

IMEI dan Validasinya dengan Algoritma Luhn

Matthew Wangsadiredja – NIM : 13507012

Jurusan Teknik Informatika ITB, Bandung 40135, email: if17012@students.if.itb.ac.id

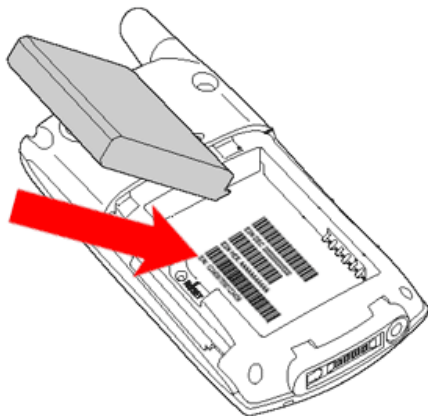
Abstract – Makalah ini membahas *International Mobile Equipment Identity* yang biasa disingkat menjadi *IMEI*[3] (arti harafiah bahasa Indonesia: identitas internasional perangkat bergerak) dan cara validasi keabsahan suatu *IMEI*. Suatu perangkat bergerak seperti ponsel GSM dan UMTS memiliki masing-masing *IMEI* yang unik. *IMEI* ini memiliki deretan angka 14 digit ditambah 1 karakter uji (check digit). Keabsahan suatu *IMEI* dapat dihitung dengan menggunakan algoritma Luhn. Selain itu setiap *IMEI* juga memiliki arti pada masing-masing bagian angkanya.

Kata Kunci: *IMEI, ponsel, karakter uji, algoritma Luhn.*

1. PENDAHULUAN

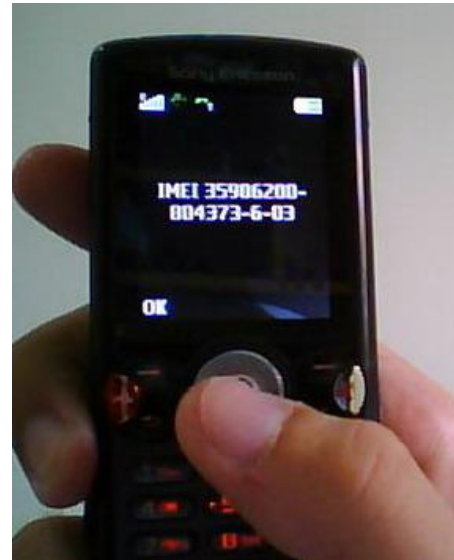
IMEI atau *International Mobile Equipment Identity* yang selama ini kita tau hanya untuk mengecek saat transaksi jual beli suatu ponsel ternyata memiliki banyak arti dan syarat penulisannya. Awalnya sertifikasi untuk *IMEI* dipegang oleh European R&TTE Directive, namun sejak April 2000, *BABT* (*British Approvals for Telecommunications*) telah memegang sertifikasi untuk *IMEI* yang resmi[6]. Dari saat itulah, dimulai era setiap ponsel yang hingga kini memiliki *IMEI* yang unik.

IMEI pada ponsel umumnya tercetak di badan ponsel di sebelah bawah baterai.



Gambar 1: Contoh lokasi *IMEI* pada sebuah ponsel

Selain dengan melihat ke bawah baterai, *IMEI* suatu ponsel juga dapat dilihat dengan menekan tombol *#06# dari ponselnya langsung[2].



Gambar 2: Contoh keluaran dari *#06# pada sebuah ponsel

Sebaliknya, jika kita mengetahui suatu nomor *IMEI*, kita juga dapat mengetahui berbagai informasi mengenai ponsel yang memiliki *IMEI* tersebut. Sebagai contoh, *IMEI* 35906200-804373-6 (03 adalah digit SVN, atau software version number yang akan dijelaskan di bab selanjutnya) memiliki deskripsi perangkat sebagai berikut

