

# Penggunaan Algoritma Modulo 10 LUHN Sebagai Validator Kartu Kredit

Otniel - 13508108

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Jl Kebon bibit utara 145 – 58 RT5 RW10 Kel. Taman sari  
otniel\_itb@yahoo.com

## ABSTRAK

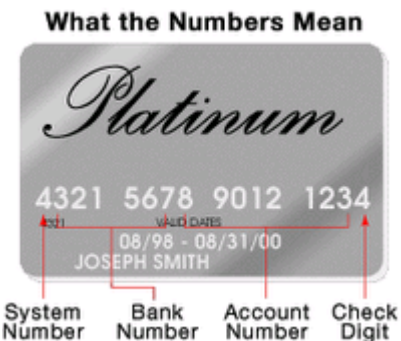
Kartu kredit adalah suatu jenis penyelesaian transaksi ritel dan system kredit yang berasal dari kartu plastic yang diterbitkan kepada pengguna system tersebut. Sebuah kartu kredit berbeda dengan kartu debit di mana penerbit kartu meminjamkan konsumen uang dan bukan mengambil uang dari rekening. Namun, untuk dapat menggunakan kartu kredit dengan aman, dibutuhkan sebuah kode pengaman kartu kredit. Kode tersebut berupa angka – angka yang mengikuti aturan tersendiri. Seperti layaknya hardware yang memiliki informasi berupa IMEI, kartu kredit juga memiliki nomor validasi yang disebut credit card number. Nomor kartu kredit baik master card maupun visa menggunakan algoritma modulo 10 atau algoritma LUHN. Dengan adanya suatu system pengkodean yang jelas, maka nomor dari kartu kredit dapat diciptakan dalam jumlah yang massif.

**Kata kunci:** LUHN, IMEI, modulo 10.

## 1. PENDAHULUAN

Sekarang ini masyarakat sudah cenderung menggunakan system transaksi elektronik dalam bertransaksi. Ada yang suka menggunakan kartu debit atau kartu kredit. Masing-masing memiliki keunggulan masing-masing. Kartu kredit, seperti telah dipaparkan di atas, adalah suatu system kredit, jadi penyedia kartu meminjamkan uang kepada pengguna, bukan mengambil uang yang terdapat dari rekening banknya.

Namun, karena kartu kredit ini dapat digunakan sebagai sarana peminjaman uang, maka dari itu dibutuhkan kode validasi kartu kredit agar proses peminjaman menjadi lebih aman. Berikut adalah gambar dari bagian-bagian kartu kredit sekaligus lokasi nomor kartu kredit (credit card number).



Gambar1. Bagian – bagian kartu kredit

Kartu kredit sebagai sarana untuk transaksi tentu dilengkapi dengan berbagai alat identifikasi. Bagian depan kartu kredit terdiri dari nama bank sebagai penerbit, nomor kartu (gambar2), logo kartu, nama pemegang kartu dan masa berlaku kartu kredit. Bagian belakang kartu kredit di lengkapi dengan pita magnetik yang di sebut dengan magstripe (gambar 3). Pita magnetik terdiri dari batangan magnet dalam ukuran mili yang disusun sejajar sepanjang kartu. Setiap merchant memiliki alat verifikasi elektronik yang bisa mengidentifikasi kartu kredit apakah masih berlaku atau tidak. Sesuatu yang menjadi patokan pemverifikasian dari kartu kredit adalah kombinasi-kombinasi angka dibelakangnya. Kode – kode tersebut dapat dibangkitkan dengan algoritma LUHN atau yang disebut juga algoritma modulo 10.

