# KRIPTOGRAFI TEKS MENGGUNAKAN MODIFIKASI SUBSTITUSI CIPHER DAN ELEMEN MUSIK

## *TEXT CRYPTOGRAPHY USING MODIFICATION OF CIPHER SUBSTITUTIONS AND MUSIC ELEMENTS*

**Yusuf Fadlila Rachman[1], Dony Ariyus[2]**
[1 2]Magister Teknik Informatika, Universitas Amikom Yogyakarta
Email: [1]yusuf.rachman@students.amikom.ac.id, [2]dony.a@amikom.ac.id

***Abstract:*** *Cryptography is a method used to facilitate data. There are two important components of cryptography, encryption and decryption. Cryptography can be applied to various things, one of which uses music. Music cryptography is the process of encrypting text characters that will produce cipher text in the form of audio. This algorithm encrypts data by combining the cipher substitution method with musical elements. Audio results from calculations of tone frequency, pitch length, and amplitude. The algorithm used has an encryption speed of 3,72 seconds and a decryption speed of 4,16 seconds.*
***Keywords:*** *Cryptography, Substitution Algorithm, Music, Tone, Chord*

**Abstrak:** Kriptografi merupakan sebuah metode yang digunakan untuk mengamankan data. Terdapat dua komponen penting kriptografi yaitu enkripsi dan dekripsi. Kriptografi dapat diaplikasikan kedalam berbagai hal, salah satunya menggunakan musik. Kriptografi musik adalah proses enkripsi karakter teks yang akan menghasilkan cipherteks berupa audio (musik). Algoritma ini mengenkripsi data dengan mengkombinasi metode substitusi cipher dengan elemen-elemen musik. Audio yang dihasilkan berasal dari perhitungan frekunsi nada, panjang nada, dan amplitudo. Algoritma yang digunakan memiliki kecepatan enkripsi sebesar 3,72 detik dan kecepatan dekripsi sebesar 4,16 detik.
**Keywords:** Kriptografi, Algoritma Substitusi, Musik, Nada, Chord

## PENDAHULUAN

Kriptografi merupakan bidang ilmu yang berisi teknik atau metode pengamanan informasi pada suatu dokumen. Kriptografi berguna untuk menjaga keaslian isi dari sebuah informasi yang dikirimkan. Kriptografi dilakukan dengan merubah isi dokumen ke dalam karakter atau simbol acak menggunakan bantuan algoritma tertentu sehingga informasi asli tidak dapat dikenali oleh pihak lain. Kriptografi terdiri dari 2 aksi penting yaitu enkripsi dan dekripsi (Stallings, 2017). Enkripsi merupakan proses penyandian informasi sehingga tidak dapat dikenali lagi (ciphertext). Sedangkan dekripsi merupakan proses menterjemahkan cipherteks hasil enkripsi kedalam teks asli atau plainteks.

Substitusi merupakan salah satu algoritma paling sederhana yang digunakan dalam kriptografi. Algoritma substitusi merubah karakter pada plain teks dengan sebuah simbol atau karakter lain sesuai aturan yang telah dibuat (Hammood et al., 2010).

Banyak hal yang dapat digunakan dalam melakukan enkripsi dalam kriptografi. Salah satunya adalah enkripsi teks menggunakan elemen-elemen musik (Lamaute et al., 2016). Musik memiliki banyak elemen yang dapat digunakan sebagai media penyandian data pada kriptografi. Musik memiliki notasi balok maupun notasi angka yang dapat digunakan untuk pengkodean informasi.

Berdasarkan penjelasan diatas, maka permasalahan yang diangkat pada penelitian ini adalah bagaimana membuat algoritma enkripsi dan dekripsi yang menghasilkan plainteks berbentuk audio. Algoritma yang diajukan mampu membuat sebuah pesan rahasia dengan mudah namun sulit untuk dideteksi.

