Penerapan Algoritma Brute Force pada Password Recovery

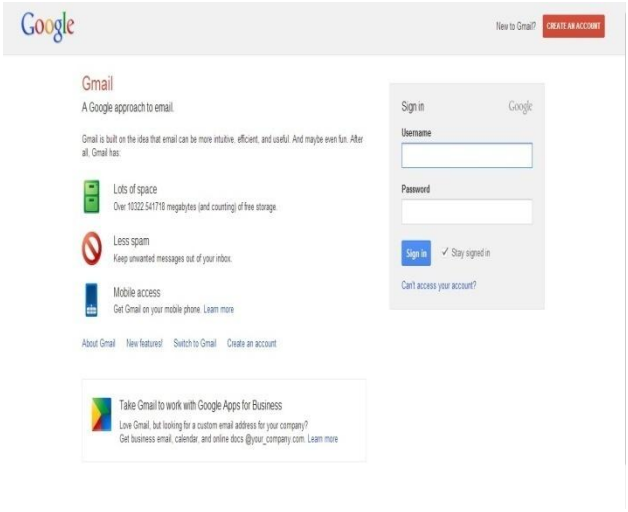
Johannes Ridho Tumpuan Parlindungan 13510103
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13510103@std.stei.itb.ac.id

***Abstrak*—Makalah ini berisi tentang penggunaan algoritma brute force pada password recovery. Password recovery adalah pemulihan password pada suatu sistem dengan metode tertentu. Salah satu metode yang sering dipakai pada proses ini adalah brute force attack. Dengan menggunakan brute force, semua kemungkinan yang mungkin akan dicoba untuk memperoleh password yang diinginkan.**

***Kata kunci*—password recovery, brute force, brute force attack.**

I. PENDAHULUAN

Penggunaan *password* tentu sudah menjadi hal biasa dalam kehidupan manusia zaman sekarang. Ketika kita ingin menggunakan e-mail, jejaring sosial, atau untuk melakukan login ke akun kita yang berada di website-website tertentu, kita memerlukan sebuah *password*. Bahkan alat komunikasi kita pun dapat diproteksi dengan sebuah *password*.



Gambar 1 Contoh Penggunaan Password [4]

Manfaat utama adanya *password* tentu adalah untuk autentikasi dan keamanan. Memiliki *password* suatu sistem berarti kita memiliki hak untuk menggunakan sistem tersebut. Seandainya tidak ada *password*, data pribadi kita tentu tidak aman. Akan sangat berbahaya jika ada orang yang tidak bertanggung jawab yang dapat dengan seenaknya membuka dan membaca email kita.

Tetapi ada masalah lain, bagaimana jika kita melupakan *password* yang kita miliki? Baik kita maupun orang lain tidak akan bisa mengakses dan menggunakan sistem. Ketika masalah itu terjadi, saat itulah kita perlu menggunakan metode *password recovery*.

II. ALGORITMA BRUTE FORCE

Brute *force* adalah pendekatan yang lempang (straightforward) untuk memecahkan suatu masalah. Istilah *brute force* dipopulerkan oleh Kenneth Thompson dengan mottonya "When in doubt, use *brute force*". Algoritma ini sangat tidak mangkus sebab menggunakan banyak langkah untuk memperoleh solusi. Tetapi algoritma ini akan memberikan solusi yang paling optimal sebab semua kemungkinan yang ada akan dicoba satu per satu. Oleh karena itu, algoritma ini sangat cocok untuk menyelesaikan masalah yang berukuran kecil. Selain itu, *brute force* adalah algoritma yang sederhana. Algoritma *brute force* juga dapat menyelesaikan hampir semua masalah, meskipun waktu pencariannya mungkin akan memakan waktu. *Brute force* sangat mudah untuk diimplementasikan pada berbagai jenis masalah.

