV	8
9	8 Oktober 2020	Revisi BAB IV	9
10	14 Oktober 2020	Revisi BAB IV	10
11	19 Oktober 2020	Revisi BAB 1, II & III	11
12	27 Oktober 2020	ACC BAB I, II & III	12
13	03 November 2020	ACC BAB IV	13
14	17 November 2020	Konsultasi Keagamaan	14
15	19 November 2020	ACC Keagamaan	15
16	23 November 2020	ACC Keseluruhan	16