Penerapan Algoritma *Backtracking* pada Proses Kriptanalisis terhadap Hasil Enkripsi *Vigenere Cipher* dengan Menggunakan Pendekatan *Dictionary Attack*

Joel THP Hutasoit[1]

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung

E-mail : if14144@students.if.itb.ac.id[1]

Abstrak

Kriptografi merupakan salah satu kajian dalam ilmu dan rekayasa komputasi yang melingkupi metode ataupun seni untuk menjaga kerahasian pesan (*plaintext*) dengan menyandikannya ke dalam bentuk yang tidak dimengerti lagi maknanya (*ciphertext*). Sejalan dengan perkembangan kriptografi, berkembang juga suatu seni ataupun ilmu yang melakukan eksplorasi dalam memecahkan *ciphertext* hasil proses kriptografi. Dalam proses kriptanalisis, hal utama yang dilakukan adalah berusaha memecahkan kunci yang dipakai selama proses kriptografi terhadap suatu *plaintext*. Salah satu pendekatan yang banyak digunakan adalah *dictionary attack* dimana selama pencarian kunci digunakan kamus sebagai sumber referensi. Masalah yang biasanya dihadapi dengan menggunakan pendekatan ini adalah lamanya proses yang dihadapi bila proses dilakukan secara beruntun terhadap setiap istilah dalam kamus. Pada makalah ini akan dibahas penggunaan salah satu strategi algoritmik di dalam membantu proses kriptanalisis yaitu algoritma *Backtracking*. Dengan penggunaan strategi ini maka diharapkan dapat mempercepat proses kriptanalisis dengan pendekatan *Dictionary Attack*.

Kata kunci : *kriptografi, kriptanalisis, vigenere cipher, dictionary attack, algoritma backtracking*

1. Pendahuluan

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dimengerti lagi maknanya. Dalam proses kriptografi terdapat 2 subjek yang terlibat yaitu pengirim serta penerima pesan dan juga terdapat 2 objek yang mengalami proses yaitu pesan asli (*plaintext*) dan pesan tersandi (*ciphertext*). Kriptografi terdiri atas 2 proses yang saling bertolakbelakang yaitu :

- Proses Enkripsi yaitu proses transformasi dari *plaintext* ke *ciphertext*.
- Proses Dekripsi yaitu proses transformasi *ciphertext* ke teks semula (*plaintext*).

Kriptografi diimplementasikan dalam berbagai metode. Salah satu metode yang ada adalah *vigenere cipher.* Metode ini merupakan kriptografi klasik karena masih beroperasi dalam karakter. Metode ini merupakan contoh terbaik dari *cipher* abjad-majemuk.

Metode *viginere cipher* menggunakan bujursangkar *vigenere* untuk melakukan enkripsi. Bujur sangkar tersebut menunjukkan huruf *ciphertext* dari huruf-huruf plainteks pada baris atas sesuai dengan huruf-huruf kunci pada kolom kiri. Huruf-huruf *ciphertext* tersebut diperoleh melalui pergeseran huruf-huruf *plaintext* sejauh nilai decimal dari huruf-huruf kunci.

Bujur sangkar *vigenere* digunakan untuk memperoleh *ciphertext* dengan menggunakan kunci yang ditentukan. Jika panjang kunci lebih pendek dari panjang *plaintext*, maka kunci diulang penggunaannya (sistem periodik). Berikut contoh proses enkripsi suatu teks dengan menggunakan kata kunci "BACK" :

Plaintext : THIS IS PLAINTEXT
Kunci : BACK BA CKBACKBAC
Ciphertext : UHKC JS RVBIPDFXV

Dari contoh di atas dapat dilihat bahwa *ciphertext* dihasilkan dari pergeseran huruf *plaintext* sejauh huruf kunci. Seperti huruf pertama "T" bergeser sejauh 1 ("B") menghasilkan huruf *ciphertext* "U". Proses dekripsi pada *vigenere cipher* merupakan kebalikan dari proses enkripsi, yaitu dengan mengurangkan huruf *ciphertext* sejauh nilai desimal dari huruf kunci.

Sejalan dengan perkembangan kriptografi, berkembang juga suatu seni ataupun ilmu yang melakukan eksplorasi dalam memecahkan *ciphertext* hasil proses kriptografi. Dalam proses kriptanalisis, hal utama yang dilakukan adalah berusaha memecahkan kunci yang dipakai selama

proses kriptografi terhadap suatu *plaintext*. Setelah kunci didapat maka *ciphertext* dapat dengan mudah dikembalikan ke bentuk *plaintext*-nya.

Salah satu pendekatan yang banyak digunakan adalah *dictionary attack* dimana selama pencarian kunci digunakan kamus sebagai sumber referensi. Pada dasarnya pendekatan ini lebih mengarah ke pendekatan *brute force attack* dengan penambahan referensi kamus sehingga tidak semua kombinasi karakter yang dicoba. Masalah yang biasanya dihadapi dengan menggunakan pendekatan ini adalah lamanya proses yang dihadapi bila proses dilakukan secara beruntun terhadap setiap istilah dalam kamus.

Untuk itulah pada makalah ini akan dibahas penggunaan salah satu strategi algoritmik di dalam membantu proses kriptanalisis yaitu algoritma *Backtracking*. Dengan penggunaan strategi ini maka diharapkan dapat mempercepat proses kriptanalisis dengan pendekatan *Dictionary Attack*.

2. Pemodelan Masalah

Masalah kriptanalisis dapat berlangsung dengan beragam kondisi seperti bahasa *plaintext* yang digunakan, apakah *plaintext* mengandung karakter selain huruf, dan sebagainya. Untuk menyederhanakan masalah maka dipilih suatu model yang dapat mewakili model lainnya secara umum dan sederhana.

Model yang dimaksud memiliki batasan (*constraint*) sebagai berikut :
- Bahasa yang digunakan *plaintext* maupun kunci adalah bahasa Indonesia
- *Plaintext* dan kunci tidak mengandung karakter selain huruf dan spasi. Karakter huruf yang dipakai hanyalah huruf kapital.
- Karakter spasi tidak ikut diproses (enkripsi maupun dekripsi).
- Ukuran *plaintext* tidak terlalu panjang, hanya dalam cakupan satu kalimat saja. Sedangkan ukuran kunci hanya mencakup satu kata saja.

Dari batasan-batasan di atas maka dapat dibuat suatu model masalah yang dijabarkan sebagai berikut :

Plaintext :
`USAHA KRIPTANALISIS DENGAN BANTUAN`
`ALGORITMA RUNUT BALIK`

Kunci :
`BARU`

Maka melalui proses enkripsi dengan menggunakan metode *vigenere cipher* dihasilkan *ciphertext* (cetak tebal) sebagai berikut :

```
USAHA KRIPTANALISIS DENGAN BANTUAN
BARUB ARUBARUBARUBA RUBARU BARUBAR
VSRBB KICQTRHBLZMJS UYOGRH CAENVAE
```

```
ALGORITMA RUNUT BALIK
UBARUBARU BARUB ARUBA
UMGFLJTDU SUEOU BRFJK
```

Selain itu, sesuai dengan deskripsi masalah maka pada model yang dibuat juga ditambahkan satu referensi kamus yang menjadi bahan referensi selama proses kriptanalisis dengan algoritma *backtrack* berlangsung.

Dari paparan di atas maka masalah yang harus diselesaikan adalah pencarian kunci dari *ciphertext* dengan menggunakan algoritma *backtrack*.

3. Algoritma Backtrack

Backtrack (runtu balik) adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih mangkus.

Runut-balik, yang merupakan perbaikan dari algoritma *brute-force*, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada.

Dengan metode runut-balik, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah ke solusi saja yang selalu dipertimbangkan. Akibatnya, waktu pencarian dapat dihemat.

Runut-balik merupakan bentuk tipikal dari algoritma rekursif.

Saat ini algoritma runut-balik banyak diterapkan untuk program *games* (seperti permainan *tic-tac-toe*, menemukan jalan keluar dalam sebuah labirin, catur, dll) dan masalah-masalah pada bidang kecerdasan buatan (*artificial intelligence*).

Adapun skema umum dari metode *backtracking* (runut-balik) ini antara lain :

1. Solusi persoalan.
   Solusi dinyatakan sebagai vektor dengan *n-tuple*:

   $X = (x_1, x_2, \ldots, x_n)$, $x_i \in$ himpunan berhingga S_i.

   Mungkin saja $S_1 = S_2 = \ldots = S_n$.

   Contoh: $S_i = \{0, 1\}$,
   $x_i = 0$ atau 1

2. Fungsi pembangkit nilai x_k
   Dinyatakan sebagai:

   $T(k)$

$T(k)$ membangkitkan nilai untuk x_k, yang merupakan komponen vektor solusi.

3. Fungsi pembatas (pada beberapa persoalan fungsi ini dinamakan fungsi kriteria)
   Dinyatakan sebagai

   $B(x_1, x_2, \ldots, x_k)$

   Fungsi pembatas menentukan apakah $(x_1, x_2, \ldots, x_k)$ mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk x_{k+1} dilanjutkan, tetapi jika tidak, maka $(x_1, x_2, \ldots, x_k)$ dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

4. Analisis Model Masalah

Pada model masalah yang sudah digambarkan pada bagian sebelumnya, maka dapat dirancang suatu model yang terdiri atas 2 komponen, yaitu :

- Pembangkit Kunci Sementara *GKs*, yang berfungsi untuk membangkitkan kunci sementara *Ks* dari referensi kamus yang ada untuk digunakan pada proses kriptanalisis.
- Pembangkit Plainteks Sementara *GPs*, yang berfungsi untuk membangkitkan plainteks sementara *Ps* sepanjang <= *n* dengan menggunakan *Ks* yang ada. *n* disini adalah panjang dari *Ks*. Panjang *Ps* yang akan dihasilkan bersesuaian dengan sejauh mana dekripsi karakter demi karakter dengan menggunakan *Ks* dapat menghasilkan teks yang sesuai dengan sebagian atau seluruh karakter dari kata yang terdapat pada referensi kamus.

Kedua komponen diatas nantinya akan berkolaborasi dalam menentukan kunci berikutnya yang akan dibangkitkan dan dicoba. Jika panjang *n* dari *Ps* sama dengan panjang *Ks* dan keseluruhan *Ps* membentuk kata yang terdapat pada referensi kamus maka *Ks* memiliki potensi menjadi kunci *K* yang diharapkan. Untuk memastikan *Ks* sebagai *K*, maka perlu dilakukan pengujian pada periodik berikutnya dari cipherteks *C* jika (panjang *C* > panjang *Ks*). Jika terdapat kondisi dimana (panjang *C* = panjang *Ks*) maka *Ks* merupakan kunci *K* yang dicari.

5. Penerapan Algoritma *Backtracking*

Dengan memperhatikan hasil analisis model masalah yang telah dikemukakan sebelumnya, maka kita dapat menentukan elemen-elemen persoalan yang bersesuaian dengan skema umum algoritma *backtracking*, yaitu :

1. Solusi persoalan.
   Solusi dari permasalahan ini adalah suatu vektor dengan *n-tuple* :

   $X = (x_1, x_2, \ldots, x_n)$, dimana

   $x_i \in$ himpunan berhingga S_i

   S_i disini adalah himpunan huruf-huruf kapital dari abjad latin (A, B, C, ..., Y, Z).

   X dinyatakan solusi jika kombinasi elemen-elemen *X* secara berurutan membentuk kata yang ada ataupun bersesuaian dengan referensi kamus.

   Sesuai dengan model masalah maka solusi yang diharapkan adalah :

   $X = \{B, A, R, U\}$

2. Fungsi pembangkit nilai x_k
   Dinyatakan sebagai:

   $T(k)$

   $T(k)$ membangkitkan nilai untuk x_k, yang merupakan komponen vektor solusi.

   Fungsi pembangkit disini didasarkan pada referensi kamus yang ada. Sesuai dengan analisis model masalah sebelumnya maka fungsi ini diperankan oleh *GKs*.

3. Fungsi pembatas
   Dinyatakan sebagai

   $B(x_1, x_2, \ldots, x_k)$

   Fungsi pembatas menentukan apakah $(x_1, x_2, \ldots, x_k)$ hasil dari fungsi pembangkit mengarah ke solusi. Mengarah ke solusi disini adalah bahwa teks plainteks yang dihasilkan bersesuaian dengan sebagian atau seluruh kata yang terdapat dalam referensi kamus. Jika ya, maka pembangkitan nilai untuk x_{k+1} dilanjutkan, tetapi jika tidak, maka $(x_1, x_2, \ldots, x_k)$ dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

Rangkaian langkah-langkah algoritma *backtracking* untuk menyelesaikan permasalahan ini yang bersesuaian dengan skema umum algoritma *backtracking* dapat dijabarkan sebagai berikut :

1. Inisialisasi vektor solusi X dan plainteks sementara *Ps* menjadi himpunan kosong.
2. Inisialisasi *Tail T* dengan 0 sebagai posisi indeks X saat ini (indeks dimulai dari 1).
3. Mulai bangkitkan elemen ke (*T*+1) dari vektor X sesuai dengan karakter ke (*T*+1) pada istilah dari referensi kamus dimana karakter ke-*n* (1<=*n*<=*T*) bersesuaian dengan vektor X. Pastikan bahwa karakter-karakter dalam X setelah pembangkitan belum pernah digunakan sebelumnya. Setelah itu lakukan juga *increment* terhadap nilai *T*.
4. Lakukan proses dekripsi pada karakter ke-*n* dari cipherteks *C* dengan menggunakan karakter ke-*n* dari X. Simpan hasil dekripsi ke plainteks sementara *Ps*.
5. Periksa apakah *Ps* merupakan subset atau keseluruhan dari kata yang terdapat dalam referensi kamus.

6. Jika ya dan *Ps* merupakan keseluruhan kata dalam kamus maka periksa apakah seluruh C sudah diproses.
   6.1. Jika sudah maka X merupakan solusi.
   6.2. Jika belum maka periksa apakah X sejauh ini sudah merupakan kata yang terdapat dalam kamus.
      6.2.1. Jika sudah maka X merupakan solusi.
      6.2.2. Jika belum maka kembali ke langkah-3
7. Jika ya dan *Ps* merupakan subset dari kata dalam kamus maka periksa apakah seluruh C sudah diproses.
   7.1. Jika sudah maka X bukan merupakan solusi, lakukan *backtrack* ke komponen vektor X sebelumnya (*T*-1) dan *decrement* nilai *T*. Kemudian kembali ke langkah 3.
   7.2. Jika belum maka periksa apakah X sejauh ini sudah merupakan kata yang terdapat dalam kamus.
      7.2.1. Jika sudah maka kembali ke langkah-4.
      7.2.2. Jika belum maka kembali ke langkah-3
8. Jika tidak maka X bukan merupakan solusi, lakukan *backtrack* ke komponen vektor X sebelumnya (*T*-1) dan *decrement* nilai *T*. Kemudian kembali ke langkah 3.
9. Lakukan langkah-langkah di atas hingga ada ditemukan vektor X yang bersesuaian dengan kesuluruhan kata yang terdapat pada referensi kamus.
10. Nilai-nilai yang tersimpan pada X merupakan solusi ataupun kunci *K* yang dicari dari permasalahan ini. Kunci *K* inilah yang selanjutnya akan dipakai untuk mendekripsi cipherteks *C*.

Pada implementasinya pada program ada baiknya proses ini menggunakan lebih dari satu *thread* (*multithreading*) dimana setiap *thread* menjalankan proses kriptanalisis dengan referensi kamus lokal merupakan subset dari referensi kamus global yang sebenarnya. Hal ini tentunya akan mempercepat proses kriptanalisis karena menggunakan pemrosesan konkuren.

6. Kesimpulan dan Saran

Masalah kriptanalisis terhadap hasil enkripsi dari *vigenere cipher* bergantung pada pemilihan kunci yang digunakan pada proses enkripsi. Jika kuncinya panjang maka proses yang dibutuhkan akan menjadi lama. Sebaliknya jika kuncinya pendek maka proses yang dibutuhkan dapat lebih cepat.

Algoritma *backtracking* (runut balik) dalam hal ini dapat dijadikan sebagai salah satu solusi dalam melakukan kriptanalisis dengan pendekatan *dictionary attack*. Hal ini lebih baik daripada penggunaan metode *brute force* yang memproses seluruh karakter dari seluruh kata dalam kamus di dalam mencari kunci yang sesuai.

Dalam proses kriptanalisis sering kali juga ditemukan keadaan dimana adanya suatu kunci *K* yang sudah di dapat hanya bersesuaian dengan sebagian cipherteks saja. Sedangkan cipherteks selanjutnya tidak bersesuaian dengan kunci *K*. Keadaan tersebut sejauh ini belum ditangani dan dapat dijadikan bahan pertimbangan untuk pengembangan ke depannya.

7. Daftar Pustaka

1. Munir, Rinaldi. 2005. *"Strategi Algoritmik"*. Departemen Teknik Informatika, Institut Teknologi Bandung
2. Munir, Rinaldi. 2006. *"Kriptografi"*. Departemen Teknik Informatika, Institut Teknologi Bandung