APLIKASI HUFFMAN CODING DALAM KRIPTOGRAFI

Yusak Yuwono Awondatu13514005
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514005@std.stei.itb.ac.id

***Abstrak* — makalah ini berisi pembahasan mengenai aplikasi Huffman coding yang umumnya dipakai untuk kompresi, yakni dalam kriptografi. Meskipun file dikompresi menjadi lebih singkat, pengkompresan file tidak berarti mengurangi tingkat keamanan file tersebut. Pembahasan ini akan melihat Huffman Coding dalam kriptografi baik kelebihan maupun kekurangannya.**

Keywords—Huffman, kriptografi, binary, ASCII

I. PENDAHULUAN

Pada era yang didominasi oleh digital seperti sekarang ini, hampir setiap orang terhubung dengan internet. Baik pengguna internet maupun kontributor dari konten yang ada didalam internet. Oleh karena itu, setiap pengguna internet tentunya memiliki data-data yang disimpan didalam internet tersebut. Misalnya seperti data social media, email, game, atau bahkan transaksi perusahaan.

Hal yang sangat krusial dari penyimpanan data yang disimpan di internet adalah kerahasiaan data pibadi. Banyak data, interaksi, dan transaksi pribadi yang penting yang disalurkan melalui internet dan menjadi sasaran utama bagi pencurian informasi dan dipakai oleh pihak yang tidak bertanggung jawab. Misalnya saja seperti foto, transaksi perusahaan, informasi pribadi.

Untuk mencegah pencurian informasi seperti ini, perlu proses penyandian atau yang lebih biasa dikenal dengan enkripsi dalam bidang kriptografi. Dengan adanya enkripsi ini, pihak yang tidak bertanggung jawab tersebut akan lebih kesulitan dalam mencuri informasi karena meskipun berhasil mencurinya, mereka tidak dapat menerjemahkan informasi yang dikirimkan tersebut.

Ada berbagai macam cara untuk melakukan proses encoding terhadap pesan-pesan tersebut. Misalnya saja Caesar Cipher, RSA. Namun salah satu alternative yang mungkin adalah dengan menggunakan *Huffman Coding.*

Huffman Coding sendiri sebenarnya adalah suatu metode coding yang membuat data yang diberikan menjadi lebih singkat sehingga sebenarnya lebih sering digunakan untuk melakukan kompresi file agar menggunakan ruang penyimpanan lebih sedikit.

Namun teknik Coding ini memiliki suatu keunikan yang menjadikan dasar penulis untuk menyampaikan idenya untuk menggunakannya dalam kriptografi.

Dengan metode Huffman file yang dikodekan memang akan menjadi lebih pendek, namun ini bukan berarti data tersebut menjadi tidak aman karena ukurannya yang lebih pendek. Dengan metode *Huffman Coding* dan konversi kedalam kode ACII, pesan yang dikompresi menjadi lebih singkat tersebut akan menjadi pesan rahasia yang kuat.

Melalui tulisan ini, akan dibahas ide singkat mengenai penggunaan *Huffman Coding* untuk kriptografi.

II. DASAR TEORI

2.1 Huffman Coding

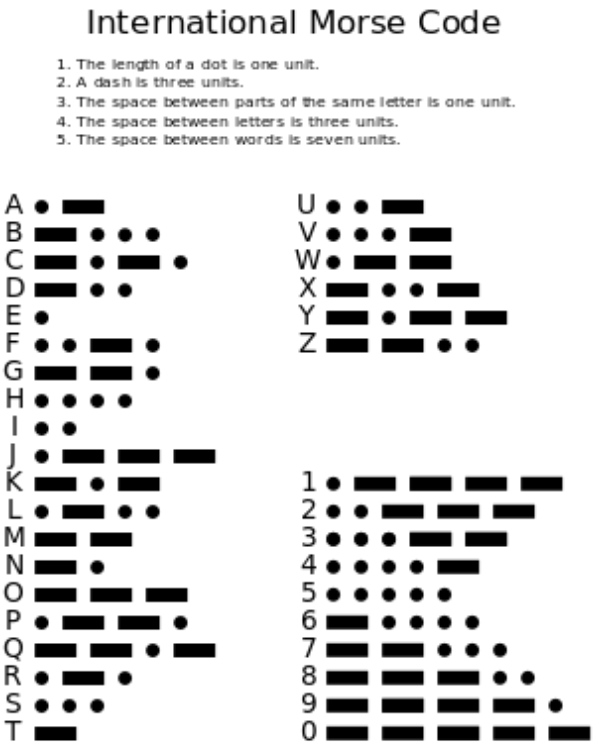
[4]Salah satu metode menyandikan pesan dengan representasi 2 simbol adalah kode morse, yaitu dengan menggunakan "–" garis dan "." titik.