Keunggulan metode *brute force* adalah [1]:

1. Metode ini dapat menyelesaikan banyak masalah.
2. Metode ini sederhana dan mudah dimengerti.
3. Metode ini mampu menghasilkan algoritma yang layak untuk beberapa masalah penting seperti pencarian, pengurutan, pencocokan *string*, perkalian matriks.
4. Metode ini dapat menghasilkan agoritma baku untuk komputasi penjumlahan n bilangan atau menentukan elemen maksimum dan minimum suatu tabel.

Kelemahan metode *brute force* adalah [1]:

1. Metode ini jarang menghasilkan algoritma yang mangkus/efisien.
2. Metode ini masih lambat dalam beberapa masalah sehingga tidak dapat diterima.
3. Metode ini terlalu sederhana dan tidak sekreatif/sekonstruktif algoritma lain.

Brute *force* attack adalah serangan terhadap suatu data yang ter-enkripsi dengan menggunakan algoritma *brute force*. Brute *force* attack akan melakukan pengecekan semua *key* secara sistematis sampai *key* yang tepat ditemukan.

III. PASSWORD RECOVERY

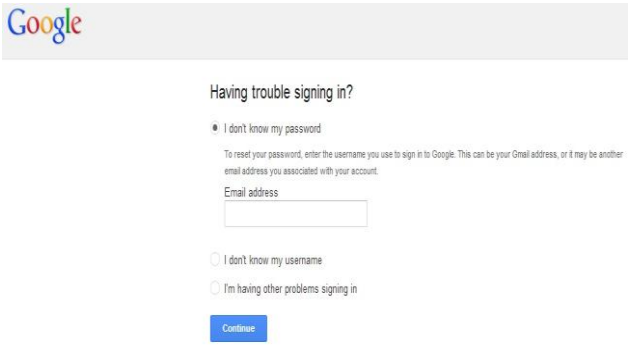
Password *recovery* adalah proses pemulihan *password* pada suatu sistem dengan metode tertentu. Algoritma yang sering digunakan pada proses ini adalah algoritma *brute force*. Pencocokan dengan *password* yang tersimpan pada sistem akan dilakukan satu per satu.

Waktu pemulihan *password* sangat bergantung pada pada panjang karakter pada *password* tersebut. Password dengan panjang satu karakter dapat dipecahkan dalam hitungan detik, tetapi untuk *password* dengan karakter yang sangat panjang akan memerlukan waktu pemecahhan yang sangat lama, atau bahkan bertahun-tahun. Penyebabnya adalah algoritma *brute force* melakukan pencarian terhadap semua kemungkinan karakter.

Gambar 2 Salah Satu Program Password Recovery yang cukup Terkenal, Hashcat [5]

Terdapat berbagai program *recovery* untuk melakukan proses *recovery password*. Program-program ini dapat bekerja dengan maksimal jika didukung oleh perangkat keras yang maksimal juga. Baru-baru ini ditemukan bahwa dengan sebuah komputer cluster yang terdiri dari 25 kartu pengolah grafis (GPU, Graphics Processing Unit) dapat melakukan *brute force* sebanyak 350 miliar kali tiap 1 detik. Dengan kata lain, komputer ini dapat menemukan solusi passsword 8 karakter hanya dalam waktu 5,5 jam. Tetapi untuk kombinasi dengan jumlah karakter yang lebih banyak, komputer ini masih memerlukan waktu yang cukup lama. Untuk *password* 9 karakter diperlukan waktu 500 jam, sedangkan untuk *password* dengan 10 karakter diperlukan waktu sekitar 5,4 tahun [3].

Selain dengan menggunakan algoritma *brute force*, cara yang sering digunakan pada pemulihan *password* adalah melakukan reset *password*. Metode ini cukup sering ditemukan pada website-website pada umumnya. Beberapa contohnya adalah www.google.com, www.yahoo.com, dan berbagai website lainnya. Ketika kita memilih untuk menggunakan metode ini, sistem akan meminta kita untuk melakukan verifikasi data pribadi untuk membuktikan bahwa kitalah yang memiliki suatu akun. Beberapa website juga dapat mengirimkan *password* baru ke email lain yang sebelumnya telah didaftarkan.



