Analisis Penggunaan Fungsi Hash pada Activation Key untuk Autentikasi Pendaftaran Akun

Sanrio Hernanto (13507019)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If17019@students.if.itb.ac.id

***Abstract*—Akun merupakan sarana otentikasi user untuk dapat mendapatkan layanan yang diberikan pada sebuah *website*. Pada pembuatan akun diperlukan sistem yang dapat mengamankan hal-hal yang tidak diinginkan pada pembuatan akun, misalnya saja pembuatan akun dengan cepat dan seenaknya. Salah satu cara pengamanan dalam pembuatan akun adalah dengan *activation key*. *Activation Key* sangat berguna untuk melakukan verifikasi keaslian dari data *e-mail* yang dimasukkan oleh pengguna dengan cara mengirimkan *activation key* yang telah dibuat sebelumnya melalui *e-mail*, dengan cara ini user harus melakukan *login* kedalam *e-mail* terlebih dahulu untuk dapat mengaktivasi akun. Dalam implementasinya, *activation key* ini mempunyai banyak cara dalam membuatnya. Cara yang paling sering digunakan untuk membuat *activation key* dalah dengan menggunakan *username* atau *user id* dari pengguna, menambahkan, memodifikasi atau mengenkripsi dengan kunci, dan menambahkan fungsi Hash kepada hasilnya. Dengan cara ini *activation key* yang dihasilkan akan mempunyai panjang yang sama dan *range* isi yang sama juga. Fungsi hash sendiri adalah sebuah fungsi yang menerima masukan sebuah string dan mengembalikan keluaran dengan panjang tetap. Sebuah fungsi hash mempunyai beberapa sifat yang berguna dalam pemanfaatannya sebagai sarana keamanan antara lain keluaran dengan panjang tetap, satu arah, dan anti-kolisi. Sedangkan fungsi hash yang sering digunakan adalah MD5 yang mempunyai keluaran 128bit dan SHA-1 yang mempunyai keluaran 160bit.**

***Index Terms*—Akun, *Activation Key*, Fungsi Hash, MD5, SHA-1.**

I. PENDAHULUAN

Sebuah akun pengguna memungkinkan seseorang untuk otentikasi ke layanan sistem. Hal ini juga umumnya menyediakan satu kesempatan yang akan diizinkan untuk mengaksesnya. Namun, otentikasi bukan berarti otorisasi. Sekali pengguna telah login, sistem operasi juga akan sering menggunakan suatu pengenal seperti *integer* untuk merujuk kepada mereka, bukan nama pengguna mereka. Pada sistem ini disebut pengidentifikasi pengguna atau *user id*.

Sebuah akun pengguna ada, tentunya setelah melakukan proses pendaftaran. Proses pendaftaran ini tentu sangat banyak digunakan karena banyaknya akun user pada sebuah *website*, apalagi untuk semua *website* yang ada. Karena penggunaan yang banyak ini, tentunya keamanannya patut dipertanyakan. Pada proses pendaftaran biasa, terlihat banyak celah-celah keamanan, untuk itulah diperlukan penambahan dari fungsi-fungsi keamanan dari pendaftaran akun. Karena itu, tentunya celah-celah yang ada ini ditangani dengan proses keamanan yang berbeda-beda. Salah satu proses keamanan yang sangat terlihat dilakukan adalah CAPTCHA, yaitu penggunaan *verifikasi* gambar yang diberikan. *Verfikasi* gambar ini bertujuan untuk melindungi server dari pendaftaran akun secara otomatis melalui bot yang bertujuan untuk membuat server *overload* karena akun dibuat secara simultan banyak dan cepat. Selain hal tersebut terdapat juga proses-proses keamanan yang simpel tetapi sangat berguna seperti pembatasan karakter pada username dan password, hal ini digunakan untuk mengantisipasi error-error yang dapat terjadi pada server jika username atau password yang diberikan mempunyai jumlah yang tidak sesuai. Selain itu terdapat juga konfirmasi password, konfirmasi ini digunakan untuk melindungi pengguna dari kesalahan pemasukan password karena pada saat pengguna memasukkan password, password tidak dapat dilihat langsung oleh mata sebagai pengamanan visual. Selain hal tersebut, terdapat juga pengamanan dengan inputan *e-mail*, pengamanan ini dilakukan dengan cara pembandingan *e-mail* masukan dengan data user, dan mengecek kesamaan, Dengan cara ini akun akan mempunyai hubugan satu-ke-satu terhadap *e-mail*,

