# IMPLEMENTASI PENYANDIAN SUPER ENKRIPSI VIGENERE CIPHER DAN RAILFENCE CIPHER MENGGUNAKAN PYTHON

**SKRIPSI**

**OLEH**
**SIGIT DENI SANTOSO**
**NIM. 15610062**

**JURUSAN MATEMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM**
**MALANG**
**2019**

# IMPLEMENTASI PENYANDIAN SUPER ENKRIPSI VIGENERE CIPHER DAN RAILFENCE CIPHER MENGGUNAKAN PYTHON

**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**
**SIGIT DENI SANTOSO**
**NIM. 15610062**

**JURUSAN MATEMATIKA**
**FAKULTAS SAINS DAN TEKNOLOGI**
**UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG**
**2019**

**IMPLEMENTASI PENYANDIAN SUPER ENKRIPSI VIGENERE CIPHER DAN RAILFENCE MENGGUNAKAN PYTHON**

**SKRIPSI**

**Oleh**
**SIGIT DENI SANTOSO**
**NIM. 15610062**

Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 30 Agustus 2019

| Pembimbing I, | Pembimbing II, |
|---|---|
| Muhammad Khudzaifah, M.Si | Mohammad Nafie Jauhari, M.Si |
| NIP. 19900511 20160801 1 057 | NIPT. 20130902 1 318 |

Mengetahui,
Ketua Jurusan Matematika

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

**IMPLEMENTASI PENYANDIAN SUPER ENKRIPSI VIGENERE CIPHER DAN RAILFENCE CIPHER MENGGUNAKAN PYTHON**

**SKRIPSI**

**Oleh**
**Sigit Deni Santoso**
**NIM. 15610062**

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 11 Oktober 2019

| | | | |
|---|---|---|---|
| Penguji Utama | : | Dr. H. Imam Sujarwo, M.Pd | ..... |
| Ketua Penguji | : | Dewi Ismiarti, M.Si | ..... |
| Sekretaris Penguji | : | Muhammad Khudzaifah, M.Si | ..... |
| Anggota Penguji | : | Mohammad Nafie Jauhari, M.Si | ..... |

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 : Sigit Deni Santoso

NIM : 15610062

Jurusan : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Penyandian Super Enkripsi *Vigenere Cipher* dan *Railfence Cipher* Menggunakan Python

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

Malang, 28 Agustus 2019
Yang membuat pernyataan,


Sigit Deni Santoso
NIM. 15610062

## MOTO

"There is only one thing that makes a dream impossible to achieve: the fear of failure."

**(Paulo Coelho - The Alchemist)**

**PERSEMBAHAN**

بِسْمِ اللهِ الرَّحْمٰنِ الرَّحِيْمِ

Alhamdulillahi Robbil'alamin, dengan mengucap syukur kepada Allah Swt., Penulis mempersembahkan skripsi ini untuk kedua orang tua, Bapak Arifin, dan Ibu Umi Toyibah yang selalu memberikan doa, dukungan dan lain sebagainya yang mungkin tidak bisa penulis balas dengan apapun, serta kakak Ahmad Mansur Fadli, yang selalu memberikan motivasi kepada penulis.

## KATA PENGANTAR

*Assalamu'alaikum Wr.Wb.*

Segala puji bagi Allah Swt. yang telah melimpahkan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini sebagai syarat untk memperoleh gelar sarjana di Jurusan Matematika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang

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

1. Prof. Dr. 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 Khudzaifah, M.Si, selaku dosen pembimbing skripsi I, yang telah menyisihkan waktu untuk memberikan arahan dan berbagi ilmunya kepada penulis.
5. M. Nafie Jauhari, M.Si, selaku dosen pembimbing skripsi II, yang telah banyak memberikan arahan, nasihat dan motivasi selama penulisan skripsi ini.
6. Segenap sivitas akademika Jurusan Matematika, terutama seluruh dosen, terima kasih atas segenap ilmu dan bimbingannya.

7. Bapak ibu serta kakak tercinta yang selalu memberikan do'a, semangat, serta motivasi kepada penulis sampai saat ini.
8. Seluruh teman – teman di Jurusan Matematika angkatan 2015 (LATTICE), khususnya Matematika B, teman – teman santri PP Anwarul Huda terutama penghuni kamar D6 dan teman – teman "Ahlul Qohwah" terima kasih atas segala pengalaman berharga dan kenangan yang terukir rapi dan abadi.
9. Semua pihak yang tidak mungkin penulis sebut satu persatu terima kasih atas keikhlasan bantuan moral, material dan spiritual yang sudah diberikan kepada penulis.

Penulis berharap semoga skripsi ini dapat memberikan manfaat kepada para pembaca khususnya bagi penulis secara pribadi. *Amin Ya Rabbal Alamin.*

*Wassalamu'alaikum Wr.Wb.*

Malang, 28 Agustus 2019

Penulis

## DAFTAR ISI

## DAFTAR GAMBAR

# ABSTRAK

Santoso, Sigit Deni. 2019. **Implementasi Penyandian Super Enkripsi *Vigenere Cipher* dan *Railfence Cipher* Menggunakan Python.** Skripsi. Jurusan Matematika Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II). Mohammad Nafie Jauhari, M.Si.

**Kata kunci:** Enkripsi, Dekripsi, Super Enkripsi, *Vigenere Cipher*, *Railfence Cipher*

Enkripsi merupakan proses menyandikan pesan teks menjadi pesan tak terbaca dan dekripsi merupakan proses kebalikanya. Penelitian ini bertujuan untuk meningkatkan tingkat keamanan pesan yang dienkripsi menggunakan metode Super Enkripsi.

Super enkripsi merupakan suatu metode enkripsi yang menggabungkan dua metode yaitu subtitusi dan transposisi. Metode subtitusi bertujuan merubah setiap entri pesan melalui operasi matematika dengan kunci yang sudah ditentukan. Metode transposisi bertujuan untuk merubah susunan entri pesan. pada penelitian ini, algoritma *Vigenere Cipher* sebagai implementasi metode subtitusi dan algoritma *Railfence Cipher* sebagai implementasi metode transposisi.

Penggunaan super enkripsi dengan *Vigenere Cipher* dan *Railfence Cipher* akan melipat gandakan kemanan dari pesan. keamanan pertama terletak dari tingkat keamanan enkripsi pesan *vigenere 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 dignakan. Selanjutnya keamanan kedua, pesan yang sudah tersandikan akan diacak susunanya menggunakan *Railfence Cipher* sehingga membuat pesan semakin sulit dipecahkan.

# ABSTRACT

Santoso, Sigit Deni. 2019. **Implementation of Super Encryption Vigenere Cipher and Railfence Cipher using Python.** Thesis. Departement of Mathematics, Faculty of Science and Technology, State Islamic University of Maulana Malik Ibrahim Malang. Advisors: (I) Muhammad Khudzaifah, M.Si. (II). Mohammad Nafie Jauhari, M.Si.

**Keyword:** Encryption, Decryption, Super Encryption, Vigenere Cipher, Railfence Cipher

Encryption is the process of encoding text messages into unread messages and decryption is the reverse process. This research aims to increase the level of security of encrypted messages using the Super Encryption method.

Super encryption is an encryption method that combines two methods, namely substitution and transposition. The substitution method aims to change each message entry through a mathematical operation with a predetermined key. The transposition method aims to change the order of message entries. in this study, the Vigenere Cipher algorithm is the implementation of the substitution method and the Railfence Cipher algorithm is the transposition method implementation.

The use of super encryption with Vigenere Cipher and Railfence Cipher will double the security of the message. The first security lies in the level of encryption security message Vigenere cipher that depends on the number of variations of characters that can be used, to improve this, in this study ASCII is used to increase the variation of characters that can be used. Furthermore, the second security, messages that have been encrypted will be encrypted stacking using the Railfence Cipher so that it makes the message more difficult to solve.

# ملخص


سانتوسو ، سيجيت ديني. ٢٠١٩. تنفيذ سوبر التشفير *Vigenere cipher* و *Railfence cipher* باستخدام *Python*. شعبة الرياضيات، كلية العلوم و التكنولوجيا، الجامعة الإسلامية الحكومية مولانا مالك إبراهيم مالانج. المشرف: (١)مُحَمَّد حذيفة الماجستير (٢)مُحَمَّد نافع جوهري الماجستير.

**الكلمات الرئيسية**: التشفير، فك التشفير، سوبر التشفير، *Vigenere cipher*، *Railfence cipher*

التشفير هو عملية ترميز الرسائل النصية في رسائل غير مقروءة وفك التشفير هو العملية العكسية. يهدف هذا البحث إلى زيادة مستوى أمان الرسائل المشفرة باستخدام طريقة التشفير الفائق. سوبر التشفير هو طريقة تشفير تجمع بين طريقتين، هما الاستبدال والتنقل. تهدف طريقة الاستبدال إلى تغيير كل إدخال رسالة من خلال عملية رياضية باستخدام مفتاح محدد مسبقًا. تهدف طريقة النقل إلى تغيير ترتيب إدخالات الرسالة. في هذه الدراسة، هي *Vigenere cipher* كتطبيق لطريقة الإحلال ولوغاريتم *Railfence cipher* هي كتطبيق لطريقة التحويل. استخدام التشفير الفائق باستخدام *Vigenere cipher* و *Railfence cipher* سيضاعف من أمان الرسالة. يكمن الأمن الأول في مستوى *Vigenere cipher* لرسالة أمان التشفير الذي يعتمد على عدد الأشكال المختلفة للأحرف التي يمكن استخدامها، لتحسين ذلك، في هذه الدراسة، يستخدم ASCII لزيادة تباين الحروف التي يمكن استخدامها. علاوة على ذلك ، يتم تشفير الرسائل التي تم ترميزها، وهي الحماية الثانية، باستخدام *Railfence cipher*، مما يجعل حل الرسالة أكثر صعوبة.

# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Keamanan merupakan hal yang sangat penting bagi setiap individu. Baik keamanan dari hal yang berupa kejahatan fisik, hingga tindak kejahatan dunia maya. Allah berfirman dalam QS. Al-An'am ayat 81:

فَأَيُّ الْفَرِيقَيْنِ أَحَقُّ بِالْأَمْنِ ۖ إِنْ كُنْتُمْ تَعْلَمُونَ

Artinya: *"Maka manakah di antara dua golongan itu yang lebih berhak mendapat keamanan (dari malapetaka), jika kamu mengetahui?".*

Ayat tersebut menjelaskan bahwa ada dua golongan, dimana salah satu golongan tersebut lebih pantas mendapatkan keamanan. Hal ini menunjukkan bahwa jika seseorang ingin mendapatkan keamanan maka orang tersebut harus mengusahakan agar dirinya pantas mendapatkan keamanan.

Bertambah majunya zaman dan semakin pesatnya perkembangan teknologi justru membuat tindak kejahatan di dunia maya semakin marak terjadi. Hal tersebut dikarenakan proses pertukaran informasi sering dilakukan melalui jaringan umum seperti contohnya login pada website, penggunaan mesin ATM (*Automatic Teller Machine*), dan lainnya. Hal ini menyebabkan tindak kejahatan seperti *scamming, carding, cracking* dan sejenisnya menjadi marak terjadi, yang akan sangat berbahaya terhadap informasi-informasi pribadi yang sifatnya rahasia. Oleh karena itu, untuk melindungi informasi yang mengalami pertukaan pada jaringan umum diperlukan teknik mengubah dan menyamarkan pesan sehingga pihak yang tidak berhak menerimanya akan

kesulitan untuk menyadap dan mencuri informasi rahasia. Sebagaimana firman Allah dalam QS An-Nisa ayat 58 :

إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمَانَاتِ إِلَىٰ أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ ۚ إِنَّ اللَّهَ نِعِمَّا يَعِظُكُمْ بِهِ ۗ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا

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

Salah satu teknik untuk mengamankan informasi adalah dengan menggunakan konsep enkripsi dan dekripsi. Enkripsi merupakan sebuah teknik yang diakukan untuk merubah informasi yang bisa dipahami menjadi bentuk informasi yang sulit dipahami, sedangkan dekripsi pesan adalah kebalikan dari enkripsi, yakni proses merubah informasi yang sulit dipahami kembali menjadi informasi yang bisa dipahami. Enkripsi pesan memungkinkan informasi yang dikirimkan sulit terbaca oleh orang yang bukan penerimanya.

Dalam prosesnya, enkripsi dan dekripsi memiliki suatu protokol kunci yaitu sebuah kunci yang disepakati oleh pihak penerima dan pihak pengirim pesan sehingga kedua pihak dapat menentukan kunci rahasia yang sama. Ada banyak jenis-jenis teknik penyandian yang bisa dilakukan. Salah satu teknik penyandian yang sederhana adalah teknik penyandian menggunakan teknik subtitusi, namun kadang kala teknik enkripsi ini masih bisa dibobol oleh pihak pihak yang tidak bertanggung jawab. Oleh karena itu untuk meminimalisir hal tersebut perlu ditingkatkan lagi tingkat keamanan dalam penyandiannya.

Super enkripsi merupakan teknik yang menggabungkan teknik penyandian subtitusi dan transposisi sehingga dapat memaksimalkan keamanan dari pesan yang akan dikirimkan. Dalam hal ini teknik penyandian *vigenere cipher* yang merupakan teknik penyandian subtitusi akan dikombinasikan dengan teknik *railfence cipher* yang merupakan teknik penyandian transposisi. Penelitian tentang penggunaan algoritma *railfence cipher* pernah dilakukan oleh Navaneethan Ramkesh tahun 2016, penelitian tersebut mengkombinasikan algoritma *railfence cipher* dengan Caesar *cipher* untuk meningkatkan kemanan informasi. Sedangkan penelitian mengenai *vigenere cipher* sudah banyak dilakukan namun sebagian besar hanya menggunakan teknik *vigenere cipher* penyandian huruf alphabet saja. Salah satu nya adalah penelitian dari Prabowo (2015). Penelitian tersebut meningkatkan keamanan *vigenere cipher* dengan melakukan penyandian *vigenere cipher* sebanyak tiga kali.