Gambar 3 Reset Password pada Google [4]

IV. PASSWORD RECOVERY DENGAN BRUTE FORCE

Pada makalah ini saya menggunakan sebuah aplikasi *brute force recovery* yang dapat digunakan pada *file* terkompresi seperti RAR dan ZIP. Aplikasi ini bernama Daossoft RAR Password Recovery. Sebenarnya terdapat beberapa cara *recovery* yang ditawarkan oleh aplikasi ini. Salah satunya adalah dictionary. Metode ini memiliki cara yang hampir sama dengan *brute force* hanya saja metode ini memanfaatkan sebuah kamus untuk memecahkan *password*. Tentu saja metode ini tidak efektif jika frasa atau kata pada *password* yang dipakai tidak ada di dalam kamus.

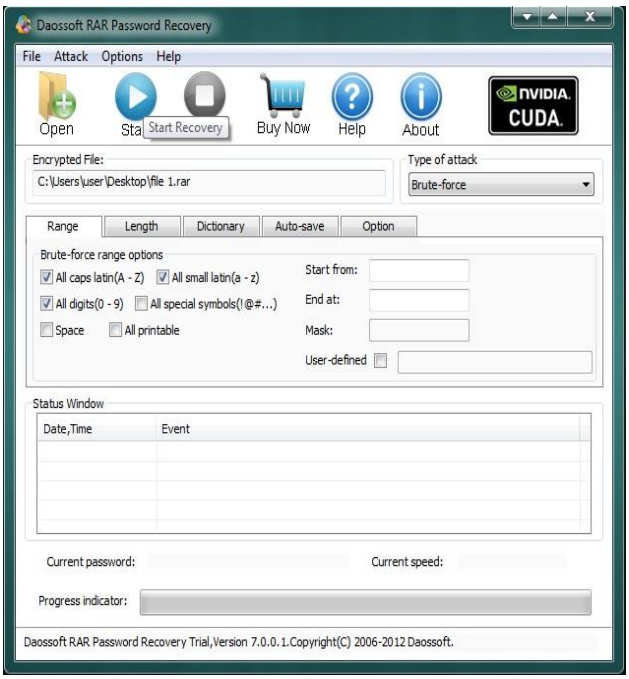
Gambar 4 Daossoft RAR Password Recovery [6]

Selain itu ada juga metode mask. Metode ini berguna jika kita mengingat bagian dari *password file*. Misalnya kita ingat *password* suatu *file* diawali dengan karakter "A", dan diakhiri dengan karakter "o". Kita juga mengingat jumlah karakternya adalah 5. Dengan kondisi ini masknya adalah A???o. Karakter "?" di sini mewakili karakter yang tidak kita ketahui. Dengan metode ini, tentu

akan mempersingkat waktu pemulihan *password*.

Metode *brute force* pada aplikasi ini akan mencari semua kemungkinan kata atau frasa sampai akhirnya ditemukan yang sesuai dengan *password* dari *file* yang telah dipilih. Caranya sederhana, yaitu dengan melakukan pencocokan dimulai dari sebuah karakter yang merupakan huruf kapital, kemudian huruf kecil, lalu angka. Pencocokan akan dilakukan dari karakter "A", lalu "B", "C", dan seterusnya sampai "9". Jika semua karakter telah dicoba, aplikasi akan melakukan pencocokan dengan memakai 2 buah karakter. Kedua karakter tersebut dimulai dari "AA", lalu "AB", "AC", dan seterusnya. Jika telah sampai ke "AZ", pencocokan akan dilakukan lagi dengan "Aa", lalu "Ab", "Ac", dan seterusnya. Setelah mencapai "Az", akan dilakukan pencocokan dengan "A0", "A1", "A2", dan seterusnya. Kemudian ini akan terus berlanjut sampai akhirnya dicoba "99". Lalu akan dilanjutkan dengan pencocokan 3 buah karakter, dan seterusnya. Sebenarnya aplikasi ini bisa melakukan pencocokan juga dengan karakter simbol-simbol dan bahkan spasi. Tetapi untuk mempersingkat waktu pencarian, saya hanya menggunakan karakter huruf kapital, huruf kecil, dan angka.