Kode morse lebih umum digunakan dalam pengiriman pesan dengan telegraf dan radio pada abad 19. Namun kode ini juga digunakan untuk mengirimkan pesan rahasia pada perang dunia II.

Kode ini bersifat prefix yang berarti telah memiliki table penerjemah yang tetap. Hal ini menjadi rawan jika pihak lawan mendapatkan table penerjemah ini.

Selain itu, kode morse memiliki kelemahan lainnya, yaitu kode yang bertumpukan. Misalnya saja pada huruf N(- .), (- .) juga terdapat didalam B C D K X Y sehingga diperlukan suatu jeda untuk menghindari huruf bertumpukan

Jika diberikan jeda untuk setiap huruf, kode akan menjadi lebih rawan lagi untuk dipecahkan oleh pihak lawan meskipun tidak mengetahui table penerjemah sandi morse tersebut.



Namun berbeda dengan kode Huffman, kode Huffman bersifat unik dan tidak akan ada karakter yang bertumpukan seperti morse sehingga dapat langsung dituliskan langsung tanpa jeda yang tentunya akan meningkatkan keamanan pesan yang dikirmkan.

[4]*Huffman Coding* diciptakan oleh seorang mahasiswa MIT, yaitu David Huffman pada 1952. *Huffman Coding* adalah metode *lossless compression scheme* atau dengan kata lain kompresi tanpa kehilangan yang berarti memperpendek data yang ada tanpa menghilangkan informasi yang ada didalamnya. Dengan metode ini, data yang ada akan menjadi lebih pendek dan tentunya akan menghemat ruang penyimpanan.

[1]Untuk menggunakan *Huffman Coding*, karakter yang ada didalam pesan diukur frekuensinya, lalu dijadikan kedalam kode binary yang lebih singkat dengan aturan tertentu sehingga karakter dengan frekuensi tertinggi memiliki jumlah bit lebih sedikit, karakter dengan frekuensi makin jarang memiliki jumlah bit lebih banyak sehingga ruang penyimpanan lebih efisien.

Misalnya diberikan sebuah pesan acak “ABACCDA”
Mula- mula, hitunglah frekuensi dari karakter yang muncul dalam pesan tersebut.

A = 3, B = 1, C = 2, D = 1

kode ASCII untuk ke-4 karakter tersebut adalah
A = 01000001 B = 01000010
C = 01000011 D = 01000100

Jika pesan ABACCDA dijadikan kedalam binary, akan menjadi

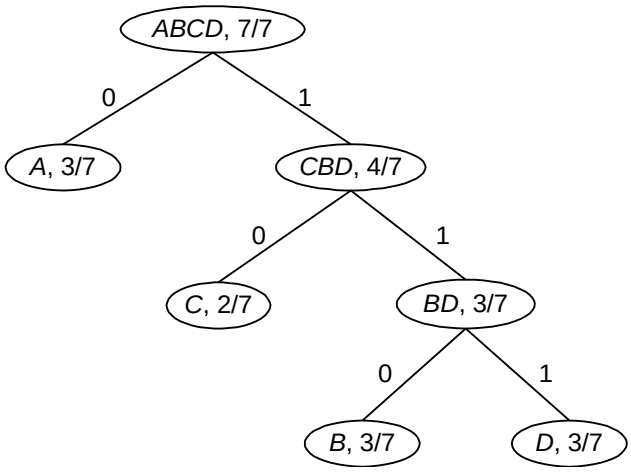
```
01000001 01000010 01000001 01000011
01000011 01000100 01000001
```

Dengan menggunakan *Huffman Coding*, dari pesan yang panjang mula-mula 7*8=56 bit, dapat dikompresi dan menjadikannya lebih pendek.