sehingga mempermudah keamanan.

Hal mencolok lain yang ada pada pengamanan pada pendaftaran akun adalah keberlanjutan penanganan keamanan *e-mail*, yaitu aktivasi akun. Aktivasi akun ini digunakan untuk autentikasi keasilan dan kepemilikan dari *e-mail* user.

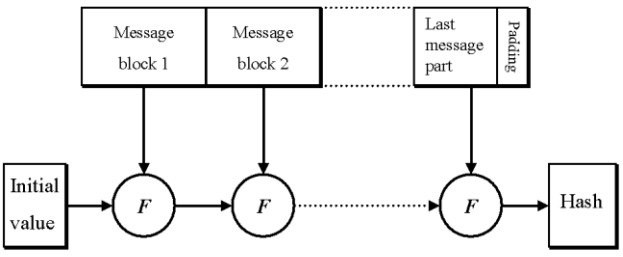
Activation Key atau bisa disebut kunci aktivasi, pada implementasinya dapat berupa berbagai macam, seperti angka random. Pada kenyataanya jika dilihat dari *website-website* saat ini, dapat dilihat kunci-kunci aktivasi yang diberikan dari berbagai macam *website* terlihat seperti hasil *message digest*. Dan dengan melihat jumlah bit kunci aktivasi makin meyakinkan penggunaan fungsi hash tersebut. Selain itu penambahan kode id pengguna dapat terlihat digunakan untuk menanangani kolisi. Dari hal-hal tersebut di atas dapat dilihat bahwa dalam pembuatan *activation key* digunakan fungsi hash yang mempunyai jumlah keluaran tertentu yang spesifik.

II. FUNGSI HASH

Sebuah fungsi hash H adalah transformasi yang mengambil sebuah m *input* dan mengembalikan sebuah string berukuran tetap, yang disebut nilai hash h atau biasa disebut *message digest*, dan bisa dilihat dengan rumus

$$h = H(m)$$

Fungsi hash dengan properti ini menggunakan berbagai komputasi umum, tapi ketika digunakan dalam kriptografi, fungsi hash biasanya dipilih karena memiliki sifat-sifat tambahannya.



