ANDROID SMS REMOTE WIPE

Ariq Bani Hardi[1], Nugroho Jati[2]
Lembaga Sandi Negara, Jakarta
(1) ariq.bani@lemsaneg.go.id, (2)nugroho.jati@lemsaneg.go.id

Abstrak

One of smart phones attacks that have a high risk impact is physical access with take over the smart phone resources and sensitive data inside. Remote wiping is a device management feature to mitigate the risk misuse of data by an unauthorized users when smart phone has been stolen or lost. Most of smart phone manufactures provide device management feature as a Data Loss Prevention (DLP) form. Google provide the Android Device Manager feature for all of google's users too. To perform remote wipe via device manager requires internet network connections. In this paper will discuss about developing remote wipe android application using GSM SMS network. This application is expected to be an alternative method of remote wipe android smart phone and become a part of Data Loss Prevention (DLP) for smart phone users.

Key word : *Remote wipe, Data Loss Prevention (DLP), Device Management, Android.*

1. Pendahuluan

Perkembangan zaman menunjukkan sangat pesatnya pemanfaatan smart phone di kehidupan sehari-hari. Smart phone menyimpan banyak data-data pengguna yang bersifat sensitif, seperti pesan SMS, chat, foto, contact, note, email, dan file-file lain [1]. Terdapat beberapa jenis kategori serangan terhadap smart phone, dimana setiap kategori serangan memiliki tingkat kerugian dan *impact* tersendiri. Kategori serangan tersebut adalah *Malicious App, Cellular Network*, *Physical Access* berupa kehilangan atau pencurian, *Physical Access* berupa penggunaan data kembali setelah kehilangan oleh pihak yang berkepentingan, dan *Malicious Email/Web Page* [2]. *Physical Access* berupa kehilangan atau pencurian memiliki tingkat kerugian level *low*, tetapi *Physical Access* berupa pemanfaatan kembali oleh pihak lain setelah kehilangan atau pencurian memiliki tingkat kerugian level *high*. Seorang *attacker* yang sengaja mengakses smart phone target secara *physical* dapat mengambil alih seluruh resource data dan network milik target. *Attacker* juga dapat menaikan *previlege* ke dalam sistem operasi smart phone, kemudian mengambil alih seluruh aktivitas yang dapat dilakukan target pada smart phone-nya. Termasuk mengakses *credential* parameter seperti informasi akun dan password dari aplikasi-aplikasi yang ada pada smart phone.

Salah satu cara untuk mengurangi risiko penyalahgunaan data oleh pihak lain ketika terjadi kehilangan adalah dengan *remote wipe*. *Remote wipe* merupakan metode untuk mengembalikan smart phone ke posisi default atau *factory reset* secara remote dengan menghapus seluruh atau sebagian data yang tersimpan di dalam smart phone. Beberapa manufaktur smart phone menyediakan fasilitas *remote wipe* kepada penggunanya untuk menghapus data secara remote apabila terjadi kehilangan atau pencurian. Google melalui *Android Device Manager* juga memberikan fasilitas tersebut, dimana pengguna dapat melakukan *control device* secara remote untuk mendeteksi lokasi device, membunyikan *panic ring tone*, perubahan password, dan *remote wipe* itu sendiri. Pemanfaatan *remote wipe* mengharuskan device pengguna tetap terhubung dengan jarigan internet, sedangkan kondisi device yang hilang tidak dapat dipastikan apakah terhubung dengan internet atau tidak.

Pada tulisan ini akan dibahas perancangan aplikasi untuk melakukan *remote wipe* tanpa jaringan internet. Rumusan masalah yang diangkat dalam tulisan ini adalah bagaimana melakukan *remote wipe* untuk smart phone dengan sistem operasi android yang hilang dengan memanfaatkan jaringan GSM SMS. Diharapkan solusi yang ditawarkan dapat menjadi alternatif metode r*emote wipe* smart phone android sebagai bentuk *Data Loss Prevention* (DLP) pengguna.

