

PENERAPAN ARITMATIKA MODULO PADA ALGORITMA LUHN UNTUK VALIDASI NOMOR SERI IMEI DAN KARTU KREDIT

Anwar Ramadha 13514013
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514013@stei.itb.ac.id

Abstract—Makalah ini bertujuan untuk membahas tentang validasi nomor IMEI pada sebuah peranti keras, misalnya telepon seluler. Validasi ini menggunakan salah satu algoritma yang dikembangkan oleh Luhn. Algoritma ini menggunakan salah satu teori bilangan yaitu aritmatika modulo. Hasil yang digunakan adalah hasil sisa pembagian dengan 10 (modulo 10). Algoritma Luhn juga telah diterapkan untuk validasi nomor kartu kredit.

Keywords—IMEI, Luhn, modulo, kartu kredit.

I. LATAR BELAKANG

Dalam kehidupan manusia, semua peralatan yang mereka beli tidak luput dari nomor seri. Hal ini digunakan untuk mempermudah pendataan vendor untuk alat tersebut. Tentu saja nomor seri ini tidak di tuliskan secara sembarangan, melainkan ada ilmu untuk menuliskannya. Salah satunya adalah menggunakan aplikasi dari aritmatika modulo. Salah satu cabang ilmu matematika diskrit ini sudah banyak diterapkan di berbagai hal, tidak hanya sebagai landasan untuk validasi nomor seri. Keamanan yang terkenal untuk enkripsi data seperti RSA (Rivest-Shamir-Adleman) juga menggunakan aplikasi dari cabang ilmu ini. Sampai saat ini belum ada algoritma yang mampu mendekripsi RSA dengan cepat.

Nomor seri telah digunakan dalam berbagai hal, contohnya adalah IMEI, ISBN, nomor kartu kredit, dan sebagainya.

Beberapa alat mensyaratkan bahwa nomor seri yang dimiliki alat tersebut harus valid agar dapat digunakan, sebagai contoh nomor seri pada kartu kredit harus valid agar dapat bertransaksi dengan lancer. Begitu juga halnya dengan IMEI, Deretan nomor ini haruslah valid agar jika terjadi pencurian pada alat tersebut, pemilik dapat mematikan piranti miliknya agar tidak dapat digunakan tanpa wewangnya.

Untuk melakukan pengecekan apakah sebuah nomor seri valid atau tidak salah satunya menggunakan algoritma Luhn. Algoritma ini didasari pada ilmu aritmatika modulo. Hans Peter Luhn, pencipta dari algoritma ini, adalah

seorang insinyur dari IBM. Algoritma Luhn menggunakan modulo 10 untuk dasar operasinya.

II. LANDASAN TEORI

1. Bilangan Bulat

Bilangan bulat adalah bilangan yang tidak memiliki pecahan decimal contohnya 1,2,3, dan lain-lain. Bilangan-bilangan ini mempunyai sifat-sifat pembagian sebagai berikut :

- Misalkan a dan b adalah bilangan bulat dengan $a \neq 0$, maka a dapat dikatakan habis membagi b jika terdapat sebuah bilangan bulat c sehingga $b = ca$ atau dapat ditulis sebagai $\{a \mid b \text{ jika } b = ac, c \in \mathbb{Z} \text{ dan } a \neq 0\}$. Contohnya $5 \mid 10$ karena $10 = 2 \times 5$.
- Misalkan a dan b adalah bilangan bulat dengan $b \neq 0$. Jika a dibagi dengan b maka terdapat 2 buah bilangan bulat unik q (quotient) dan r (remainder), sedemikian sehingga $a = bq + r$ dengan $0 \leq r < b$. Pelu diingat bawa sisa pembagian selalu lebih besar sama dengan nol dan lebih kecil dari pembagi. Contoh jika kita mempunyai bilangan 10 dibagi dengan 3, maka hasil bagi adalah 3 dengan sisa pembagian 1.
 $10 = 3 \cdot 3 + 1$.
- Faktor Persekutuan Terbesar (FPB) bilangan terbesar bilangan bulat yang dapat digunakan untuk membagi 2 bilangan bulat. Contoh 45 mempunyai factor 1,3,5,9,15,45 dan 36 mempunyai factor 1,2,3,4,6,9,12,18,36, maka FPB dari 45 dan 36 adalah 9. FPB dapat dicari juga dengan menggunakan algoritma Euclidean.