Information on IMEI 359062008043736	
Type Allocation Holder	SonyEricsson
Mobile Equipment Type	SonyEricsson W810i
GSM Implementation Phase	2/2+
IMEI Validity Assessment	Very likely
Information on range assignment	
Est. Date of Range Issuance	Around Q1 2006
Reporting Body	British Approvals Board of Telecommunications (BABT)
Primary Market	Europe
Legal Basis for Allocation	EU R&TTE Directive

Tabel 1. Informasi dari *IMEI* 359062008043736

Cara untuk mendapatkan keterangan dari IMEI ada beberapa cara, namun yang termudah adalah dengan menggunakan perangkat yang disediakan *INTERNATIONAL numbering plans*, yakni dengan membuka halaman web di alamat <http://www.numberingplans.com/?page=analysis&sub=imei> [4] dan memasukan IMEI yang hendak diperiksa.

2. IMEI

Sebuah IMEI tidak dibuat dengan sembarangan, melainkan memiliki struktur yang teratur sesuai. IMEI yang standar memiliki 14 digit angka ditambah 1 digit karakter uji. Sedangkan IMEISV memiliki 16 digit, yaitu ditambah 2 digit SVN atau *Serial Version Number*. Namun, pada makalah yang akan dibahas adalah IMEI standar dengan 14 digit ditambah 1 digit karakter uji. Setiap digit dari IMEI pun bukan merupakan deretan angka yang unik saja, namun juga memiliki arti dibalik deretan-deretan angka tersebut.

2.1. Struktur IMEI

Sejak tahun 2004, struktur IMEI dibentuk menjadi AA-BBBBBB-CCCCC-D, meski belum tentu ditampilkan seperti ini. Struktur IMEI ini seperti yang dispesifikasikan pada 3GPP TS 23.003[1].

Berikut adalah tabel mengenai struktur IMEI

AA	BBBBBB	CCCCC	D
Laporan pengenalan tubuh ponsel	Deskripsi dari TAC (Type Allocation Code)	Seri kode yang menunjukkan model ponsel	Karakter uji untuk algoritma Luhn

Tabel 2. Deskripsi Struktur IMEI

Sebagai contoh, untuk IMEI 359062008043736 akan di jelaskan secara per bagian. Pertama-tama, diambil 2 digit pertama, yaitu 35. 35 adalah kode bahwa IMEI ini terdaftar di BAPT (*British Approvals for Telecommunications*). Selanjutnya 9062 adalah nomor alokasi TAC yang diberikan oleh BAPT. 00 menunjukkan kode FAC (Final Assembly Code) yang sejak 1 January 2003 seluruh perangkat memiliki FAC 00. Sedangkan 804373 adalah nomor serial ponsel yang menunjukkan bahwa ponsel dengan IMEI ini adalah ponsel merk SonyEricsson dengan model W810i, dirakit pada sekitar quartal pertama tahun 2006. Sedangkan 6 adalah karakter uji dari untuk algoritma Luhn.

Berikut adalah tabel keterangan mengenai ponsel dengan IMEI 359062008043736.

Information on number format	
Full IMEI Presentation	359062-00-804373-6
Reporting Body Identifier	35
Type Allocation Code	35906200
Serial Number	804373
Check Digit	6

Tabel 3. Informasi dari IMEI **359062008043736**

2.2. CEIR (*Central Equipment Identity Register*)

Salah satu kegunaan IMEI adalah untuk melaporkan ponsel yang hilang akibat dicuri, lalu mendaftarkan ke CEIR. Pada kasus kehilangan, disertai dengan bukti yang kuat, maka sang pemilik ponsel dapat memblokir ponsel dengan IMEI tersebut. Hal ini disebabkan, semua IMEI telah disimpan pada basis data CEIR. Sehingga setiap ponsel yang sudah didaftarkan untuk di masukan ke list ponsel curian tidak akan dapat digunakan lagi dengan kartu SIM apapun.

Namun, untuk beberapa kasus, ada alat khusus yang dapat merubah IMEI secara paksa, maka dengan begitu ponsel dapat kembali digunakan seperti biasa[7].

