

Laporan Proyek Algoritma dan Pemrograman

Enigmoo – Word Encryption and Decryption



Disusun oleh:

Ida Bagus Krishna Yoga Utama (1506716983)

Nurian Satya Wardana (1506717071)

**DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNIK UNIVERSITAS INDONESIA**

2017

1. Tujuan Program

Dewasa ini semakin banyak kejahatan digital yang terjadi. Contohnya adalah penyadapan pesan, baik email, sms, maupun pesan WhatsApp atau Line, kejahatan digital lain yang terjadi adalah pencurian password dan pencurian notes penting. Maka dari itu, keamanan atas suatu data menjadi hal yang penting karena data-data tersebut merupakan hal yang penting dan bersifat pribadi, dimana tidak boleh ada orang lain yang tahu selain pemilik data. Salah satu cara untuk menghindari tindak pencurian data atau penyadapan pesan adalah dengan mengaplikasikan metode enkripsi dan dekripsi pada saat transfer data dan penyimpanan data. Dengan dilakukannya enkripsi dan dekripsi terhadap data-data tersebut, orang yang tidak memiliki akses terhadap data tersebut tidak akan dapat membaca isi dari data tersebut karena saat proses enkripsi data dilakukan perubahan isi data menjadi bentuk tertentu sehingga menjadi data yang tidak terbaca. Agar dapat membaca isi data yang telah dienkripsi, digunakan metode dekripsi untuk mengubah data yang telah terenkripsi menjadi isi data yang asli.

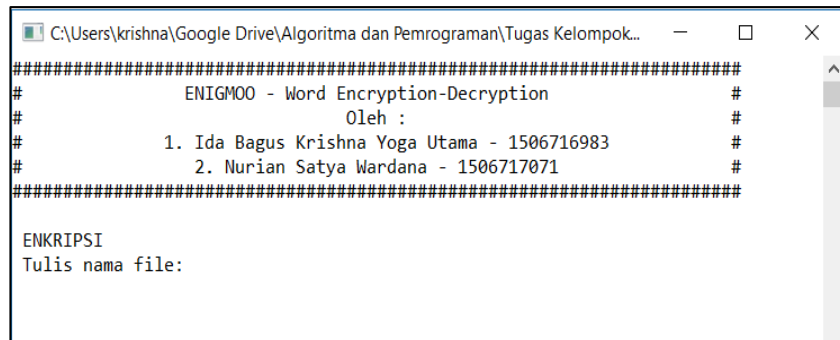
Program Enigmoo ini dibuat dengan tujuan untuk melindungi data-data berupa teks agar aman dan isinya tidak dapat diketahui oleh orang lain. Enigmoo menggunakan enkripsi data dengan cara memasukkan kode rahasia dalam proses enkripsi dan dekripsi. Kode rahasia pada proses enkripsi dan dekripsi data harus sama karena jika berbeda akan menyebabkan data tidak dapat terbaca seperti bentuk semula. Program Enigmoo diciptakan sebagai program enkripsi dan dekripsi sederhana yang cukup reliabel dan data pengguna aman dari tindak pencurian data.

2. Cara Kerja Program

Program Enigmoo memiliki dua fungsi yaitu sebagai program untuk mengenkripsi data dan program untuk mendekripsi data. Masing – masing fungsi tersebut memiliki cara kerja yang berbeda.

1. Enigmoo sebagai encryptor data

Untuk mengenkripsi data, awalnya Enigmoo akan meminta pengguna untuk memasukkan nama file dengan format .txt dimana isi dari file .txt tersebut akan dienkripsi oleh Enigmoo.



Gambar 1. Tampilan menu enkripsi dari Enigmoo

Sesaat setelah pengguna memasukkan nama file yang akan dienkripsi, Enigmoo akan membuka file tersebut dan membaca isi dari file dan memasukkan isi file ke dalam sebuah variable array. Jika seluruh isi file telah dipindahkan ke dalam suatu variable array, maka Enigmoo akan menutup file tersebut dan mulai mengolah data. Setiap kata atau karakter yang terdapat dalam variable array diubah ke dalam bentuk ASCII code masing masing.