## 2 Aritmatika Modulo

Dalam matematika, aritmatika modular adalah suatu system aritmatika untuk integer atau bilangan bulat . Perhitungan ini adalah perhitungan yang menghasilkan sisa pembagian (ditulis  $a \text{ mod } b$ ). Misal,  $5 \text{ mod } 2$  mnghasilkan nilai satu karena sisa pembagian dari 5 bagi 2 adlah satu. Perhitungan aritmatik semacam ini terbatas pada bilangan bulat saja. Meskipun kemampuannya terbatas pada bilangan bulat saja, namun perhitungan

macam ini dapat dikembangkan menjadi sebuah algoritma, misalnya algoritma pengecekan kartu kredit. Aritmatika modulo dapat dicontohkan sebagai berikut : missal, a dan m bilangan bulat, ( $m > 0$ ). Operasi  $a \bmod m$  memberikan sisa jika a dibagi dengan m. Biasanya, aritmatika modulo ini disebut juga jam aritmatika karena penggunaan yang umum dari aritmatika modula adalah jam dengan system 24 jam.

## 2.1 Kekongruenan

Jika dua buah bilangan a dan b yang dibagi dengan bilangan m memiliki sisa yang sama, dikatakan a dan b kongruen dalam modulo m, Dapat kita tuliskan seperti berikut

$$a \equiv b \pmod{m} \quad (1)$$

Jika a tidak kongruen dengan b dalam modulus m, maka dapat dituliskan seperti berikut

$$a \not\equiv b \pmod{m} \quad (2)$$

Misalnya,  $38 \bmod 5 = 3$  dan  $13 \bmod 5 = 3$  maka dikatakan  $38 \equiv 13 \pmod{5}$

$a \equiv b \pmod{m}$  dalam bentuk sama dengan dapat dituliskan sebagai berikut

$$a = b + km \quad (k \text{ adalah bilangan bulat})$$

## 3 Cara Kerja Kartu Kredit

Kartu kredit umumnya menggunakan ISO 7812 sebagai standard internasional penomoran kartu identitas. **ISO 7812**, pertama kali diterbitkan oleh ISO International Organization for Standardization pada tahun 1989, adalah mengenai pengaturan standar internasional pita magnetik kartu identitas, kartu-2 akses, kartu ATM, dan kartu kredit. ISO 7812 disediakan dalam dua bagian, ISO 7812-1 dan ISO 7812-2. masing-2 berisikan spesifikasi lengkap dimana dapat terpakai pada penggunaannya. penomoran kartu kredit adalah bagian dari penomoran ISO 7812.

Pada ISO 7812 penggunaan satu digit untuk MII major industry identifier , enam digit untuk IIN issuer identifier number, disusul rangkaian angka dari sebuah nomor rekening, dan satu digit kemudian dipakai untuk pengontrolan sebagian besar industri pengidentifikasi merupakan bagian dari penerbit nomor pengidentifikasi, dan besaran jumlah angka tidak lebih dari 19 digit

### 3.1 Major Industry Identifier

MII *major industry identifier* pengaturan pada digit pertama dari penomoran menurut ISO 7812 adalah pernyataan tujuan penggunaan atau pemakaian dari kartu adalah sbb

Nilai Digit MII	Panjang Nomor
0	ISO/TC 68 and other industry assignments
1	Airlines
2	Airlines and other industry assignments
3	Travel and entertainment
4	Banking and financial
5	Banking and financial
6	Merchandising and banking
7	Petroleum
8	Telecommunications and other industry assignments
9	National assignment

umpama [MII] *major industry identifier* adalah angka 9 maka tiga angka kemudian adalah menerangkan asal negara dengan pemakaian ketentuan dari ISO 3166-1 yang terdiri dari tiga angka.

### 3.2 Issuer Identifier Number (IIN)

Tabel 3 deskripsi MII

Jenis kartu	Jumlah angka	Prefix (IIN)	Symbol	Rumus Pengontrolan
American Express	15	34/37	AmEx	Luhn Algorithm
Diners Club Carte Blanche	14	300, 301, 302, 303, 304, 305	DC-CB	Luhn Algorithm
Diners Club International	14	36	DC-Int	Luhn Algorithm
Diners Club Enroute	14	2014-2149	dc-Er	Luhn Algorithm
Diners Club US dan Canada	16	55	DC-UC	Luhn Algorithm
Maestro(Debit Card)	16,18	5020, 5038, 6304, 6759	Maes	Luhn Algorithm
Discover Card	16	6011,65	Disc	Luhn Algorithm
Master Card	16	51,52,53,54,55	MC	Luhn Algorithm
Visa Electron	16	417500, 4917, 4913, 4508, 4844	Visa	Luhn Algorithm
Visa	13,16	4	Visa	Luhn Algorithm