Kedua teknik ini dipilih karena ralatif mudah untuk di lakukan namun memiliki tingkat keamanan informasi yang mumpuni karena dengan menggunakan dua jenis teknik penyandian maka akan melipat gandakan tingkat keamananya, dan nantinya akan diimplementasikan menggunakan Python. Berdasarkan uraian di atas maka penulis menyusunnya dalam penelitian dengan judul "Implementasi Penyandian Super Enkripsi *Vigenere cipher* dan *Railfence cipher* Menggunakan Python"

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah dalam penelitian ini adalah untuk mengetahui bagaimanakah proses penyandian menggunakan super enkripsi *vigenere cipher* dan *railfence cipher*, analisa keamanan super enkripsi *vigenere cipher* dan *railfence cipher* serta implementasinya menggunakan Python?

## 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah di atas, maka tujuan yang ingin dicapai dalam penelitian ini adalah untuk mengetahui proses penyandian menggunakan super enkripsi *vigenere cipher* dan algoritma railfence, analisis keamanan *vigenere cipher* dan algoritma railfence serta implementasi *vigenere cipher* dan algoritma railfence menggunakan Python.

## 1.4 Manfaat Penelitian

Manfaat penelitian ini adalah untuk memperdalam pemahaman pada kajian kriptografi super enkripsi, pengetahuan analisis keamanan super enkripsi serta pengetahuan tentang cara impelementasi kriptografi super enkripsi *vigenere cipher* dan *railfence cipher* mengggunakan Python.

## 1.5 Batasan Masalah

Untuk mendekati sasaran yang diharapkan maka permasalahan dilakukan pembatasan antara lain:

1. Metode yang digunakan adalah *vigenere cipher* dan *railfence cipher*.
2. Teknik penyandian menggunakan subtitusi kode ASCII (UTF-8).
3. Karakter spasi (“ “) dianggap mempunyai nilai sesuai dengan tabel ASCII.
4. Diimplementasikan menggunakan aplikasi Python.

## 1.6 Metode Penelitian

Metode penelitian yang dilakukan dalam penelitian ini adalah studi literatur dengan mengkaji buku buku serta jurnal yang berkaitan dengan topik enkripsi dan dekripsi algoritma *vigenere cipher*, *railfence cipher*, dan super enkripsi

Adapun langkah-langkah penelitian ini adalah sebagai berikut:

1. Memaparkan proses penyandian *Vigenere cipher* dan *Railfence cipher*
2. memberikan contoh serta langkah – langkah enkripsi dan dekripsi *vigenere cipher* dan *railfence cipher*,
3. menganalisis tingkat keamanan algoritma *vigenere cipher* dan *railfence cipher* serta kombinasi kedua algoritma.
4. Mengimplementasikan alur penyandian super enkripsi kedalam bentuk program dengan menggunakan bahasa Python.

## 1.7 Sistematika Penulisan

Adapun sistematika penulisan yang digunakan penulis terdiri dari empat bab yang masing-masing terdapat beberapa subbab seperti berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metode penelitian dan sistematika penulisan.

Bab II Kajian Pustaka

Bab ini berisi tentang definisi maupun teorema teorema yang mendukung topik yaitu aritmatika modulo, kriptografi, Super Enkripsi, Python, serta kajian agama.

Bab III Pembahasan

Bab ini berisi tentang penjabaran teknik super enkripsi *vigenere cipher* dan *railfence cipher* menggunakan ASCII secara manual, implementasinya menggunakan Python, dan kajian keagamaan

Bab IV Penutup

Bab ini menyajikan poin poin hasil dari pembahasan secara garis besar berupa kesimpulan dan saran untuk penelitian selanjutnya.

# BAB II

# KAJIAN PUSTAKA

## 2.1 Teori Bilangan

Dalam pengertian yang ketat, kajian tentang sifat-sifat bilangan asli disebut dengan teori bilangan. Dalam pengertian yang lebih luas, teori bilangan mempelajari bilangan dan sifat-sifatnya. Sebagai salah satu cabang matematika, teori bilangan dapat disebut sebagai "aritmetika lanjut *(advanced aritmetics*)" karena terutama berkaitan dengan sifat-sifat bilangan asli (Muhsetyo, 1997). Teori bilangan merupakan dasar perhitungan dan menjadi salah satu teori yang mendasari pemahaman kriptografi, khususnya sistem kriptografi kunci publik. Bilangan yang dimaksud adalah bilangan bulat.

### 2.1.1 Bilangan Bulat

Bilangan bulat adalah bilangan yang ada dalam himpunan berikut {…, −3, −2, − 1,0,1,2,3 … }. Himpunan semua bilangan bulat dinotasikan dengan $\mathbb{Z}$ yang diambil dari kata *Zahlen* dari bahasa Jerman atau dinotasikan dengan I yang diambil dari huruf pertama kata Integer dari bahasa Inggris. Himpunan bilangan bulat dibagi tiga, yaitu bilangan bulat positif, yaitu bilangan bulat yang lebih besar dari nol yang dituliskan $\mathbb{Z}^+$, nol, dan bilangan bulat negatif, yaitu bilangan bulat yang lebih kecil dari nol yang dituliskan $\mathbb{Z}^-$ (Abdussakir, 2009).

Himpunan bilangan bulat dilengkapi dengan dua buah operasi yaitu operasi penjumlahan dan perkalian, dilambangkan $(\mathbb{Z}, +, \cdot)$ membentuk suatu

sistem matematika yang disebut gelanggan atau ring (Abdussakir, 2009), yaitu memenuhi beberapa sifat - sifat berikut:

1. Tertutup terhadap penjumlahan (+)

   Misalkan $a$ dan $b$ adalah anggota $\mathbb{Z}$, berlaku $a + b \in \mathbb{Z}$

2. Komutatif terhadap penjumlahan (+)

   Misalkan $a, b \in \mathbb{Z}$, maka $a + b = b + a$

3. Asosiatif terhadap penjumlahan (+)

   Misalkan $a, b, c \in \mathbb{Z}$, maka $(a + b) + c = a + (b + c)$

4. Adanya elemen identitas terhadap penjumlahan (+)

   Terdapat $0 \in R$, sehingga $a + 0 = 0 + a = a$, $\forall\, a \in \mathbb{Z}$

5. Adanya unsur invers terhadap penjumlahan (+)

   Misal $a \in \mathbb{Z}$, Terdapat$-a \in \mathbb{Z}$, sehingga $a + (-a) = (-a) + a = 0$

6. Tertutup terhadap perkalian (.)

   Misalkan $a$ dan $b$ adalah anggota $\mathbb{Z}$, berlaku $a.b \in \mathbb{Z}$

7. Asosiatif terhadap perkalian (.)

   Misalkan $a, b, c \in \mathbb{Z}$, maka $(a.b).c = a.(b.c)$

8. Distributif perkalian (.) terhadap penjumlahan (+)

   Misalkan $a, b, c \in \mathbb{Z}$, maka $a.(b + c) = (a.b) + (a.c)$

   dan $(a + b).c = (a.c) + (b.c)$ (Gallian, 2010).

Suatu ring disebut ring komutatif jika memiliki tambahan sifat komutatif terhadap perkalian. Lebih lanjut $\mathbb{Z}$ adalah ring komutatif dan memiliki unsur kesatuan $1 \in \mathbb{Z}$ sehingga $1 \cdot a = a \cdot 1 = a$, $\forall\, a \in \mathbb{Z}$ (Gallian, 2010)

Himpunan bilangan bulat berperan sangat penting dalam kriptografi karena banyak algoritma kriptografi yang menggunakan sifat-sifat himpunan bilangan bulat dalam melakukan proses penyandiannya.

### 2.1.2 Keterbagian

Sifat-sifat yang berkaitan dengan keterbagian (*divisibility*) merupakan dasar pengembangan teori bilangan. Penjelasan mengenai definisi dan teorema yang berkaitan dengan keterbagian dibahas dalam banyak buku. Berikut ini adalah definisi dan teorema keterbagian.

**Definisi 2.1**

*Misal $a, b \in \mathbb{Z}$ dengan $a \neq 0$. Bilangan bulat $a$ dikatakan membagi $b$, ditulis $a|b$, jika dan hanya jika $b = ax$, untuk suatu $x \in \mathbb{Z}$* (Abdussakir, 2009).

Ada beberapa hal yang dapat diambil dari definisi keterbagian di atas yaitu:

1. $1|x$, untuk setiap $x \in \mathbb{Z}$, karena ada $x \in \mathbb{Z}$, sehingga $x = 1 \cdot x$
2. $x|0$, untuk setiap $x \in \mathbb{Z}$, dengan $x \neq 0$, karena ada $0 \in \mathbb{Z}$, sehingga $0 = x \cdot 0$
3. $x|x$, untuk setiap $x \in \mathbb{Z}$, dengan $x \neq 0$, karena ada $1 \in \mathbb{Z}$, sehingga $x = x \cdot 1$
4. $x|(-x)$, untuk setiap $x \in \mathbb{Z}$, dengan $x \neq 0$, karena ada $-1 \in \mathbb{Z}$, sehingga $-x = x \cdot (-1)$

Contoh:

1. $4|12$, sebab ada $3 \in \mathbb{Z}$, sehingga $12 = 4 \cdot 3$
2. $15|60$, sebab ada $4 \in \mathbb{Z}$, sehingga $60 = 15 \cdot 4$

**Teorema 2.1**

Diberikan $a, b, c \in \mathbb{Z}$.

1. Jika $a|b$ maka $a|bx$ untuk setiap bilangan bulat $x$
2. Jika $a|b$ dan $b|c$ maka $a|c$
3. Jika $a|b$ dan $a|c$ maka $a|(bx + cy)$ untuk setiap $x,y \in \mathbb{Z}$
4. Jika $a|b$ dan $b|a$ maka $a = \pm b$; $a \neq 0, b \neq 0$
5. Jika $a|b$, $a > 0$, dan $b > 0$, maka $a \leq b$
6. Untuk setiap bilangan bulat $m \neq 0$, $a|b$ jika dan hanya jika $ma|\, mb$

(Irawan, 2014).

**Bukti:**

1. Jika $a|b$, maka ada $y \in \mathbb{Z}$, sehingga $b = ay$. Akibatnya, untuk setiap $x \in \mathbb{Z}$ diperoleh $bx = (ay)\, x = a(yx)$. Karena pada bilangan bulat berlaku sifat tertutup pada perkalian maka terdapat $p = yx \in \mathbb{Z}$. Sehingga berlaku $bx = ap$ jadi, $a|bx$.
2. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Dan $b|c$, maka $c = by$ untuk $y \in \mathbb{Z}$. Diperoleh $c = by = (ax)y = a(xy)$, untuk suatu $xy \in \mathbb{Z}$. Jadi, $a|c$.
3. Jika $a|b$ maka $b = ap$ untuk $p \in \mathbb{Z}$. Dan $a|c$, maka $c = aq$ untuk $q \in \mathbb{Z}$. Akibatnya $bx = (ap)x$ untuk setiap $x \in \mathbb{Z}$ dan $cy = (aq)\, y$ untuk setiap $y \in \mathbb{Z}$. Diperoleh $bx + cy = (ap)x + (aq)\, y = a\,(px + qy)$. Jadi, $a|(bx + cy)$.
4. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Dan $b|a$, maka $a = by$ untuk $y \in \mathbb{Z}$. Diperoleh $b = ax = (by)x = b(yx)$ maka $b - b(yx) = b(1 - yx) = 0$ karena $b \neq 0$, maka $1 - yx = 0$ atau $xy = 1$. Diperoleh $x = y = 1$ atau $x = y = -1$ sehingga didapatkan $a = \pm b$.

5. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Jika $a > 0$, $b > 0$ dan $b = ax$ maka $x > 0$. Untuk $x = 1$ maka dipenuhi $a = b$. Sedangkan untuk $x > 1$ maka $b > a$. Jadi $a \leq b$.
6. Jika $a|b$, maka $b = ax$ untuk $x \in \mathbb{Z}$. Akibatnya untuk $m \in \mathbb{Z}$ dan $m \neq 0$ maka berlaku $mb = m(ax) = (ma)x$. Jadi $ma|mb$. Jika $ma|mb$ dan $m \neq 0$, maka $mb = (ma)\,x$ untuk suatu $x \in \mathbb{Z}$. $mb = (ma)\,x = m(ax)$ atau $mb - m(ax) = m(b - ax) = 0$. Karena $m \neq 0$, maka $b - ax = 0$ atau $b = ax$ untuk suatu $x \in \mathbb{Z}$. Jadi $a|b$.

### 2.1.3 Algoritma Pembagian

Algoritma pembagian merupakan salah satu pokok bahasan dari Teori Bilangan yang berkaitan dengan sifat pembagian dalam matematika. Berikut teorema yang menjelaskan tentang algoritma pembagian.

**Teorema 2.2**

*Misalkan a dan b bilangan bulat dan b > 0, maka ada bilangan bulat q dan r yang unik (tunggal) yang memenuhi a = qb + r dengan 0 ≤ r < b. dimana q disebut hasil bagi dan r disebut sisa dari pembagian a oleh b* (Judson & Beezer, 2016).

**Bukti:**

Akan dibuktikan bahwa q dan r benar benar ada. Dan jika ada dua bilangan berbeda $q'$ dan $r'$ dengan $a = q'b + r'$ maka $q = q'$dan $r = r'$.