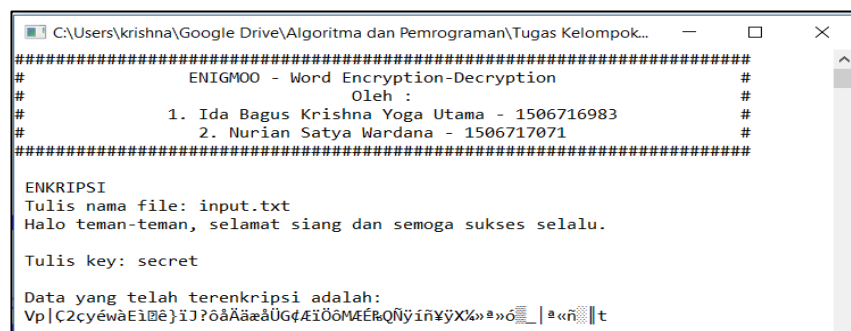
Kemudian, pengguna akan memasukkan kode rahasia sebagai kunci dari proses enkripsi ini. Kode rahasia yang dimasukkan oleh pengguna dapat berupa angka, satu kata atau satu kalimat. Contoh dari kode rahasia adalah “10”, “a”, “secret”, “rahasia”, dan lain – lain.

Enigmoo akan memproses kode rahasia yang dimasukkan oleh pengguna dengan cara melakukan sorting kode rahasia secara ascending lalu mengambil median dari hasil sorting tersebut untuk dijadikan *key* yang digunakan dalam proses enkripsi data.

Key yang didapatkan lalu dikirim ke proses enkripsi dimana setiap ASCII code dari setiap karakter pada data akan dimasukkan ke dalam perhitungan dengan rumus:

$$\text{Hasil Enkripsi} = \text{ASCII code data} - (100 - (\text{key} + i))$$

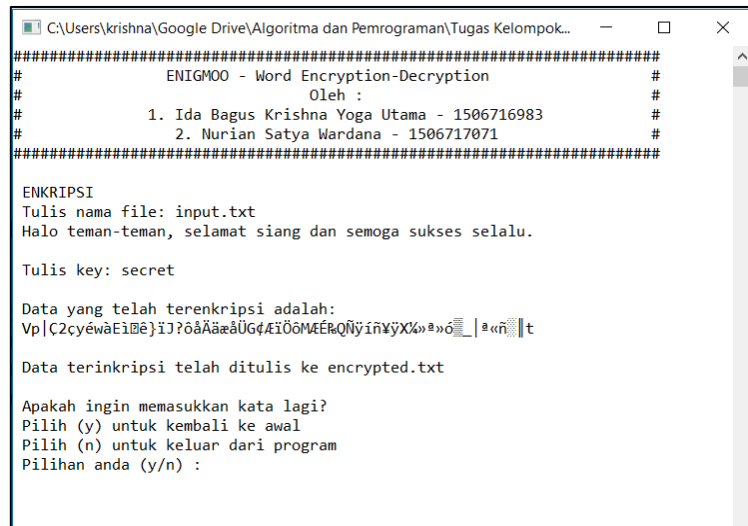
Dimana *ASCII code data* adalah nilai ASCII code per karakter dari data awal, *key* adalah hasil sorting dari kode rahasia yang dimasukkan oleh pengguna, dan *i* adalah nomor iterasi dari proses tersebut. *Hasil Enkripsi* adalah hasil perhitungan tersebut dimana akan menjadi nilai tertentu yang berbeda dibanding data awal.



Gambar 2. Tampilan data yang telah terenkripsi

Proses perhitungan tersebut dilakukan berulang – ulang kepada setiap karakter dari data awal sehingga didapatkan data baru yang berbeda total dibanding data awal. Setelah seluruh