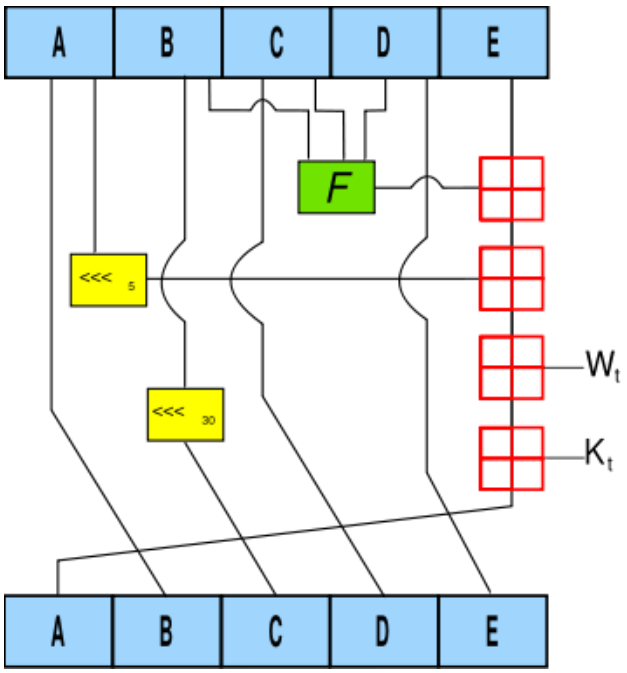
Gambar 1 - Proses Blok pada fungsi Hash

Sifat yang diperlukan pada fungsi hash untuk kriptografi adalah bebas kolisi yaitu tidak dapat menghasilkan dua *message digest* yang sama dari dua masukan yang berbeda. Keunikan *message digest* yang dihasilkan juga merupakan faktor penting dalam suatu fungsi hash dan hal ini masih terus menjadi bahan yang terus diteliti hingga sekarang karena pada kenyataanya banyak fungsi hash yang sudah ditemukan kolisinya.

Sifat lain yang diperlukan adalah fungsi hash bersifat satu arah. Ketika kita membentuk suatu *message digest* dari suatu masukan tetapi kita tidak dapat mendapatkan informasi masukan sebelum diproses dari message digestnya. Hal ini memberikan keuntungan dan kerugian pada fungsi hash. Keuntungannya adalah penggunaan fungsi hash dengan aman karena data tidak dapat diambil dari *message digest*-nya. Tetapi karena hal ini kita tidak dapat bertukar informasi dengan menggunakan fungsi hash karena tidak dapat diubah kembali menjadi data. Oleh karena inilah fungsi hash lebih sering digunakan sebagai alat autentikasi data digital, bukan untuk pertukaran data digital tersebut.

Fungsi hash sendiri banyak digunakan dalam penyimpanan *database* karena kemudahannya dalam penyimpanan beredasarkan sifatnya yang menghasikan keluaran yang panjangnya sama. Salah satu penggunaan yang sering adalah pada penyimpanan *password*. Hal ini memberikan keuntungan, yaitu panjang penyimpanan *password* yang sama dan keamanan data yang dikirim karena berupa *message digest*.

Amerika serikat memiliki lembaga yang mengeluarkan standar untuk fungsi hash yang bernama *National Institute of Standards and Technology*. Lembaga tersebut telah mengeluarkan lima standar algoritma fungsi hash, yaitu SHA-1 dan SHA-2 yang terdiri dari SAH-256, SHA-224, SHA-512, SHA-384



Gambar 2- Iterasi yang dilakukan pada SHA-1

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Word size (bits)	Rounds	Operations	Collisions found
SHA-0		160	160	512	$2^{64}-1$	32	80	+,and,or,xor,rot	Yes
SHA-1		160	160	512	$2^{64}-1$	32	80	+,and,or,xor,rot	None (2^{52} attack)
SHA-2	SHA-256/224	256/224	256	512	$2^{64}-1$	32	64	+,and,or,xor,shr,rot	None
	SHA-512/384	512/384	512	1024	$2^{128}-1$	64	80	+,and,or,xor,shr,rot	None

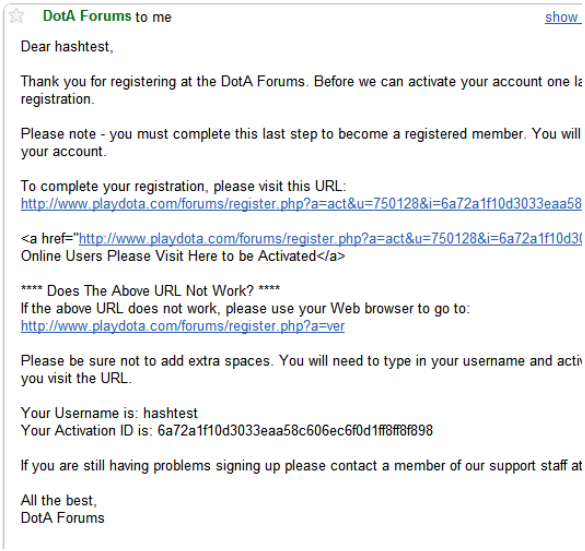
Tabel 1 - Perbandingan Algoritma Hash SHA