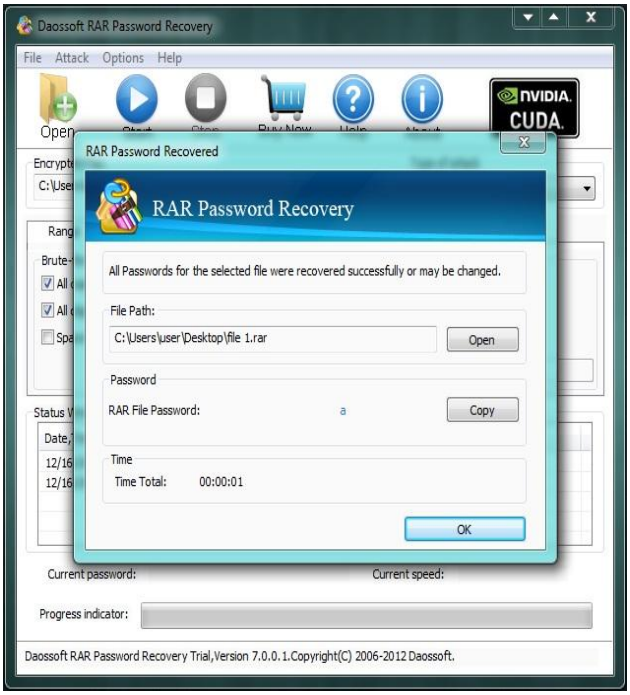
Di sini saya membuat sebuah *file* RAR bernama file 1.rar yang diberi *password* a. Kemudian dengan program *recovery*, saya mencoba membongkar *password*-nya.



Gambar 5 Tampilan Awal Program

Kemudian program dijalankan. Selama program mencari *password*, akan ditampilkan *current password*, sehingga kita dapat melihat kombinasi string apa saja yang sedang dicocokkan. Selain itu, ditampilkan juga current speed yang menunjukkan kecepatan pencarian saat itu. Kecepatan ini sangat bergantung pada kemampuan perangkat keras yang dipakai. Semakin bagus perangkat keras yang kita pakai, semakin cepat kecepatannya. Pada *status window* kita dapat melihat *log* kegiatan yang telah dilakukan. Di sini akan ditampilkan berbagai *event* seperti *autosave*, atau ketika aplikasi telah menemukan *password* suatu *file*, hasilnya akan dituliskan di *status window*.

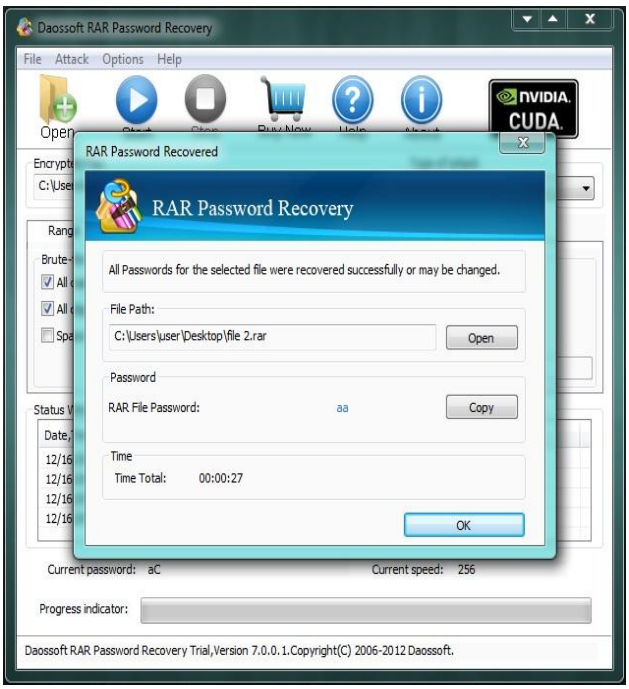
Setelah *password* ditemukan, program akan menampilkan layar baru. Berikut tampilannya.