Akan ditunjukkan keberadaan q dan r, misalkan

$$S = \{a - bk : k \in \mathbb{Z} \text{ dan } a - bk \geq 0\}$$

Jika $0 \in S$, maka $b$ membagi $a$,dengan $q = \frac{a}{b}$ dan $r = 0$. Jika $0 \notin S$, dan ditunjukkan bahwa $S$ tidak kosong. Jika $a > 0$, maka $a - b \cdot 0 \in S$. Jika $a < 0$, maka $a - b(2a) = a(1 - 2b) \in S$. Sehingga $S$ tidak kosong $(S \neq \emptyset)$. Berdasarkan *Well Ordering Principle*, himpunan *S* memuat unsur terkecil sebut saja r. berdasarkan definisi *S*, maka ada bilangan bulat q yang memenuhi $r = a - bq$. Sehingga $a = bq + r$ dengan $r \geq 0$.

Kemudian ditunjukkan $r < b$. Misal $r > b$ maka $a - b(q + 1) = a - bq - b = r - b > 0$. Sehingga didapatkan $a - b(q + 1)$ dalam himpunan *S*. Namun kemudian $a - b(q + 1) < a - bq$, yang akan bertentangan dengan fakta bahwa $r = a - bq$ adalah anggota terkecil dari himpunan S. Dengan demikian pengandaian tersebut salah, maka $r \leq b$. Karena $0 \notin S$,$r \neq b$ sehingga $r < b$.

Selanjutnya akan ditunjukkan keunikan (ketunggalan) dari q dan r, diandaikan q dan r tidak unik akan didapatkan $a = bq + r, 0 \leq r \leq b$ dan $a = bq' + r', 0 \leq r' < b$, untuk suatu bilangan bulat $q', r'$.

Sehingga $bq + r = bq' + r'$diasumsikan $r' \geq r$. Diperoleh $b(q - q') = r' - r$. Karena itu $b$ harus membagi $r' - r$ dan $0 \leq r' - r \leq r' < b$. Hal ini hanya mungkin jika $r' - r = 0$, sehingga $r = r'$ dan $q = q'$

### 2.1.4 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). Contohnya pada kriptografi klasik yang

menggunakan alfabet latin dari "A" sampai "Z", langkah pertama adalah memetakan {A,...,Z} ke {0,…,25}. Aritmatika modulo digunakan dalam kriptografi klasik agar transformasi penyandian selalu bernilai {0,…,25} sehingga hasil penyandian memiliki pasangan simbol yang digunakan (Sadikin, 2012).

### 2.1.5 Operator Modulo

Operator modulo memerlukan dua masukan antara lain sebuah integer $a$ dan modulus $n$ dimana $n > 0$. Operasi ini mengembalikan $r$, dimana $r$ adalah sisa dari pembagian $a$ oleh $n$ (Sadikin, 2012). Operator yang digunakan dalam modulo dinotasikan dengan $mod$, operator $mod$ memberikan sisa pembagian (Munir, Matematika Diskrit, 2010).

**Definisi 2.2** (Munir, Matematika Diskrit, 2010)

Misal $a$ adalah bilangan bulat dan $n$ adalah bilangan bulat positif. Operasi $a \bmod n$ memberikan sisa $r$ jika $a$ dibagi $m$. Dengan kata lain, $a \bmod n = r$ sedemikian sehingga $a = nq + r$, dengan $0 \leq r < n$.

Notasi:$a \bmod n = r$ sedemikian sehingga $a = nq + r$, dengan $0 \leq r < n$.

Contoh:

a) Hasil bagi 31 oleh 7 adalah 4 dengan sisa pembagian 3 atau dapat ditulis $31 = (7 \times 4) + 3$ sehingga $31 \bmod 7 = 3$

b) Hasil bagi 17 oleh 2 adalah 8 dengan sisa pembagian 1 atau dapat ditulis $17 = (2 \times 8) + 1$ sehingga $17 \bmod 2 = 1$.

## 2.2 Himpunan dan Fungsi (Pemetaan)

Selain teori bilangan, istilah himpunan dan konsep fungsi (pemetaan) juga merupakan dasar dalam algoritma kriptografi. Hal ini dikarenakan kriptografi menerapkan konsep memetakan *plaintext* menjadi *ciphertext*. Berikut ini beberapa definisi dari himpunan dan fungsi

**Definisi 2.3** Himpunan

Himpunan adalah kumpulan dari obyek atau elemen. Misalkan $A$ adalah himpunan, jika x adalah sebuah obyek pada $A$, maka $x$ dikatakan anggota dari $A$ dan dapat ditulis $x \in A$. Jika $x$ bukan anggota dari $A$, maka ditulis $x \notin A$ (Bhattacharya, 1994).

**Definisi 2.4** Fungsi atau Pemetaan

Fungsi $f$ dari himpunan $A$ ke $B$ adalah aturan yang memasangkan setiap elemen $a \in A$ tepat satu elemen $b \in B$. Himpunan $A$ disebut domain dari $f$ dan himpunan $B$ disebut kodomain dari $f$ (Gallian, 2010).

Dalam penulisannya, dapat ditulis dengan $f: A \to B$ yang berarti bahwa $f$ adalah pemetaan dari $A$ ke $B$.

**Definisi 2.5** Fungsi satu - satu

Suatu fungsi $f$ dari himpunan A ke himpunan B disebut fungsi satu - satu (injektif) jika untuk setiap $a_1, a_2 \in A, f(a_1) = f(a_2)$ mengakibatkan $a_1 = a_2$.

Istilah satu - satu memastikan bahwa satu elemen B hanya dapat merupakan peta dari hanya satu elemen A. Fungsi $f$ adalah satu - satu jika dan hanya jika $a_1 \neq a_2$ mengakibatkan $f(a_1) \neq f(a_2)$. Artinya, elemen-elemen yang

berbeda dari A memetakan elemen-elemen yang berbeda dari B (Gallian, 2010).

**Definisi 2.6** Fungsi *onto*

Suatu fungsi f dari himpunan A ke himpunan B dikatakan onto jika setiap elemen B adalah peta dari paling sedikit satu elemen A. Dengan simbol $f: \mathrm{A} \to \mathrm{B}$ adalah into jika untuk setiap $b \in B$ terdapat setidalnya satu $a \in A$ sedemikian sehingga $f(a) = b$ (Gallian, 2010).

**Definisi 2.7** Fungsi *bijektif*

Fungsi $f$ dari himpunan A ke himpunan B dikatakan bijektif jika dan hanya jika $f$ satu – satu dan f onto (Bhattacharya, 1994).

Konsep himpunan dan fungsi diterapkan dalam kriptografi terutama terkait proses enkripsi dan dekripsi. Enkripsi dan dekripsi merupakan suatu proses yang memetakan elemen-elemen antara himpunan *plaintext* dan *ciphertext*. Misalkan P adalah himpunan *plaintext*, dan C adalah himpunan *ciphertext*, maka fungsi enkripsi E memetakan P ke C, ditulis $E(P) = C$. Dan fungsi dekripsi D memetakan C ke P, ditulis $D(C) = P$ (Munir, Kriptografi, 2006).

**Definisi 2.8** Fungsi *invers*

Dimisalkan $f$ adalah fungsi satu – satu dari himpunan A ke himpunan B. fungsi invers dari $f$ adalah fungsi yang memasangkan sebuah elemen $b \in B$ kepada elemen $a \in A$ sedemikian sehingga $f(a) = b$. Fungsi invers dari $f$ di notasikan dengan $f^{-1}$. Oleh sebab itu, $f^{-1}(b) = a$ jika dan hanya jika $f(a) = b$ (Rosen, 2012).

Fungsi yang memiliki *invers* disebut fungsi *invertible*. Dalam penyandian pesan, proses dekripsi mengembalikan pesan ke pesan asal sebelum proses

enkripsi sehingga harus memenuhi persamaan *D(E(P))=P* dimana *D* adalah proses dekripsi, *E* adalah proses enkripsi, dan *P* adalah pesan. Sehingga fungsi enkripsi dan dekripsi dalam penyandian pesan haruslah fungsi *invertible* (Munir, Kriptografi, 2006).

## 2.3 Kriptografi

### 2.3.1 Pengertian Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu *Crypto* yang berarti rahasia dan *Grapho* yang berarti menulis. Sehingga secara umum kriptografi merupakan ilmu dan seni untuk menjaga keamanan pesan (Ariyus, 2006). Dalam pengertian lain kriptografi merupakan ilmu yang bersandarkan pada teknik matematika yang memberikan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas (Sadikin, 2012). Sehingga tidak hanya menjadi teknik dalam penyembunyian pesan, kriptografi modern dapat digunakan sebagai teknik untuk pengamanan informasi.

Di dalam kriptografi, banyak ditemukan berbagai istilah (terminologi). Adapun istilah-istilah yang kerap kali digunakan adalah sebagai berikut (Sadikin, 2012).

a. Pesan, *Plaintext*, dan *Ciphertext*

Pesan adalah data ataupun suatu informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan ialah *plaintext*, atau teks jelas. *Ciphertext* adalah suatu bentuk pesan yang tersandikan yang berupa pesan yang tidak memiliki makna. Disandikannya suatu pesan adalah agar pesan tersebut tidak dapat dimengerti oleh pihak lainnya.

b. Pengirim dan Penerima

Suatu aktivitas komunikasi data, akan melibatkan pertukaran antara dua entitas, yakni pengirim dan penerima. Pengirim adalah entitas yang mengirim pesan kepada entitas lainnya. Sedangkan penerima adalah entitas yang menerima pesan (Munir, Kriptografi, 2006). Dalam suatu pengiriman pesan, pengirim tentu menginginkan pesan dapat dikirim secara aman. Untuk mengamankannya, pengirim biasanya akan menyandikan pesan yang dikirimkan tersebut.

c. Enkripsi dan Dekripsi

Suatu proses untuk menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi *(encryption).* Sedangkan proses pengembalian dari *ciphertext* manjadi *plaintext* dinamakan dekripsi *(decription).* Enkripsi dan dekripsi merupakan suatu proses yang memetakan elemen-elemen antara himpunan *plaintext* dan *ciphertext*.

d. *Cipher* dan Kunci

Algoritma kriptografi disebut juga *cipher* yaitu aturan atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi (Munir, Kriptografi, 2006). Pengiriman pesan dalam kriptografi modern membutuhkan kunci untuk menjaga kerahasiaan pesan. Kunci (*key*) merupakan parameter yang digunakan untuk proses pengenkripsian dan pendekripsian pesan. Kunci dapat berupa deretan bilangan maupun string. Dengan menggunakan kunci K maka proses

enkripsi dan dekripsi dapat ditulis sebagai $EK(P) = C$ dan $DK(C) = P$, dan kedua fungsi tersebut memenuhi $DK(EK(P)) = P$

### 2.3.2 Tujuan kriptografi

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

1. Kerahasiaan (*confidentiality*), merupakan suatu layanan yang digunakan untuk menjaga isi dari informasi dari pihak-pihak yang tak berhak untuk mendapatkannya.
2. Integritas data (*data integrity*), merupakan suatu layanan dimana menjamin bahwa pesan masih asli, dan belum dimanipulasi oleh pihak-pihak yang tidak berhak. Realisasi layanan ini di dalam kriptografi, adalah dengan menggunakan tanda tangan digital.
3. Otentifikasi (*authentication*), merupakan suatu layanan yang berhubungan dengan identifikasi.
4. Nirpenyangkalan (*non-repudiation*), merupakan suatu layanan untuk mencegah entitas yang saling berkomunikasi melakukan penyangkalan. Misalkan salah satu dari entitas menyangkal telah mengirim maupun menerima pesan.

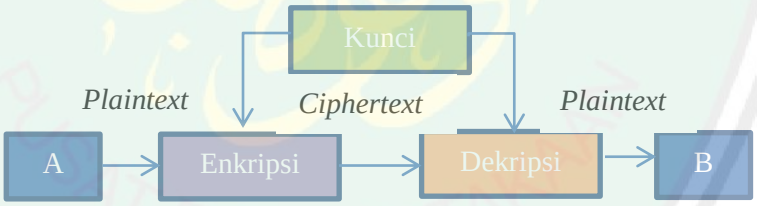
### 2.3.3 Algoritma Kriptografi

Algoritma kriptografi berdasarkan jenis kunci yang digunakan dapat dibedakan menjadi dua jenis yaitu :

1. Algoritma Simetris

Algoritma simetris merupakan algoritma kriptografi yang menggunakan kunci yang sama dalam proses enkripsi dan dekripsinya. Dalam kriptografi kunci simetris dapat diasumsikan bahwa penerima dan pengirim pesan telah terlebih dahulu berbagi kunci sebelum pesan dikirimkan. Keamanan dari sistem ini terletak pada kerahasiaan kuncinya. (Kamil, 2016)

Pada umumnya yang termasuk ke dalam kriptografi simetris ini beroperas dalam mode blok (*block cipher*), yaitu setiap kali proses enkripsi atau dekripsi dilakukan terhadap satu blok data (yang berukuran tertentu), atau beroperasi dalam mode aliran (*stream cipher*), yaitu setiap kali enkripsi atau dekripsi dilakukan terhadap satu bit atau satu byte data. (Kamil, 2016). Proses dari skema kriptografi simetris dapat dilihat pada gambar 2.1.




**Gambar 2. 1** Algoritma Simetris

Kelebihan kriptografi simetris adalah (Kamil, 2016):

a. Proses enkripsi atau dekripsi kriptografi simetris membutuhkan waktu yang singkat.

b. Ukuran kunci simetris relatif lebih pendek.

c. Otentikasi pengiriman pesan langsung dari *ciphertext* yang diterima, karena kunci hanya diketahui oleh penerima dan pengirim saja.