Algoritma	Ukuran *message digest* (bit)	Ukuran blok pesan	Kolisi
MD2	128	128	Ya
MD4	128	512	Hampir
MD5	128	512	Ya
RIPEMD	128	512	Ya
RIPEMD-128/256	128/256	512	Tidak
RIPEMD-160/320	160/320	512	Tidak
SHA-0	160	512	Ya
SHA-1	160	512	Ada cacat
SHA-256/224	256/224	512	Tidak
SHA-512/384	512/384	1024	Tidak
WHIRLPOOL	512	512	Tidak

Tabel 2 - Tabel perbandingan algoritma-algoritma hash

III. ACTIVATION KEY

Activation Key, sesuai dengan namanya, merupakan kunci untuk melakukan aktivasi pada akun. Pengaktivasian ini dilakukan agar akun yang didaftarkan dapat digunakan oleh pengguna. Kegunaan *activation* akun sendiri pada *server/host* adalah untuk melakukan salah satu pengamanan, yaitu pengamankan dan pemferifikasian terhadap *e-mail* yang digunakan oleh user. Dengan melakukan pengiriman kepada *e-mail* user, user harus melakukan aktivasi dengan membuka *e-mail*nya sehingga user tidak dapat memalsukan *e-mail* yang digunakan. Selain itu dengan melakukan pengaktivasikan akun dengan cara ini, user tidak dapat sembarangan dalam membuat akun karena *e-mail* yang digunakan harus asli dan belum dipakai dalam pembuatan akun pada tempat yang sama. Selain itu dengan membuat keharusan dalam aktivasi akun, pengguna dibatasi dengan melakukan akses terhadap *e-mail* nya sehingga membatasi kemungkinan spam akun dan batas kecepatan dalam pembuatan akun itu sendiri. Meski terlihat menyulitkan user, pengaktivasian akun sangatlah mudah dilakukan karena *website-website* yang ada pada saat ini memberikan *link* langsung yang berisi aktivasi langsung pada *e-mail* sehingga pengguna tinggal mengakses situs lewat *link* tersebut dan akun akan teraktivasi secara langsung, selain itu biasanya user akan melakukan *login* secara otomatis setelah melakukan aktivasi akun tersebut.


DotA Forums to me show

Dear hashtest,

Thank you for registering at the DotA Forums. Before we can activate your account one la registration.

Please note - you must complete this last step to become a registered member. You will your account.

To complete your registration, please visit this URL:
http://www.playdota.com/forums/register.php?a=act&u=750128&i=6a72a1f10d3033eaa58

<a href="http://www.playdota.com/forums/register.php?a=act&u=750128&i=6a72a1f10d3(
Online Users Please Visit Here to be Activated

**** Does The Above URL Not Work? ****
If the above URL does not work, please use your Web browser to go to:
http://www.playdota.com/forums/register.php?a=ver

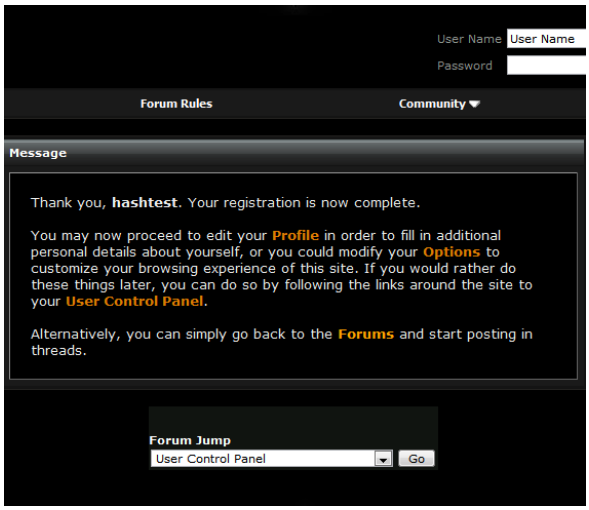
Please be sure not to add extra spaces. You will need to type in your username and activ you visit the URL.