Tabel 2 deskripsi MII

### 3.3 Nomor Rekening

Keberadaan nomor rekening berjumlah maksimum 7 angka digit berada pada urutan kedua dari belakang setelah digit pengontrolan.

### 3.4 Digit Pengontrolan

Angka digit pengontrolan check digit berada pada urutan paling akhir dan dihitung berdasarkan perumusan Algoritma Luhn.

### 4 Algoritma LUHN

Ada tiga langkah untuk membuktikan apakah suatu kartu kredit memenuhi algoritma cek digit luhn atau tidak. Untuk kartu dengan jumlah digit genap (mis Visa [16 digit], MasterCard[16 digit], dan Novus[16 digit]) caranya sebagai berikut :

1. untuk setiap digit pada posisi yang ganjil (saya menghitung digit mulai dari kiri dengan yang paling kiri adalah digit ke-1), kalikan nilainya dengan dua, jika hasilnya lebih dari 9, kurangi hasilnya dengan 9. Jumlahkan semua angka yang telah didapat itu.
2. untuk setiap digit pada posisi genap, jumlahkan semua nilainya dan tambahkan hasilnya dengan hasil langkah pertama.
3. jika hasil pada langkah kedua habis dibagi 10, berarti nomor kartu tersebut sah

Untuk kartu dengan jumlah digit yang ganjil, (misalnya Visa [13 digit], Amex [15 digit]) caranya sama, hanya saja pada langkah pertama yang dikalikan adalah digit pada posisi genap, dan pada langkah kedua yang dijumlahkan adalah digit pada posisi ganjil

Contoh, 7889-8594-5435-5413

dan saya ingin tahu apakah angka ini benar-benar nomor kartu kredit yang sah atau tidak saya bisa melakukan langkah sbb (perhatikan jumlah digit ada 16):

1) mengalikan semua angka pada digit ganjil dengan dua dan mengurangkan hasilnya dengan 9 jika lebih dari 9, lalu dijumlahkan

inilah digit-digit pada posisi ganjil

- D01 = 7
- D03 = 8
- D05 = 8
- D07 = 9
- D09 = 5
- D11 = 3

- D13 = 5
- D15 = 1

Jika dilakukan operasi kali dua dan kurangi 9 (jika lebih dari 9) didapat

- D01' = 7 x 2 = 14, karena lebih dari maka hasilnya dikurangi 9 D02' = 14 - 9 = 5
- D03' = 8 x 2 = 16, karena lebih dari maka hasilnya dikurangi 9 D02' = 16 - 9 = 7
- D05' = 8 x 2 = 16, karena lebih dari maka hasilnya dikurangi 9 D02' = 16 - 9 = 7
- D07' = 9 x 2 = 18, karena lebih dari maka hasilnya dikurangi 9 D02' = 18 - 9 = 9
- D09' = 5 x 2 = 10, karena lebih dari maka hasilnya dikurangi 9 D02' = 10 - 9 = 1
- D11' = 3 x 2 = 6, kurang dari 9 jadi tetap 6
- D13' = 5 x 2 = 10, karena lebih dari maka hasilnya dikurangi 9 D02' = 10 - 9 = 1
- D15' = 1 x 2 = 2, kurang dari 9 jadi tetap 2

Jumlah D01' + D03' + D05' + D07' + D09' + D11' + D13' + D15' = 5 + 7 + 7 + 9 + 1 + 6 + 1 + 2 = 38

2) Menjumlahkan semua digit pada posisi genap

- D02 = 8
- D04 = 9
- D06 = 5
- D08 = 4
- D10 = 4
- D12 = 5
- D14 = 4
- D16 = 3

Jumlah = 8 + 9 + 5 + 4 + 4 + 5 + 4 + 3 = 42

Bila hasil langkah 1 dan langkah 2 dijumlahkan didapat 38 + 42 = 80, karena 80 habis dibagi 10 maka nomor tersebut sah sebagai nomor kartu kredit.