2. Metodologi

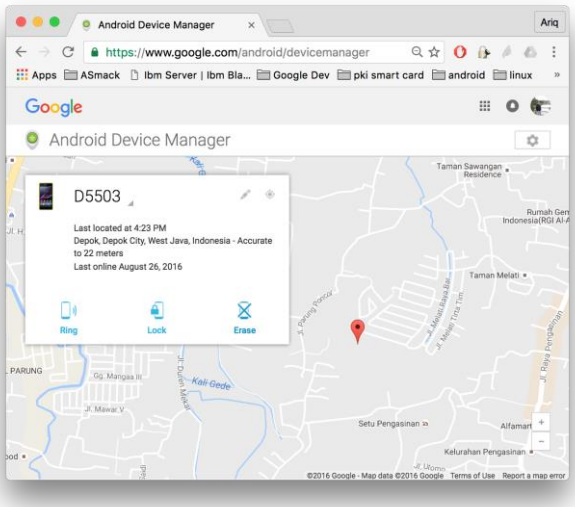
Metode yang digunakan dalam perancangan Aplikasi Android SMS *Remote Wipe* adalah dengan melakukan studi literatur dan pembuatan prototype rancangan aplikasi. Fungsi *messaging* SMS dan *remote wiping* memanfaatkan *Application Programming Interface* (API) yang disediakan oleh Google. Prototype aplikasi diujikan pada emulator smart phone Android dan perangkat android. Langkah pengembangan aplikasi dilakukan dengan pembuatan desain, implementasi, dan pengujian. Perangkat yang digunakan untuk pengembangan aplikasi adalah Android Studio dan Genymotion Android Emulator, sedangkan perangkat smart phone yang digunakan untuk pengujian adalah Samsung Galaxy A5.

3. Pembahasan

1. Remote Wipe

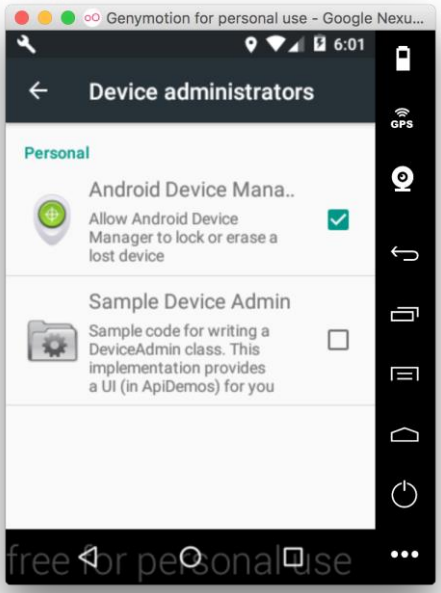
Hasil riset statistik *Annual State of The Net Survey by Consumer Reports National Researh Center* [4] menunjukkan sebanyak 34% pengguna smart phone tidak menerapkan pengamanan pada device yang digunakanya, 36 % menggunakan pengamanan minimun burupa 4 digit PIN. Sangat sedikit presentase pengguna yang memanfaatkan fitur pengamanan pada device-nya, seperti menggunakan password yang panjang, enkripsi device, atau mengaktfikan fitur *remote wipe* sebagai salah satu metode *Data Loss Prevention* (DLP) terhadap data-data pribadi di smart phone ketika terjadi kehilangan. Sedangkan pemanfaatan data personal oleh pihak yang tidak berhak pada kasus kehilangan smart phone saat ini semakin meningkat [5]. Sebagian besar pengguna tidak menyadari bahwa mereka menyimpan data-data sensitif di dalam smart phonenya seperti pesan SMS, chat, foto, contact, note, email, file-file pekerjaan, file-file perusahaan, dan lain sebagainya.

Remote wipe adalah salah satu metode untuk meminimalisir penyalahgunaan data oleh pihak lain ketika terjadi kehilangan atau pencurian smart phone pengguna [7]. *Remote wipe* merupakan bentuk penerapan *device manager* pada smart phone. Google menyediakan fitur *remote wipe* melalui *Android Device Manager* yang memungkinkan pengguna melakukan *remote device* berupa deteksi keberadaan device, membunyikan *panic ring tone*, perubahan password, dan *remote wipe* data yang ada di dalamnya. Pengguna dapat mengakses dashboard *android device manager* untuk melakukan control terhadap smart phone android yang digunakan. Berikut adalah gambar Dashboard *Android Device Manager* yang disediakan oleh Google.



