**TUGAS KRIPTOGRAFI**



**Disusun oleh:**

Maya Setiana

205150707111021

Keamanan Jaringan

Teknologi Informasi - C

**UNIVERSITAS BRAWIJAYA**

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**MALANG**

**2022**

**BAB I**

**Pendahuluan**

* 1. **Latar Belakang**

Perkembangan teknologi informasi di era modern ini telah mengalami kemajuan yang sangat pesat. Teknologi informasi menawarkan kemudahan melalui berbagai bidang, salah satunya dalam penggunaan media komunikasi. Hal ini tentunya akan berdampak bagi keamanan informasi karena pesan akan menjadi sangat rentan untuk diakses oleh pihak yang tidak berkepentingan. Oleh karena itu teknologi informasi memiliki peranan dalam mengatasi masalah pada bidang keamanan informasi.

Keamanan jaringan merupakan hal yang penting untuk mengimbangi perkembangan teknologi informasi. Keamanan jaringan menjadi salah satu bagian dari sistem informasi yang berperan penting dalam menjaga integritas dan validitas data. Untuk menjaga integritas dan validitas data, proses pengiriman pesan harus memenuhi prinsip keamanan jaringan, yaitu *confidentiality, integrity,* dan *availaibility*. *Confidentiality* adalah suatu usaha pencegahan akan pengaksesan terhadap informasi yang dilakukan oleh pihak yang tidak berkepentingan. *Integrity* berkaitan dengan pencegahan akan modifikasi pada informasi yang dilakukan oleh pihak yang tidak berhak. *Availability* adalah usaha pencegahan akan penguasaan informasi atau sumber daya yang dilakukan oleh pihak yang tidak berhak. Terdapat berbagai usaha yang dilakukan untuk memenuhi prinsip keamanan jaringan dalam menjaga integritas dan validitas data, salah satunya dengan algoritma kriptografi.

Kriptografi berasal dari bahasa Yunani yang terdiri dari kata *cryptos* dan *graphien. Crypto* berarti rahasia dan *graphien* berarti tulisan. Menurut (Schneier, 1996) kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Sedangkan menurut Menezes, kriptografi adalah ilmu yang memperlajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi (Menezes et al., 1996). Kriptografi adalah ilmu mengenai proses enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi data yang sulit dibaca oleh seseorang yang tidak memiliki kunci deskripsi (Sentot Kromodimoeljo, 2010). Proses enkripsi adalah proses pengacakan atau penyandian pesan asli yang dapat dibaca (*plain text*) menjadi teks cipher yang tidak dapat dibaca (*cipher text*). Sedangkan proses mengembalikan teks ke bentuk semula yaitu menjadi pesan asli yang dapat dibaca disebut dengan proses dekripsi.

Berdasarkan kunci yang digunakan dalam proses enkripsi dan dekripsi, algoritma kriptografi dapat dibedakan menjadi algoritma simetri dan algoritma asimetri. Algoritma simetri dapat disebut dengan algoritma kriptografi klasik atau algoritma kriptografi konvensional. Hal ini dikarenakan kunci yang digunakan pada proses enkripsi dan dekripsi adalah sama (Hallim et al., 2010). Algoritma kriptografi klasik terdiri dari beberapa teknik, seperti teknik substitusi, teknik permutasi, teknik *blocking,* dan teknik penempatan. Di dalam teknik substitusi juga   terdapat beberapa macam cara, yaitu *Caesar cipher, Playfair Cipher, Hill Cipher,* dan *Vigenere Cipher .* Pada laporan ini, akan dijelaskan secara lebih lanjut mengenai analisis algoritma *Caesar cipher* dengan javascript..

* 1. **Permasalahan**

*Caesar cipher* adalah sistem persandian klasik yang berbasis algoritma substitusi yang digunakan oleh kaisar Romawi untuk bertukar pesan kepada gubernurnya.. Teknik penyandian yang sederhana pada *caesar cipher* yaitu dengan teknik substitusi dan transposisi akan rentan terhadap ancaman penyerangan (Sholehah, 2022). Metode *brute force* dan statistika frekuensi huruf yang paling sering muncul dalam suatu kalimat adalah salah satu teknik yang sering digunakan untuk memperoleh pesan asli pada penggunaan *caesar cipher.* Berdasarkan penelitian yang berjudul *“Enhancing Security of Caesar cipher Using Divide And Conquer Approach”,* algoritma *caesar cipher*  mudah dipecahkan karena secara umum hanya dilakukan pergeseran tiga kali pada teknik substitusinya (Singh, n.d.). Proses modifikasi diperlukan agar algoritma *caesar cipher* tidak mudah dipecahkan oleh kriptanalis (Sholehah, 2022).