3. ALGORITMA LUHN

Algoritma Luhn juga yang dikenal sebagai algoritma modulus 10 adalah sebuah algoritma yang memeriksa keabsahan suatu deretan angka dengan algoritma modulus. Algoritma Luhn dipakai untuk memeriksa keabsahan suatu deretan angka yang harus benar-benar valid.

3.1. Sejarah dan Penggunaan Algoritma Luhn

Algoritma Luhn dibuat oleh seorang ilmuwan IBM, yaitu Hans Peter Luhn dan dipatenkan di paten Amerika pada 23 Agustus 1960[8]. Sejak saat itu, algoritma Luhn ini digunakan untuk memvalidasi suatu nomor kartu kredit dan nomor identitas pemerintahan di beberapa negara. Setelah itu, kemudian muncul nomor IMEI untuk ponsel, dan algoritma Luhn juga digunakan untuk memvalidasi nomor IMEI. Dengan algoritma Luhn, setiap kesalahan baik 1 digit pun akan terdeteksi.

3.2. Algoritma Luhn dan cara penggunaannya

Suatu deretan angka termasuk karakter ujinya akan memenuhi syarat dari algoritma Luhn bila memenuhi ujicoba berikut:

1. Menghitung mulai dari digit terkanan dan bergerak ke kiri. Lalu, setiap digit genap dari kanan, nilai nya dikali dua. Setelah itu, hasil nilai yang bernilai lebih besar dari 10, kedua

- angkanya dijumlahkan lagi.
- Jumlahkan semua angka yang telah melewati proses 1 tersebut.
 - Nilai penjumlahan dari no 2 jika dimodulus 10 harus kongruen dengan 0, yakni harus merupakan kelipatan 10. Sehingga bila nilai jumlah yang diperoleh adalah 60,70,80, dsb., maka nilai deretan digit tersebut adalah valid memenuhi syarat algoritma Luhn

Sebagai contoh, untuk menghitung digit 490154203237518 apakah valid menurut algoritma Luhn, maka akan dibuat sebuah ujicoba berikut:

- (**8**; 1x2=**2**; **5**; 7x2=14→1+4=**5**; **3**; 2x2=**4**; **3**; 0x2=**0**; **2**; 4x2=**8**; **5**; 1x2=**2**; **0**; 9x2=18→1+8=**9**; **4**) maka angka baru yang diperoleh adalah 490258203435528
- Nilai yang diperoleh dari proses no 1 dijumlahkan, yaitu:
4+9+0+2+5+8+2+0+3+4+3+5+5+2+8 = **60**
- Karena hasil penjumlahan no 2 merupakan kelipatan 10 ($60 \bmod 10 \equiv 0$) maka deretan angka 490258203435528 valid memenuhi algoritma Luhn.

Dewasa ini, karena semakin canggihnya dunia bidang informatika, maka dibuatlah algoritma untuk dapat langsung memeriksa apakah sebuah deretan digit memenuhi algoritma Luhn atau tidak. Untuk memeriksanya, dapat mengakses alamat web <http://www.intermedia-online.com/luhn/#checkform> [9] dan memasukkan deretan angka yang mau diperiksa. Algoritmanya dalam bahasa *python* adalah sebagai berikut[11]:

```
def
cardLuhnChecksumIsValid(DIGITS):
    """ checks to make sure that
    the card passes a luhn mod-10
    checksum """

    sum = 0
    num_digits = len(DIGITS)
    oddeven = num_digits & 1

    for count in range(0,
num_digits):
        digit =
int(DIGITS[count])

        if not (( count & 1 ) ^
oddeven ):
            digit = digit * 2
        if digit > 9:
            digit = digit - 9
        sum = sum + digit

    return ( (sum % 10) == 0 )
```