Gambar 1. *Dashboard Android Device Manager* [3]

Device Android pengguna harus terhubung dengan jaringan internet agar dapat diakses melalaui *Android Device Manager*. Secara konsep Sistem Operasi Android memiliki agent berupa service pada perangkat pengguna, dimana service tersebut harus di-enable terlebih dahulu apabila pengguna ingin memanfaatkan fitur tersebut. Gambar 2 adalah settingan *enable/disable* agent *Android Device Manager* pada perangkat Android pengguna. Apabila pengguna memberikan perintah *erase / remote wipe* dari dashboard *Android Device Manager*, service agent *Android Device Manager* pada perangkat pengguna akan menjalankan perintah *factory reset* dan seluruh data yang tersimpan akan terhapus.

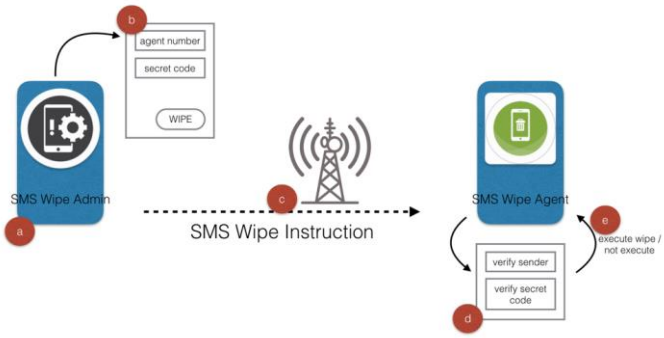


Gambar 2. *Setting Android Device Manager*

Google menyediakan berbagai macam *Application Programming Interface* (API) yang dapat dimanfaatkan oleh para pengembang untuk dapat mengembangkan aplikasi pada sistem operasi android. Beragam API yang disediakan termasuk di dalamnya API device administrator yang memiliki fungsi *wiping data*. API *wiping data* tersebut yang akan dimanfaatkan untuk membangun prototype Aplikasi Android SMS Remote Wipe pada tulisan ini.

2. Desain Aplikasi Android SMS *Remote Wipe*

Terdapat 2 (dua) entitas yang digunakan dalam desain sistem Android SMS *Remote Wipe*, yaitu Aplikasi Administrator dan Aplikasi Agent yang terinstall pada device yang berbeda. Desain sistem SMS *Remote Wipe* adalah sebagai berikut.



Gambar 3. *Desain Sistem SMS Wipe Android*

a. Pengguna melalui aplikasi administrator memberi perintah *wiping* kepada device yang dimilikinya

b. Aplikasi administrator menggenerate format SMS untuk *wiping* dengan inputan kode rahasia yang hanya diketahui oleh pengguna

c. Aplikasi administrator mengirimkan perintah *wiping* ke perangkat pengguna melalui jaringan GSM SMS

d. Aplikasi agent menerima perintah *wiping* kemudian melakukan verifikasi kode rahasia

e. Aplikasi agent menjalankan perintah *wiping* data

3. Implementasi

Pengembangan aplikasi administrator dan agent memanfaatkan IDE Android Studio dan bahasa pemrograman Java. Fungsi utama pada aplikasi administrator adalah ganerator format SMS dan SMS Sender, dan diimplementasikan pada *source code* berikut.

Generator format SMS :

```
String WipeInstruction =
       "WIPE";
String SecretCode = "Y0URS3CR3TC0D3";
WipeInstruction = WipeInstruction+ "||" +
SecretCode;
```

Fungsi generator format SMS digunakan untuk membuat format SMS yang dapat diterima oleh aplikasi agent. Format SMS terdiri dari instruksi berupa String "WIPE" yang di concate dengan kode rahasia yang hanya diketahui oleh pengguna.

SMS Sender :

```
SmsManager smsManager =
       SmsManager.getDefault();
String agentNumber = "NoTelp.DeviceAgent";
byte[] smsBody = WipeInstruction.getBytes();
short port = 1234;

smsManager.sendDataMessage(agentNumber, null,
port, smsBody, null, null);
```

Fungsi SMS Sender yang digunakan adalah binary SMS dengan memanfaatkan port custom yaitu 1234. Binary SMS merupakan salah satu bentuk SMS yang mengijinkan pengiriman berbagai *content* seperti ring tones, system setting, dan WAP Push melalui text messaging [6].