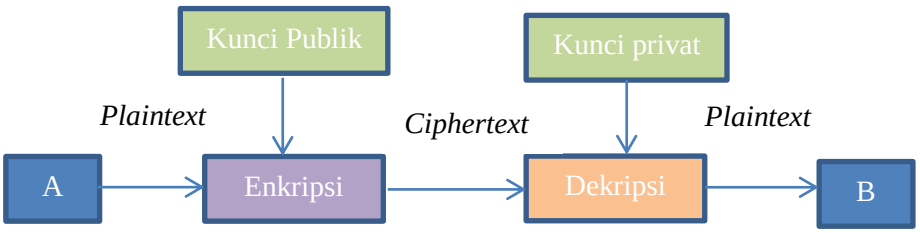
Kekurangan kriptografi simetris adalah (Kamil, 2016):

a. Kunci simetris harus dikirim melalui saluran komunikasi yang aman, dan kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci.

b. Kunci harus sering diubah, setiap kali melaksanakan komunikasi. Apabila kunci tersebut hilang atau lupa, maka pesan tersebut tidak dapat dibuka.

2. Algoritma Asimetris

Berbeda dengan kriptografi kunci simetris, kriptografi kunci publik memiliki dua buah kunci yang berbeda pada proses enkripsi dan dekripsinya. Kunci yang digunakan untuk proses enkripsi atau sering disebut publik *Key* dan dekripsi atau sering disebut *Private key* menggunakan kunci yang berbeda. Entitas pengirim akan mengenkripsi dengan menggunakan kunci publik, sedangkan entitas penerima mendekripsi menggunakan kunci Privat (Kamil, 2016). Skema dari kriptografi dapat dilihat pada Gambar 2.2.




**Gambar 2. 2** Algoritma Asimetris

Kelebihan kriptografi asimetris adalah (Kamil, 2016):

a. Hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi. Tidak ada kebutuhan mengirim kunci private sebagaimana kunci simetri.
b. Pasangan kunci privat dan kunci publik tidak perlu diubah dalam jangka waktu yang sangat lama.
c. Dapat digunakan dalam pengaman pengiriman kunci simetris.

Kelemahan kriptografi asimetris adalah (Kamil, 2016):

a. Proses enkripsi dan dekripsi umumnya lebih lambat dari algortima simetri, karena menggunakan bilangan yang besar dan operasi bilangan yang besar.
b. Ukuran *ciphertext* lebih besar dari *plaintext*.
c. Ukuran kunci relative lebih besar daripada ukuran kunci simetris.

## 2.4 *Vigenere Cipher*

*Vigenere cipher* adalah suatu algoritma kriptografi klasik dengan teknik substitusi yang ditemukan oleh Giovan Battista Bellaso. Beliau menuliskan metodenya tersebut pada bukunya yang berjudul *La Cifra del. Sig. Giovan Battista Bellaso* pada tahun 1553. Nama *Vigenere* sendiri diambil dari seorang yang bernama Blaise de *Vigenere*. *Vigenere cipher* menggunakan suatu kunci yang memiliki panjang kunci tertentu (McAndrew, 2011).

*Vigenere cipher* ini adalah suatu metode yang dirancang untuk memperbaiki kelemahan dari algoritma subtitusi tunggal. *Vigenere cipher*

merupakan teknik kriptografi sederhana yang lebih aman. Dikembangkan dari metode *caesar cipher*, metode ini menggunakan karakter huruf sebagai kunci enkripsi. *Vigenere cipher* juga merupakan polyalphabetic substition *cipher* (McAndrew, 2011).

Teknik *vigenere cipher* bisa dilakukan dengan dua cara yaitu dengan angka dan dengan huruf (Ariyus, 2006)

### 2.4.1 *Vigenere cipher* dengan angka

Teknik ini dilakukan dengan cara mensubtitusikan huruf dengan angka, sehingga teknik ini hampir mirip dengan *cipher* pergeseran (Katz & Lindell, 2015)

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Secara matematis dapat dituliskan sebagai persamaan

$$Ei = (Pi + Ki)\ mod\ 26 \quad (1)$$

Keterangan : $Ei$ = Enkripsi karakter ke-$i$

$Pi$ = Karakter ke-$i$ pada pesan

$Ki$ = karakter ke-$i$ pada kunci

Sedangkan dekripsi *vigenere cipher* dapat diketahui dengan menggunakan persamaan (2):

$$Di = (Ci - Ki) \ mod \ 26 \quad (2)$$

Keterangan : $Di$ = Dekripsi karakter ke-$i$

$Ki$ = karakter ke-$i$ pada kunci

$Ci$ = karakter ke-$i$ pada *ciphertext*

### 2.4.2 *Vigenere cipher* dengan Huruf

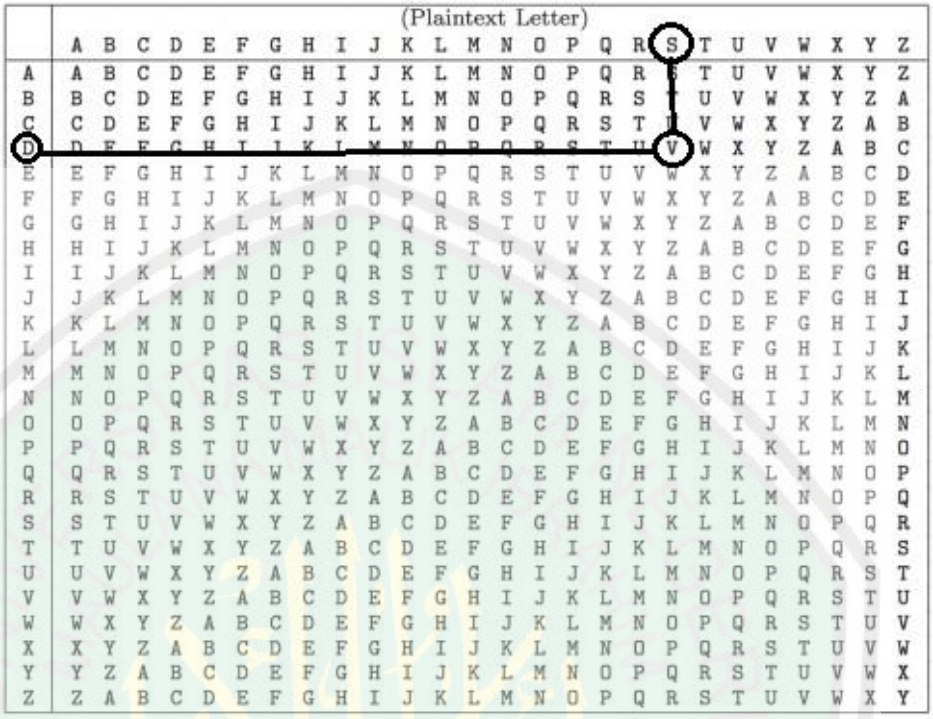
Teknik subtitusi *vigenere* dengan huruf dilakukan menggunakan tabula recta seperti dibawah ini (Katz & Lindell, 2015)

| | (Plaintext Letter) | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

**Gambar 2. 3** Tabula Recta

Pada tabula recta, barisan horizontal merupakan deretan karakter untuk *plaintext* sedangkan posisi vertikal merupakan kunci. Dan untuk mendapatkan *ciphertext* dilakukan dengan menarik garis lurus antara

karakter *plaintext* dan kunci contoh: huruf S disandikan dengan huruf D maka didapatkan hasil seperti berikut




**Gambar 2. 4** Penggunaan Tabula Recta

Sehingga didapat hasil *ciphertext* dari penyandian *plaintext* huruf S menggunakan kunci huruf D adalah huruf V.

## 2.5 *Railfence cipher*

*Railfence cipher* atau yang dikenal dengan *zigzag cipher* merupakan salah satu teknik enkripsi dengan mengubah posisi karakter menjadi bentuk diagonal ke bawah dan ke atas. *Cipher* ini menggunakan perubahan posisi dan susunan karakter *cipher* ini tidak memiliki kunci tertentu. Sehingga untuk memecahkannya kita harus memperhatikan tingkatan dari tulisan tersebut, dikarenakan *cipher* ini biasanya sistematis (Ramkesh, 2016).

*Railfence cipher* membentuk sebuah lintasan. Lintasan ini berbentuk Zig Zag. Itulah sebabnya metode ini dapat juga disebut Kriptografi Zig-Zag.

*Railfence cipher* adalah contoh sederhana dari *cipher* transposisi. Konsep kunci dari *cipher* ini hanyalah sebuah nilai yang sudah disepakati pengirim dan penerima untuk menentukan jumlah baris sedangkan jumlah kolom ditentukan berdasarkan jumlah karakter yang di enkripsi

Contoh:

*Plaintext* : SIGIT DENI SANTOSO

*Railfence cipher* kunci 2

- Proses enkripsi, buatlah *plaintext* menggunakan pola zig-zag dalam dua baris, kemudian *ciphertext* nya dibaca dari baris pertama dan diikuti oleh baris kedua.

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | | G | | T | | D | | N | | | | A | | T | | S | |
| | I | | I | | | | E | | I | | S | | N | | O | | O |

*Ciphertext* : SGTDN ATSII EISNOO

- Proses dekripsi, *ciphertext* di inputkan kedalam bentuk matrik dengan baris sejumlah nilai kunci dan kolom sebanyak jumlah karakter kemudian dibaca secara zigzag

  Contoh

  *Ciphertext* : SGTDN ATSII EISNOO

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S | | G | | T | | D | | N | | | | A | | T | | S | |
| | I | | I | | | | E | | I | | S | | N | | O | | O |

*Plaintext* : SIGIT DENI SANTOSO

Enkripsi pesan dengan teknik *railfence cipher* sangat mudah dilakukan dan bisa mengenkripsi dengan cepat namun juga rentan terjadinya penyadapan pesan oleh pihak ketiga karena algoritma ini secara umum relatif mudah untuk dianalisis (Ramkesh, 2016).

## 2.6 Super Enkripsi

Super enkipsi (*multiple encryption*) merupakan salah satu teknik kriptografi yang meggabungkan dua atau lebih teknik subtitusi dan *cipher* permutasi untuk mendapatkan *cipher* yang lebih kuat dan susah untuk dipecahkan (Ariyus, 2006).

Teknik ini tidak terlalu susah untuk dilakukan jika sudah memahami konsep *cipher* subtitusi dan permutasi. Enkripsi dan dekripsi bisa dilakukan dengan urutan *cipher* subtitusi kemudian permutasi, ataupun sebaliknya. Dengan prosesnya yaitu yang pertama dengan mengenkripsi *plaintext* menjadi *ciphertext* kemudian *ciphertext* dienkripsi kembali menggunakan *cipher* dan kunci yang lain.

## 2.7 Python

Python adalah satu dari bahasa pemrograman tingkat tinggi yang mudah dipelajari dikarenakan menggunakan *sintax* yang jelas yang dapat dikombinasikan dengan penggunaan modul-modul yang mempunyai struktur data tingkat tinggi, efisien, dan siap langsung digunakan. Python bersifat *interpreter*, interaktif, *object-oriented* dan dapat beroperasi di hampir semua *platform*, seperti keluarga Linux, Windows, Mac, dan *platform* lainnya. yang dikombinasikan dengan penggunaan modul-modul yang mempunyai struktur data tingkat tinggi, efisien, dan siap langsung digunakan (Rosmala & Dwipa, 2012). Kelemahan dari Python terletak pada kecepatan eksekusi yang tidak secepat bahasa pemrograman yang dikompilasi seperti C dan C++.

Kelebihan dari Python menurut (Lutz, 2013) antara lain:

1. Kualitas software Bahasa pemrograman Python dirancang agar mudah dibaca, sehingga mendukung penggunaan kembali *source code (code reusability*) dan jika perlu dilakukan perubahan, programmer juga dimudahkan untuk mengatur kembali source code tersebut.
2. Produktivitas pengembang produktivitas *developer* pengguna bahasa pemrograman Python dapat lebih baik dibandingkan pengguna bahasa pemrograman lain seperti C, C++, dan Java. *Source code* Python biasanya juga memiliki ukuran file lima kali lebih kecil dari besar file *source code* bahasa pemrograman C++ atau Java. Hal ini berarti mengurangi besarnya *source code* yang harus ditulis oleh developer sehingga proses debugging aplikasi akan lebih mudah.
3. Portabilitas program Sebagian besar program yang dikembangkan dengan bahasa pemrograman Python berjalan tanpa adanya perubahan pada perangkat yang berbeda-beda. Jika programmer ingin menjalankan program Python pada perangkat yang menjalankan Linux dan Windows, programmer dapat dengan mudah menjalankan program tersebut tanpa modifikasi.
4. Dukungan *library* Bahasa pemrograman Python memiliki Python standard *library*, yaitu kumpulan fungsionalitas yang bersifat portabel. *Library* ini sangat mendukung berbagai fungsionalitas dasar sampai kompleks yang portabel. Selain itu, *library* Python juga dapat menggunakan *library* yang dikembangkan oleh pihak ketiga untuk memperluas lagi cakupannya.

5. Integrasi komponen Bahasa pemrograman Python memiliki kemampuan untuk dapat berintegrasi dengan bagian lain dari sebuah aplikasi. Integrasi ini membuat Python memiliki kapabilitas untuk dapat dipakai sebagai alat ekstensi, contohnya bahasa pemrograman Python dapat memanggil *library* C dan C++, dan sebaliknya.