Your Username is: hashtest
Your Activation ID is: 6a72a1f10d3033eaa58c606ec6f0d1ff8ff8f898

If you are still having problems signing up please contact a member of our support staff at

All the best,
DotA Forums

Gambar 3 - Contoh e-mail yang berisi activation key



Gambar 4 - Contoh pengaktivasian akun dengan sukses

Gambar 5 - Contoh pengaktifan akun yang gagal karena pengeditan activation key

IV. UJI ACTIVATION KEY

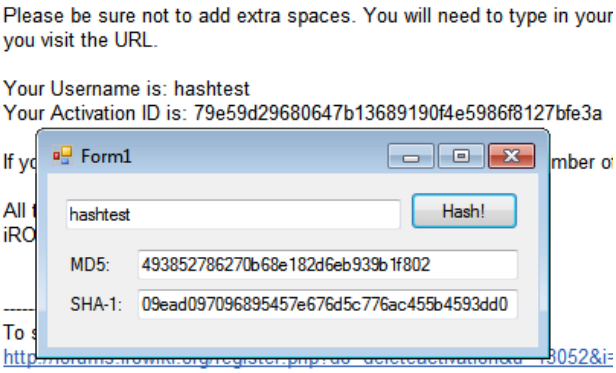
Activation Key sudah banyak digunakan di berbagai *website* yang ada pada pendaftaran akun, salah satu penggunaan yang pasti adalah pada *website* forum. Penggunaan fungsi hash pada *website-website* ini pun menggunakan variasi yang cukup beragam dan unik, untuk mengetahui lebih lanjut mengenai *activation key* yang digunakan untuk tiap *website* penulis melakukan pengujian terhadap beberapa *website* dengan mendaftar akun baru dan melihat *activation key* dari masing-masing web tersebut.

Setelah menguji beberapa *website* forum dan mendapatkan *activation key* yang diberikan, dari *activation key* yang diberikan penulis mencoba menggunakan fungsi hash terhadap masukan pada pendaftaran akun untuk melihat keterhubungan secara langsung dari input masukan user pada pendaftaran akun dengan *activation key*, fungsi hash yang digunakan dalam pengetesan keterhubungan adalah fungsi hash yang sesuai dengan jumlah bit pada activation key. Berikut adalah table hasil tes pada beberapa website.

Website	Jumlah Bit Activation Key	Keterhubungan Langsung
www.playdota.com/forum	160bit	Tidak
forums.irowiki.org	160bit	Tidak
www.mangatraders.com	128bit	Tidak
passport.asiasoft.com	128bit	Tidak
www.forum.nokia.com	160bit	Tidak
www.eathena.ws	128bit	Tidak
board.limit-ro.net	128bit	Tidak

Tabel 3 - Hasil Uji Coba Activation Key pada Beberapa Website

Dari tabel dapat dilihat bahwa penggunaan fungsi hash yang sering digunakan adalah MD5(128bit) dan SHA-1(160 bit), selain itu tidak ada keterhubungan langsung antara masukan akun dengan *activation key*. Selain itu, dengan masukan data akun yang sama pada *website* yang berbeda, dihasilkan *activation key* yang berbeda. Ini berarti pembuatan *activation key* antara suatu web dengan web lainnya berbeda, variasi perbedaan yang terjadi adalah pada *input* masukan. Dari penelitian yang dilakukan, variasi *input* yang sering digunakan adalah penambahan waktu pendaftaran pada *input* masukan dari *user*. Dengan penambahan waktu, input masukan dapat lebih unik satu sama lain, sehingga *activation key* yang dihasilkan dapat terlihat lebih acak. Selain itu, pada hampir seluruh *website* pada autentikasi akun digunakan juga *id* dari akun pada autentikasinya selain *activation key*, hal ini diperkirakan untuk menanggulangi kolisi dari fungsi hash sehingga tidak terjadi kesalahan pengaktivasian akun.