**BAB II**

**Pembahasan**

1. **Dasar Teori**

*Caesar cipher* merupakan teknik enkripsi substitusi yang pertama kali dikenal dan paling sederhana yang ditemukan oleh Julius Caesar (Siambaton, 2016). *Caesar cipher* merupakan algoritma yang digunakan oleh Julius Caesar untuk mengirimkan pesan pada gubernurnya, Marcus Cicero, pada tahun 50 SM. Julius Caesar mengkodekan pesannya dengan mengubah setiap karakter alfabet dalam pesannya menjadi tiga huruf yang terletak setelah pesan asli dalam urutan alfabet (Ariyus, 2008).Algoritma ini digunakan jauh sebelum sistem kriptografi *public key* ditemukan sehingga sangat sederhana dan terlalu mudah untuk dipecahkan. Rumus enkripsi dan dekripsi dari *Caesar cipher* adalah sebagai berikut (Rahim, 2016) :

Cipher text = E(P) = (Plain text + Key) mod 26

Plain text = D(C) = (Cipher text - Key) mod 26

Seiring dengan perkembangan teknologi, *caesar cipher* adalah salah satu jenis algoritma kriptografi klasik dengan menggunakan teknik substitusi yaitu dengan melakukan pergeseran terhadap semua karakter pada *plaintext* dengan menggunakan *key* yang sudah ditentukan. Dalam perkembangannya, algoritma *caesar cipher* tidak hanya mengandalkan pergeseran huruf, tetapi juga dapat diacak dan disubstitusikan dengan *plaintext.* Salah satu modifikasi dari algoritma *cipher text* adalah dengan mengganti modulus yang digunakan dari 26 menjadi 256 untuk dapat digunakan di semua karakter ASCII (Prayitno & Nurdin, 2017). Selain itu, ada juga yang melakukan perubahan dengan mengganti modulus menjadi 95 dikarenakan karakter ASCII yang digunakan adalah karakter indeks 32 sampai 126 (Utomo et al., 2019).

1. **Solusi**

Algoritma *caesar cipher* berikut dimodifikasi dengan sistem kriptografi public key, dimana pengguna akan diminta untuk memasukkan plaintext untuk di enkripsi dan key untuk mengkodekan plaintext tersebut. Jadi key pada algoritman berikut bersifat dinamis karena key pada proses pengodean akan bergantung pada key yang dimasukkan pengguna. Hal ini akan meminimalisir ancaman penyerangan dan kebocoran pesan oleh kriptanalis.

Proses enkripsi

|  |
| --- |
| function isUpperCase(inputKata) {    return inputKata === inputKata.toUpperCase();  }  function enkripsi (inputKata, key) {      let hasilEnkripsi = "";    for (let i = 0; i < inputKata.length; i++) {      if (isUpperCase(inputKata[i])) {        hasilEnkripsi += String.fromCharCode(          ((inputKata.charCodeAt(i) + key - 65) % 26) + 65        );      } else {        hasilEnkripsi += String.fromCharCode(          ((inputKata.charCodeAt(i) + key - 97) % 26) + 97        );      }    }    return hasilEnkripsi;  };  console.log(enkripsi('MESIR', 3)) |

Pada algoritma enkripsi, terdapat fungsi enkripsi dengan parameter inputKata dan Key sehingga pengguna akan diminta untuk memasukkan data berupa kata atau kalimat dan nilai key. Kemudian akan dilakukan proses *looping*, dimana terdapat percabangan yang akan menjalankan fungsi isUpperCase. Fungsi isUpperCase digunakan untuk melakukan pengecekan penggunaan huruf kapital pada setiap karakter pada inputKata. Jika huruf yang dimasukkan oleh pengguna adalah huruf kapital maka akan masuk ke proses percabangan yang pertama. Dimana pada variabel hasilEnkripsi akan dilakukan konversi string dari setiap index karakter pada inputKata menjadi kode. Kode dari setiap index ini akan ditambah dengan key yang dimasukkan pengguna, dikurangi 65, dimodulus 26, dan ditambah 65. Kemudian method String.fromCharCode akan mengembalikan nilai Unicodes dari hasil enkripsi menjadi format string untuk nantinya ditampilkan.