## KAJIAN TEORI

Penelitian berjudul "*Musical Cryptography Using Multiple Note Substitution Algorithm"* menjelaskan bahwa penerapan algoritma substitusi mampu menghasilkan cipher teks berbentuk musik yang enak didengar dan mengurangi peluang pesan tersebut terbongkar

(Raghav & John, 2016). Penelitian berjudul *"A Hybrid Polybius-Playfair Music Cipher"* menjelaskan bahwa kriptografi musikal dapat digunakan untuk mengganti steganografi audio dengan mengurangi upaya pencarian ukuran file yang tepat sebagai cover pesan (Kumar et al., 2015). Penelitian berjudul *"A Symmetric Key Algorithm for Cryptography using Music"* menjelaskan bahwa algoritma yang diajukan mampu merubah karakter plainteks menjadi potongan musik dengan merubah karakter teks dengan not musik yang telah digenerate (Dutta et al., 2013). Penelitian berjudul *"Music Cryptography based on Carnatic Music"* menjelaskan bahwa metode yang diajukan bertujuan untuk merubah pesan berformat teks kedalam klip musik. Setiap karakter pesan akan di rubah berdasarkan kombinasi dari *swara* dan menghasilkan output enkripsi berupa musik yang sesuai dengan Musik Klasik Carnatic (Rao & Koolagudi, 2019). Penelitian berjudul *"A Substitution Cipher for Musical Cryptograhy"* menjelaskan bahwa algoritma yang diajukan memiliki kemampuan untuk mengenkripsi plainteks secara efisien. Hasil enkripsi berupa suara dan sulit untuk dideteksi sebagai cipherteks (Lamaute et al., 2016). Penelitian berjudul *"Encrypting Messages Using Musical Notes by Genetic Algorithm"* menjelaskan bahwa proses pendeteksian pesan rahasia lebih sulit dilakukan dalam enkripsi pesan menggunakan notasi musik. Permutasi dan kombinasi yang digunakan untuk dekripsi pesan membuat hampir mustahil dilakukan tanpa menggunakan kunci (Patil et al., 2017). Penelitian berjudul *"Genetic Algorithm Based Cryptographic Approach Using Karnatic Music"* menjelaskan bahwa penggunaan algoritma genetik untuk menghasilkan pesan rahasia tidak hanya bertujuan untuk menyembunyikan pesan ke dalam sebuah musik, tetapi mengurangi kemungkinan pesan tersebut teridentifikasi (Surya & H, 2017).

- Nada

Nada merupakan suara yang beraturan dan memiliki frekuensi tertentu (Satrianingsih, 2006). Setiap nada memiliki tinggi nada tertentu berdasarkan frekuensi tiap-tiap nada.

Nada pada musik terdiri dari gabungan antara frekuensi dasar, volume nada, dan bentuk nada. Secara matematis, sebuah nada sederhana dapat dibuat menggunakan gelombang sine. Hubungan antara nada dan frekuensi dapat dinyatakan dalam persamaan $A\sin(2\pi ft)$ dimana f adalah frekuensi, t adalah waktu nada tersebut dimainkan, dan A adalah amplitudo. (Dutta et al., 2013). Tabel nada beserta frekuensinya dapat dilihat pada tabel 1:

**Tabel 1.** Nada dan Frekuensi

| Note/Octave | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| C | 65.07 | 130.14 | 260.29 | 520.58 | 1041.16 | 2082.31 |
| C#/Db | 69.46 | 138.92 | 277.84 | 555.69 | 1111.37 | 2222.75 |
| D | 73.27 | 146.54 | 293.08 | 586.17 | 1172.34 | 2344.68 |
| D#/Eb | 77.42 | 154.83 | 309.66 | 619.33 | 1238.65 | 2477.30 |
| E | 82.57 | 165.14 | 330.28 | 660.56 | 1321.12 | 2642.24 |
| F | 86.72 | 173.45 | 346.90 | 693.80 | 1387.60 | 2775.19 |
| F#/Gb | 93.05 | 186.10 | 372.19 | 744.39 | 1488.78 | 2977.56 |
| G | 97.65 | 195.30 | 390.61 | 781.21 | 1562.43 | 3124.86 |
| G#/Ab | 103.70 | 207.41 | 414.82 | 829.64 | 1659.28 | 3318.56 |
| A | 110.00 | 220.00 | 440.00 | 880.00 | 1760.00 | 3520.00 |
| A#/Bb | 115.58 | 231.16 | 462.33 | 924.65 | 1849.31 | 3698.61 |
| B | 123.96 | 247.92 | 495.84 | 991.68 | 1983.36 | 3966.72 |

- Not

Not atau notasi merupakan representasi dari nada. Not berguna untuk menterjemahkan bunyi suatu nada ke dalam sebuah angka ataupun simbol gambar (Lamaute et al., 2016). Manfaat dari not adalah untuk mengetahui dan memahami nada yang sedang dibunyikan. Terdapat dua jenis not yang digunakan pada musik, yaitu not balok dan not angka. Not angka merepresentasikan bunyi nada kedalam angka, dimulai dari do = 1, re = 2, mi = 3, fa = 4, sol = 5, la = 6, dan si = 7. Sedangkan not balok berguna untuk merepresentasikan bunyi nada kedalam simbol-simbol musik.