Sedangkan algoritma Luhn yang ditulis dalam bahasa C adalah seperti berikut[10]:

```
#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. KASUS PEMBAHASAN VALIDASI IMEI DENGAN ALGORITMA LUHN

Untuk memeriksa apakah sebuah IMEI merupakan IMEI yang valid atau tidak, dapat digunakan pengujian dengan algoritma Luhn. Cara yang digunakan sama dengan cara untuk memeriksa keabsahan suatu deretan angka dengan menggunakan algoritma Luhn seperti yang sudah dijelaskan pada bab sebelumnya.

Pada contoh kasus ini, diberikan suatu persoalan dimana uji kasus yang akan dicoba adalah mengecek sebuah digit yang hilang pada IMEI jika diketahui 14 digit yang lainnya.

Dibawah ini adalah sebuah contoh kasus cara mengecek jika diketahui 14 digit awal (digit TAC dan serial nomor HP).

Contoh Kasus 1

Diketahui IMEI sebuah ponsel adalah 35906200-804373-X (X merupakan karakter uji algoritma Luhn), berapakah nilai X, agar IMEI ponsel tersebut merupakan sebuah IMEI yang valid?

Jawab

Maka, langkah pertama yang harus dilakukan adalah menjumlahkan seluruh digit yang diketahui menurut kaidah algoritma Luhn.

(X ; 3x2=6 ; 7 ; 3x2=6 ; 4 ; 0x2=0 ; 8 ; 0x2=0 ; 0 ; 2x2=4 ; 6 ; 0x2=0 ; 9 ; 5x2=10→1+0=1 ; 3) maka angka baru yang diperoleh adalah 31906400804676X

Langkah kedua adalah menjumlahkan deretan angka baru yang diperoleh setelah mendapat dari proses nol yakni:

$$3+1+9+0+6+4+0+0+8+0+4+6+7+6+X = \underline{54 + X}$$

Setelah didapat nilai 54+X, kita kembalikan ke syarat awal dari kaidah algoritma Luhn, bahwa untuk mendapatkan digit nilai yang valid, nilai penjumlahan pada proses kedua haruslah kongruen dengan 0 bila di modulus 10, atau dapat dikatakan juga nilai merupakan kelipatan 10. Sehingga pada persoalan ini, nilai X yang memungkinkan ada 6, karena 54 + 6 = 60 (kelipatan 10).

Sehingga, IMEI dari ponsel tersebut diperoleh 35906200-804373-6

Selain untuk menguji karakter uji, algoritma Luhn ini juga dapat memeriksa 1 karakter lain pada IMEI ponsel jika diketahui ke-14 digit lainnya. Berikut ini adalah sebuah contoh kasus bila 1 karakter IMEI ditengah yang hilang.

Contoh Kasus 2

David hendak mendaftarkan IMEI ponselnya yang hilang dicuri ke CEIR, namun karena box ponsel SonyEricssonnya sudah kotor, 1 digit IMEInya terhapus. IMEI yang masih terbaca adalah 357Y0401-567581-6-50 dengan Y adalah digit yang terhapus. Berapakan seharusnya IMEI yang dilaporkan David ke CEIR?

Jawab

Maka, sama dengan contoh 1, langkah pertama yang harus dilakukan adalah menjumlahkan seluruh digit yang diketahui menurut kaidah algoritma Luhn. Namun yang perlu diperhatikan di sini adalah 2 digit terakhir, yaitu -50. 50 disini adalah nilai SVN, sehingga tidak perlu dimasukkan ke dalam perhitungan.

(6 ; $1 \times 2 = \underline{2}$; 8 ; $5 \times 2 = 10 \rightarrow 1 + 0 = \underline{1}$; 7 ; $6 \times 2 = 12 \rightarrow 1 + 2 = \underline{3}$; 5 ; $1 \times 2 = \underline{2}$; 0 ; $4 \times 2 = \underline{8}$; 0 ; $Y \times 2 = \underline{2Y}$; 7 ; $5 \times 2 = 10 \rightarrow 1 + 0 = \underline{1}$; 3) maka angka baru yang diperoleh adalah 317(2Y)08025371826

Langkah kedua adalah menjumlahkan deretan angka baru yang diperoleh setelah mendapat dari proses no1 yakni:

$$3+1+7+2Y+0+8+0+2+5+3+7+1+8+2+6 = \underline{53 + 2Y}$$