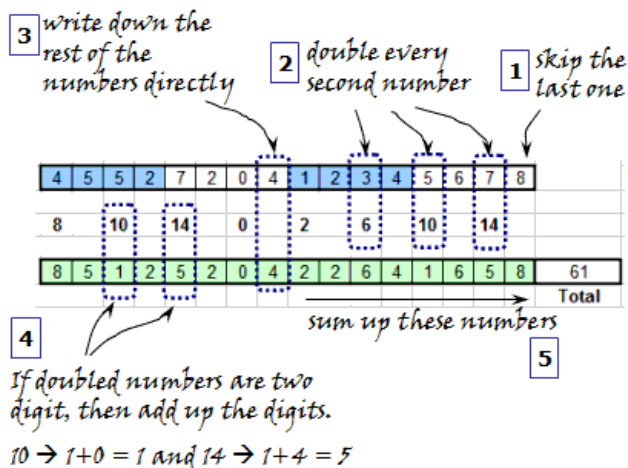
Berikut adalah spesifikasi dari jeni-jenis kartu kredit.

**Tabel 4 Spesifikasi kartu kredit**

Prefix	Panjang Nomor	Jenis Kartu
1800	15	JCB
2131	15	JCB
300	14	Diners
301	14	Diners
302	14	Diners
303	14	Diners
304	14	Diners
305	14	Diners
34	15	Amex
36	14	Diners
37	15	Amex

38	14	Diners
3	16	JCB
4	13/16	Visa
51	16	MasterCard
52	16	MasterCard
53	16	MasterCard
54	16	MasterCard
55	16	MasterCard
56	16	BankCard
6011	16	Discover

Berikut ini adalah ilustrasi dari pengecekan dogot dengan algoritma luhn atau modulo 10.



Gambar2. Ilustrasi algoritma modulo 10 atau LUHN

Berikut ini adalah kode dalam bahasa C yang merepresentasikan pengecekan algoritma pengecekan digit angka pada kartu kredit dengan algoritma LUHN.

```

Luhn.c
#include <ctype.h>
#include <stdio.h>
#include <string.h>

static int isValidNumber(const char *);

/*
 * Test harness for an implementation
 of the Luhn algorithm that checks the
 * validity of a credit card number.
 */
int
main(int argc, char *argv[])
{
    int i;

    if (argc < 2) {
        fprintf(stderr, "Usage: luhn
<number>, ... \n");
        return 1;
    }

    for (i = 1; i < argc; ++i)

```

```

        printf("Number '%s' is%s a
valid credit card number\n",
            argv[i],
            isValidNumber(argv[i]) ? "" : " not");

    return 0;
}

/*
 * Checks whether a string of digits is
 a valid credit card number according to
 * the Luhn algorithm.
 *
 * 1. Starting with the second to last
 digit and moving left, double the value
 * of all the alternating digits.
 For any digits that thus become 10 or
 more,
 * add their digits together. For
 example, 1111 becomes 2121, while 8763
 * becomes 7733 (from (1+6)7(1+2)3).
 *
 * 2. Add all these digits together.
 For example, 1111 becomes 2121, then
 * 2+1+2+1 is 6; while 8763 becomes
 7733, then 7+7+3+3 is 20.
 *
 * 3. If the total ends in 0 (put
 another way, if the total modulus 10 is
 0),
 * then the number is valid
 according to the Luhn formula, else it
 is not
 * valid. So, 1111 is not valid (as
 shown above, it comes out to 6), while
 * 8763 is valid (as shown above, it
 comes out to 20).
 */
static int
isValidNumber(const char *number)
{
    int n, i, alternate, sum;

    if (!number)
        return 0;

    n = strlen(number);

    if (n < 13 || n > 19)
        return 0;

    for (alternate = 0, sum = 0, i = n
- 1; i > -1; --i) {
        if (!isdigit(number[i]))
            return 0;

        n = number[i] - '0';

        if (alternate) {