Fungsi utama pada aplikasi agent adalah SMS receiver, verifikator, dan wipe data. Aplikasi Agent mendaftarkan meta-data device admin pada file manifestnya agar aplikasi dapat menggunakan *Device Administration* API. Berikut adalah definisi XML resource pendefinisian meta-data device admin yang digunakan.

```
<?xml version="1.0" encoding="utf-8"?>
<device-admin>
    <uses-policies>
        <wipe-data />
    </uses-policies>
</device-admin>
```

SMS Receiver :

```
@Override
public void onReceive(Context context, Intent
intent) {
    Bundle bundle = intent.getExtras();
    SmsMessage[] msgs = null;
    String str = "";
       if (bundle != null){
       Object[]     pdus     =     (Object[])
          bundle.get("pdus");
       msgs = new SmsMessage[pdus.length];
       for (int i=0; i<msgs.length; i++) {
          byte[] data = null;
              msgs[i] =
          SmsMessage.createFromPdu((byte[])
             pdus[i]);
          data = msgs[i].getUserData();
          for    (int    index=0;    index    <
             data.length; index++) {
          str += Byte.toString(data[index]);
          for    (int    index=0;    index    <
             data.length; index++) {
          str   +=   Character.toString((char)
             data[index]);}
          str += "\n";
       }
```

SMS Receiver meng-*override* fungsi *onReceive* dari Class *BroadcastReceiver*. Fungsi *onReceive* digunakan untuk menghandle pesan SMS binary yang masuk ke device pengguna. SMS binary yang masuk dengan port 1234 akan diabaikan oleh aplikasi *messaging* default dari smart phone. SMS tersebut akan dihandle oleh aplikasi agent untuk diverifikasi.

SMS Verifikator :

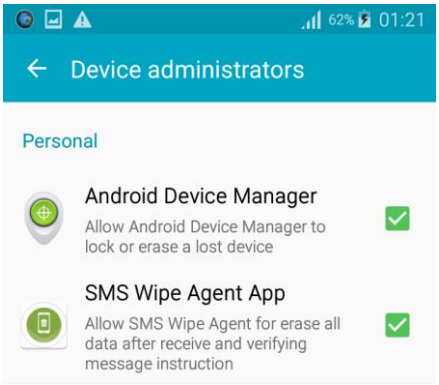
```
String adminNumber = number;
String messages = str;
if(adminNumber = SenderNumber &&
   messages.getSecretCode = SecretCode){
   wipe();
} else {
   //Do Nothing - secret code rejected
}
```

SMS Verifikator melakukan ekstraksi isi SMS yang masuk kemudian melakukan pencocokan nomor administrator dan kode rahasia yang diterima. Apabila kode rahasia yang diterima sama dengan kode rahasia yang tersimpan di dalam aplikasi agent, maka fungsi wipe akan dijalankan.

Wipe Data :

```
DevicePolicyManager mDPM;
mDPM.wipeData(0);
```

Untuk dapat menjalankan fungsi wipe data, aplikasi agent harus memiliki akses ke dalam *Device Administrator*. Manifest aplikasi harus memiliki receiver yang memiliki permission BIND_DEVICE_ADMIN. Sehingga aplikasi agent muncul di dalam setting Sistem Operasi sebagai aplikasi yang dapat mengakses device administrator android. Hasil implementasi aplikasi agent untuk mengakses device administrator ditunjukkan pada gambar berikut.