2. Aritmatika Modulo

Banyak persolaan yang dapat diselesaikan dengan menggunakan aritmatika modulo, sebagai contoh untuk validasi IMEI pada piranti keras, validasi nomor ISBN, kriptografi, dan lain-lain.

Aritmatika modulo adalah salah satu operasi bilangan bulat yang menghasilkan sisa pembagian. Operator yang digunakan dalam aritmatika modulo adalah **mod**. Sebagai contoh 10 dibagi 3 sisa 1, maka $10 \text{ mod } 3 = 1$. Notasi modulo adalah $\{a \text{ mod } m = r \mid a = mq + r, 0 \leq r < m\}$. Perlu diperhatikan bahwa sisa pembagian selalu lebih besar sama dengan nol, perhatikan contoh berikut ini :
 $-41 \text{ mod } 9 \neq -5$, tetapi $-41 \text{ mod } 9 = 4$ karena $-41 = 9(-5) + 4$.

Terkadang dua buah bilangan bulat a dan b jika dibagi dengan sebuah bilangan bulat m, akan menghasilkan sisa pembagian yang sama, hal ini dapat dikatakan bahwa a dan b kongruen dalam modulo m, dan dinotasikan sebagai $a \equiv b \pmod{m}$. Jika a tidak kongruen dengan b maka dapat ditulis sebagai $a \not\equiv b \pmod{m}$. Misalnya $38 \text{ mod } 5 = 3$ dan $13 \text{ mod } 5 = 3$, maka 38 dan 13 kongruen dalam modulo 5.

Definisi dari kongruen adalah “Misalkan a dan b adalah bilangan bulat dan m adalah bilangan bulat > 0 , maka $a \equiv b \pmod{m}$ jika m habis membagi $a - b$ ”. Contoh $17 \equiv 2 \pmod{3}$ karena 3 habis membagi $17 - 2$. Kekongruenan dapat juga ditulis sebagai $a = b + km$, sehingga hasil diatas dapat juga ditulis sebagai $17 = 2 + 5 \cdot 3$.

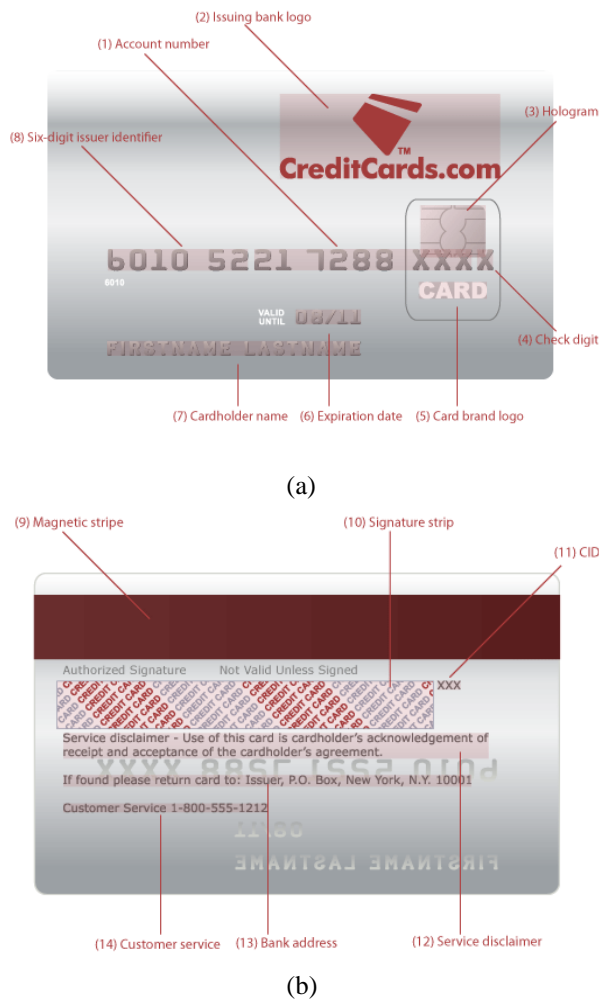
Sifat-sifat yang dimiliki aritmatika modulo adalah sebagai berikut :
 Misal m adalah bilangan bulat positif

