Studi dan Implementasi Digital Signature untuk Fotografi pada Device Android

Danang Tri Massandy (13508051)
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
danangstei08161@students.itb.ac.id

Kegemaran masyarakat akan dunia fotografi semakin mendunia. Foto-foto yang terekam baik secara tidak sengaja maupun sengaja menjadi suatu bukti sejarah untuk masa depan nantinya. Tidak semua orang mempunyai bakat untuk mengabadikan suatu momen secara sempurna, namun tidak semua orang mempunyai niat baik untuk tidak merebut karya fotografi milik orang lain. Pada dasarnya, orang lebih suka menggunakan kamera digital saku untuk mengabadikan suatu momen. Dan apalagi saat ini, device handphone telah disisipi fitur kamera dengan kualitas setara dengan kamera digital saku tersebut. Pada kamera digital, keamanan akan foto-foto yang ada di dalamnya dirasakan masih kurang aman. Seseorang bisa saja dengan mudah menyalin foto dan menghapus foto aslinya sehingga kepemilikan akan foto tersebut berubah. Akan tetapi, untuk meningkatkan keamanan pada device seperti ini diperlukan usaha yang tinggi karena akan secara langsung berhubungan dengan hardware dan firmware dari kamera digital tersebut. Lain halnya dengan device handphone/smartphone. Aplikasi keamanan dapat dikembangkan lebih mudah di lingkungan ini. Sekarang ini, smartphone yang mempunyai peluang tinggi menjangkau pasaran adalah smartphone berbasikan android. Hal ini dikarenakan smartphone ini memiliki rata-rata harga jual yang dapat dijangkau masyarakat menengah, mudah digunakan, dan mudah dan banyak dikembangkan. Android merupakan sistem operasi yang open source sehingga banyak developer mengembangkan aplikasi-aplikasi pada sistem operasi ini. Device android sendiri sudah dilengkapi fitur-fitur yang cukup lengkap termasuk kamera beresolusi tinggi. Kamera ini cukup untuk mengabadikan suatu peristiwa yang penting sehingga keamanan akan hasilnya sangat diutamakan. Oleh karena itu, penulis mengembangkan aplikasi bernama DSDro yang merupakan aplikasi fotografi dengan fitur menambahkan digital signature pada foto hasil pemotretan. Selain itu, terdapat juga fitur autentikasi yang digunakan untuk mengautentikasi foto apakah diambil pada device tertentu asli atau bukan. Aplikasi ini diharapkan membuat karya seorang fotografer baik yang professional ataupun tidak dihargai oleh semua orang.

Kata kunci : Android, kriptografi, Digital Signature, fotografi, DSDro, kamera, autentikasi

I. PENDAHULUAN

Fotografi adalah suatu hal yang membuat masa sekarang menjadi suatu sejarah di masa depan. Dengan fotografi, kejadian-kejadian menyenangkan, baik, lucu, riang gembira, ataupun sedih dicatat dan diabadikan. Hasil fotografi adalah karya dari fotografernya sendiri. Karya-karya ini tentunya harus dijaga dan dihormati karena suatu hal yang dinamakan karya adalah suatu perjuangan dan pengorbanan dari orang.

Kejahatan dalam dunia fotografi adalah merampas hak/karya dari seseorang. Orang lain mengaku bahwa foto tersebut miliknya, padahal yang memotret bukanlah dirinya sendiri. Kasus-kasus seperti ini lebih sering terdengar di jurnalisme. Seorang wartawan/orang biasa yang sudah susah payah mengabadikan suatu kejadian penting, ternyata hasilnya dimiliki orang lain secara paksa. Dengan begitu, orang tersebut meraih kepopuleran ataupun keuntungan lainnya dari hasil fotografi tersebut.

Fotografi paling mudah dilakukan dengan menggunakan kamera pada *handphone*. Kamera-kamera yang disediakan *handphone* saat ini sudah memiliki resolusi yang tinggi sehingga dapat menghasilkan gambar yang cukup bagus. *Smartphone* memiliki kemampuan proses pengolahan gambar yang lebih baik dari *handphone* biasa karena memiliki processor sendiri. Saat ini, kebanyakan *Smartphone* yang beredar di pasaran adalah *Smartphone* dengan bersistem operasi Android. Android memang banyak dikembangkan oleh developer-developer di dunia sehingga menyebabkan perkembangannya cukup pesat dalam 2 tahun terakhir ini.

Sistem android sendiri berbasikan linux yang dalam pengembangan aplikasi-aplikasinya dalam lingkungan bahasa java. Pengembangan aplikasi android tidak terlalu ribet dan tidak membutuhkan license khusus agar aplikasi yang dibuat dapat dijalankan secara langsung pada device Android. Hal ini dikarenakan Android bersifat *open source*. Namun, dengan status *open source* ini, mengembangkan aplikasi Android bukan tidak menghasilkan apa-apa. Dengan mengembangkan aplikasi Android, pengembangnya dapat bekerjasama dengan google, misalnya dengan *googlemobs*, sehingga pengembang mendapatkan keuntungan dari aplikasi yang dibuatnya.