Namun, jika huruf yang dimasukkan pada inputData adalah huruf kecil, maka akan masuk ke proses percabangan berikutnya dimana variabel hasilEnkripsi akan melakukan konversi string dari setiap index karakter pada inputKata menjadi kode. Kode dari setiap index ini akan ditambah dengan key yang dimasukkan pengguna, dikurangi 97, dimodulus 26, dan ditambah 97. Kemudian method String.fromCharCode akan mengembalikan nilai Unicodes dari hasil enkripsi menjadi format string untuk nantinya ditampilkan.

Proses dekripsi

|  |
| --- |
| function isUpperCase(inputKata) {    return inputKata === inputKata.toUpperCase();  }  function dekripsi(inputKata, key) {    let hasilDekripsi = "";    for (let i = 0; i < inputKata.length; i++) {      if (isUpperCase(inputKata[i])) {        hasilDekripsi += String.fromCharCode(          ((inputKata.charCodeAt(i) - key - 65) % 26) + 65);      } else {        hasilDekripsi += String.fromCharCode(          ((inputKata.charCodeAt(i) - key - 97) % 26) + 97);      }    }    return hasilDekripsi;  }  console.log(dekripsi("PHVLU", 3)); |

Pada algoritma dekripsi, terdapat fungsi dekripsi dengan parameter inputKata dan Key sehingga pengguna akan diminta untuk memasukkan data berupa kata atau kalimat dan nilai key. Kemudian akan dilakukan proses *looping*, dimana terdapat percabangan yang akan menjalankan fungsi isUpperCase. Fungsi isUpperCase digunakan untuk melakukan pengecekan penggunaan huruf kapital pada setiap karakter pada inputKata. Jika huruf yang dimasukkan oleh pengguna adalah huruf kapital maka akan masuk ke proses percabangan yang pertama. Dimana pada variabel hasilDekripsi akan dilakukan konversi string dari setiap index karakter pada inputKata menjadi kode. Kode dari setiap index ini akan ditambah dengan key yang dimasukkan pengguna, dikurangi 65, dimodulus 26, dan ditambah 65. Kemudian method String.fromCharCode akan mengembalikan nilai Unicodes dari hasil dekripsi menjadi format string untuk nantinya ditampilkan.

Namun, jika huruf yang dimasukkan pada inputData adalah huruf kecil, maka akan masuk ke proses percabangan berikutnya dimana variabel hasilDekripsi akan melakukan konversi string dari setiap index karakter pada inputKata menjadi kode. Kode dari setiap index ini akan ditambah dengan key yang dimasukkan pengguna, dikurangi 97, dimodulus 26, dan ditambah 97. Kemudian method String.fromCharCode akan mengembalikan nilai Unicodes dari hasil dekripsi menjadi format string untuk nantinya ditampilkan.

**Daftar Pustaka**

Ariyus, D. (2008). *Pengantar Kriptografi dan Implementasi*. 16. http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/Pengantar Kriptografi.pdf

Hallim, A., Nadhori, I. U., & Setiawardhana. (2010). *Pembuatan perangkat lunak media pembelajaran kriptografi klasik*. 1–11.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. *Handbook of Applied Cryptography*, 1–780. https://doi.org/10.2307/2589608

Prayitno, A., & Nurdin, N. (2017). Analisa Dan Implementasi Kriptografi Pada Pesan Rahasia. *Jurnal Elektronik Sistem Informasi Dan Komputer (JESIK)*, *3*(1), 1–11. nnurdin69@gmail.com

Rahim, R. (2016). Penyisipan Pesan Dengan Algoritma Pixel Value Differencing Dengan Algoritma Caesar Cipher Pada Proses Steganografi. *Times*, *5*(1), 6–11.

Schneier, B. (1996). Applied Cryptography. *Electrical Engineering*, *1*([32), 429–455. https://doi.org/10.1.1.99.2838

Sentot Kromodimoeljo. (2010). *Teori & Aplikasi Kriptografi*.

Sholehah, A. (2022). *MODIFIKASI CAESAR CIPHER UNTUK MENGHASILKAN READABLE CIPHERTXT*.

Siambaton, M. Z. (2016). Kombinasi Algoritma Pixel Value Differencing Dengan Algoritma Caesar Cipher Pada Proses Steganografi. *CESSJournal Of Computer Engineering, System And Science*, *1*(2), 19–25.

Singh, P. (n.d.). *Enhancing Security of Caesar Cipher Using Divide and Conquer Approach*. 144–150.

Utomo, I. W., Latifah, R., & Risanty, R. D. (2019). Aplikasi Kriptografi Berbasis Android Menggunakan Algoritma Caesar Cipher & Vigenere Cipher. *Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, *9*(2), 142–149.