Gambar 6 - Uji coba perbandingan activation key dengan input maukan user secara langsung

V. IMPLEMENTASI ACTIVATION KEY

Pada bab ini, penulis melakukan implementasikan generator *activation key* yang berbasiskan kunci khusus pada pembuatannya. Pada pembuatan *activation key* dengan kunci khusus, hasil dari pembuatan *activation key* tidak perlu dimasukkan kedalam database sehingga dapat menghemat memori, tetapi jika diulik, kunci tersebut dapat ditemukan dengan mencoba melakukan fungsi hash pada *input* masukan dan kunci percobaan. Pada pemferifikasiannya *activation key* hanya perlu di generasi dengan melakukan fungsi hash terhadap masukan yang sudah dimodifikasi dengan kunci sebelumnya. Berikut adalah kode hasil implementasi.

index.html

```
<html>
 <head>
  <title>
   Implementation - Activation Key Generator
  </title>
  <script type="text/javascript">
```

```
        function generateAK(val)
        {
         inid.innerHTML = val;
         var qry = 'val='+val;

         //get activation key(md5)
         var xmlhttp = new XMLHttpRequest();
         var url = 'akmd5.php';
         xmlhttp.open('GET', url+"?"+qry, true);
         xmlhttp.onreadystatechange = function()
         {
          if(xmlhttp.readyState == 4)
          {
           if(xmlhttp.status == 200)
           {
            akmd5.innerHTML = xmlhttp.responseText;
           }
          }
         };
         xmlhttp.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
         xmlhttp.send(null);

         //get activation key(sha-1)
         var xmlhttp2 = new XMLHttpRequest();
         var url2 = 'aksha1.php';
         xmlhttp2.open('GET', url2+"?"+qry, true);
         xmlhttp2.onreadystatechange = function()
         {
          if(xmlhttp.readyState == 4)
          {
           if(xmlhttp.status == 200)
           {
            aksha1.innerHTML = xmlhttp2.responseText;
           }
          }
         };
         xmlhttp2.setRequestHeader('Content-Type',
'application/x-www-form-urlencoded');
         xmlhttp2.send(null);
        }
      </script>
    </head>
    <body>
      <input id="inp" name="inp" value="">
       <br/>
      <input type="button" value="generate"
onClick="generateAK(inp.value)">
       <br/>
      ID:
       <br/>
      <span id="inid" style="color:red;"></span>
       <br/>
       <br/>
      Activation Key
       <br/>
      128bit (MD5) :
       <br/>
      <span id="akmd5" style="color:red;"></span>
       <br/>
      160bit (SHA-1) :
       <br/>
      <span id="aksha1" style="color:red;"></span>
       <br/>
       <br/>
    </body>
</html>
```

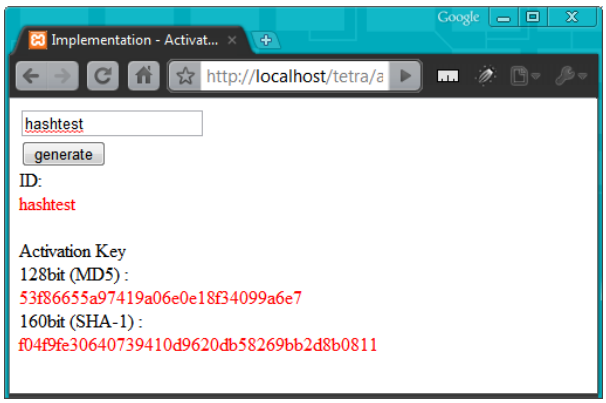
akmd5.php

```
<?php
        $val = $_GET["val"];
        $key = "actkey";
        $result = $val." ".$key;
        echo md5($result);
 ?>
```

aksha1.php

```
<?php
        $val = $_GET["val"];
        $key = "actkey";
        $result = $val." ".$key;
        echo sha1($result);
?>
```