Algoritma pembentukan pohon Huffman

1. Pilih dua simbol dengan peluang (*probability*) paling kecil (pada contoh di atas simbol *B* dan *D*). Kedua simbol tadi dikombinasikan sebagai simpul orangtua dari simbol *B* dan *D* sehingga menjadi simbol *BD* dengan peluang 1/7 + 1/7 = 2/7, yaitu jumlah peluang kedua anaknya.
2. Selanjutnya, pilih dua simbol berikutnya, termasuk simbol baru, yang mempunyai peluang terkecil.
3. Ulangi langkah 1 dan 2 sampai seluruh simbol habis.

Dengan menerapkan langkah tersebut :
A = 3/7, B = 1/7, C = 2/7, D = 1/7
A = 3/7, C = 2/7, BD = 2/7.
A = 3/7, BCD = 4/7.
ABCD = 7/7.
.



dengan *Huffman Coding*

A = 0 B = 110
C = 10 D = 111

representasi ABACCDA kini menjadi
0110010101110 yang panjangnya hanyalah 13 bit.

Keunikan dari Huffman Code ini adalah,berubah-ubah untuk setiap pesan yang diberikan sehingga memiliki pola yang tidak tetap. Hal inilah yang memberinya nilai bonus untuk menggunakannya dalam kriptografi karena meningkatkan keamanan dari pesan yang diberikan. Pencuri pesan tidak dapat membuat kode penerjemahan yang tetap karena setiap pesan akan memiliki kode terjemahan yang berbeda-beda.

Selain itu kode yang unik untuk setiap karakter juga menutupi kelemahan dari morse yang memerlukan jeda di tiap huruf untuk menghindari penumpukan.

Untuk proses decipher/menerjemahkan, dapat dilakukan kembali dengan menjadikannya dulu kedalam ASCII dulu, lalu menerjemahkannya dari kiri ke kanan

```
0110010101110
A110010101110
A B 010101110
A B A10101110
A B A C101110
A B A C C1110
A B A C C D 0
A B A C C D A
```

Kode kembali diterjemahkan menjadi ABACCDA

2.2 Kriptografi

Kriptografi berasal dari istilah yunani κρυπτός (baca : kryptós) yang berarti tersembunyi atau rahasia, dan γράφειν (baca : graphein) yang berarti tulisan. Dengan begitu kriptografi berarti tulisan yang dirahasiakan. Proses mengrahasiakan tulisan ini bertujuan agar pesan tersebut hanya dapat dibaca oleh pihak pengirim dan pihak penerima pesannya.

Ada banyak macam kriptografi yang digunakan sejak `\jaman dahulu. Misalnya dari yang paling sederhana seperti Caesar Cipher dimana menggeser urutan 26 huruf dalam alphabet dengan n bilangan, lalu menggunakan modulo. Ada pula metode enkripsi lain yang cukup terkenal seperti Enigma yang digunakan oleh tentara jerman pada perang dunia II. Lalu yang terbaik sejauh ini adalah kode RSA yang menggunakan 2 key dan 2 bilangan unik untuk menggeser huruf dalam kode ASCII.

Semakin rumit algoritma yang digunakan untuk kriptografi, maka sandi akan semakin aman. Namun karena setiap algoritma memiliki pola, maka cepat atau lambat kode tersebut juga akan dapat dipecahkan meskipun tidak mengetahui kuncinya.

2.3 Penggunaan Huffman Coding dalam Kriptografi

Meskipun *Huffman coding* umumnya dipakai dalam melakukan kompresi yang berarti data yang diberikan menjadi lebih pendek, bukan berarti data tersebut menjadi lebih rentan keamanannya.

Huffman code memiliki pengkodean yang unik sehingga dapat menjadi faktor yang meningkatkan kerahasiaan dari informasi yang diberikan. Keuntungan lainnya adalah *Huffman Coding* membuat file menjadi lebih pendek, file tersebut menjadi memiliki multipretasi yang juga menyulitkan untuk decrypting.

Dengan proses Huffman, karakter-karakter yang ada dapat dipersingkat menjadi kode binary yang lebih pendek. Jika kode-kode pendek tersebut dikelompokkan setiap 8 bit dan menerjemahkannya kedalam ASCII kembali, maka akan menghasilkan karakter lain yang muncul secara acak dan tidak memiliki arti.

