APLIKASI FUNGSI *HASH* SEBAGAI GENERATOR NOMOR IDENTIFIKASI KEPENDUDUKAN

Muhammad Amrimirza – NIM : 13506003

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if116003@students.if.itb.ac.id

Abstrak

Makalah ini akan membahas tentang bagaimana mengimplementasikan fungsi Hash untuk menghasilkan rangkaian kombinasi huruf dan angka yang bisa digunakan sebagai tanda identifikasi untuk kependudukan. Fungsi Hash merupakan fungsi yang mampu menerima sebuah masukan data, yang umumnya berupa *String*, dan menghasilkan suatu data keluaran yang juga berupa *String* acak. Fungsi *Hash* yang akan digunakan adalah fungsi *Hash* yang bersifat satu arah, atau bisa dikatakan juga *non-reversibel*, yaitu data yang dimasukkan ke dalam fungsi ini tidak akan bisa diubah kembali ke bentuk semulanya.

Namun, pada makalah ini hanya akan dibatasi pada bagaimana sidik jari, yang berupa masukan pertama, diproses menggunakan fungsi *Hash* sehingga menghasilkan *String* hasil, yaitu nomor identifikasi kependudukan yang diharapkan. Untuk fungsi *Hash* yang digunakan, penulis memutuskan akan menggunakan fungsi *WHIRLPOOL*, yang akan dijelaskan secara singkat pada makalah ini. Diharapkan asil yang dihasilkan akan merupakan *String* yang unik untuk setiap sidik jari manusia yang juga unik. Sehingga menghasilkan nomor identifikasi yang unik untuk setiap individunya.

***Kata kunci**:* Fungsi *Hash, Message digest, WHIRLPOOL,* Nomor ID Kependudukan

1. Pendahuluan

Indonesia merupakan negara yang sangat besar, khususnya dalam hal jumlah penduduk. Tercatat pada data statisitik Indonesia, tahun 2005, Indonesia memiliki jumlah penduduk 218.868.791 jiwa. Namun, di negara yang besar ini timbul beberapa masalah dalam pendataan kependudukan. Yaitu salah satunya adalah tidak adanya tanda identifikasi yang unik dan tunggal dimiliki oleh seorang individu penduduk. Nomor kependudukan yang ada di Indonesia hanya berdasarkan pada tempat tinggal dan waktu di mana mereka membuat Kartu Tempat Penduduk. Hal ini menyebabkan, adanya seorang penduduk yang memiliki 2 nomor identifikasi yang berbeda.

Dalam makalah ini, penulis akan memberikan salah satu solusi agar bisa menghasilkan nomor identifikasi penduduk yang unik untuk tiap orangnya. Hal ini dapat dilakukan dengan mengaplikasikan salah satu fungsi dalam kriptografi, yaitu fungsi *Hash.* Fungsi *Hash* sebenarnya merupakan fungsi yang digunakan untuk aplikasi otentikasi dan keamanan pesan. Namun, dalam makalah ini fungsi *Hash* akan digunakan untuk tujuan berbeda, yaitu sebagai generator nomor identifikasi kependudukan.

2. Nomor Identifikasi Kependudukan

Sudah banyak negara maju yang mengimplementasikan ID kependudukan yang unik kepada penduduknya. ID ini memiliki berbagai macam nama dan cara implementasi masing-masing, tergantung dari negaranya sendiri, antara lain:

a. Di Amerika Serikat disebut sebagai *Social Security Number (SSN)*
b. Di Prancis disebut sebagai kode *INSEE*
c. Di Inggris disebut sebagai *National Insurance Number*

Pada intinya kode ini dimiliki oleh setiap penduduk, dan bersifat unik. Selain itu, kode ini diambil berdasarkan data unik dan permanen dari tiap penduduk (tergantung dari negara), sehingga menghasilkan nomor ID tunggal yang dimiliki oleh setiap penduduk.

Berbeda dengan di Indonesia, di mana seorang penduduk bisa memiliki 2 macam ID, karena basis dari pengambilan nomor ID ini bukanlah data yang bersifat permanen melekat pada penduduk, yaitu: waktu dan tempat tinggal saat pembuatan nomor ID.

Sehingga salah satu masalah yang cukup sering menjadi perhatian adalah ketika datangnya Pemilihan Umum di Indonesia. Di mana terdapat seorang penduduk yang memiliki 2 nomor identifikasi sehingga bisa menggunakan hak suaranya sebanyak 2 kali juga. Hal ini merupakan salah satu bentuk masih kurang diperhatikannya masalah nomor kependudukan di Indonesia.

Selain terhindar dari salah satu masalah di atas, ada beberapa keuntungan yang bisa diperoleh jika penduduk memiliki nomor ID yang unik, antara lain:

a. Akan kecil kemungkinan terjadinya duplikasi dalam pendataan penduduk.
b. Lebih mudah dalam pemrosesan berbagai masalah pendataan kependudukan, meliputi: pendataan pajak, pendataan jaminan kesehatan, dan sebagainya

3. Fungsi *Hash*

Fungsi *Hash* adalah fungsi yang menerima masukan *string* dengan panjang sembarang dan mengonversikannya menjadi *string* keluaran yang panjangnya tetap (umumnya berukuran lebih kecil dari ukuran *string* masukan).