1. Jika $a \equiv b \pmod{m}$ dan c adalah sembarang bilangan bulat, maka
 - a. $a + c \equiv (b + c) \pmod{m}$.
 - b. $ac \equiv bc \pmod{m}$.
 - c. $a^p \equiv b^p \pmod{m}$.
2. Jika $a \equiv b \pmod{m}$ dan $c \equiv d \pmod{m}$, maka
 - a. $a + c \equiv (b + d) \pmod{m}$
 - b. $ac \equiv bd \pmod{m}$

3. **IMEI (International Mobile Equipment Identity)**
 IMEI adalah sekumpulan nomor unik yang tertera di setiap piranti keras komunikasi GSM, UMTS, iDEN, dan beberapa telepon satelit. Nomor ini dapat digunakan oleh jaringan untuk mengidentifikasi piranti yang sah, sehingga apabila piranti hilang, maka pemilik piranti dapat meminta pemblokiran agar piranti tidak dapat digunakan lagi. Nomor IMEI tidak menyimpan data pelanggan, tetapi data pelanggan disimpan di dalam data operator seluler.

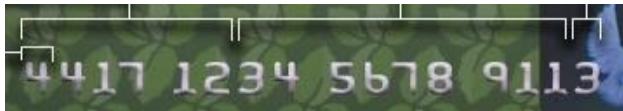
IMEI terdiri dari 15 digit angka, 14 angka memuat informasi negara asal, model, dan nomor seri piranti. 1 digit terakhir adalah angka untuk cek digit. 8 digit pertama memuat model dan asal piranti atau sering dikenal sebagai *Type Allocation Code (TAC)*., sedangkan sisanya tergantung vendor piranti.

4. **Kartu Kredit**
 Kartu kredit adalah alat transaksi yang dapat digunakan oleh pemiliknya tanpa harus membayar pada saat itu juga. Kartu kredit dapat diidentifikasi dari nama vendor yang mengeluarkannya.



Gambar 1 : Kartu kredit tampak depan (a) dan tampak belakang (b).[3]

Pada bagian bawah kartu kredit terdapat nomor seri kartu tersebut. Nomor seri kartu kredit mempunyai panjang maksimal yaitu 19 digit.



Gambar 2 : Contoh nomor seri pada kartu kredit [3]

Untuk mengidentifikasi kategori lembaga yang mengeluarkan kartu tersebut, dapat dilihat dari digit pertama nomor kartu kredit. Digit ini disebut juga MII (*Major Industry Identifier*). Daftar lembaga dapat dilihat di table dibawah ini

Tabel 1 : Lembaga penerbit kartu kredit berdasarkan nomor MII. [3]

MII	Lembaga yang mengeluarkan
0	ISO/TC 68 dan industry lain
1	Penerbangan
2	Penerbangan dan indutri lain
3	Pariwisata dan hiburan
4	Bank dan lembaga finansial lain
5	Bank dan lembaga finansial lain
6	Merchandizing dan bank
7	Petroleum
8	Telekomunikasi dan industry lain
9	Nasional

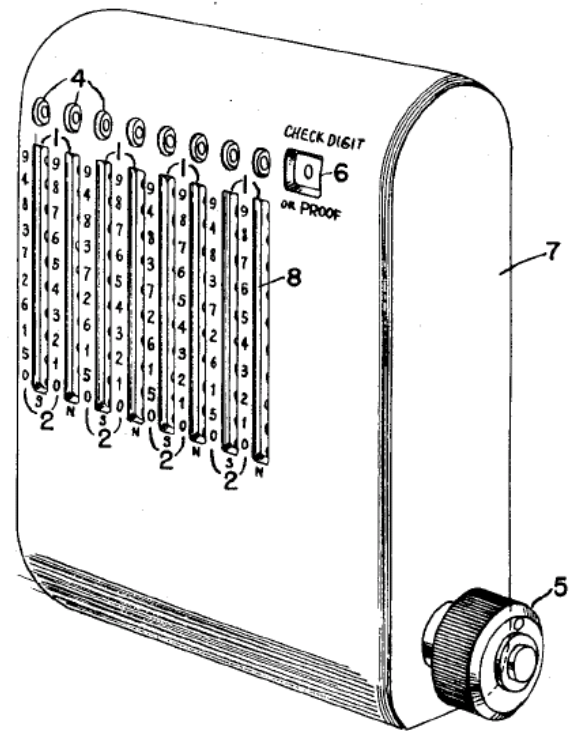
Selanjutnya sampai digit ke-6 adalah nomor MII yang digunakan untuk mengidentifikasi perusahaan yang menerbitkan kartu terebut. Batas maksimal perusahaan yang dapat diidentifikasi oleh 6 digit tersebut adalah 1 juta perusahaan. Untuk kartu kredit yang dikeluarkan oleh lembaga nasional, 3 digit setelah angka 9 adalah kode negara penerbit, sedangkan 2 sisanya tergantung dari peraturan negara yang bersangkutan.