# BAB III

# PEMBAHASAN

## 3.1 Proses Penyandian Super Enkripsi *Vigenere Cipher* dan *Railfence Cipher*

### 3.1.1 Teknik Penyandian *Vigenere cipher*

Bentuk umum penyandian *vigenere cipher* adalah menggunakan 26 huruf alfabet dari A sampai Z

- Enkripsi:

  *Plaintext*: COBAVIGENERECIPHER

  Kunci: TES

  *Plaintext* dan kunci disubtitusikan kedalam bentuk angka berdasarkan urutan alfabet

| C | O | B | A | V | I | G | E | N | E | R | E | C | I | P | H | E | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 14 | 1 | 0 | 21 | 8 | 6 | 4 | 13 | 4 | 17 | 4 | 2 | 8 | 15 | 7 | 4 | 17 |
| T | E | S | T | E | S | T | E | S | T | E | S | T | E | S | T | E | S |
| 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 |

  kedua variabel ini dioperasikan menggunakan persamaan 1 dan Hasil operasi persamaan 1 dikembalikan ke bentuk huruf

| 21 | 18 | 19 | 19 | 25 | 26 | 25 | 8 | 31 | 23 | 21 | 22 | 21 | 12 | 33 | 26 | 8 | 35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| V | S | T | T | Z | A | Z | I | F | X | V | W | V | M | H | A | I | J |

  Sehingga didapatkan *ciphertext* VSTTZAZIFXVWVMHAIJ

- Dekripsi

  *Ciphertext*: VSTTZAZIFXVWVMHAIJ

  Kunci: TES

  *Ciphertext* dan kunci disubtitusikan kedalam bentuk angka

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| V | S | T | T | Z | A | Z | I | F | X | V | W | V | M | H | A | I | J |
| 21 | 18 | 19 | 19 | 25 | 26 | 25 | 8 | 31 | 23 | 21 | 22 | 21 | 12 | 33 | 26 | 8 | 35 |
| T | E | S | T | E | S | T | E | S | T | E | S | T | E | S | T | E | S |
| 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 | 19 | 4 | 18 |

kedua variabel ini dioperasikan menggunakan persamaan 2 dan Hasil operasi persamaan 2 dikembalikan ke bentuk huruf

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 14 | 1 | 0 | 21 | 8 | 6 | 4 | 13 | 4 | 17 | 4 | 2 | 8 | 15 | 7 | 4 | 17 |
| C | O | B | A | V | I | G | E | N | E | R | E | C | I | P | H | E | R |

Sehingga akan didapatkan *plaintext* COBAVIGENERECIPHER

Untuk meningkatkan keamanan teknik *vigenere cipher* dapat dilakukan dengan memodifikasi teknik penyandiannya menggunakan subtitusi berdasarkan tabel ASCII yang memiliki variasi karakter lebih banyak. Berdasarkan tabel kode ASCII, karakter yang bisa digunakan dimulai dari no 32 sampai 127 yang mana sebanyak 95 karakter, sehingga persamaan untuk enkripsi pesan menggunakan *vigenere cipher* akan dimodifikasi menjadi:

$$C = ((P + K) - 64) mod\ 95) + 32 \quad (3)$$

Keterangan: $C$ = *Ciphertext*
$P$ = *Plaintext*
$K$ = *Key*

Dan persamaan dekripsi pesan dari *vigenere cipher* modifikasi adalah::

$$P = ((C - K) mod\ 95) + 32 \quad (4)$$

Keterangan: $P$ = *Plaintext*
$C$ = *Ciphertext*
$K$ = *Key*

Pembuktian persamaan (3) dan (4) dilakukan dengan sebuah contoh Misalkan nilai P = 32, K = 34 kemudian disubstitusikan kedalam persamaan (3) sehingga:

$$\begin{aligned} C &= \big((32 + 34) - 64\ mod\ 95\big) + 32 \\ C &= (66 - 64) mod\ 95) + 32 \\ C &= (2\ mod\ 95) + 32 \\ C &= 34 \end{aligned}$$

Sehingga didapatkan nilai C = 34. Kemudian ketiga nilai disubtitusikan ke persamaan (4) sehingga:

$$\begin{aligned} 32 &= \big((34 - 34)\ mod\ 95\big) + 32 \\ 32 &= 0\ mod\ 95 + 32 \\ 32 &= 0 + 32 \\ 32 &= 32 \end{aligned}$$

Nilai C sama dengan nilai perhitungan menggunakan persamaan (4). Sehingga persamaan (3) dan (4) terbukti dapat digunakan untuk enkripsi dan dekripsi vigenere cipher berdasarkan nilai ASCII.

- Enkripsi:

  *Plaintext*: COBAVIGENERECIPHER

  Kunci: TES

  - *Plaintext* dan kunci disubtitusikan kedalam bentuk angka berdasarkan tabel ASCII

| C | O | B | A | V | I | G | E | N | E | R | E | C | I | P | H | E | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 67 | 79 | 66 | 65 | 86 | 73 | 71 | 69 | 78 | 69 | 82 | 69 | 67 | 73 | 80 | 72 | 69 | 82 |
| T | E | S | T | E | S | T | E | S | T | E | S | T | E | S | T | E | S |
| 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 |

  - kedua variabel ini dioperasikan menggunakan persamaan 3 dan Hasil operasi persamaan 3 dikembalikan ke bentuk huruf

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 119 | 116 | 117 | 117 | 123 | 124 | 123 | 106 | 34 | 121 | 119 | 120 | 121 | 110 | 36 | 124 | 106 | 38 |
| w | t | u | u | { | \| | { | j | “ | y | w | x | w | n | $ | \| | j | & |

- *Ciphertext* yang didapatkan adalah wtuu{|{j"ywxwn$|j&

- Dekripsi

  *Ciphertext*: wtuu{|{j"ywxwn$|j&

  Kunci : TES

  - *Plaintext* dan kunci disubtitusikan kedalam bentuk angka berdasarkan tabel ASCII

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| w | t | u | u | { | \| | { | j | “ | y | w | x | w | n | $ | \| | j | & |
| 119 | 116 | 117 | 117 | 123 | 124 | 123 | 106 | 34 | 121 | 119 | 120 | 121 | 110 | 36 | 124 | 106 | 38 |
| T | E | S | T | E | S | T | E | S | T | E | S | T | E | S | T | E | S |
| 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 | 84 | 69 | 83 |

  - kedua variabel ini dioperasikan menggunakan persamaan (4) dan Hasil operasi persamaan 3 dikembalikan ke bentuk huruf

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 67 | 79 | 66 | 65 | 86 | 73 | 71 | 69 | 78 | 69 | 82 | 69 | 67 | 73 | 80 | 72 | 69 | 82 |
| C | O | B | A | V | I | G | E | N | E | R | E | C | I | P | H | E | R |

  - *Plaintext* yang didapatkan adalah COBAVIGENERECIPHER

### 3.1.2 Teknik Penyandian *Railfence cipher*

- Enkripsi

  *Plaintext*: COBA RAILFENCE CIPHER

  Kunci: 3

  - Buat matriks dengan jumlah baris sebanyak nilai kunci dan jumlah kolom sebanyak panjang pesan

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |

- Input *plaintext* secara zigzag

| C | | | | | | | | L | | | | C | | | | I | | | | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | O | | A | | R | | I | | F | | N | | E | | C | | P | | E | |
| | | B | | | | A | | | | E | | | | | | | | H | | |

- *Ciphertext* didapatkan dengan mengambil karakter dari urutan baris ke-1, baris ke-2, dan baris ke-3. diperoleh C LCIROARIFNECPEBAE H

- Dekripsi

  *Ciphertext*: C LCIROARIFNECPEBAE H

  Kunci: 3

  - Buat matriks dengan jumlah baris sebanyak nilai kunci dan jumlah kolom sebanyak panjang pesan

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | |

  - Buat suatu tanda yang mana tanda ini menjadi patokan dalam penginputan pesan. tanda ini dibuat bergerak secara zig zag ke arah kanan

| X | | | | X | | | | X | | | | X | | | | X | | | | X |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | X | | X | | X | | X | | X | | X | | X | | X | | X | | X | |
| | | X | | | | X | | | | X | | | | X | | | | X | | |

- Input *ciphertext* kedalam matrik yang dibuat dengan cara pengitputan berdasarkan urutan baris ke-1, baris ke-2, dan baris ke-3 sesuai dengan tanda yang sudah disediakan.

| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C | | | | | | | | L | | | | C | | | | I | | | | R |
| | O | | A | | R | | I | | F | | N | | E | | C | | P | | E | |
| | | B | | | | A | | | | E | | | | | | | | H | | |

- *Plaintext* didapatkan dengan mengambil karakter dengan urutan baris ke-1, baris ke-2 hingga baris ke-n dengan n adalah nilai dari panjang pesan. Diperoleh COBA RAILFENCE CIPHER.

### 3.1.3 Teknik Penyandian Super Enkripsi *Vigenere* dan *Railfence cipher*

Teknik penyandian pesan dimulai dengan proses enkripsi menggunakan *Vigenere cipher*, kemudian pesan hasil enkripsi *vigenere cipher* dienkripsi lagi menggunakan *Railfence cipher* sehingga akan terbentuk keamanan dua lapis, untuk mengembalikan pesan agar terbaca kembali maka dilakukan dekripsi menggunakan *Railfence cipher* kemudian pesan didekripsi menggunakan *vigenere cipher*. Proses enkripsi dan dekripsi pesan dilakukan menggunakan kunci yang sama

Contoh :

Sigit akan mengirimkan pesan "**Sigit Mahasiswa UIN**" kepada Deni.Namun karena sigit ingin agar pesan nya tidak diketahui oleh pihak lain, maka Sigit akan menggunakan super enkripsi untuk menyandikan pesan. Teknikyang digunakan adalah *Vigenere cipher* dengan kunci enkripsi "@$1tEs):" dan *Railfence cipher* dengan kunci 5.

### 3.1.3.1 Proses Enkripsi Pesan

- ***Vigenere cipher***

  ➢ Langkah pertama adalah mensubtitusikan teks pesan dan kunci kedalam bentuk ASCII

  Berikut adalah hasil subtitusi kedalam bentuk karakter ASCII

  - Teks Pesan

| S | i | g | i | t | | M | a | h |
|---|---|---|---|---|---|---|---|---|
| 83 | 105 | 103 | 105 | 116 | 32 | 77 | 97 | 104 |

| A | s | i | s | w | a | | U | I | N |
|---|---|---|---|---|---|---|---|---|---|
| 97 | 115 | 105 | 115 | 119 | 97 | 32 | 85 | 73 | 78 |

  - Kunci

| @ | $ | 1 | t | E | s | ) | : |
|---|---|---|---|---|---|---|---|
| 64 | 36 | 49 | 116 | 69 | 115 | 41 | 58 |

  ➢ Proses penyandian pesan

  Dengan menggunakan perhitungan di dapatkan:

| *Plaintext* | *Key* | ((*plaintext*+*key*)-64 mod 95)+32 | *Ciphertext* |
|---|---|---|---|
| 83 | 64 | 115 | s |
| 105 | 36 | 109 | m |
| 103 | 49 | 120 | x |
| 105 | 116 | 94 | ^ |
| 116 | 69 | 58 | : |
| 32 | 115 | 115 | s |
| 77 | 41 | 86 | V |
| 97 | 58 | 123 | { |
| 104 | 64 | 41 | ) |

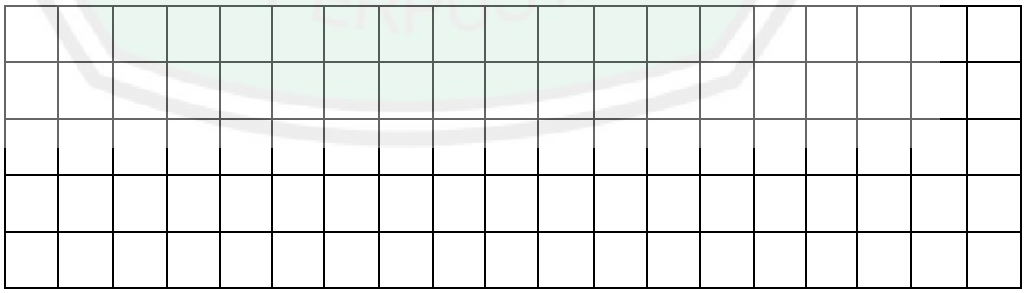
| | | | |
|---|---|---|---|
| 97 | 36 | 101 | e |
| 115 | 49 | 37 | % |
| 105 | 116 | 94 | ^ |
| 115 | 69 | 57 | 9 |
| 119 | 115 | 75 | k |
| 97 | 41 | 106 | j |
| 32 | 58 | 58 | : |
| 85 | 64 | 117 | u |
| 73 | 36 | 77 | M |
| 78 | 49 | 95 | _ |

Sehingga *Ciphertext* yang didapat

**smx^:sV{)e%^9kj:uM_**

*ciphertext* tersebut selanjutnya akan di enkripsi kembali menggunakan *railfence cipher*

- ***Railfence cipher***
  - Langkah pertama yaitu membuat matriks dengan jumlah baris sebanyak nilai dari kunci yang dalam kasus ini bernilai 5 dan jumlah kolom sebanyak jumlah dari karakter pesan yang disandikan yang dalam kasus ini sebanyak 19 karakter



  - Pesan **smx^:sV{)e%^9kj:uM_** diinputkan kedalam matriks 5x19dengan format penginputan zigzag ke kanan.

| S | | | | | | | | ) | | | | | | | | u | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | m | | | | | | { | | e | | | | | | : | | M | |
| | | x | | | | V | | | | % | | | | j | | | | _ |
| | | | ^ | | s | | | | | | ^ | | k | | | | | |
| | | | | : | | | | | | | | 9 | | | | | | |

- Kemudian mengelompokkan pesan menjadi seperti berikut

| | | | | | |
|---|---|---|---|---|---|
| Kolom 1 | s | ) | u | | |
| Kolom 2 | m | { | e | : | M |
| Kolom 3 | x | V | % | j | _ |
| Kolom 4 | ^ | s | ^ | k | |
| Kolom 5 | : | 9 | | | |

Untuk mendapatkan hasil enkripsi dari *railfence cipher* dengan cara membaca karakternya secara kolom sehingga Sehingga akan didapatkan hasil enkripsi dari *Railfence cipher* yaitu **s)um{e:MxV%j_^s^k:9**

### 3.1.3.2 Proses Dekripsi Pesan

Setelah deni menerima pesan dari sigit berupa *ciphertext* **s)um{e:MxV%j_^s^k:9**,maka diperlukan teknik dekripsi pesan agar *ciphertext* kembali menjadi *plaintext* yang bisa dibaca dan dipahami. pada proses dekripsi dilakukan teknik yang berlawanan dengan proses enkripsi disebabkan teknik super enkripsi tidak bersifat komutatif karena perbedaan urutan teknik yang dipakai akan berpengaruh pada hasil penyandian,sebagai contoh apabila pesan disandikan menggunakan *vigenere cipher* dahulu baru disandikan menggunakan *Railfence cipher* hasil penyandianya akan berbeda dengan teknik penyandian *Railfence cipher* kemudian disandikan dengan

*vigenere cipher*. Maka, dilakukan teknik dekripsi *Railfence cipher* dahulu kemudian dilanjutkan dengan teknik dekripsi *vigenere cipher* untuk mendapatkan pesan yng sesuai dengan *plaintext*.