perhitungan selesai, hasil enkripsi data disimpan ke dalam file encrypted.txt, dimana file tersebut dibuat oleh program Enigmoo.



```

#####
#           ENIGMOO - Word Encryption-Decryption           #
#                   Oleh :                                   #
#                   1. Ida Bagus Krishna Yoga Utama - 1506716983   #
#                   2. Nurian Satya Wardana - 1506717071           #
#####

ENKRIPSI
Tulis nama file: input.txt
Halo teman-teman, selamat siang dan semoga sukses selalu.

Tulis key: secret

Data yang telah terenkripsi adalah:
Vp|C2cyewàEiBē}iJ?ôãÄääÜG4EiÖ6MÉRQñyñvYX«»#»ö_|««ñ||t

Data terenkripsi telah ditulis ke encrypted.txt

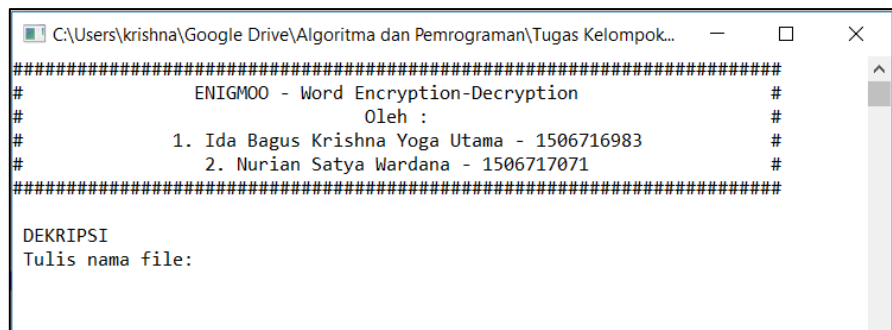
Apakah ingin memasukkan kata lagi?
Pilih (y) untuk kembali ke awal
Pilih (n) untuk keluar dari program
Pilihan anda (y/n) :

```

Gambar 3. Tampilan lengkap dari proses enkripsi

2. Enigmoo sebagai decryptor data

Cara kerja dari Enigmoo sebagai decryptor data ini adalah proses yang hampir sama dengan proses enkripsi data namun memiliki perbedaan dibanding proses enkripsi pada bagian perhitungan hasil. Untuk melakukan dekripsi data, awalnya Enigmoo akan meminta pengguna untuk memasukkan nama file dengan format .txt dimana isi dari file .txt tersebut akan didekripsi oleh Enigmoo.



```

#####
#           ENIGMOO - Word Encryption-Decryption           #
#                   Oleh :                                   #
#                   1. Ida Bagus Krishna Yoga Utama - 1506716983   #
#                   2. Nurian Satya Wardana - 1506717071           #
#####

DEKRIPSI
Tulis nama file:

```

Gambar 4. Tampilan menu dekripsi dari Enigmoo

Setelah pengguna memasukkan nama file yang akan didekripsi, Enigmoo akan membaca file tersebut dan menyimpan setiap karakter di dalam file ke dalam sebuah variabel array. Jika seluruh isi data telah dibaca dan dipindahkan, Enigmoo akan menutup file tersebut dan mengubah isi data ke dalam ASCII code.

Kemudian, pengguna memasukkan kode rahasia yang sama dengan kode rahasia yang digunakan dalam proses enkripsi. Kode rahasia tersebut akan disorting dan median dari hasil sorting tersebut menjadi *key* yang digunakan dalam proses dekripsi data. Setelah mendapat variable array yang berisi ASCII code dari data dan *key*, Enigmoo akan melakukan perhitungan dengan rumus:

$$\text{Hasil Dekripsi} = \text{ASCII code data} + (100 - (\text{key} + i))$$

Perhitungan diatas dilakukan kepada setiap karakter dari data yang akan didekripsi sehingga didapatkan data baru yang telah terdekripsi. Setelah seluruh data terdekripsi, Enigmoo membuat sebuah file baru bernama decrypted.txt dan menulis data yang telah terdekripsi ke dalam file tersebut sehingga pengguna dapat melihat hasil dekripsi pada file tersebut.

Gambar 5. Tampilan lengkap dari proses dekripsi

```

C:\Users\krishna\Google Drive\Algoritma dan Pemrograman\Tugas Kelompok...
#####
#          ENIGMOO - Word Encryption-Decryption          #
#                   Oleh :                                #
#          1. Ida Bagus Krishna Yoga Utama - 1506716983   #
#          2. Nurian Satya Wardana - 1506717071          #
#####

DEKRIPSI
Tulis nama file: encrypted.txt
Vp|C2cyewàEi@è}i?ôâÄäâÜGçÆiÖöMÆEßQñYññYX%»»ö||«ñ||t

Tulis key: secret

Data yang telah terdekripsi adalah:
Halo teman-teman, selamat siang dan semoga sukses selalu.

Data terdekripsi telah ditulis ke decrypted.txt

Apakah ingin memasukkan kata lagi?
Pilih (y) untuk kembali ke awal
Pilih (n) untuk keluar dari program
Pilihan anda (y/n) :

```

3. Penjelasan Program

Program ini dibuat dengan memperhatikan aspek modularitas sehingga terdiri dari lima komponen:

1. source.c

Program utama yang berisi tampilan menu awal dan bertugas untuk mereferensikan keempat file header lain. Pada file source.c ini, fungsi yang dipanggil hanyalah fungsi-fungsi yang berada pada file header.h.