Selain itu dengan memperpendek pesan yang dikirim, pengiriman akan menjadi lebih mudah pula karena memakan data yang lebih sedikit..

III. HUFFMAN CODING UNTUK KRIPTOGRAFI

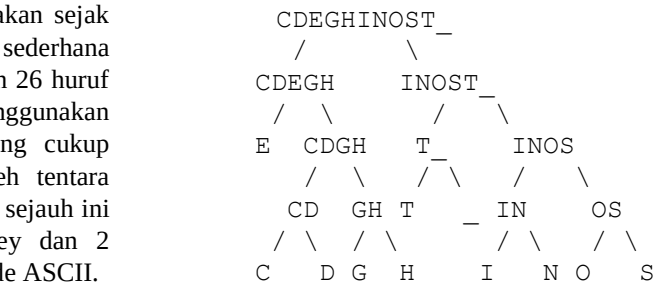
Dalam *Huffman coding*, pesan yang diberikan memang lebih singkat, namun bukan berarti pesan tersebut tidak aman. Dengan mengkombinasikannya dengan konversi kedalam binary, hasil pesan dengan kriptografi menggunakan kode Huffman akan cukup aman.

Misalkan diberikan sebuah pesan teks singkat

TESTING THE CODE (spasi dihitung)

Untuk menggunakan metode *Huffman Coding*, mula-mula akan dicari kemunculan karakter tersebut

```
C = 1          D = 1          E = 3
G = 1          H = 1          I = 1
N = 1          O = 1          S = 1
T = 3          _(spasi) = 2
Arah kiri = 1, arah kanan =0
```



Hasil karakter sesudah kode Huffman.

```
C = 0100       D = 0101       E = 00
G = 0110       H = 0111       I = 1100
N = 1101       O = 1110       S = 1111
T = 100        _(spasi) = 101
```

Jika teks semula diterjemahkan kedalam binary[2]

```
01010100 01000101 01010011 01010100
01001001 01001110 01000111 00100000
01010100 01001000 01000101 00100000
01000011 01001111 01000100 01000101
```

Teks akan memiliki panjang awal 16*8=128 bit

Setelah penerjemahan dengan Huffman code,

```
100 00 1111 100 1100 1101 0110 101
100 0111 00 101 0100 1110 0101 00
```

Setelah menggunakan Huffman code, panjang teks tinggal 53 bit.

Setelah melakukan penerjemahan dengan Huffman Code, kode yang ada dipotong setiap 8 bit.

```
10000111 11001100 11010110 10110001
11001010 10011100 10100___
```

Setelah pemotongan dilakukan, karakter diterjemahkan kembali kedalam ASCII.
Namun karena hasil hanyalah 53 bit, bagian yang tidak genap 8 perlu diisi dengan suatu karakter dummy. Karakter dummy haruslah yang tidak terdapat didalam kode Huffman yang telah dibuat. Dalam contoh ini, akan diberikan 111.

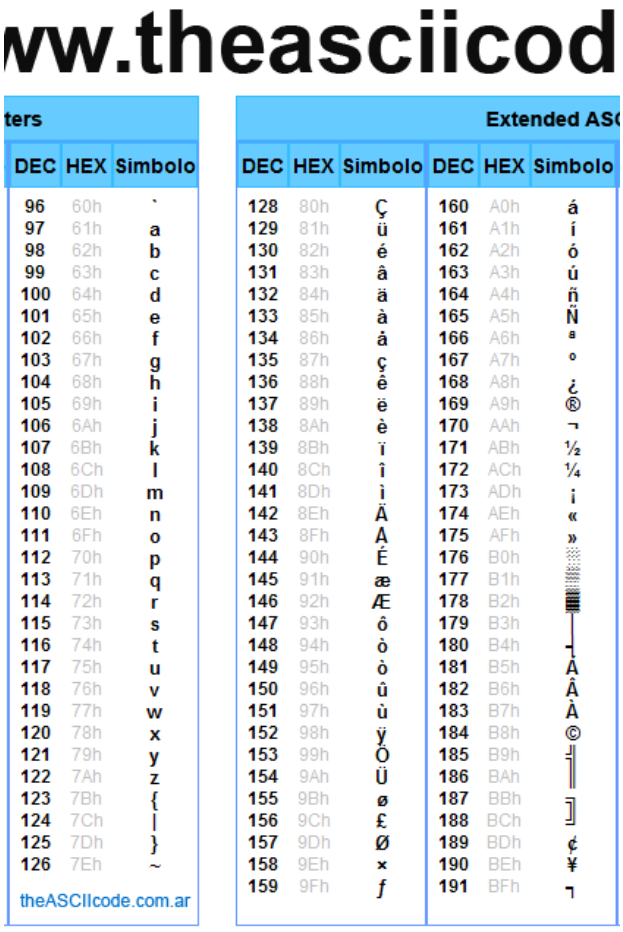
Namun karena pada kode 0-32, 127 dan 255 merupakan karakter yang tidak memiliki representasi. Symbol karakter, hal ini dapat menyulitkan pengguna apabila sampai terbentuk karakter yang seperti ini.