- ***Railfence cipher***
  - Seperti pada proses enkripsi pesan *railfence cipher*, pertama adalah membuat sebuah matriks dengan ukuran nilai kunci enkripsi × jumlah karakter pesan. kemudian dimulai dari kolom 1 dan baris 1 di beri suatu tanda yang mana tanda ini menjadi patokan dalam penginputan pesan. tanda ini dibuat bergerak secara zig zag ke arah kanan

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | | | | | | | | X | | | | | | | | X | | |
| | X | | | | | | X | | X | | | | | | X | | X | |
| | | X | | | | X | | | | X | | | | X | | | | X |
| | | | X | | X | | | | | | X | | X | | | | | |
| | | | | X | | | | | | | | X | | | | | | |

  - Berbeda dengan cara pengitputan pada proses Enkripsi *Railfence cipher*, proses dekripsi *Railfence cipher* penginputan teks dilakukan secara horizontal sesuai dengan letak tanda yang sudah dibuat, Sehingga untuk *ciphertext* **s)um{e:MxV%j_^s^k:9** akan menjadi seperti berikut ini

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s | | | | | | | | ) | | | | | | | | u | | |
| | m | | | | | | { | | e | | | | | | : | | M | |
| | | x | | | | V | | | | % | | | | j | | | | _ |
| | | | ^ | | s | | | | | | ^ | | k | | | | | |
| | | | | : | | | | | | | | 9 | | | | | | |

- Kemudian mengelompokkan pesan menjadi seperti berikut

| | | | | | |
|---|---|---|---|---|---|
| Kolom 1 | s | ) | u | | |
| Kolom 2 | m | { | e | : | M |
| Kolom 3 | x | V | % | j | _ |
| Kolom 4 | ^ | s | ^ | k | |
| Kolom 5 | : | 9 | | | |

Untuk mendapatkan hasil enkripsi dari *railfence cipher* dengan cara membaca karakternya secara baris ke baris. Sehingga didapatkan hasil dekripsi *Railfence cipher* **smx^:sV{)e%^9kj:uM_** kemudian teks hasil dekripsi akan di dekripsikan lagi menggunakan teknik *vigenere cipher*.

- ***Vigenere cipher***
  - Teks pesan dan kunci disubtitusikan kedalam bentuk nilai karakter berdasarkan tabel ASCII
    - Teks pesan

| s | m | x | ^ | : | s | V | { | ( |
|---|---|---|---|---|---|---|---|---|
| 115 | 109 | 120 | 94 | 58 | 115 | 86 | 123 | 41 |

| e | % | ^ | 9 | k | j | : | u | M | _ |
|---|---|---|---|---|---|---|---|---|---|
| 101 | 37 | 94 | 57 | 75 | 106 | 58 | 117 | 77 | 95 |

    - Kunci

| @ | $ | 1 | t | E | s | ) | : |
|---|---|---|---|---|---|---|---|
| 64 | 36 | 49 | 116 | 69 | 115 | 41 | 58 |

  - Proses dekripsi *ciphertext* menggunakan kunci dilakukan dengan persamaan (4). Sehingga diperoleh:

| *Ciphertext* | *Key* | ((*Ciphertext*-*key*)mod95)+32 | *Plaintext* |
|---|---|---|---|
| 115 | 64 | 83 | S |
| 109 | 36 | 105 | i |
| 120 | 49 | 103 | g |
| 94 | 116 | 105 | i |
| 58 | 69 | 116 | t |
| 115 | 115 | 32 | |
| 86 | 41 | 77 | M |
| 123 | 58 | 97 | a |
| 41 | 64 | 104 | h |
| 101 | 36 | 97 | a |
| 37 | 49 | 115 | s |
| 94 | 116 | 105 | i |
| 57 | 69 | 115 | s |
| 107 | 115 | 119 | w |
| 106 | 41 | 97 | a |
| 58 | 58 | 32 | |
| 117 | 64 | 85 | U |
| 77 | 36 | 73 | I |
| 95 | 49 | 78 | N |

Setelah pesan terdekripsi maka deni akan dapat membaca isi pesan asli yang dikirimkan sigit yaitu **Sigit Mahasiswa UIN**.

## 3.2 Analisa Keamanan Super Enkripsi *Vigenere Cipher* dan *Railfence Cipher*

### 3.2.1 Analisa Keamanan *Vigenere cipher*

Keamanan dari teknik *vigenere cipher* sangat bergantung pada panjang pesan dan besarnya semesta jumlah karakter yang bisa digunakan untuk *plaintext* dan kunci nya dan panjang *plaintext* serta kunci yang digunakan.

Salah satu teknik yang umum digunakan untuk memecahkan pesan terenkripsi adalah algoritma *brute-force*. Teknik ini merupakan metode pemecahan sandi yang paling dikenal, teknik ini menggunakan setiap kombinasi karakter yang memungkinkan sebagai kata sandi. Teknik ini hampir memungkinkan menyerang kunci privat pada hamper semua skema kriptografi, tipe serangan ini bergantung pada ukuran kunci dan mekanisme pada enkripsi yang digunakan (LastBit, 2005). Sehingga semakin panjang sandi akan semakin lama proses pemecahan sandi menggunakan *brute-force*.

Teknik penyandian *vigenere cipher* umumnya sangat susah dibobol menggunakan teknik brute-force. teknik *vigenere cipher* ini memanfaatkan sejumlah 26 karakter huruf untuk teks pesan dan kunci nya, dikarenakan teknik *vigenere cipher* sangat bergantung dari panjang kata kunci yang digunakan. Maka kemungkinan sebuah pesan yang sudah tersandikan menggunakan *vigenere cipher* bisa dipecahkan adalah sebanyak $26^k$ kemungkinan, dimana *k* adalah nilai dari panjang kunci. Misalkan sebuah pesan memiliki kunci 7 karakter maka kemungkinan pesan terpecahkan adalah $26^7 = 8$ milyar kemungkinan (Sweigart, 2013). Namun, karakter

yang digunakan untuk *vigenere cipher* yang sudah dimodifikasi berdasarkan tabel ASCII yang memiliki jumlah karakter sebanyak 95 karakter. Menyebabkan kemungkinan terbobolnya akan menjadi $95^k$ kemungkinan dengan k adalah panjangnya kunci. Sehingga diperlukan waktu yang lama untuk membobol sebuah pesan yang sudah tersandikan dengan teknik ini.

### 3.2.2 Analisa Keamanan *Railfence cipher*

Teknik penyandian *railfence cipher* adalah teknik penyandian sederhana yang merupakan penyandian dengan teknik transposisi yang merubah posisi dari tiap karakter huruf berdasarkan nilai kunci. Brute-force menjadi sangat efektif untuk memecahkan pesan yang tersandikan menggunakan teknik *railfence cipher* yaitu dengan mencoba semua kemungkinan kunci dimana kemungkinan kunci dari teknik *railfence cipher* sangatlah terbatas, yaitu sejumlah bilangan bulat yang kurang dari jumlah nilai panjang *plaintext* yang ada. Sehingga teknik ini sangat rentan untuk dipecahkan. Misalkan pesan disandikan menggunakan Railfece *cipher* dengan kunci 3, maka hanya membutuhkan 3 kali percobaan agar *plaintext* bisa didapatkan.

### 3.2.3 Analisa Keamanan Penyandian *Vigenere cipher* dan *Railfence cipher*

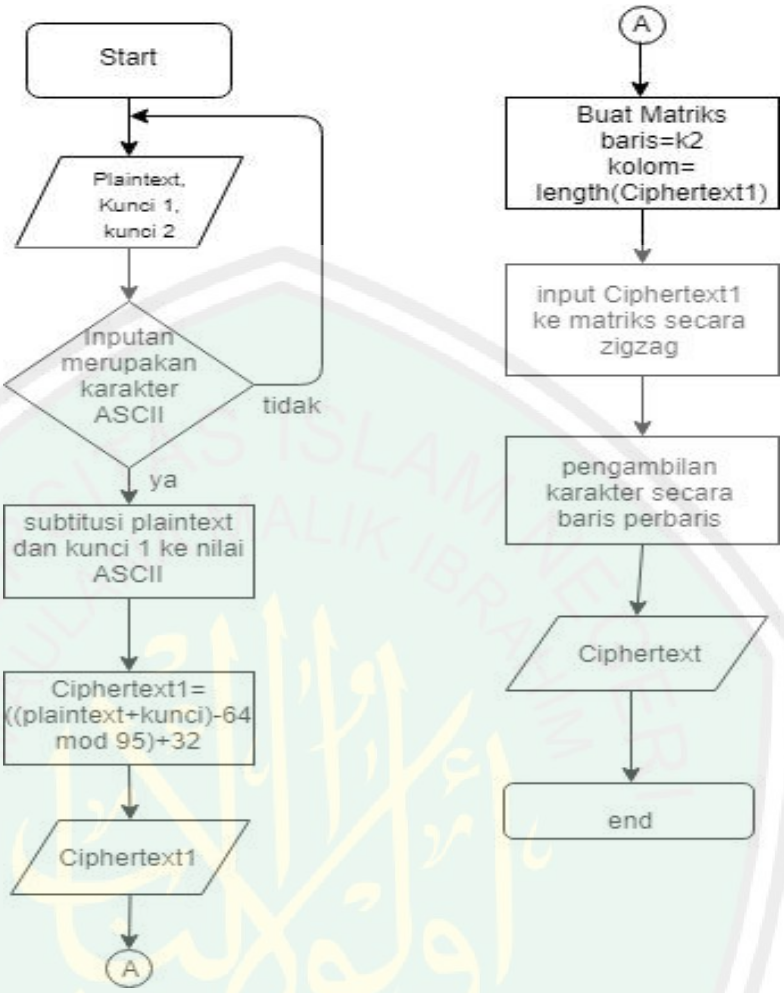
Penggabungan teknik penyandian *vigenere cipher* dengan *railfence cipher* akan melipatgandakan tingkat keamanan pesan dari teknik pembobolan pesan brute-force yang menggunakan konsep mencoba semua kemungkinan yang ada sehingga peluang terbobolnya teknik gabungan ini adalah kombinasi dari peluang terbobolnya *vigenere cipher* menggunakan ASCII sebanyak $95^{K1}$ dan peluang terbobolnya *railfence cipher* sebanyak $k2$, dimana k1 adalah panjang kunci *vigenere cipher* dan k2 adalah nilai kunci dari *railfence cipher* maka tingkat keamananya menjadi sebanyak $95^{K1} \times k2$,. Sehingga terlihat jelas jika teknik penyandian menggunakan teknik super enkripsi akan meningkatkan keamanan dari pesan yang disandikan.

## 3.3 Implementasi Super Enkripsi Dengan Python

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

Berikut ini adalah rancangan *flowchart* penyandian super enkripsi *vigenere cipher* dan *railfence cipher* sesuai dengan langkah langkah yang telah dibahas pada pembahasan sebelumnya.

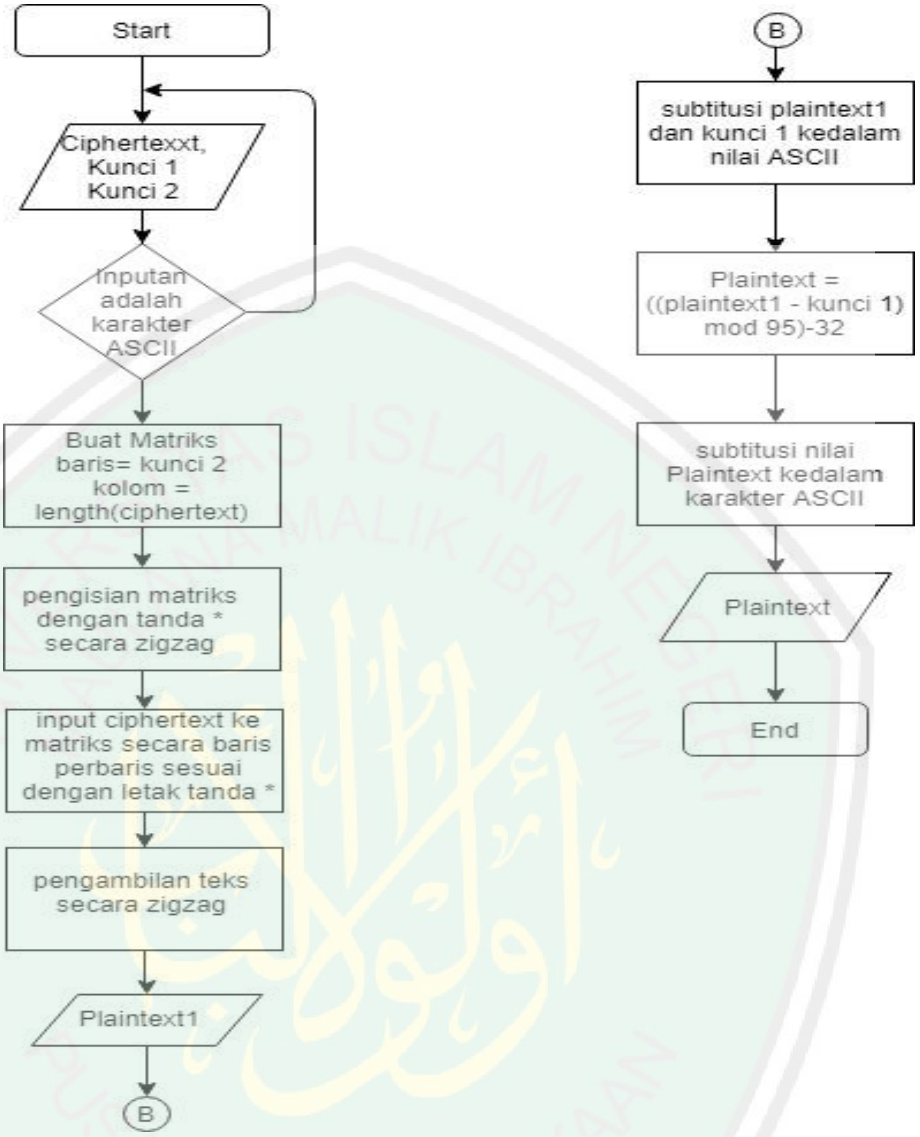
- Flowchart enkripsi




**Gambar 3. 1** Flowchart Enkripsi

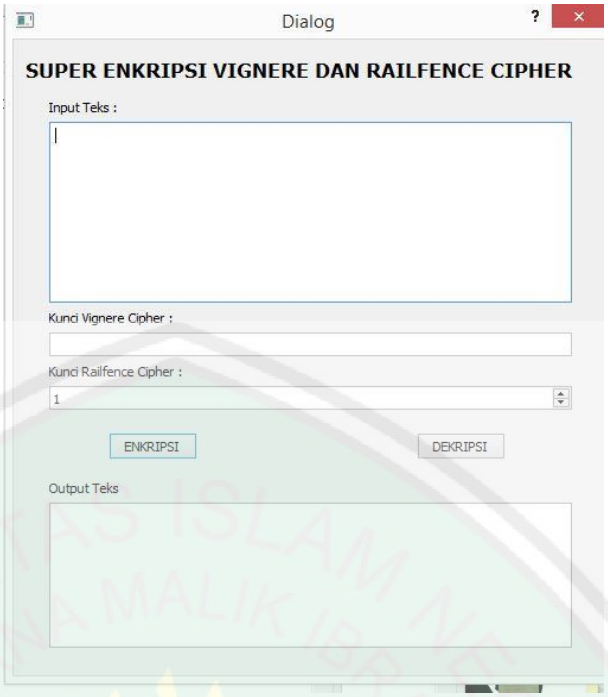
Kemudian dibuat *flowchart* dekripsi pesan sehingga memungkinkan pesan yang disandikan dapat kembali menjadi pesan semula yang sama dengan keadaan ketika belum tersandikan

- Flowchart dekripsi



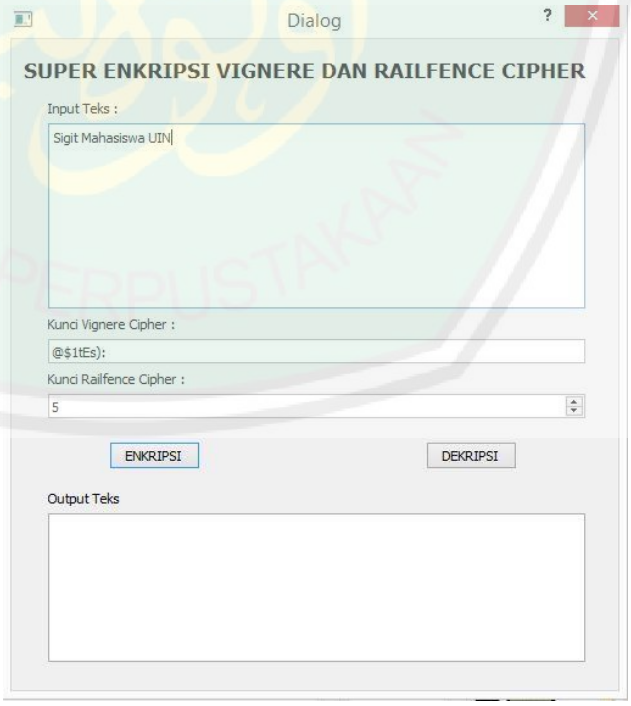

**Gambar 3. 2** Flowchart Dekripsi

Langkah berikutnya adalah mengimpelementasikan rangkaian *flowchart* tersebut kedalam bentuk skrip bahasa Python. Untuk memudahkan penggunaan aplikasi maka perlu dibuat suatu GUI (*Graphical User Interface*) di Python seperti pada gambar berikut




**Gambar 3. 3** Tampilan GUI

Untuk menyandikan suatu pesan, pengguna aplikasi hanya perlu menginputkan pesan, kunci *vigenere cipher* dan kunci *railfence cipher*




**Gambar 3. 4** Input *Plaintext* dan Kunci

User mengklik tombol enkripsi maka hasil penyandianya akan ditampilkan pada kolom *output*


**Gambar 3. 5** *Output* Teks

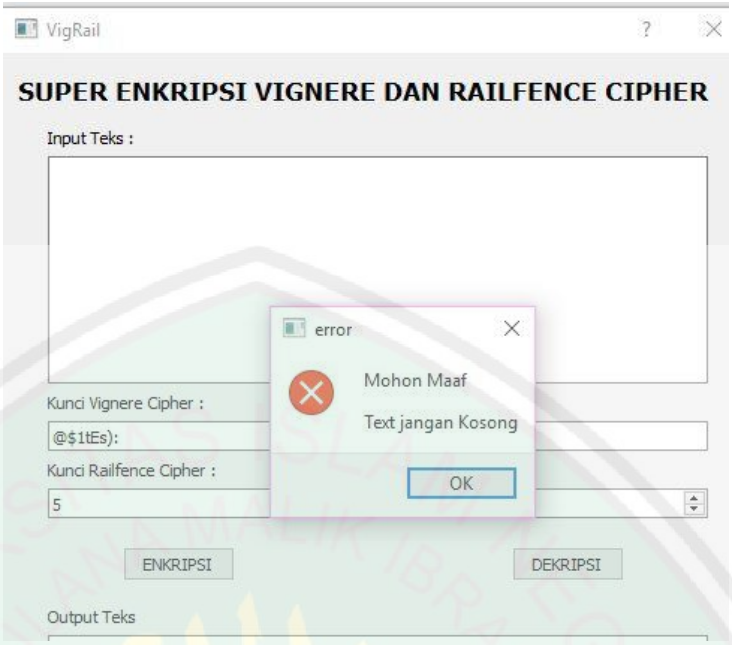
*Output* teks pada aplikasi menunjukkan hasil yang sama dengan hasil penyandian dengan cara perhitungan manual. Sedangkan untuk dekripsi pesan *ciphertext* dimasukkan ke kolom input, kemudian kunci di masukkan sesuai dengan kunci yang sama dengan enkripsi kemudian dengan menekan tombol dekripsi maka kolom *output* akan menunjukkan hasil dekripsi seperti berikut ini


**Gambar 3. 6** Hasil Dekripsi

Proses dekripsi menggunakan aplikasi berhasil mengembalikan teks sesuai dengan *plaintext*. Aplikasi ini juga didesain agar menampilkan pesan *Error* jika mengalami kondisi berikut ini:

- Kolom input dikosongi
- Kolom kunci vigenere cipher dikosongi
- Nilai kunci railfence cipher lebih dari sama dengan jumlah karakter pada kolom input

Berikut ini adalah tampilan dari pesan *error* yang muncul




**Gambar 3. 7** Tampilan Pesan *Error*

Untuk mencegah munculnya pesan *error* maka harus menghindari terjadinya beberapa kondisi diatas.

## 3.4 Kajian Agama Islam

### 3.4.1 Penyampaian Pesan dan Pengamananya

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-Mumtahanah ayat 1:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَتَّخِذُوا عَدُوِّي وَعَدُوَّكُمْ أَوْلِيَاءَ تُلْقُونَ إِلَيْهِمْ بِالْمَوَدَّةِ وَقَدْ كَفَرُوا بِمَا جَاءَكُمْ مِنَ الْحَقِّ يُخْرِجُونَ الرَّسُولَ وَإِيَّاكُمْ ۙ أَنْ تُؤْمِنُوا بِاللَّهِ رَبِّكُمْ إِنْ كُنْتُمْ خَرَجْتُمْ جِهَادًا فِي سَبِيلِي وَابْتِغَاءَ مَرْضَاتِي ۚ تُسِرُّونَ إِلَيْهِمْ بِالْمَوَدَّةِ وَأَنَا أَعْلَمُ بِمَا أَخْفَيْتُمْ وَمَا أَعْلَنْتُمْ ۚ وَمَنْ يَفْعَلْهُ مِنْكُمْ فَقَدْ ضَلَّ سَوَاءَ السَّبِيلِ

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 disabda kan oleh rasulullah dari hadits al-hasan dari samurah bahwa rasulullah saw bersabda bahwa

أَدِّ الأَمَانَةَ إِلَى مَنِ ائْتَمَنَكَ، وَلاَ تَخُنْ مَنْ خَانَكَ

Artinya: *"sampaikanlah amanat itu kepada orang yang mempercayaimu dan janganlah kamu berkhianat kepada orang yang berkhianat kepadamu"(HR. Imam Ahmad)*

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 disepakati 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:

يَا أَيُّهَا الَّذِينَ آمَنُوا لَا تَخُونُوا اللَّهَ وَالرَّسُولَ وَتَخُونُوا أَمَانَاتِكُمْ وَأَنْتُمْ تَعْلَمُ

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

Seperti 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 karea itu, setidaknya kita melakukan ikhtiar untuk mengamankan pesan itu dengan salah satu caranya adalah dengan menyandikanya menggunakan teknik penyandian yang berlapis lapis.

### 3.4.2 Kajian Keagamaan tentang Persatuan

Konsep penyandian super enkripsi menekankan penggunaan gabungan dari dua jenis teknik penyandian yang berbeda dengan maksud untuk meningkatkan keamanan dari penyandian agar tidak mudah dipecahkan. Allah berfirman dalam surah Ali Imron ayat 103:

وَاعْتَصِمُوا بِحَبْلِ اللَّهِ جَمِيعًا وَلَا تَفَرَّقُوا ۚ وَاذْكُرُوا نِعْمَتَ اللَّهِ عَلَيْكُمْ إِذْ كُنْتُمْ أَعْدَاءً فَأَلَّفَ بَيْنَ قُلُوبِكُمْ فَأَصْبَحْتُمْ بِنِعْمَتِهِ إِخْوَانًا وَكُنْتُمْ عَلَىٰ شَفَا حُفْرَةٍ مِنَ النَّارِ فَأَنْقَذَكُمْ مِنْهَا ۗ كَذَٰلِكَ يُبَيِّنُ اللَّهُ لَكُمْ آيَاتِهِ لَعَلَّكُمْ تَهْتَدُونَ

Artinya: *"Dan berpeganglah kamu semuanya kepada tali (agama) Allah, dan janganlah kamu bercerai-berai". [Ali Imran: 103]*

Ayat ini menjelaskan perintah untuk menjaga persatuan, secara tidak langsung mengisyaratkan bahwa sebuah persatuan akan membuat menjadi lebih kokoh. Nabi juga bersabda tentang pentingnya sebuah persatuan

الْمُؤْمِنُ لِلْمُؤْمِنِ كَالْبُنْيَانِ يَشُدُّ بَعْضُهُ بَعْضًا

Artinya: *"Seorang mukmin terhadap mukmin lainnya seperti satu bangunan, sebagiannya menguatkan yang lainnya."(HR. Bukhori dan Muslim).*

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

# BAB IV

# PENUTUP

## 4.1 Kesimpulan

Dari analisa dan pembahasan dapat diambil beberapa kesimpulan sebagai berikut:

1. Teknik super enkripsi menggunakan *vigenere cipher* dengan modifikasi subtitusi ASCII dilakukan dengan menggunakan persamaan $C = ((P + K) - 64) mod\ 95) + 32$ hasil penyandiannya kemudian disandikan kembali menggunakan *Railfence cipher* dengan memanfaatkan perubahan posisi karakter secara zig-zag. Proses pengembalian pesan dilakukan menggunakan dekripsi *Railfence cipher* kemudian didekripsi dengan *vigenere cipher* menggunakan persamaan $P = ((C - K) mod\ 95) + 32$.
2. Keamanan teknik super enkripsi terletak pada panjang karakter kunci *vigenere cipher* dan besar nilai kunci *Railfence cipher* yang digunakan. penggunaan dua jenis *cipher* memungkinkan keamanan pesan menjadi dua kali lipat.
3. Implementasi sederhana dilakukan menggunakan Python dengan menggunakan prosedur penyandian super enkripsi sehingga hasil hitung aplikasi akan sesuai dengan hasil hitung manual. Namun, aplikasi ini masih terbatas penggunaanya untuk menyandikan karakter pesan yang berada dalam cakupan ASCII karakter nomor 32 sampai 127.

## 4.2 Saran

Pada penelitian ini membahas tentang super enkripsi menggunakan teknik *vigenere cipher* dan *railfence cipher* serta implementasinya menggunakan Python. Untuk penelitian selanjutnya disarankan untuk memodifikasi program penyandian sehingga memungkinkan untuk mennyandikan file non teks dan juga pesan karakter bahasa arab. selain itu, disararnkan menggunakan teknik penyandian lain untuk mengetahui teknik penyandian yang lebih kuat lagi tingkat keamananya.

## DAFTAR RUJUKAN


Abdussakir. (2009). *Matematika 1 Kajian Integratif Matematika dan Al-Qur'an.* Malang: UIN Malang Press.

Ariyus, D. (2006). *Kriptografi Keamanan Data dan Komunikasi.* Yogyakarta: Graha Ilmu.

Bhattacharya, P. B. (1994). *Basic Abstract Algebra.* New York: Cambridge University Press.

Gallian, J. A. (2010). *Contemporary Abstract Algebra (7th ed.).* Belmont: Brooks/Cole.

Irawan, W. H. (2014). *Pengantar Teori Bilangan.* Malang: UIN-Maliki Press.

Judson, T. W., & Beezer, R. (2016). *Abstract Algebra Theory and Applications.* Texas: PWS Publishing.

Kamil, F. (2016). *Implementasi Kriptografi dengan Menggunakan Algoritma Advanced Encryption Standard (AES 256) dan Lempel Ziv Welch (LZW).* Tangerang: STMIK Raharja.

Katz, J., & Lindell, Y. (2015). *Introduction to Modern Cryptography. 2nd ed.* Boca Raton: CRC Press.

LastBit. (2005). *Brute Force Attack.* Retrieved Agustus 2, 2019, from http://www.lastbit.com/rm_bruteforce.asp

Lutz, M. (2013). *Learning Python 5th Edition.* Sebastopol: O'Reilly Media, Inc.

Manggala, R. (2010). Analisis Kriptografi dalam penentuan Cipherteks kode ASCII melalui metode. *MAKALAH IF3058 KRIPTOGRAFI*, 3-4.

McAndrew, A. (2011). *Introduction to Cryptography With Open-Source Software.* Florida: CRC Press.

Muhsetyo, G. (1997). *Dasar-Dasar Teori Bilangan.* Jakarta: PGSM.

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.

Rosen, K. H. (2012). *Discrete Mathematics and Its Apllications 7th ed.* New York: McGraw-Hill.

Rosmala, D., & Dwipa, G. (2012). PEMBANGUNAN WEBSITE CONTENT MONITORING SYSTEM MENGGUNAKAN DIFFLIB PYTHON. *Informatika*, 20.

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

Setyaningsih, E. (2015). *Kriptografi Dan Implementasinya Menggunakan Matlab.* Yogyakarta: ANDI.

Sweigart, A. (2013). *Hacking Secret Cipher with Python.* California: CreateSpace Independent Publishing.

# LAMPIRAN

```
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtWidgets import QMessageBox
import base64
class Ui_Dialog(object):
    def setupUi(self, Dialog):
        Dialog.setObjectName("Super Enkripsi")
        Dialog.resize(498, 534)
        self.Push1 = QtWidgets.QPushButton(Dialog)
        self.Push1.setGeometry(QtCore.QRect(80, 330, 75, 23))
        self.Push1.setObjectName("encrypt")
        self.Push1.clicked.connect(self.terklik)
        self.Push2 = QtWidgets.QPushButton(Dialog)
        self.Push2.setGeometry(QtCore.QRect(340, 330, 75, 23))
        self.Push2.setObjectName("decrypt")
        self.Push2.clicked.connect(self.terklik2)
        self.Push3 = QtWidgets.QPushButton(Dialog)
        self.Push3.setGeometry(QtCore.QRect(470, 80, 30, 20))
        self.Push3.setObjectName("open")
        self.Push3.clicked.connect(self.terklik3)
        self.Push4 = QtWidgets.QPushButton(Dialog)
        self.Push4.setGeometry(QtCore.QRect(470, 400, 30, 20))
        self.Push4.setObjectName("save")
        self.Push4.clicked.connect(self.terklik4)
        self.verticalLayoutWidget = QtWidgets.QWidget(Dialog)
        self.verticalLayoutWidget.setGeometry(QtCore.QRect(30, 50, 441, 261))
        self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout.setObjectName("verticalLayout")
        self.label = QtWidgets.QLabel(self.verticalLayoutWidget)
        self.label.setObjectName("label")
        self.verticalLayout.addWidget(self.label)
        self.txt = QtWidgets.QPlainTextEdit(self.verticalLayoutWidget)
        self.txt.setObjectName("txt")
        self.verticalLayout.addWidget(self.txt)
        self.label_2 = QtWidgets.QLabel(self.verticalLayoutWidget)
        self.label_2.setObjectName("label_2")
        self.verticalLayout.addWidget(self.label_2)
        self.key = QtWidgets.QLineEdit(self.verticalLayoutWidget)
        self.key.setObjectName("key")
        self.verticalLayout.addWidget(self.key)
```

```
        self.label_3 = QtWidgets.QLabel(self.verticalLayoutWidget)
        self.label_3.setObjectName("label_3")
        self.verticalLayout.addWidget(self.label_3)
        self.k = QtWidgets.QSpinBox(self.verticalLayoutWidget)
        self.k.setMinimum(2)
        self.k.setObjectName("k")
        self.verticalLayout.addWidget(self.k)
        self.verticalLayoutWidget_2 = QtWidgets.QWidget(Dialog)
        self.verticalLayoutWidget_2.setGeometry(QtCore.QRect(30, 370, 441, 141))
        self.verticalLayoutWidget_2.setObjectName("verticalLayoutWidget_2")
        self.verticalLayout_2 = QtWidgets.QVBoxLayout(self.verticalLayoutWidget_2)
        self.verticalLayout_2.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout_2.setObjectName("verticalLayout_2")
        self.label_4 = QtWidgets.QLabel(self.verticalLayoutWidget_2)
        self.label_4.setObjectName("label_4")
        self.verticalLayout_2.addWidget(self.label_4)
        self.out = QtWidgets.QTextEdit(self.verticalLayoutWidget_2)
        self.out.setObjectName("out")
        self.verticalLayout_2.addWidget(self.out)
        self.label_5 = QtWidgets.QLabel(Dialog)
        self.label_5.setGeometry(QtCore.QRect(10, 10, 481, 31))
        font = QtGui.QFont()
        font.setFamily("Verdana")
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.label_5.setFont(font)
        self.label_5.setObjectName("label_5")
        self.retranslateUi(Dialog)
        QtCore.QMetaObject.connectSlotsByName(Dialog)

    def retranslateUi(self, Dialog):
        _translate = QtCore.QCoreApplication.translate
        Dialog.setWindowTitle(_translate("Dialog", "VigRail"))
        self.Push1.setText(_translate("Dialog", "ENKRIPSI"))
        self.Push2.setText(_translate("Dialog", "DEKRIPSI"))
        self.Push3.setText(_translate("Dialog", "OPEN"))
        self.Push4.setText(_translate("Dialog", "SAVE"))
        self.label.setText(_translate("Dialog", "Input Teks :"))
        self.label_2.setText(_translate("Dialog", "Kunci Vignere Cipher :"))
        self.label_3.setText(_translate("Dialog", "Kunci Railfence Cipher :"))
```

```
        self.label_4.setText(_translate("Dialog", "Output Teks"))
        self.label_5.setText(_translate("Dialog", "SUPER ENKRIPSI VIGNERE DAN RAILFENCE CIPHER"))

    def terklik(self):
        txt = self.txt.toPlainText()
        key = self.key.text()
        k   = self.k.value()

        if len(txt) < 1:
            msg = QMessageBox()
            msg.setIcon(QMessageBox.Critical)
            msg.setText("Mohon Maaf")
            msg.setInformativeText('Text jangan Kosong')
            msg.setWindowTitle("error")
            msg.exec_()
        elif len(key) < 1:
            msg = QMessageBox()
            msg.setIcon(QMessageBox.Critical)
            msg.setText("Mohon Maaf")
            msg.setInformativeText('Key jangan Kosong')
            msg.setWindowTitle("error")
            msg.exec_()
        else:
            if k >= len(txt):
                msg = QMessageBox()
                msg.setIcon(QMessageBox.Critical)
                msg.setText("Kunci Railfence Cipher")
                msg.setInformativeText('harus kurang dari panjang pesan')
                msg.setWindowTitle("error")
                msg.exec_()
            else:
                self.encrypt(txt,key,k)

    def terklik2(self):
        txt = self.txt.toPlainText()
        key = self.key.text()
        k   = self.k.value()

        if len(txt) < 1:
            msg = QMessageBox()
            msg.setIcon(QMessageBox.Critical)
```

```
            msg.setText("Mohon Maaf")
            msg.setInformativeText('Text jangan Kosong')
            msg.setWindowTitle("error")
            msg.exec_()
        elif len(key) < 1:
            msg = QMessageBox()
            msg.setIcon(QMessageBox.Critical)
            msg.setText("Mohon Maaf")
            msg.setInformativeText('Key jangan Kosong')
            msg.setWindowTitle("error")
            msg.exec_()
        else:
            if k >= len(txt):
                msg = QMessageBox()
                msg.setIcon(QMessageBox.Critical)
                msg.setText("Kunci Railfence Cipher")
                msg.setInformativeText('harus kurang dari panjang pesan')
                msg.setWindowTitle("error")
                msg.exec_()
            else:
                self.decrypt(txt,key,self.k.value())
    def terklik3(self):
        directory = str(QtWidgets.QFileDialog.getOpenFileUrl())
        self.txt.setText(directory)
    def terklik4(self):
        simpan = str(QtWidgets.QFileDialog.getSaveFileName())
        self.txt.setText(simpan)
    #script
    def vign(self, txt=' ',key=' ',typ='d'):
        semesta=[c for c in (chr(i)for i in range (32,127))]
        len_semesta=len(semesta)

        if not txt:
            print ('masukkan teks')
            return
        if not key:
            print ('masukkan kunci')
            return
        if typ not in ('d','e'):
            print ('d = dekripsi atau e = enkripsi')
            return
```

```
    if any (t not in semesta for t in key):
        print('masukkan kode ASCII')
        return
    ret_txt=''
    len_key =len(key)
    for i, l in enumerate(txt):
        if l not in semesta:
            ret_txt+=1
        else:
            txt_idx=semesta.index(l)

            k=key[i% len_key]
            key_idx=semesta.index(k)
            if typ =='d':
                key_idx*=-1

            kode = semesta[(txt_idx+key_idx)%len_semesta]

            ret_txt+=kode
    print(ret_txt)
    return ret_txt

def encrypts(self,p,k):
  fence = [[] for i in range(k)]
  rail  = 0
  var   = 1

  for char in p:
    fence[rail].append(char)
    rail += var

    if rail == k-1 or rail == 0:
      var = -var

  res = ''
  for i in fence:
    for j in i:
      res += j
  return res
```

```
def decrypts(self,c,k):
    fence = [[] for i in range(k)]
    rail  = 0
    var   = 1

    for char in c:
        fence[rail].append(char)
        rail += var

        if rail == k-1 or rail == 0:
            var = -var

    rFence = [[] for i in range(k)]

    i = 0
    l = len(c)
    c = list(c)
    for r in fence:
        for j in range(len(r)):
            rFence[i].append(c[0])
            c.remove(c[0])
        i += 1

    rail = 0
    var  = 1
    r = ''
    for i in range(l):
        r += rFence[rail][0]
        rFence[rail].remove(rFence[rail][0])
        rail += var

        if rail == k-1 or rail == 0:
            var = -var
    return r
def encrypt (self,txt,key,k):
    x=self.vign(txt,key,'e')
    y=self.encrypts(x,k)
    self.out.setText(y)
    return y

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    Dialog = QtWidgets.QDialog()
    ui = Ui_Dialog()
    ui.setupUi(Dialog)
    Dialog.show()
    sys.exit(app.exec_())
```

## RIWAYAT HIDUP

Sigit Deni Santoso, lahir di Banywangi 01 Juli 1996, tinggal di Desa Sempu, Kecamatan Sempu, Kabupaten Banyuwangi. Anak bungsu dari dua bersaudara, putra dari bapak Arifin dan ibu Umi Toyibah.

Pendidikan dasar ditempuh di MI NU Salafiyah Tugung dan lulus pada tahun 2008, melanjutkan pendidikan menengah pertama di SMP Negeri 1 Genteng dan lulus tahun 2011, kemudian melanjutkan pendidikan menengah atas di SMA Negeri 1 Genteng dan lulus tahun 2014. Selanjutnya pada tahun 2015 menempuh pendidikan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang mengambil jurusan Matematika Fakultas Sains dan Teknologi. Selama menjadi mahasswa pernah mengikuti Mathematic English Club pada 2016/2017 dan pernah menjadi asisten laboratorium selama satu semester.

Selain pendidikan formal, dia juga menempuh pendidikan non formal di madrasah at-toyibah ketika duduk dibangku MI sampai SMP, menempuh pendidikan non formal Ma'had Sunan Ampel Al-Aly periode 2015/2016, kemudian melanjutkan pendidikan nonformal di Pondok Pesantren Anwarul Huda Malang.

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 : Sigit Deni Santoso
NIM : 15610071
Fakultas/Jurusan : Sains dan Teknologi/Matematika
Judul Skripsi : Implementasi Penyandian Super Enkripsi *Vigenere Cipher* dan *Railfence Cipher* Menggunakan Python
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Mohammad Nafie Jauhari, M.Si

| No | Tanggal | Hal | Tanda Tangan |
|---|---|---|---|
| 1. | 13 Maret 2019 | Konsultasi Bab I dan Bab II | 1. |
| 2. | 20 Maret 2019 | Konsultasi Bab II | 2. |
| 3. | 27 Maret 2019 | ACC Bab I dan Bab II | 3. |
| 4. | 28 Maret 2019 | Konsultasi Kajian Keagamaan | 4. |
| 5. | 5 April 2019 | Konsultasi Kajian Keagamaan | 5. |
| 6. | 25 April 2019 | Konsultasi Bab III | 6. |
| 7. | 20 Agustus 2019 | Konsultasi Bab III dan Bab IV | 7. |
| 8. | 27 Agustus 2019 | Konsultasi Kajian Keagamaan | 8. |
| 9. | 29 Agustus 2019 | ACC Kajian Keagamaan | 9. |
| 10. | 29 Agustus 2019 | ACC Keseluruhan | 10. |

Malang, 29 Agustus 2019
Mengetahui,
Ketua Jurusan Matematika

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