2. header.h

File header ini berisi tampilan menu untuk enkripsi-dekripsi. Pada file header ini lah fungsi encrypt() dan decrypt() dipanggil. File ini juga berisi tampilan untuk panduan penggunaan program.

3. encrypt.h

Algoritma dari enkripsi terdapat pada file header ini. Untuk mendapatkan key yang digunakan untuk proses enkripsi, fungsi encrypt() memanggil fungsi getKey() di getKey.h. Penjelasan untuk setiap baris pada program ini ditulis sebagai line comment di file encrypt.h.

4. decrypt.h

File header ini berisi algoritma untuk dekripsi data. Seperti encrypt.h, pada file ini juga memanggil fungsi getKey() dari getKey.h.

5. getKey.h

Pada file ini berisi algoritma untuk menentukan key yang digunakan pada proses enkripsi dan dekripsi, dengan input berupa string (gabungan beberapa karakter) dari pengguna. Algoritma

tersebut menggunakan bubble sort. Tujuan dibuatnya file header ini ialah agar saat ada perubahan pada algoritma pencarian key, baik fungsi encrypt() dan fungsi decrypt() tidak perlu diubah. Penjelasan setiap bagian pada fungsi ini ditulis sebagai line comment di file getKey.h.

Cuplikan program dan penjelasan

```
for (i = 0; i < lastIndex; i++) {
    for (j = lastIndex; j > i; j--) {
        if(keys[j] < keys[j-1]) {
            temp = keys[j];
            keys[j] = keys[j-1];
            keys[j-1] = temp;
        }
    }
}

return keys[keysLength/2];
```

Potongan program tersebut berada di file getKey.h pada baris 20-31. Di atas merupakan algoritma bubble sort yang digunakan untuk mencari median dari string yang ditulis oleh user. User menginput key sebagai string (beberapa karakter), program kemudian mengkonversi karakter tersebut menjadi representasi kode ASCII. Kode ASCII yang berupa bilangan bulat (integer) itulah yang selanjutnya diurutkan oleh algoritma bubble sort.

Dalam hal ini, input string dari user akan dimasukkan ke array `keys[]`, dimana panjang string pada array tersebut disimpan pada variabel `keysLength`. Variabel `lastIndex` merupakan `keysLength - 1`

Hasil pengurutan tersebut selanjutnya merupakan bilangan bulat yang digunakan untuk algoritma enkripsi dan dekripsi. Sesuai dengan metode enkripsi yang telah ditulis sebelumnya:

$$\text{Hasil Enkripsi} = \text{ASCII code data} - (100 - (\text{key} + i))$$

Maka dengan telah ditemukannya bilangan bulat key, maka hasil enkripsi dapat ditentukan.

```
printf("\n Data yang telah terdekripsi adalah: \n ");
for(i = 0; i < length; i++){
    decrypted[i] = isiFile[i] + (100-(key+i));
    printf("%c", decrypted[i]);
    fprintf(ofp, "%c", decrypted[i]);
}
```

Kemudian di atas merupakan cuplikan program pada file decrypt.h baris 50-55. Pada bagian program tersebut, terlihat bahwa metode untuk dekripsi ialah:

$$\text{Hasil Dekripsi} = \text{ASCII code data} + (100 - (\text{key} + i))$$

Metode dekripsi tersebut ialah kebalikan dari metode enkripsi. Variabel `key` adalah variabel hasil keluaran dari pemanggilan fungsi `getKey()`. Pada bagian program di atas, dilakukan pengulangan sebanyak jumlah karakter pada file yang ingin didekripsikan. Banyaknya jumlah karakter yang ingin didekripsikan disimpan di variabel `length`.

Untuk penjelasan yang lebih mendalam mengenai enkripsi dan pencarian key, ditulis lebih lanjut di file `encrypt.h` dan `getKey.h`.

4. Flowchart Program