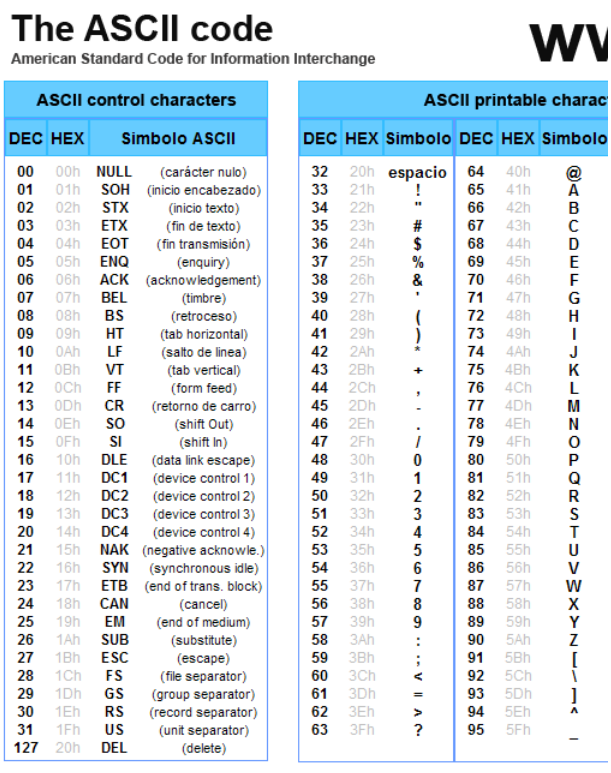
Dengan penerjemahan kembali kedalam ASCII, teks akan menjadi :

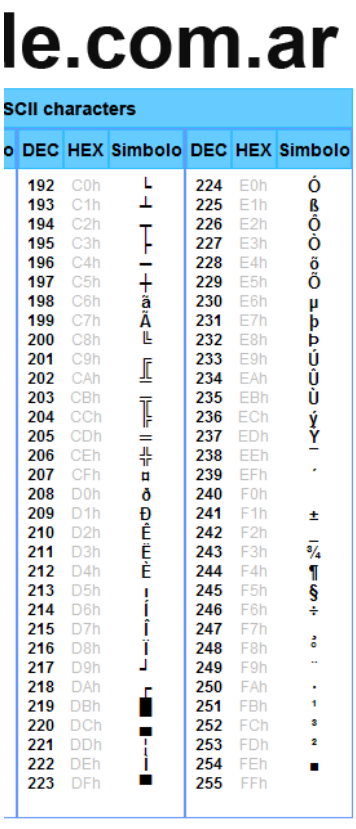
‡ Ì Ö ± Ê œ §

Dari bentuk awal
TESTING THE CODE, panajng 128 bit
Setelah enkripsi dengan *Huffman coding* menjadi[2]
‡ Ì Ö ± Ê œ § , panjang 56 bit
Dengan metode seperti ini, pesan akan menjadi tidak dapat dimengerti meskipun memiliki panjang yang lebih singkat dibandingkan dengan pesan semula karena hanya terdiri dari simbol-simbol dan yang tidak bermakna jika disambungkan.

Table ASCII dan *extended symbol*nya [3]







Untuk proses decipher, kembalikan ‡ Ì Ö ± Ê œ Ş menjadi kode ASCII lagi
1000011111001100110101101011000111001010100111001010100111

Terjemahkan kembali berdasarkan table Huffman yang telah dibuat.
100(T)00(E)1111(S)100(T)1100(I)
1101(N)0110(G)101(_)100(T)0111(H)00(E)
101(_)0100(C)1110(O)0101(D)00(E)111(dummy)
Kode berhasil diterjemahkan kembali menjadi
TESTING_THE_CODE(dummy)
tanpa ada kode yang bertumpukan seperti morse.

IV. PEMBAHASAN

4.1 KELEBIHAN

Melalui metode yang telah diberikan pada bab II, dapat dilihat bahwa meskipun teks menjadi lebih pendek, teks juga menjadi sulit untuk dibaca. Hal ini membuktikan bahwa teks yang lebih pendek belum tentu tidak aman.

Setiap kode Huffman juga memiliki bentuk yang unik dan tidak tetap untuk setiap pesan. Karena itu setiap pesan juga akan memiliki kode dan tingkat keamanannya tersendiri dan lebih susah untuk diterjemahkan kembali jika tidak mengetahui Kode Huffman untuk setiap karakter pada pesan tersebut.

Meskipun pihak luar juga mengetahui sistem kode Huffman, tetap akan sulit untuk menerjemahkan kode yang diberikan karena terdapat beberapah huruf yang memiliki panjang bit yang sama dan dapat ditukar-tukar oleh pengirim pesan.

Pesan yang dikirimkan pengguna tidak hanya memiliki ukuran yang lebih kecil sehingga lebih mudah dikirimkan, namun tetap terjaga kerahasiaannya.

4.1 KELEMAHAN

Terdapat beberapa kelemahan dari metode ini. Salah satunya adalah menggunakan ASCII dimana didalamnya terdapat sejumlah karakter yang tidak punya representasi, misalnya pada karakter no 0-32, 127, dan 255 sehingga jika pesan membentuk karakter tersebut, maka tidak punya representasi karakter.

Selain karakter kosong, kelemahan lainnya adalah jika hasil penerjemahan dengan Huffman Code tidak kelipatan delapan, maka sisa karakternya terpaksa diisi dengan dummy yang memiliki risiko untuk mengacaukan isi pesan.

Karena kode Huffman unik untuk setiap pesan, maka hal ini juga dapat menyulitkan proses untuk mengodekan dan merjemahkannya kembali dan akan memiliki table decipher yang berbeda untuk setiap pesan yang juga dapat menyusahkan pengguna. Selain itu penerima pesan juga perlu menerima table penerjemahan pesan karena setiap pesan punya kode unik dan tidak bias dihafalkan seperti morse atau semacamnya.

V. KESIMPULAN

Kode Huffman dapat digunakan didalam kriptografi. Penggunaan ini memiliki keuntungan dalam segi keefektifan panjang data yang dikirimkan serta keamanan karena memiliki panjang yang singkat, namun tetap sulit dibaca pula. Selain itu juga memiliki penerjemahan yang berbeda dan unik untuk setiap pesan yang berbeda sehingga tidak memiliki pola yang merupakan kelemahan dari setiap kebanyakan kode.

Namun hal ini dapat memberi kelemahan karena menyulitkan pihak pengguna juga. Selain itu meskipun bisa menghemat space, kode yang telah diterjemahkan dengan Huffman kedalam ASCII terkadang tidak memiliki representasi fisik yang tepat karena bukan karakter (backspace, NULL, spasi, enter, dan lainnya), misalnya untuk kode 0-32, 127, 129, 141, 144, 157, dan 160.

REFERENSI

[1] Munir, Rinaldi, 2006, *Matematika Diskrit*. Bandung : Penerbit Informatika ITB.
[2] http://www.ascii-code.com/ diakses pada 8 Desember 2015, 20.40
[3] http://www.theasciicode.com.ar/ diakses pada 9 Desember 2015, 15.00
[4] http://rosettacode.org/wiki/Huffman_coding diakses pada 9 Desember 2015, 18.00
[5] http://www.cs.utsa.edu/~wagner/laws/huffman.html diakses pada 9 Desember 2015, 20.00
[6] http://www.huffmancoding.com/my-uncle/scientific-american diakses pada 9 Desember 2015, 21.00
[7] http://www.staff.science.uu.nl/~leeuw112/huffman.pdf diakses pada 9 Desember 2015, 21.15

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, 8 Desember 2015

ttd

Yusak Yuwono Awondatu
13514005