Jika fungsi *Hash h* dapat menerima *string* sembarang *M*, akan dihasilkan sebuah *string* keluaran *h*. Perhatikan notasi berikut:

$$h = H(M)$$

Keluaran fungsi *Hash* ini disebut juga dengan nilai *hash* atau *message digest.*

3.1 Fungsi *Hash* satu arah

Fungsi *Hash* satu-arah (*one-way Hash*) adalah fungsi *Hash* yang bekerja dalam satu arah, atau dengan kata lain pesan yang sudah diubah menjadi *message diggest* tidak akan dapat dikembalikan lagi menjadi pesan semula.

Sifat-sifat fungsi *hash* satu-arah adalah sebagai berikut:

1. Fungsi *H* dapat diterapkan pada blok data berukuran berapa saja.
2. *H* menghasilkan nilai (*h*) dengan panjang tetap (*fixed-length output*).
3. $H(x)$ mudah dihitung untuk setiap nilai *x* yang diberikan.
4. Untuk setiap *h* yang dihasilkan, tidak mungkin dikembalikan nilai *x* sedemikian sehingga $H(x) = h$. Itulah sebabnya fungsi *H* dikatakan fungsi *hash* satu-arah (*one-way hash function*).
5. Untuk setiap *x* yang diberikan, tidak mungkin mencari $y \neq x$ sedemikian sehingga $H(y) = H(x)$.
6. Tidak mungkin mencari pasangan *x* dan *y* sedemikian sehingga $H(x) = H(y)$.

Keenam sifat di atas penting, sebab suatu fungsi *Hash* juga memiliki peran sebagai fungsi acak. Dan sebuah fungsi *Hash* dianggap tidak aman lagi jika:

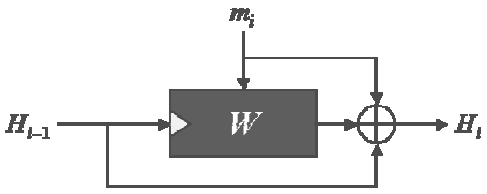
a. Secara komputasi dimungkinkan menemukan pesan yang bersesuaian dengan pesan masukannya.
b. Terjadi kolisi (*collision*), yaitu: adanya beberapa pesan berbeda yang memiliki *message digest* yang sama.

3.3 Fungsi *Hash WHIRLPOOL*

WHIRPOOL merupakan sebuah fungsi *Hash* yang dirancang oleh *Vincent Rijmen* dan *Paulo S. L. M. Barreto*. Fungsi *WHIRLPOOL* ini bekerja pada pesan dengan panjang kurang dari 2^{256} bit, dan menghasilkan *message digest* sepanjang 512 bit.

Fungsi *WHIRLPOOL* menggunakan penguatan Merkle-Damgård dan metode *Hash Miyaguchi-Preneel* dengan *block-cipher dedicated* sepanjang 512-bit yang disebut *W*. Berikut cara kerjanya:

a. Bit *string* masukan di tambahkan dengan bit-‘1’, deretan bit-‘0’, dan terakhir dengan panjang asli dari *string* masukan tersebut, hingga panjang akhir dari *string* menjadi nilai kelipatan dari 512.
b. Kemudian pesan yang dihasilkannya akan dibagi menjadi bagian-bagaian dengan panjang 512 bit (m_1, m_2, ... m_t). Kemudian pesain-pesan ini akan digunakan untuk menghasilkan sederetan *message digest* perantara (H_0, H_1, H_2, ... H_t). Dan secara definisi H_0 merupakan string bit-‘0’ sepanjang 512 bit.
c. Komputasi pada H_i dilakukan dengan mengenkripsikan m_i dengan menggunakan H_{i-1} sebagai kuncinya
d. Kemudian melakukan XOR terhadap *ciphertext* hasil tersebut dengan H_{i-1} dan m_i.
e. Terakhir hasil *message digest* fungsi *WHIRLPOOL* adalah H_t



Gambar 1. Fungsi Kompresi Miyaguchi-Preneel

Penulis memilih fungsi *WHIRLPOOL* sebagai fungsi *Hash* yang digunakan karena beberapa faktor berikut:

a. Fungsi *WHIRLPOOL* merupakan salah satu fungsi *Hash* yang hingga saat ini belum memiliki kolisi. Dan merupakan fungsi yang lolos *ISO* (*International Organization for Standardization*), dalam standar *ISO/IEC 10118-3:2004*
b. Fungsi *WHIRLPOOL* merupakan fungsi yang bersifat *open source*, sehingga lebih mudah dalam implementasinya ke depan.

3.3 Fungsi *Hash* sebagai Generator ID Kependudukan

Dalam implementasinya, fungsi *Hash* akan digunakan pada salah satu atribut tetap yang dimiliki oleh semua penduduk, yaitu sidik jari. Sidik jari merupakan sebuah tanda identifikasi unik yang dimiliki oleh semua manusia. Dan dengan memproses gambar sidik jari manusia ke dalam fungsi *Hash* akan dihasilkan deretan *string* yang bisa digunakan sebagai nomor ID penduduk, di mana nomor tersebut unik (berbeda antara satu individu dan individu lainnya).