Digit 7 sampai (n-1), dengan n adalah jumlah digit nomor kartu kredit, adalah nomor akun pemilik kartu. Sedangkan digit terakhir adalah check digit.

5. Algoritma Luhn

Algoritma ini dibuat oleh seorang insinyur IBM, Hans Peter Luhn dan dipatenkan di Amerika Serikat pada 23 Agustus 1960. Algoritma Luhn adalah algoritma modulus 10 yang digunakan untuk validasi nomor kartu kredit, ISBN, dan IMEI.

Pada awalnya, Luhn menggunakan sebuah alat untuk menghitung operasi modulus secara mekanik.

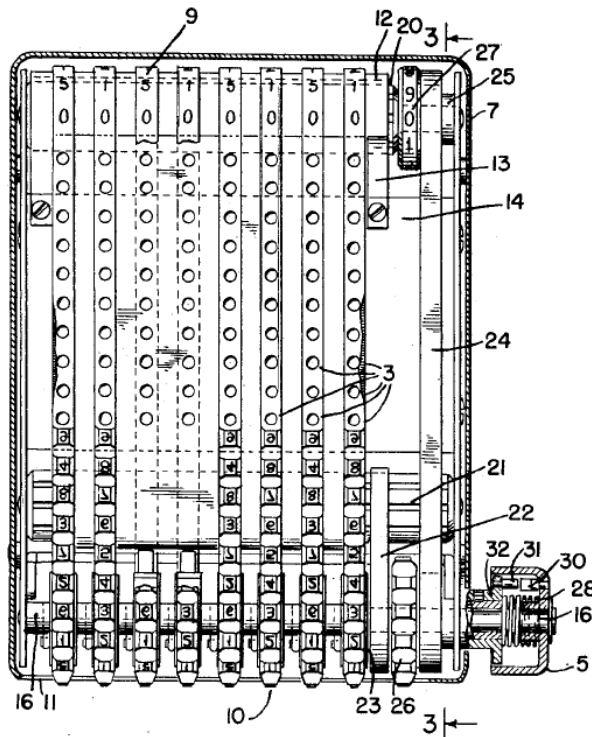


Gambar 1 : Alat penghitung modulus yang dibuat oleh Luhn.[2]

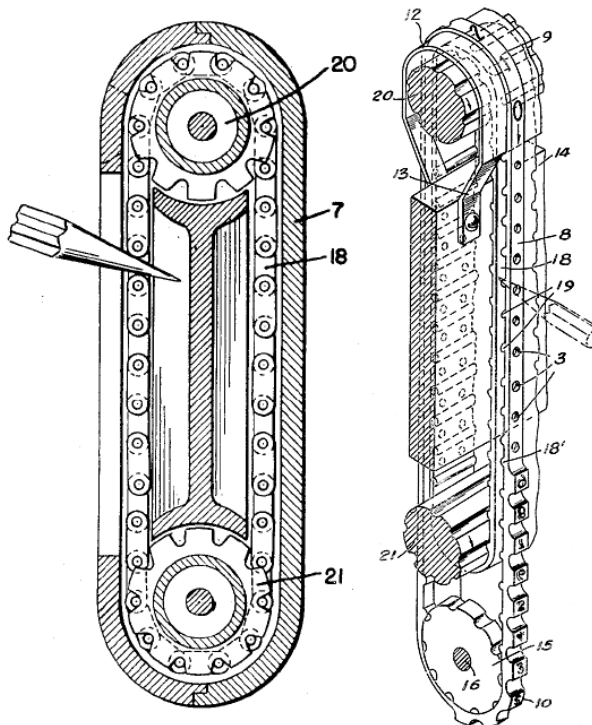
Cara kerja alat ini adalah dengan memasukkan nomor seri yang akan diperiksa, kemudian hasil perhitungan akan muncul di bagian check digit.