- Tangga Nada

Tangga nada merupakan elemen yang sangat penting dalam penyusunan sebuah musik. Tangga nada adalah kumpulan nada-nada yang tersusun dalam sebuah sistem nada (Green, 2014). Tangga nada dapat dilihat pada gambar 1:

| Nada | Do | Re | Mi | Fa | Sol | La | Si | Do' |
|---|---|---|---|---|---|---|---|---|
| C | C | D | E | F | G | A | B | C' |
| C# | C♯ | D♯ | E | F♯ | G♯ | A♯ | C | C♯' |
| D♭ | D♭ | E♭ | F♭ | G♭ | A♭ | B♭ | C♭ | D♭' |
| D# | D♯ | F | G | G♯ | A♯ | C | D | D♯' |
| E | E | F♯ | G♯ | A | B | C♯ | D♯ | E' |
| E♭ | E♭ | F | G | A♭ | B♭ | C | D | E♭' |
| F# | F♯ | G♯ | A♯ | B | C♯ | D♯ | F | F♯' |
| G | G | A | B | C | D | E | F♯ | G' |
| G# | G♯ | A♯ | C | C♯ | D♯ | F | G | G♯' |
| A | A | B | C♯ | D | E | F♯ | G♯ | A' |
| A# | A♯ | C | D | D♯ | F | G | A | A♯' |
| B | B | C♯ | D♯ | E | F♯ | G♯ | A♯ | B' |

**Gambar 1.** Tangga Nada

- Kunci nada/Akord/Chord

Kunci nada atau chord merupakan kumpulan dari 2 atau lebih not yang dimainkan secara bersamaan. Semua chord yang dibuat menggunakan not tertentu akan menghasilkan sebuah tangga nada chord tersebut. Sebagai contoh, chord C mayor terdiri dari 3 not yaitu C-G-E (1-3-5). Terdapat 4 tipe chord yaitu chord mayor, minor, augmented, dan deminished (Lamaute et al., 2016). Beberapa chord dan kombinasi not dapat dilihat pada gambar 2:

| | | | | |
|---|---|---|---|---|
| 0 | C | C | E | G |
| 1 | C# | C# | G# | F |
| 2 | D | D | F# | A |
| 3 | D# | D# | A# | G |
| 4 | E | E | G# | B |
| 5 | F | F | A | C |
| 6 | F# | F# | A# | C |
| 7 | G | G | B | D |
| 8 | G# | G# | C | D# |
| 9 | A | A | C# | E |
| 10 | A# | A# | D | F |
| 11 | B | B | D# | F# |

**Gambar 2.** Kombinasi Chord

- Substitution Cipher

Salah satu metode yang paling tua dalam kriptografi adalah metode substitusi. Metode ini dilakukan dengan merubah alfabet teks ke dalam simbol atau karakter yang telah ditentukan (Hammood et al., 2010). Perubahan alfabet kedalam karakter dilakukan dengan bantuan sebuah algoritma tertentu. Terdapat beberapa algortima yang dapat digunakan dalam substitusi cipher yaitu, *Caesar Cipher* dan *Hill Cipher*. *Caesar cipher* termasuk dalam metode substitusi cipher yang sederhana dimana proses pengkodean teks berdasarkan substitusi atau perpindahan (Singh et al., 2014). *Caesar cipher* termasuk dalam kriptografi simetris. Caesar cipher membutuhkan kunci yang sama dalam proses enkripsi dan dekripsi.

Hill cipher termasuk dalam blok cipher atau polyalphabetic cipher. Proses enkripsi dan dekripsi pada hill cipher akan dilakukan pada tiap blok karakter, dimana karakter dalam satu blok akan berpengaruh pada karakter lainnya. Salah satu keunggulan algoritma in adalah tidak dapat dipecahkan menggunakan teknik analisis frekuensi. Algoritma ini pertama kali

ditemukan oleh Lester S. Hill pada tahun 1929. Proses enkripsi dan dekripsi algoritma *Hill Cipher* memakai kunci berbentuk matriks berukuran m x m. Dasar perhitungan dalam algoritma hill cipher menggunakan modulo untuk tiap karakter yang akan dienkripsi ataupun dekripsi (Susanto et al., 2020).