```

```

        n *= 2;
        if (n > 9)
            n = (n % 10) + 1;
    }
    alternate = !alternate;

    sum += n;
}

return (sum % 10 == 0);
}

```

#### 4.1. Pembangkitan Kode LUHN

Sebuah sistem yang melibatkan aturan LUHN harus dapat memperbanyak kode tersebut. Namun, jika kita membuat varian dari nomor tersebut dengan cara manual akan menghabiskan waktu yang sangat lama tentunya. Oleh karena itu, harus ada suatu algoritma yang dapat membangkitkan nomor kartu kredit. Tentunya, pembangkitan tersebut harus memenuhi algoritma cek digit luhn.

Berikut ini adalah sebuah contoh pembangkitan IMEI number dengan algoritma LUHN.

##### IMEI Generator

```

// Javascript code copyright 2009 by
Fiach Reid : www.webtropy.com
// This code may be used freely, as
long as this copyright notice is intact.
function Calculate(Luhn)
{
    var sum = 0;
    for (i=0; i<Luhn.length; i++ )
    {
        sum +=
parseInt(Luhn.substring(i,i+1));
    }
    var delta = new Array
(0,1,2,3,4,-4,-3,-2,-1,0);
    for (i=Luhn.length-1; i>0; i-=2
)
    {
        var deltaIndex =
parseInt(Luhn.substring(i,i+1));
        var deltaValue =
delta[deltaIndex];
        sum += deltaValue;
    }
    var mod10 = sum % 10;
    mod10 = 10 - mod10;
    if (mod10==10)
    {
        mod10=0;
    }
    return mod10;
}

```

```

function Validate(Luhn)
{
    var LuhnDigit =
parseInt(Luhn.substring(Luhn.length-
1,Luhn.length));
    var LuhnLess =
Luhn.substring(0,Luhn.length-1);
    if
(Calculate(LuhnLess)==parseInt(LuhnDigit
))
    {
        return true;
    }
    return false;
}

```

#### 4.2. Penerapan LUHN Pada Aplikasi Online

Dalam suatu transaksi, dibutuhkan sebuah alat yang dapat mengamankan proses transaksi. Misal pada sebuah sistem belanja online, tentunya dibutuhkan sistem yang dapat mengidentifikasi kevalidan kartu kredit tentunya dengan mengeceknya dengan algoritma LUHN. Selain pada situs belanja, situs seperti bank onlinepun seperti paypal menggunakan pengecekan kartu kredit.

#### 5. KESIMPULAN

Suatu deretan angka dapat dikodekan menjadi sebuah kode verifikasi dengan menggunakan perhitungan modulo 10. Dengan menggunakan table – table yang ada di halaman atas, kita dapat menentukan dan membuat suatu alat pemverifikasian kode untuk beberapa macam tipe kartu kredit. Tiap – tiap macam kartu memiliki kode tersendiri, misalnya yang terdapat pada table 4, untuk setiap jenis kartu memiliki kode prefix yang berbeda – beda.

Dengan adanya suatu system pengecekan nomor kartu kredit, dapat memudahkan dan membantu dalam melakukan transaksi online. Selain itu, terdapat standar yang jelas terhadap penomoran kartu kredit sehingga setiap orang yang ingin membuat took online dapat dengan mudah menciptakan suatu kode pengidentifikasian nomor kartu kredit.

#### REFERENSI

- [1] <http://www.klik-kanan.com>, “Algoritma cek digit LUHN”
- [2] Wikipedia, “ISO 7812”
- [3] Wikipedia, “LUHN Algorithm”
- [4] <http://berita-ipetek.blogspot.com>, cara kerja kartu kredit
- [5] <http://www.chriswareham.demon.co.uk>
- [6] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung