

# Optimasi Kasiski Examination pada Studi Kasus SPOJ The Bytelandian Cryptographer (Act IV)

Freddy Hermawan Y., Rully Soelaiman dan Wijayanti Nurul Khotimah

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: freddy.yuwono@gmail.com, rully@is.its.ac.id, wijayanti@if.its.ac.id

**Abstrak**—Pada era digitalisasi ini, tingkat kebutuhan masyarakat akan informasi semakin meningkat. Hal ini menyebabkan pertukaran informasi menjadi sangat mudah. Hal ini membuat informasi yang bersifat sensitif dapat terjadi kebocoran informasi kepada pihak - pihak yang tidak berkepentingan. Kebocoran informasi terbagi menjadi dua apabila dilihat dari keutuhan informasi yang didapat, yaitu sebagian dan seutuhnya. Kebocoran informasi yang bersifat sebagian, membuat pihak-pihak yang tidak berkepentingan tetapi yang menginginkan informasi tersebut, berusaha untuk mendapatkan informasi yang utuh dari potongan-potongan informasi yang telah didapatkan.

Permasalahan dalam buku tugas akhir ini adalah permasalahan untuk mendapatkan plaintext sebanyak-banyaknya dari ciphertext dan batas atas panjang kunci pada metode enkripsi yang menggunakan teknik Vigenere Cipher. Dalam permasalahan ini diberikan plaintext dan ciphertext, akan tetapi terdapat informasi yang hilang pada keduanya. Diberikan batas atas panjang kunci, dimana batas atas ini belum tentu panjang kunci yang sesungguhnya. Untuk dapat merekonstruksi plaintext dari kepingan informasi yang didapatkan diperlukan untuk mencari panjang kunci yang di dapatkan dengan cara memodifikasi Kasiski Examination dan Intersection. Beberapa hal yang perlu diperhatikan seperti mempercepat dari Kasiski Examination dan juga Intersection terhadap hasil yang diperoleh dari pencarian panjang kunci.

Hasil dari metode ini telah berhasil untuk menyelesaikan permasalahan yang telah diangkat dengan benar. Waktu yang diperlukan untuk dapat menyelesaikan masukan sebesar 2MB rata-rata dalam 4,42 detik dengan alokasi memori sebesar 26,5MB.

**Kata Kunci**—Plaintext, Ciphertext, Kasiski Examination, Optimisasi.

## I. Pendahuluan

Ketergantungan seseorang terhadap informasi tidak terlepas dari kebutuhan manusia akan informasi yang berada disekitarnya. Informasi yang diterima seseorang pada masa sekarang dapat melalui media fisik dan media digital. Media fisik seperti koran dan majalah, sedangkan media digital seperti facebook dan twitter. Media-media tersebut sanggup untuk menyebarkan informasi sangat cepat, sehingga orang-orang dengan cepat mengetahui informasi yang berada disekitarnya.

Informasi digital yang beredar di dunia maya pun tidak lepas dari penyalahgunaan informasi. Dibutuhkan

suatu teknik penyandian terhadap data yang dimiliki agar data yang bersifat rahasia itu tidak diketahui dengan orang – orang yang tidak berkepentingan. Akan tetapi hal ini menarik perhatian dari pihak-pihak yang menginginkan informasi tersebut tetapi informasi yang diperoleh hanyalah terbatas dengan kepingan-kepingan data saja. Seperti contohnya adalah studi kasus SPOJ The Bytelandian Cryptographer(Act IV). Pada studi kasus ini diketahui bahwa metode enkripsi yang digunakan adalah Vigenere Cipher. Diberikan sejumlah kasus ujicoba, untuk setiap kasus ujicoba diberikan sejumlah potongan-potongan informasi dari plaintext dan ciphertext, batas atas dari panjang kunci yang digunakan(Panjang Kunci yang diberikan bukan panjang kunci yang sesungguhnya). Diharapkan dari studi kasus tersebut adalah merekonstruksi ulang plaintext dari apa yang telah tersedia. Batasan masalah jumlah inputan tidak akan melebihi dari 2 MB, batas atas panjang kuncinya bernilai  $1 \leq M \leq 100,000$ , dan jumlah kasus ujicoba  $1 \leq T \leq 200$ . Solusi dari studi kasus yang akan dibahas akan diimplementasikan dalam bentuk kode.

Hasil dari Tugas Akhir ini diharapkan dapat merekonstruksi ulang pesan dari kepingan-kepingan informasi yang telah didapatkan sebelumnya. Sehingga diharapkan memberikan kontribusi pada pengembangan ilmu pengetahuan dan komunikasi.

## II. Tinjauan Pustaka

### A. Kasiski Examination

Kasiski Examination adalah suatu teknik yang digunakan untuk mencari panjang kunci dari suatu ciphertext dari suatu Polyalphabetic Cipher dan turunannya, dengan memanfaatkan kelemahan dari Polyalphabetic Cipher, tanpa harus mengetahui plaintext dan himpunan kunci yang digunakan. Kelemahannya yaitu apabila suatu substring plaintext yang sama dienkripsi dengan substring dari kunci yang digunakan akan menghasilkan pola yang sama[1]. Metode pencarian panjang kunci yang dilakukan adalah sebagai berikut.

- 1) Mencari semua substring yang berulang pada suatu kalimat.
- 2) Mencari panjang dimana substring tersebut berulang kembali.

- 3) Mencari semua faktor dari nilai yang diperoleh dari tahap dua.
- 4) Mencari faktor persekutuan terbesar dari hasil yang diperoleh pada tahap tiga.

#### B. Intersection

Intersection adalah himpunan  $A$  dan himpunan  $B$ , dimana ada bagian dari  $A$  juga merupakan bagian dari  $B$ . [2].

### III. Metode Penyelesaian

Dalam bagian pendahuluan telah dijelaskan bahwa enkripsi yang digunakan adalah Vigenere Cipher. Dimana enkripsi yang dilakukan mengikuti aturan

$$y_i = x_i + k_{1+(i-1) \bmod M} \bmod 26 \quad (1)$$

Pada studi kasus inputan yang bernilai A sampai dengan Z, dapat direpresentasikan menjadi 0 sampai dengan 25. Tahapan-tahapan untuk menyelesaikan studi kasus ini sebagai berikut:

- 1) Menyimpan posisi indeks karakter, dimana pada indeks tersebut baik ciphertext maupun plaintext tidak bernilai "\*", beserta menyimpan hasil perhitungan selisih antara ciphertext dan plaintext.[3].
- 2) Menyimpan posisi indeks, apabila ciphertext diketahui dan plaintext tidak diketahui. Untuk mengurangi running time dari program[3].
- 3) Pada Tahapan ini adalah modifikasi dari Kasiski Examination, apabila menggunakan Kasiski Examination pada umumnya yang hanya mencari posisi berulang dari suatu sub kalimat dalam suatu kalimat tidak bisa digunakan untuk menyelesaikan permasalahan ini. Oleh karena itu dirubah menjadi mengiterasi  $M$  dari 1 sampai dengan  $M$ , yang akan digunakan untuk membagi selisih yang diketahui antara plaintext dan ciphertext sebesar posisi iterasi yang telah berjalan. Melihat apakah dalam blok-blok yang telah terbentuk ini terdapat collision dan indeks yang saling bertabrakan memiliki value yang sama atau tidak, apabila sama maka tidak terjadi tabrakan sebaliknya jika terjadi tabrakan maka harus mencari panjang kunci yang baru. Mengiterasi panjang kunci dari  $\frac{M}{2} + 1 \leq N \leq M$ , alasannya dimulai dari  $\frac{M}{2} + 1$  tidak dari 1 karena apabila suatu panjang kunci bernilai benar maka kelipatan dari panjang kunci itu pun juga pasti benar dan untuk mempersingkat waktu running time yang seharusnya terjadi. Pada bagian ini dilakukan untuk mencari panjang kunci yang benar dengan cara mengiterasi hasil yang diperoleh pada tahap satu dimodulo dengan posisi iterasi yang dilakukan, apabila tidak terjadi konflik maka panjang kunci tersebut benar jika sebaliknya yang terjadi maka panjang kunci tersebut tidak salah. Pada bagian ini memungkinkan bahwa bisa jadi lebih dari satu panjang kunci yang bernilai benar.
- 4) Melakukan intersection terhadap himpunan dari kunci yang telah di hasilkan pada tahapan

sebelumnya[3]. intersection yang dilakukan berada didalam perulangan panjang kunci pada waktu generate setiap karakter yang terdapat dalam penyimpanan tahap dua pada panjang kunci tersebut dengan ketentuan:

- a) Panjang kunci harus benar.
- b) Apabila terdapat indeks yang tidak dapat dipastikan isinya maka posisinya harus di buang dari penyimpanan dan hasilnya pasti "\*".
- c) Apabila plaintext bernilai "\*" dan himpunan kunci yang telah terbentuk tidak kosong pada indeks tersebut, maka plaintext akan bernilai sesuai dengan kunci yang terbentuk pada indeks tersebut.

### IV. Ujicoba dan Analisis

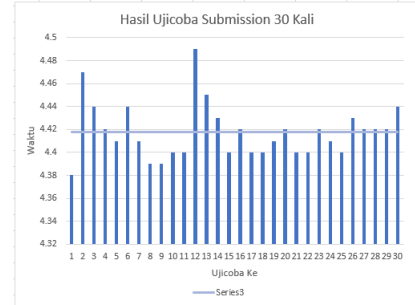
#### A. Uji coba Kebenaran

Ujicoba kebenaran dilakukan dengan cara mengumpulkan berkas kode implementasi kedalam daring penilaian online SPOJ. Studi kasus yang diselesaikan adalah The Bytelandian Cryptographer(Act IV) dengan code CRYPTO4. Hasil uji kebenaran dan waktu eksekusi program pada situs SPOJ ditunjukkan pada gambar



Gambar 1. Hasil uji kebenaran dengan melakukan submission ke situs penilaian daring SPOJ

Ujicoba Kinerja dari implementasi program yang dihasilkan dengan cara mengumpulkan berkas kode implementasi kedalam daring penilaian online SPOJ sebanyak 30 kali dengan mencatat waktu dan memori yang dibutuhkan, dapat dilihat pada gambar 2 dan table 1.



Gambar 2. Hasil Uji Coba Submission ke situs penilaian daring SPOJ sebanyak 30 kali

Waktu Maksimal	4,49 detik
Waktu Minimal	4,38 detik
Waktu Rata-Rata	4.418 detik
Memori Maksimal	27 MB
Memori Minimal	26 MB
Memori Rata-Rata	26.5 MB

Tabel 1. Kecepatan Maksimal, Minimal, dan Rata-Rata dari Hasil Uji Coba Pengumpulan 30 Kali pada Situs Pengujian Daring Spoj

## B. Analisa Kompleksitas

Berdasarkan algoritma yang telah dibentuk pada bagian metode penyelesaian maka, didapatkan algoritma dengan kompleksitas waktu  $\mathcal{O}(T * \frac{M}{2} * (N + S))$  Dimana  $n$  adalah panjang karakter plaintext atau ciphertext,  $\frac{M}{2}$  adalah batas atas kunci dibagi dengan 2,  $N$  adalah jumlah posisi karakter yang terdapat pada tahap 2 pada subbab ??, dan  $S$  adalah jumlah posisi karakter yang terdapat pada tahap 1 pada subbab ??. Pada kondisi worst case  $T * (N + S) = 1.000.000$ , sedangkan  $\frac{M}{2}$  adalah 50.000. Hasilnya 50 miliar perulangan, dengan asumsi 1 detik adalah 1 miliar perulangan maka waktu yang dibutuhkan adalah 50 detik. Oleh karena itu hal ini tidak mungkin bisa dilakukan begitu saja, diperlukan pruning pada sumber kode yang ada untuk memangkas waktu eksekusi program. Pruning yang dapat dilakukan ketiga menggunakan Kasiski Examination dan Intersection. Apabila semuanya ini telah dilakukan pasti mendapatkan seperti gambar 1 pada daring online SPOJ.

## V. Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap perancangan dan implementasi algoritma untuk menyelesaikan studi kasus SPOJ 20 The Bytelandian Cryptographer (Act IV) dapat diambil kesimpulan sebagai berikut:

- 1) Implementasi algoritma dengan menggunakan teknik Kasiski Examination dengan adanya optimasi tidak dapat menyelesaikan permasalahan SPOJ The Bytelandian Cryptographer (Act IV) dengan benar. Dengan adanya metode Intersection yang dilakukan setelah teknik Kasiski Examination dapat menyelesaikan studi kasus tersebut dengan benar.
- 2) Kompleksitas waktu  $\mathcal{O}(T * \frac{M}{2} * (N + S))$  masih dapat menyelesaikan permasalahan SPOJ The Bytelandian Cryptographer (Act IV)
- 3) Waktu yang dibutuhkan oleh program untuk menyelesaikan SPOJ The Bytelandian Cryptographer (Act IV) minimum 4,38 detik, maksimum 4,49 detik dan rata-rata 4.418 detik. Memori yang dibutuhkan berkisar antara 26-27 MB.

Saran-saran yang dapat diambil dari metode yang telah di bahas sebagai berikut:

- 1) Dengan teknik Kasiski Examination masih cenderung lambat, karena masih menggunakan teknik brute force sehingga hasil yang diperoleh kurang optimal, perlu adanya optimisasi lanjutan yang dapat mencari suatu panjang kunci.
- 2) Perlu adanya Optimisasi dalam hal pencarian suatu indeks yang perlu dirubah atau tidak. Dengan teknik yang dipakai oleh penulis tidak dapat memenuhi ekspektasi jika berharap dengan hasil yang sangat cepat.

## Ucapan Terima Kasih

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas pimpinan, penyertaan, dan karunia-Nya sehingga penulis dapat menyelesaikan penelitian ini. Penulis

juga mengucapkan terima kasih kepada orang tua dan keluarga penulis, juga kepada Bapak Rully Soelaiman dan Ibu Nurul Wijayanti K. selaku dosen pembimbing penulis dan kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama penulis mengerjakan penelitian ini.

## References

- [1] "Kasiski Method." [Online]. Available: <http://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/Vig-Kasiski.html>
- [2] K. Devlin, The Joy of Sets: Fundamentals of Contemporary Set Theory, 2nd ed. New York: Springer, 1993.
- [3] john\_jones, "SPOJ Discussion Board," 2009. [Online]. Available: <http://discuss.spoj.com/t/problemset-3/242/13>