Kode tersebut dapat digunakan untuk melakukan pembuatan *activation key* yang nantinya akan dikirimkan kepada *e-mail* user secara otomatis dengan menyertakan juga *username* dari user atau *user id* nya sebagai pengantisiapasian anti-kolisi dari *activation key*. Sedangkan untuk melakukan pemverifikasiannya tinggal melakukan generasi *activation key* dengan input yang sama dan melakukan perbandingan dengan *activation key* yang berasal dari *e-mail* user, jika sama berarti user telah terautentikasi dengan baik.



Gambar 7 - Screenshot implementasi program generator activation key

VI. KESIMPULAN

Dari hasil pengujian dan anlisis yang dilakukan, didapatkan beberapa kesimpulan sebagai berikut:

- Fungsi hash mempunyai banyak sifat yang dapat digunakan dan dimanfaatkan antara lain satu arah, anti-kolisi, dan menghasilkan *message digest* dengan panjang tetap,
- Fungsi hash merupakan cara yang tepat dan efektif dalam implementasi *activation key* dalam pengautentikasian pendaftaran akun
- Pada saat ini, hampir semua *website* yang menggunakan pendaftaran akun melakukan autentikasi pendaftaran dengan *activation key*
- Pengimplementasian *activation key* yang digunakan oleh *website-website* yang ada dilakukan dengan menggunakan fungsi hash yang kebanyakan adalah fungsi hash MD5 dan SHA-1
- Pada pengimplementasiannya website-website yang ada menggunakan variasi pada input fungsi hash yang inputnya sendiri berasal dari masukan user. Variasi ini dapat berupa penambahan atau pemodifikasian input awal dengan sebuah kunci.
- Pada implementasinya, selain *activation key*, *user id* atau *username* dikirimkan juga kepada *user* untuk menghindari kolisi yang bisa terjadi karena pada fungsi hash yang digunakan masih dapat terjadi kolisi.
- Terdapat beberapa variasi pada pengimplementasian pembuatan *activation key* antara lain waktu pembuatan akun, kunci, dan pemanipulasian data masukan pada akun dengan kunci.
- Terdapat dua perbedaan utama dalam pengimplementasian, yaitu dengan kunci dinamik (co: menggunakan waktu pembuatan) dan kunci statik (co: menggunan kunci yang telah ditentukan sebelumnya) yang masing-masing mempunyai keunggulan tersendiri yaitu lebih aman karena generasi kunci secara dinamis atau lebih hemat dalam penyimpanan database karena penggenerasian kembali yang tidak berubah-ubah pada kunci statik.

DAFTAR PUSTAKA

[1] Munir, Rinaldi. (2004). Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.
[2] http://playdota.com/forum
Waktu akses: 15 Mei 2010 pukul 10.00
[3] http://forums.irowiki.org
Waktu akses: 15 Mei 2010 pukul 10.00
[4] http://passport.asiasoft.com
Waktu akses: 15 Mei 2010 pukul 10.00
[5] http://www.forum.nokia.com
Waktu akses: 15 Mei 2010 pukul 10.00
[6] http://www.mangatraders.com
Waktu akses: 15 Mei 2010 pukul 10.00
[7] http://www.eathena.ws
Waktu akses: 15 Mei 2010 pukul 10.00
[8] http://board.limit-ro.net
Waktu akses: 15 Mei 2010 pukul 10.00
[9] http://webmasterworld.com/forum88/11371.htm
Waktu akses: 16 Mei 2010 pukul 13.00
[10] http://www.astahost.com/info.php/Activation-Code t19109.html
Waktu akses: 16 Mei 2010 pukul 13.00
[11] http://www.plus2net.com/
Waktu akses: 16 Mei 2010 pukul 13.00

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, 16 May 2010

Sanrio Hernanto, 13507019