Gambar 6 Tampilan setelah Password pada file 1.rar Ditemukan

Dari gambar di atas dapat kita lihat, untuk memperoleh *password* "a" hanya perlu waktu 1 detik. Hal ini tentu sangat wajar sebab *password* hanya 1 karakter. Aplikasi akan melakukan pencocokan dimulai dari karakter "A" sampai karakter "a". Dapat kita lihat bahwa karakter "a" adalah karakter ke 27 yang dicoba.

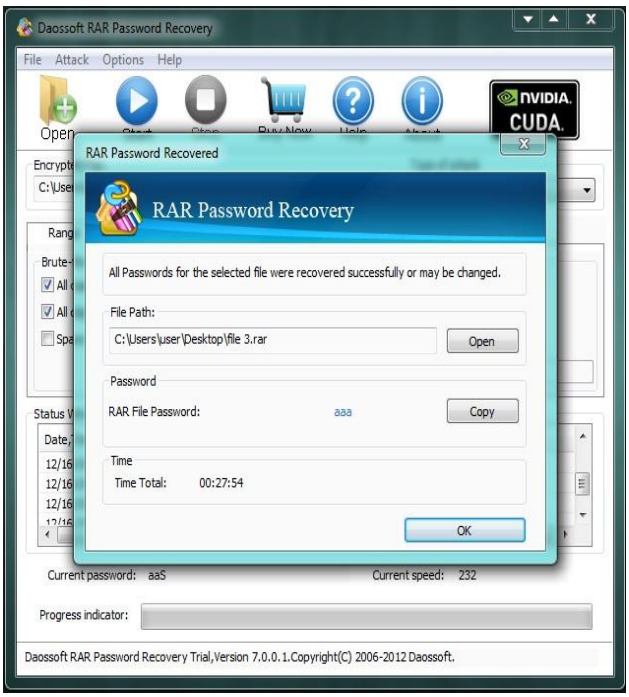
Kemudian saya membuat sebuah *file* bernama file 2.rar dengan *password* "aa". Lalu saya menjalankan program dan kemudian program berhasil menemukan *password*nya.



Gambar 7 Tampilan setelah Password pada file 2.rar Ditemukan

Waktu yang diperlukan untuk memecahkan *password* "aa" adalah 27 detik. Dapat kita lihat, hanya dengan menambah sebuah karakter saja lonjakkannya menjadi 27 kali lipat.

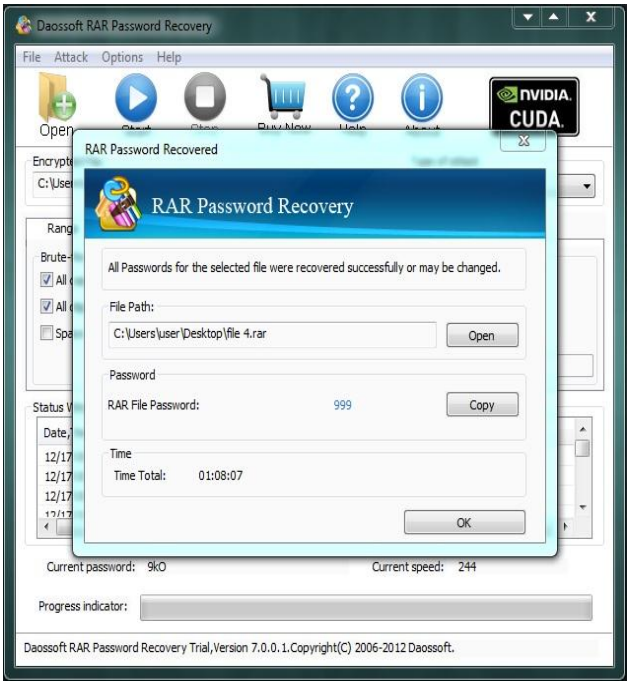
Selanjutnya saya membuat file 3.rar dengan *password* "aaa". Kemudian program kembali dijalankan dan diperoleh hasil berikut.



Gambar 8 Tampilan setelah Password pada file 3.rar Ditemukan

Waktu yang diperlukan untuk menemukan *password* "aaa" adalah 27 menit 54 detik, jauh lebih lama dari *password* sebelumnya.

Kemudian saya kembali membuat sebuah *file* rar dengan nama file 4.rar. Password yang saya gunakan untuk *file* ini adalah "999". Dengan menggunakan karakter "999" ini, saya dapat mengetahui kemungkinan terburuk pada percobaan dengan 3 buah karakter.



Gambar 9 Tampilan setelah Password pada file 4.rar Ditemukan

Dapat kita lihat, untuk memulihkan *password* dengan karakter "999" diperlukan waktu selama 1 jam 8 menit 7 detik. Perbedaannya dengan pemulihan *password* dengan *password* "aaa" cukup besar yaitu dua kali lipatnya. Hal ini terjadi karena string "999" merupakan **worst case** dari *recovery password* dengan string tiga digit. Untuk menemukan string ini diperlukan pengecekan sampai kemungkinan terakhir. Oleh karena itu jumlah pengecekan yang dilakukan paling besar. Selain itu waktu pencarian solusi juga paling lama. Sementara "aaa" dapat dicapai tanpa perlu mengecek sampai kemungkinan terakhir. Oleh karena itu waktu pencarian solusinya hanya 27 menit 54 detik.

Password	Waktu Pencarian (s)
a	1
aa	27
aaa	1674
999	4087

Tabel 1 Perbandingan Waktu Pencarian

Dari percobaan di atas, dapat kita lihat bahwa algoritma *brute force* memang pasti menemukan *password* yang kita cari. Tetapi waktu yang diperlukan akan sangat lama terutama untuk *password* yang sangat panjang. Hanya untuk memperoleh *password* dengan jumlah karakter 3 saja memerlukan waktu sekitar satu jam. Padahal jumlah kemungkinan kata maksimal pada *password* 3 karakter hanya 242.234 kemungkinan. Belum lagi untuk melakukan percobaan pada *password* yang karakternya jauh lebih banyak. *Recovery password* dengan metode ini sangat tidak efisien.

V. KESIMPULAN

1. Algoritma *brute force* dapat digunakan untuk melakukan recovery *password*.
2. Algoritma *brute force* masih kurang mangkus untuk *recovery password*.
3. Semakin panjang karakter suatu *password*, semakin lama proses pemulihan dilakukan. Bahkan perbedaan 1 karakter saja dapat membuat perbedaan waktu puluhan kali lipat.

REFERENCES

[1] Munir, Rinaldi, 2009, *Diktat Kuliah IF3051 Strategi Algoritma*, Program Studi Teknik Informatika ITB

[2] www.informatika.org/~rinaldi

[3] http://tekno.kompas.com/read/xml/2012/12/13/09250070/Digempu D.25.Kartu.Grafis..Password.Kuat.Windows.Bobol

[4] www.mail.google.com

[5] hashcat.net/oclhashcat-plus

[6] http://download.cnet.com/RAR-Password-Recovery/3000-18501_4-75609764.html

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 20 Desember 2012

Johannes Ridho T. P.
13510103