## METODE PENELITIAN

Berdasarkan permasalahan yang telah dijabarkan, maka terdapat alur penelitian yang digunakan untuk menyelesaikan permasalahan tersebut. Alur penelitian yang dilakukan pada penelitian ini yaitu:

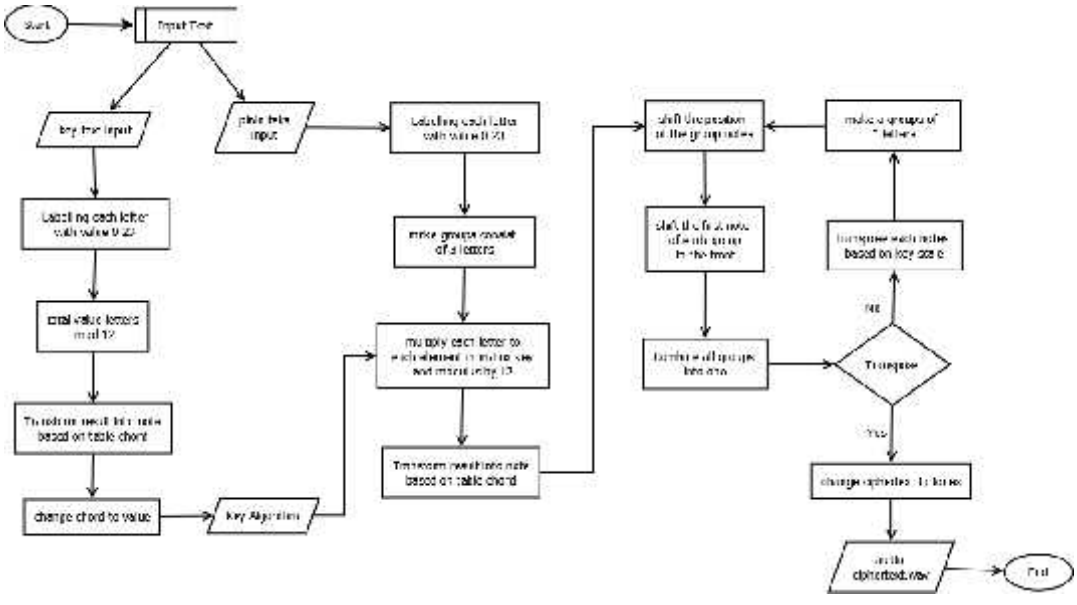
1. Studi Literatur: studi literatur dilakukan dengan melakukan observasi dan analisa terkait penyelesaian masalah yang didapat dari berbagai sumber seperti jurnal ilmiah ataupun buku dengan tema kriptografi.
2. Perancangan Algoritma: proses perancangan algoritma dilakukan dengan mengkombinasikan beberapa algoritma yang sudah ada dengan algoritma hasil pemikiran penulis.
3. Uji Coba Algoritma: uji coba dilakukan untuk mengetahui bagaimana performa algoritma dalam melakukan proses enkripsi ataupun dekripsi.

## HASIL PENELITIAN

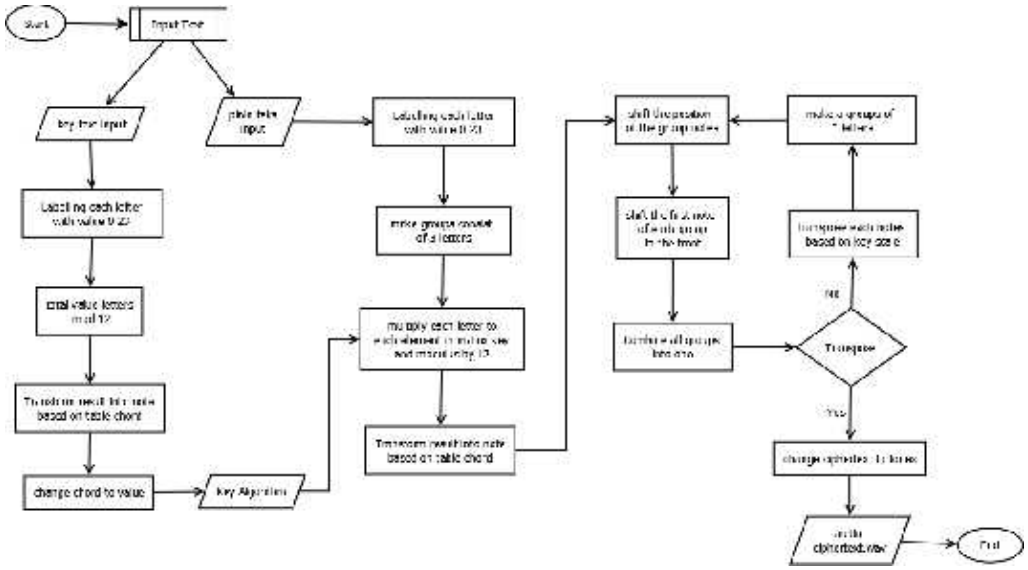
Hasil yang diperoleh pada penelitian ini adalah dengan terbentuknya sebuah algoritma yang mampu melakukan enkripsi teks ke audio dan dekripsi audio ke teks. akan di selesaikan menggunakan algoritma yang diajukan. Algoritma yang diajukan memiliki dua proses, yaitu enkripsi dan dekripsi. Alur proses enkripsi dapat dilihat pada gambar 3. Sedangkan alur proses dekripsi dapat dilihat pada gambar 4. Algoritma yang diajukan akan melalui uji coba performa untuk mengetahui kecepatan proses komputasi. Hasil akhir proses uji coba dapat dilihat pada tabel 2.