Setelah didapat nilai $53+2Y$, kita kembalikan ke syarat awal dari kaidah algoritma Luhn, bahwa untuk mendapatkan digit nilai yang valid, nilai penjumlahan pada proses kedua haruslah kongruen dengan 0 bila di modulus 10, atau dapat dikatakan juga nilai merupakan kelipatan 10. Sehingga pada persoalan ini, nilai $2Y$ yang memungkinkan ada Y , karena $53 + 7 = 60$ (kelipatan 10).

Namun, karena Y bernilai 7, dipastikan hasil perkalian Y menjadi lebih dari 10 sehingga nilainya dijumlahkan. Penjumlahan bilangan dari 10 hingga 18 yang menghasilkan 7 adalah 16, oleh karena itu $2Y = 16$, sehingga diperoleh Y bernilai 8

Maka, nilai Y pada IMEI 357Y0401-567581-6-50 adalah 8. Sehingga IMEI pada ponsel david adalah 35780401-567581-6-50

5. KESIMPULAN

1. Setiap ponsel GSM memiliki IMEI yang unik dengan struktur tertentu yang memiliki arti masing-masing.
2. BAPT (*British Approvals Telecommunications*) adalah pemegang sertifikasi untuk IMEI yang resmi, selain itu setiap IMEI juga didaftarkan pada CEIR. Sehingga setiap ponsel yang hilang dicuri dapat didaftarkan ke CEIR dan di blokir.
3. Dengan memanfaatkan halaman web <http://www.numberingplans.com/?page=analysis&sub=imeinr> info ponsel bisa diperoleh jika IMEI diketahui.
4. Algoritma Luhn adalah sebuah algoritma yang memeriksa sederetan angka dan menjumlahkan dengan aturannya, dan hasil penjumlahannya harus merupakan kelipatan 10. Algoritma Luhn disebut juga algoritma modulus 10.
5. Deretan angka IMEI merupakan deretan angka yang memenuhi persyaratan dari algoritma Luhn, dan angka terakhirnya (ke 15) disebut karakter uji Luhn.

DAFTAR REFERENSI

- [1] Wikipedia, *International Mobile Equipment Identity* <<http://en.wikipedia.org/wiki/IMEI>>, diakses tanggal 31 Desember 2008 pukul 11.24
- [2] *Wireless Technology Terms, Glossary and Dictionary* <<http://www.javvin.com/wireless/IMEI2.html>>, diakses tanggal 31 Desember 2008 pukul 11.25
- [3] GSM Security, *What is an IMEI?*, <<http://www.gsm-security.net/faq/imei-international-mobile-equipment-identity-gsm.shtml>>, diakses tanggal 31 Desember 2008 pukul 11.25
- [4] International Numbering Plan, *Analysis of IMEI numbers*, <<http://www.numberingplans.com/?page=analysis&sub=imeinr>>, diakses tanggal 31 Desember 2008 pukul 11.25
- [5] BAPT, *about us*, <<http://www.babt.com/about-us.asp>>, diakses tanggal 31 Desember 2008 pukul 13.00
- [6] Wikipedia, *British Approvals Board for Telecommunications* <http://en.wikipedia.org/wiki/British_Approvals_Board_for_Telecommunications>, diakses tanggal 31 Desember 2008 pukul 13.00

- [7] Wikipedia, *Central Equipment Identity Register* <http://en.wikipedia.org/wiki/Central_Equipment_Identity_Register>, diakses tanggal 31 Desember 2008 pukul 13.00
- [8] Wikipedia, *Luhn_algorithm*, <http://en.wikipedia.org/wiki/Luhn_algorithm>, diakses tanggal 31 Desember 2008 pukul 13.00
- [9] InterMedia, *Luhn (or MOD 10) Check Algorithm* <<http://www.intermedia-online.com/luhn/#checkform>>, diakses tanggal 1 Januari 2008 pukul 11.15
- [10] <<http://www.chriswareham.demon.co.uk/software/luhn.c>> , diakses tanggal 1 Januari 2008 pukul 11.35
- [11] ActiveState Code, *Python Luhn checksum validation*, <<http://code.activestate.com/recipes/172845/>>, diakses tanggal 1 Januari 2008 pukul 11.35