Oleh karena sidik jari manusia itu sangat beraneka ragam, maka diperlukan sebuah fungsi *Hash* yang cukup *powerful* untuk menghasilkan *string* hasil yang juga unik antara satu sama lainnya tanpa ada kolisi. Hal inilah menjadi salah satu faktor yang menyebabkan penulis memilih fungsi *WHIRLPOOL* sebagai fungsi *Hash* yang akan digunakan untuk mengimplementasikan generator ID penduduk ini.

4. Implementasi

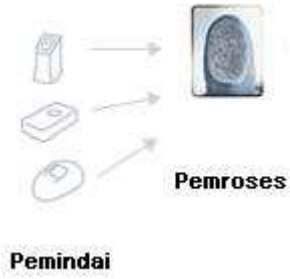
Dalam Implementasinya, Generator ID Kependudukan ini akan memiliki setidaknya 2 Modul, yaitu Modul Pembaca (*reader*) dan Modul *Hash*.

4.1 Modul Pembaca (*reader*)

Modul ini bertanggung jawab dalam pembacaan sidik jari hingga bisa menghasilkan data sidik jari dalam bentuk bit. Data yang sudah dalam bentuk bit inilah yang kemudian akan di proses dalam Modul *Hash* untuk dijadikan serangkaian *string* berupa *message digest* hasil fungsi *WHIRLPOOL*.

Modul Pembaca ini meliputi:

a. Pemindai sidik jari (*Fingerprint reader*)
   Merupakan perangkat keras yang mampu membaca sidik jari masukan.
b. Pemroses sidik jari (*Fingerprint processing*)
   Bagian yang memproses dan menjadikan sidik jari masukan menjadi data mentah yang siap digunakan dalam Modul *Hash* kemudian.



Gambar 2. Skema Modul Pembaca

Data sidik jari yang di dapat dari modul pembaca ini diharapkan akan selalu sama jika diberikan sidik jari yang sama secara konsisten (keadaan jari saat dipindai tidak akan memeberikan pengaruh pada hasil pemindaian).

4.2 Modul *Hash*

Modul ini bertanggung jawab untuk memroses data sidik jari yang sudah dalam bentuk bit untuk dijadikan *string message digest* yang akan digunakan sebagai nomor ID penduduk.

Modul *Hash* ini relatif sederhana, karena modul ini cukup menangani keluaran dari Modul Pembaca dan memasukkan data tersebut ke dalam fungsi *Hash WHIRLPOOL* sehingga akan dihasilkan *string* yang kemudian dapat digunakan sebagai ID Kependudukan.

Nomor ID Kependudukan yang akan dihasilkan dengan menggunakan fungsi *WHIRLPOOL* ini adalah *string* dengan panjang 512 bit (terdiri atas 128 karakter).

5. Kesimpulan

Kesimpulan yang dapat dimbil dari studi fungsi *Hash* ini adalah:

1. Fungsi *Hash merupakakan fungsi* yang digunakan untuk aplikasi keamanan dan otentikasi pesan. Namun, dengan karakterisitiknya yang dimilikinya, kita bisa memanfaatkan fungsi *Hash* pada hal yang sama sekali tidak berhubungan dengan hal yang disebutkan sebelumnya.
2. Sifat *string message digest* yang dihasilkan oleh fungsi *Hash* adalah unik, sesuai dengan masukan yang diberikan. Dari sifat inilah penulis bisa mengambil fungsi *Hash* sebagai generator dari ID kependudukan yang juga dituntut unik sesuai dengan masukan yang diberikan untuk menghasilkannya.

6. Saran untuk Pengembangan ke Depannya

Berikut beberapa saran yang masih perlu diperhatikan untuk pengembangan makalah ini ke depannya:

1. *Message digest* yang dihasilkan oleh fungsi *WHIRLPOOL* masih terlalu panjang untuk digunakan sebagai ID yang mudah diingat oleh pemiliknya (128 karakter). Oleh karena itu, masih diperlukan suatu cara agar *message digest* ini bisa dipersingkat, dengan tetap mempertahankan keunikannya terhadap masukan sidik jari.
2. Perlunya dilakukan uji coba hasil Implementasi generator ID kependudukan ini terhadap masukan data sidik jari dalam jumlah yang besar. Sesuai dengan target dari hasil implementasi yang akan digunakan untuk menggenerate ID penduduk yang mencapai hingga dari 200 juta jiwa lebih.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2006). Kriptografi. Program Studi Teknik Informatika, Institut Teknologi Bandung.

[2] Rijmen, Vincent, Paulo S.L.M. Baretto. (2003). *The WHIRLPOOL Hash Function* *http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html*. Tanggal akses: 20 Mei 2009 pukul 21.30 WIB.

[3] National Identification Number *en.wikipedia.org*. Tanggal akses: 18 Mei 2009 pukul 20.00 WIB

[4] *www.datastatistik-indonesia.com*. Tanggal akses: 20 Mei 2009 pukul 22.45 WIB.