**Tabel 2.** Pengujian Algoritma

| Panjang Karakter | Waktu Enkripsi | Waktu Dekripsi |
|---|---|---|
| <10 | 3,9 detik | 3,7 detik |
| 20 | 3,7 detik | 4,3 detik |
| 30 | 3,5 detik | 4,0 detik |
| 40 | 3,6 detik | 4,2 detik |
| 50< | 3,9 detik | 4,6 detik |



**Gambar 3.** Algoritma Enkripsi



**Gambar 4.** Algoritma Dekripsi

## PEMBAHASAN

Algoritma yang digunakan pada penelitian ini merupakan gabungan dari beberapa algoritma kriptografi substitusi yaitu caesar cipher dan hill cipher. Algoritma memiliki kemampuan enkripsi 26 huruf menjadi audio. Plainteks yang diberikan akan dienkripsi kedalam audio yang memiliki nada sesuai tangga nada musik. Setiap tangga nada terdiri dari 24 not yang terbagi menjadi 2 oktaf.

Proses pertama adalah identifikasi 12 nada pada satu oktaf. Langkah ini dibuat berdasarkan *chord* musik. Kunci ini digunakan untuk menentukan tangga nada yang digunakan pada proses translasi teks. Kunci algoritma didapatkan melalui perhitungan modulo pada kunci teks. Huruf yang dapat dienkripsi berjumlah 24 dengan menggabungkan huruf I/J dan U/V. Penggabungan dilakukan untuk mempermudah pelabelan alfabet, sesuai dengan jumlah not pada satu oktaf. Alfabet kemudian dilabeli nilai mulai dari 0 hingga 23. Proses pertama ini dapat dilihat pada gambar 5.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| C | D | E | F | G | A | B | C' | D' | E' | F' | G' | A' | B' | C# | D# | F# | G# | A# | C#' | D#' | F#' | C#' | A#' |
| C# | D# | E | F# | G# | A# | C | C#' | D#' | E' | F#' | G#' | A#' | C' | D | F | G | A | B | D' | F' | G' | A' | B' |
| D | E | F# | G | A | B | C# | D' | E' | F#' | G' | A' | B' | C#' | C | D# | F | G# | A# | C' | D#' | F' | G#' | A#' |
| D# | F | G | G# | A# | C | D | D#' | F' | G' | G#' | A#' | C' | D' | C# | F | F# | A | B | C#' | F' | F#' | A' | B' |
| E | F# | G# | A | B | C# | D# | E' | F#' | G#' | A' | B' | C#' | D#' | F | G | A# | C | D | F' | G' | A#' | C' | D' |
| F | G | A | A# | C | D | E | F' | G' | A' | A#' | C' | D' | E' | F# | G# | B | C# | D# | F#' | G#' | B' | C#' | D#' |
| F# | G# | A# | B | C# | D# | F | F#' | G#' | A#' | B' | C#' | D#' | F' | G | A | C | D | E | G' | A' | C' | D' | E' |
| G | A | B | C | D | E | F# | G' | A' | B' | C' | D' | E' | F#' | G# | A# | C# | D# | F | G#' | A#' | C#' | D#' | F' |
| G# | A# | C | C# | D# | F | G | G#' | A#' | C' | C#' | D#' | F' | G' | A | B | D | E | F# | A' | B' | D' | E' | F#' |
| A | B | C# | D | E | F# | G# | A' | B' | C#' | D' | D' | F#' | G#' | A# | C | D# | F | G | A#' | C' | D#' | F' | G' |
| A# | C | D | D# | F | G | A | A#' | C' | D' | D#' | F' | G' | A' | B | C# | E | F# | G# | B' | C#' | E' | F#' | G#' |
| B | C# | D# | E | F# | G# | A# | B' | C#' | D#' | E' | F#' | G#' | A#' | C | D | F | G | A | C' | D' | F' | G' | A' |