Algoritma ini mempunyai kekurangan, yaitu tidak bisa mendeteksi adanya dua digit yang ditukar secara berurutan, missal '13' dengan '31' atau sebaliknya. Terlepas dari itu semua, algoritma ini dapat mendeteksi kesalahan nomor seri dengan baik.

Pemrosesan angka pada algoritma Luhn adalah dari bagian kanan ke kiri dan angka nol hanya akan memengaruhi perhitungan jika mengakibatkan pergeseran pada nomor. Penambahan nol dibagian kiri untuk keperluan normalisasi, tidak akan mengubah hasil perhitungan sebagai contoh hasil 135 den 000135 hasilnya akan sama.



Gambar 2 : Penampang dalam dari alat yang dibuat Luhn [2]



Gambar 3 : Bagian untuk menggeser angka. Terlihat bahwa untuk menggeser angka dibutuhkan alat seperti pensil.[2]

Algoritma sederhana untuk validasi modulus 10 adalah dengan menjumlahkan seluruh digit angka sebelum check digit, kemudian hasil

penjumlahan dimoduluskan dengan 10, maka sisa pembagian itulah yang akan dijadikan sebagai check digit. Sebagai ilustrasi, misalnya diberikan nomor seri 13514013165140?. Langkah pertama yang harus dilakukan untuk mencari nilai yang sesuai untuk menggantikan “?” adalah menjumlahkan semua angka yang berada di bagian kiri “?”
 $1+3+5+1+4+0+1+3+1+6+5+1+4+0+? = 35 + ?$.
 Langkah selanjutnya adalah memoduluskan hasil dengan 10
 $35 \text{ mod } 10 = 5$, maka 5 adalah angka check digit, sehingga nomor seri lengkapnya menjadi 135140131651405.

Algoritma ini menggunakan pembobotan untuk mengalikan setiap digit angka pada nomor seri. Misalnya diberikan nomor seri 135140131651405. Jika penjumlahan setiap digit dilakukan dengan mengalikan setiap digit dengan konstanta yang berbeda misalnya 1 dan 2, digit yang berpasangan dan ditukar tempatnya tidak akan menjadi masalah. Jika hasil perkalian lebih dari 9, maka digit tersebut harus dikurangi 9. Digit terakhir dijumlahkan sehingga hasil penjumlahan menghasilkan kelipatan dari 10.

Secara umum algoritma Luhn mempunyai langkah-langkah sebagai berikut :

1. Kalikan setiap digit angka pada IMEI dengan 2 dimulai dari digit paling kanan secara selang-seling (digit cek adalah angka pertama). Contohnya132, maka akan menjadi234.
 2. Jika ada hasil kali yang lebih dari 9, maka kurangi hasil kali tersebut dengan 9. Contoh135 akan menjadi231 ($5 \times 2 = 10$, jika dikurangi dengan 9 akan menghasilkan 1).
 3. Jumlahkan seluruh digit termasuk digit cek, jika hasilnya kongruen dengan modulo 10, maka IMEI valid, jika tidak maka IMEI tidak valid.
6. Validasi Nomor IMEI

Menurut Menurut *IMEI Allocation and Approval Guidelines* [4]:

The Check Digit is calculated according to Luhn formula (ISO/IEC 7812). See GSM 02.16 / 3GPP 22.016. The Check Digit shall not be transmitted to the network. The Check Digit is a function of all other digits in the IMEI. The Software Version Number (SVN) of a mobile is not included in the calculation. The purpose of the Check Digit is to help guard against the possibility of incorrect entries to the CEIR and EIR equipment [registries]. The presentation of the Check Digit (CD), both electronically and in printed form on the label and packaging, is very important. Logistics (using bar-code reader) and

EIR/CEIR administration cannot use the CD unless it is printed outside of the packaging, and on the ME IMEI/Type Accreditation label. The check digit shall always be transmitted to the network as "0".

Digit terakhir pada IMEI adalah cek digit yang di hitung berdasarkan algoritma Luhn.

Untuk melakukan validasi IMEI, lakukan langkah-langkah algoritma Luhn. Misalnya diberikan nomor IMEI 357874063902602, maka akan menghasilkan tabel sebagai berikut

Tabel 1 : Hasil Perkalian setiap digit pada IMEI yang dilakukan secara selang seling

3	5	7	8	7	4	0	6	3	9	0	2	6	0	?
3	10	7	16	7	8	0	12	3	18	0	4	6	0	?
3	1	7	7	7	8	0	3	3	9	0	4	6	0	?

Ket :

Baris 1 : nomor IMEI sebelum dikali dengan 2

Baris 2 : nomor IMEI sesudah dikali dengan 2 tetapi masih ada angka yang lebih dari 9.

Baris 3 : semua angka yang lebih dari 9 sudah dikurangi dengan 9.

Jika angka pada baris ketika dijumlahkan, maka akan menghasilkan :

$$3+1+7+7+7+8+0+3+3+9+0+4+6+0+? = 58 + ?$$

Dari sini dapat dilihat bahwa jika “?” diganti nilainya dengan cek digit, maka hasil yang di dapatkan adalah 60. Sedemikian sehingga 60 mod 10 adalah 0, maka IMEI diatas adalah valid.

7. Validasi Nomor Kartu Kredit

Untuk menguji kevalidan nomor kartu kredit, lakukan langkah-langkah sebagai berikut :

1. Untuk nomor kartu kredit berjumlah genap, kalikan semua digit nomor seri ganjil, kecuali check digit, kartu tersebut dengan 2. Jika ada hasil perkalian yang lebih dari 9, maka kurangi hasil tersebut dengan 9. Kemudian jumlahkan seluruh angka yang telah dikalikan dengan 2 tadi. Lalu bagi dengan 10, jika sisa pembagian adalah 0, maka kartu kredit valid.
2. Untuk nomor kartu kredit berjumlah ganjil, lakukan hal seperti diatas, hanya saja nomor genap saja yang dikalikan dengan 2.

Misalnya diberikan nomor kartu kredit :
4408041234567893

Maka untuk menguji kevalidan nomor tersebut, lakukan langkah-langkah yang telah disebutkan diatas.

Tabel : Nomor kartu kredit setelah dikalikan dengan 2 dan dikurangi 9 jika hasil kali lebih dari 9

4	2	0	8	0	4	2	2	6	4	1	6	5	7	9	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Lalu jumlah semua angka diatas. Ternyata hasil dari penjumlahan angka-angka diatas, menghasilkan 60. Jika 60 dibagi dengan 10, maka sisa pembagian yang dihasilkan adalah 0, maka nomor kartu kredit tersebut adalah valid.

III. CONCLUSION

Algoritma Luhn adalah algoritma yang dapat digunakan untuk memvalidasi banyak nomor seri, contohnya adalah IMEI dan nomor pada kartu kredit. Algoritma ini masih dipakai hingga sekarang. Tetapi masih ada kekurangan pada algoritma ini yaitu tidak dapat mendeteksi adanya digit yang ditukar secara berurutan. Tapi secara *overall* algoritma ini dapat mendeteksi kesalahan dengan baik.

REFERENSI

- [1] Munir, Rinaldi (2006). Diktat Kuliah IF2120 Matematika Diskrit. Penerbit : ITB.
- [2] Jiwandono, Adityo. Penerapan Teori Bilangan Bulat untuk Pemeriksaan Keabsahan Nomor IMEI. <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2008-2009/Makalah2008/Makalah0809-072.pdf>
- [3] Azrina, Kania (2011). Aplikasi Teori Bilangan Pada Nomor Kartu Kredit. <http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2011-2012/Makalah2011/Makalah-IF2091-2011-069.pdf>
- [4] GSMA Association. IMEI Allocation and Approval Guidelines <http://gsmworld.com/documents/twg/tw06.pdf> diakses pada 08-12-2015 16:41

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



Anwar Ramadha 13514013