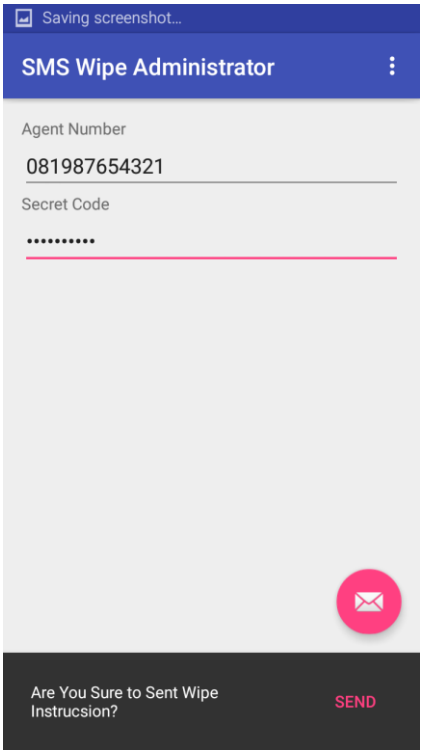


Gambar 4. *Setting Akses Device Administrator Aplikasi SMS Wipe Agent*

4. Pengujian

Pengujian prototype aplikasi dilakukan pada device Samsung Galaxy A5 dengan sistem operasi Android 5.0.2. Gambar 5 adalah aplikasi administrator untuk mengirimkan perintah wipe.

Pada pengujian yang dilakukan aplikasi administrator berhasil mengirimkan perintah wipe melalui jaringan GSM SMS kepada device yang terinstall aplikasi agent. Biaya yang digunakan untuk mengirimkan binary SMS melalui aplikasi Administrator sama dengan biaya pengiriman satu kali SMS normal sesuai tarif provider GSM yang digunakan. Aplikasi agent dapat menerima instruksi yang dikirimkan dan meverifikasi kode rahasia. Aplikasi agent berhasil menjalankan perintah wipe, sehingga seluruh data yang tersimpan di dalam smart phone terhapus dan kembali pada kondisi default (*factory reset*).



Gambar 4. *Aplikasi Administrator untuk Mengirimkan SMS Perintah Wipe*

4. Kesimpulan

Salah satu metode *Data Loss Prevention* (DLP) pada hilangnya smart phone adalah *remote wipe*. *Remote wipe* mampu menghapus seluruh data yang tersimpan di dalam sebuah smartphone secara remote dan mengembalikan kondisi smartphone pada posisi defaultnya / *factory reset*. Aplikasi Android SMS *Remote Wipe* dapat menjadi solusi alternatif *remote wipe* smart phone android dengan memanfaatkan jaringan GSM SMS. Aplikasi SMS *Remote Wipe* terdiri dari dua entitas yaitu Aplikasi Administrator dan Aplikasi Agent. Aplikasi Administrator digunakan sebagai pengirim instruksi wipe melalaui jaringan GSM SMS, sedangkan aplikasi agent ditanam di device pengguna untuk menerima instruksi, melakukan verifikasi, dan menjalankan wipe data.

Sistem yang dikembangkan masih berupa prototype aplikasi administrator dan agent. Dari hasil pengujian dan analisis terdapat beberapa materi yang dapat dikembangkan lebih lanjut, yaitu :

a. Belum adanya aplikasi manajemen administrator dan agent untuk mengelola informasi terkait nomor pengirim administrator, agent, serta kode rahasia pengguna.

b. Proses verifikasi hanya dilakukan pada kode rahasia dan nomor pengirim / administrator. Dibutuhkan metode otentikasi yang lebih baik untuk membuktikan pengirim instruksi wipe adalah pihak yang tepat.

Daftar Pustaka

[1] Ildar Muslukhov, dkk, "Know Your Enemy: The Risk of Unauthorized Access in Smartphones by Insiders," *15th International Conference on Human-Computer Interaction with Mobile Devices and Services*, Agustus, 2013.
[2] The Mitigations Group, "New Smartphones and the Risk Picture," *The Information Assurance Mission at NSA*, April, 2012.
[3] www.google.com/android/devicemanager
[4] www.consumerreports.org
[5] Laurent Simon, Ross Anderson, "Security Analysis of Consumer-Grade Anti-Theft Solutions Provided by Android Mobile Anti-Virus Apps," *4th Mobile Security Technologies Workshop*, Mei, 2015.
[6] Jeff Brown, Bill Shipman, and Ron Vetter, "SMS: The Short Messages Service," *How Things Work*, Desember, 2007.
[7] Ernst & Young, "*Data loss prevention Keeping your sensitive data out of the public domain*", Insights on governance, risk and compliance, Oktober, 2011.