**Gambar 5**. Tabel Tangga Nada dan Pelabelan Alfabet

Langkah kedua merubah *key* algoritma ke dalam chord sesuai dengan tabel chord. Misal, akan *key* yang digunakan HJAUFBI. Maka masing-masing huruf akan dijumlahkan sesuai nilai yang dimiliki kemudian di modulo 12. Berdasarkan perhitungan tersebut, maka chord dan tangga nada yang digunakan adalah C.

Langkah ketiga akan dibentuk matriks berdasarkan kunci nada tersebut. Berdasarkan gambar 2, kunci nada C merupakan kombinasi dari nada C, E, dan G. Kemudian ubah kombinasi tersebut sesuai nilai yang dimiliki oleh masing-masing nada. Nada C = 0, nada E = 4, dan nada G = 7. Hasil akhir matriks yang didapat adalah [0, 4, 7]. Matriks ini digunakan untuk merubah plainteks input kedalam ciphertext.

Plainteks yang akan dienkripsi dikelompokkan setiap 3 karakter. Contoh plainteks RAMADAN DIRUMAH akan dibagi menjadi RAM, ADA, NDI, RUM, AH. Jika terdapat karakter sisa, maka akan ditambah dengan X sehingga tepat memiliki 3 karakter per kelompok. Kelompok karakter dapat dilihat pada tabel 3.

**Tabel 3.** Kelompok karakter

| Kelompok 1 | Kelompok 2 | Kelompok 3 | Kelompok 4 | Kelompok 5 |
|---|---|---|---|---|
| RAM | ADA | NDI | RUM | AHX |

Langkah keempat adalah mengalikan masing-masing kelompok dengan matriks kunci nada. Hasil perkalian nantinya akan diubah sesuai dengan kunci nada yang telah didefinisikan sebelumnya. Rumus yang digunakan adalah: Nada[i] = (Karakter[i] x matriks[j]) mod 12, dimana i merupakan nilai karakter ke-i dan matriks[j] merupakan matriks kunci nada indeks ke j. Contoh perhitungannya adalah

R = (16 x 0) mod 12 = 0 = C
A = (0 x 4) mod 12 = 0 = C
M = (11 x 7) mod 12 = F
dst

Maka hasil akhirnya adalah C, C, F, C, C, C, C, C, G#, C, E, F, C, E, D#

Langkah kelima adalah melakukan pergeseran kelompok karakter sesuai jumlah kelompok. Terdapat 5 kelompok karakter yang didapat, maka akan digenerate nilai mulai 1-5 sesuai posisi kelompok. Contoh susuan nilai yang diberikan adalah 4,3,5,1,2. Kelompok karakter tersebut akan digeser posisinya dengan nilai yang telah diberikan. Maka hasilnya dapat dilihat pada tabel 4.

**Tabel 4.** Pergeseran Karakter

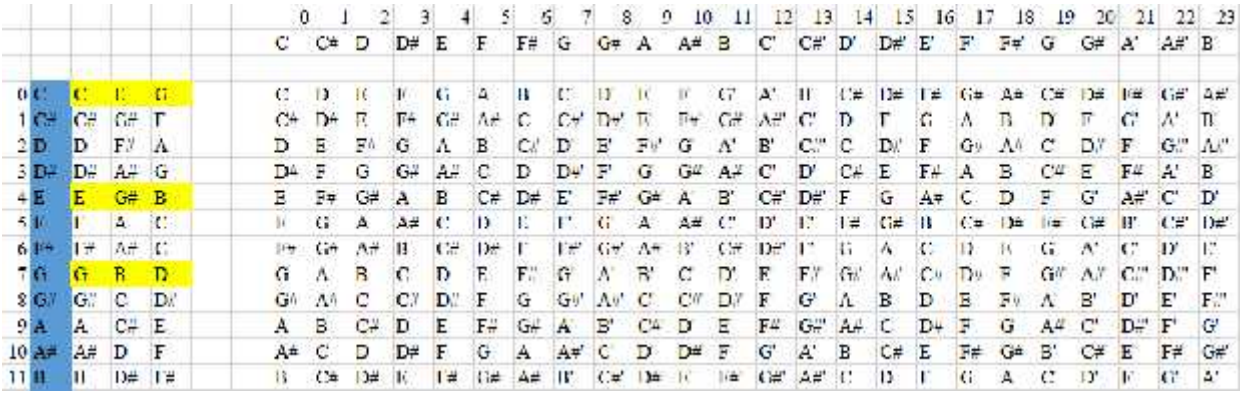
| Kelompok 4 | Kelompok 3 | Kelompok 5 | Kelompok 1 | Kelompok 2 |
|---|---|---|---|---|
| CEF | CCG# | CED# | CCC | CCF |

Selanjutnya dilakukan pergeseran karakter pertama tiap kelompok ke depan berdasarkan posisi kelompok karakter pada langkah sebelumnya. Misal diberikan nilai 5,1,3,2,4, maka yang bergeser ke depan adalah karakter C pada kelompok 4, C pada kelompok 3, dan seterusnya. Hasil pergeseran dapat dilihat pada tabel 5.

**Tabel 5.** Pergeseran Karakter Pertama

| Kelompok 1 | Kelompok 2 | Kelompok 3 |
|---|---|---|
| CCCCC | ECCCE | D#CG#FF |

Langkah keenam adalah melakukan perubahan karakter. Proses ini diawali dengan mengidentifikasi nada-nada yang terdapat pada 2 oktaf yang digunakan. Susunan tangga nada dapat dilihat pada gambar 3:



**Gambar 6**. Susunan kunci dan tangga nada

Terdapat 24 nada yang digunakan sesuai dengan jumlah alfabet yang digunakan mulai C hingga B'. Hasil yang diperoleh pada langkah 5 kemudian akan diubah sesuai dengan kombinasi kunci pada gambar 3. Kombinasi kunci nada yang digunakan adalah C E G. Perubahan dilakukan dengan setiap 3 karakter sesuai dengan kombinasi matriks. Terdapat 3 karakter pertama yaitu CCC, maka C1 akan di ubah menggunakan kombinasi kunci C. Sedangkan C2 akan diubah menggunakan kombinasi kunci E dan C3 diubah menggunakan kunci G. Langkah - langkah perubahannya adalah sebagai berikut:

- Tangga nada kunci C = C -> C
- Tangga nada kunci E = Cari nada C berada pada not ke 17 yaitu F'
- Tangga nada kunci G = F' -> D#

  Dengan proses yang sama, maka akan didapatkan ciphertext baru, yaitu:
  "D#, G, G#, D#, G, D#, D#, G, G#, A#, C#, G#, F, E, G"

Langkah ketujuh adalah mengulangi proses pergeseran yang dilakukan pada langkah kelima. Maka hasilnya dapat dilihat pada tabel 6 dan 7.

**Tabel 6.** Translasi posisi kelompok nada

| **Kelompok 3** | **Kelompok 4** | **Kelompok 1** | **Kelompok 5** | **Kelompok 2** |
|---|---|---|---|---|
| D# G G# | A# C# G# | D# G G# | F' E G | D# G D# |

**Tabel 7.** Translasi nada pertama

| **Kelompok 1** | **Kelompok 2** | **Kelompok 3** |
|---|---|---|
| D# A# D# E' D# | G C# G E G | G# G# G# G D# |

Berdasarkan tabel 7, maka hasil akhir ciphertext telah didapat yaitu D#, A#, D#, E', D#, G, C#, G, E, G, G#, G#, G#, G, D#.

Langkah terakhir adalah dengan merubah ciphertext tersebut kedalam bentuk audio. Perubahan kedalam audio dilakukan berdasarkan tabel 1 frekuensi nada. Frekuensi nada tersebut kemudian akan diubah menjadi nada menggunakan rumus $A\sin(2\pi ft)$.

Adapun proses dekripsi yang dilakukan adalah sebagai berikut.

1. Input audio akan diidentifikasi frekuensi tiap nada kemudian dikonversi kedalam not dalam bentuk teks.
2. Ubah teks kunci kedalam matriks kunci.
3. Hasil langkah 1 kemudian dibagi menjadi 3 bagian. Buat kelompok baru berisi 3 huruf dari masing-masing bagian.
4. Translasi kelompok karakter sesuai dengan nilai random yang telah terismpan.
5. Translasi masing-masing karakter sesuai dengan matriks kunci
6. Ulangi langkah 3 dan 4.
7. Ubah tiap karakter sesuai nilai. Kalikan tiap kelompok karakter dengan invers matriks dan di modulo 12.
8. Ubah nilai hasil langkah 7 kedalam bentuk alfabet.

## KESIMPULAN

Berdasarkan penjelasan diatas, dapat disimpulkan bahwa algoritma yang ditawarkan berbeda dengan algoritma substitusi pada umumnya. Algoritma yang diajukan dapat melakukan enkripsi teks dan merubah kedalam bentuk audio dan dekripsi cipherteks audio kedalam teks. Kinerja algoritma berdasarkan pengujian menunjukkan bahwa proses enkripsi karakter memiliiki rata-rata waktu sebesar 3,72 detik. Sedangkan untuk proses dekripsi karakter memiliki waktu rata-rata sebesar 4,16 detik

Adapun kekurangan dari algoritma ini adalah karakter yang dapat dienkripsi terbatas. Algoritma ini hanya mampu melakukan enkripsi pada alfabet latin yaitu A-Z. Sedangkan karakter lain seperti tanda baca ataupun angka belum dapat dienkripsi oleh algoritma.

Pengembangan lebih lanjut dapat dilakukan penambahan karakter yang akan dienkripsi seperti tanda baca, angka, ataupun simbol lainnya. Penambahan karakter yang dapat digunakan berarti akan menambah jumlah chord, not, atau tangga nada yang digunakan pada algoritma enkripsi ataupun dekripsi.

## DAFTAR PUSTAKA

Dutta, S., Kumar, C., & Chakraborty, S. (2013). A symmetric key algorithm for cryptography using music. *International Journal of Engineering and Technology*, *5*(3), 3109–3115.

Green, R. (2014). *Music Theatre : Concepts , Theories & Practices Ryan Thomas Green*.

Hammood, D. A., Omran, S. S., & Al-Khalid, A. S. (2010). Using genetic algorithm to cryptanalyse a simple substitution cipher. *11th Middle Eastern Simulation Multiconference, MESM 2010 - 1st GAMEON-ARABIA Conference, GAMEON-ARABIA 2010*, *June*, 93–97.

Kumar, C., Dutta, S., & Chakraborty, S. (2015). A hybrid polybius-playfair music cipher. *International Journal of Multimedia and Ubiquitous Engineering*, *10*(8), 187–198. https://doi.org/10.14257/ijmue.2015.10.8.19

Lamaute, N., Piccoli, A., Chen, L., & Cotoranu, A. (2016). A Substitution Cipher for Musical Cryptography. *Proceedings of Student-Faculty Research Day*, 1–6.

Patil, S. N., Parab, D. H., Nambly, A., & John, L. M. (2017). Encrypting Messages Using Musical Notes By Genetic Algorithm. *International Journal of Research in Engineering Application and Management*, *3*(01), 1–4.

Raghav, A., & John, B. (2016). Musical Cryptography Using Multiple Note Substitution Algorithm. *Intenational Journal of Innovative Research in Science Engineering and Technology*, *5*(6), 9851–9860. https://doi.org/10.15680/IJIRSET.2016.0506046

Rao, D., & Koolagudi, S. (2019). Music cryptography based on carnatic music. *International Journal of Engineering and Advanced Technology*, *9*(1), 5107–5114. https://doi.org/10.35940/ijeat.A1358.109119

Singh, J., Shyam, P., & Yadav, S. (2014). *Implementation of Caesar Cipher and Chaotic Neural network by using MATLAB Simulator*. *2*(6), 16–20.

Stallings, W. (2017). *Cryptography and Network Secuirty*.

Surya, S., & H, M. I. (2017). Genetic Algorithm Based Cryptographic Approach Using Karnatic Music. *International Research Journal of Engineering and Technology(IRJET)*, *4*(6). https://irjet.net/archives/V4/i6/IRJET-V4I6614.pdf

Susanto, A., Dian, U., Semarang, N., Febrian, M. R., Dian, U., Semarang, N., Wm, I. U., Dian, U., & Semarang, N. (2020). *A Combination of Hill Cipher and LSB for Image Security A Combination of Hill Cipher and LSB for Image Security*. *June*. https://doi.org/10.15294/sji.v7i1.24393