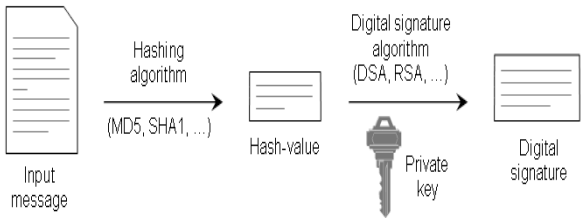
Untuk menjaga keamanan di bidang fotografi pada device Android, dapat diterapkan konsep *digital signature* atau tanda tangan digital untuk sebuah foto/gambar. Tanda tangan digital sebenarnya digunakan untuk mengautentikasi apakah pesan yang dikirim benar dari pengirimnya atau tidak. Namun, dalam hal ini terdapat beberapa perubahan sedikit. Dengan aplikasi yang dibuat, fotografer dapat membuktikkan apakah foto yang diambilnya merupakan hasil pemotretannya sendiri atau tidak. Tindak kejahatan seperti mengakui foto orang lain sebagai foto miliknya pun diharapkan dapat berkurang dengan adanya aplikasi ini.

II. KONSEP DIGITAL SIGNATURE SECARA UMUM

Tanda tangan sudah digunakan untuk otentikasi dokumen cetak sejak dari zaman dulu. Tanda tangan mempunyai karakterisktik sebagai berikut,

- Tanda tangan adalah bukti otentik
- Tanda tangan tidak dapat dilupakan
- Tanda tangan tidak dapat dipindah untuk digunakan ulang
- Dokumen yang telah ditandatangani tidak dapat diubah
- Tanda tangan tidak dapat disangkal

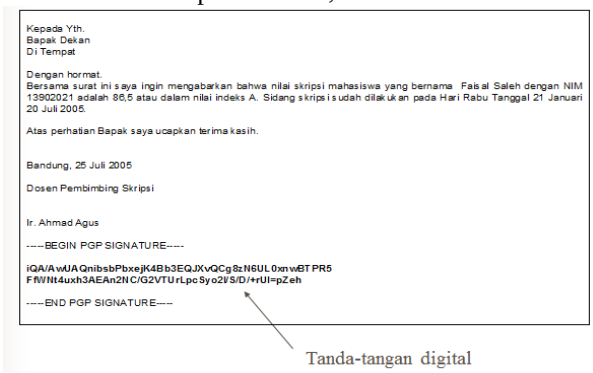
Berikut adalah proses penyisipan tanda tangan digital ke dalam suatu pesan,



Gambar 1Penyisipan tanda tangan digital pada pesan

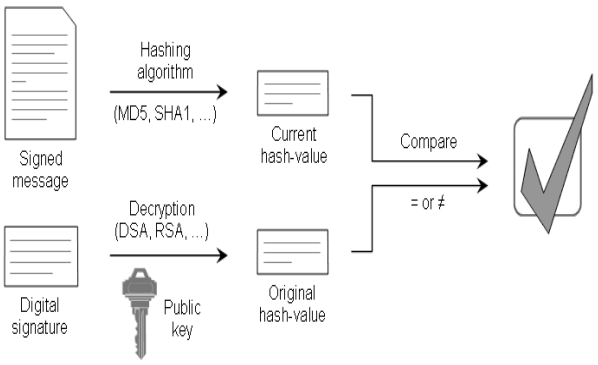
Langkah pertama yang dilakukan adalah dengan mencari nilai hash dari pesan. Nilai hash dapat ditentukan dengan menggunakan dengan salah satu algoritma hash, misalnya MD2, MD4, MD5, SHA1, SHA-256, dan lainnya. Nilai hash untuk file yang berbeda selalu berbeda juga walalupun sebenarnya ada kemungkinan yang sangat kecil bahwa dapat menghasilkan nilai hash yang sama.

Setelah mendapat nilai hash dari pesan tersebut, maka nilai hash tersebut dienkripsi dengan menggunakan algoritma simetri untuk digital signature. Yang paling digunakan adalah algoritma RSA(berdasarkan teori bilangan), DSA(berdasarkan teori logaritma diskrit), dan ECDSA(berdasarkan teori kurva melengkung). Dalam kasus ini, enkripsi menggunakan kunci private sedangkan kunci public yang digunakan untuk dekripsi. Sehingga, kunci private hanya dimiliki oleh pengirim pesan, namun pengirim pesan memberikan kunci publiknya ke orang lain dengan tujuan agar pesan tersebut nantinya dapat diotentikasi. Setelah dienkripsi, hasil enkripsinya ditambahkan ke dalam pesan tersebut dengan mekanisme tertentu. Contoh penambahan digital signature adalah pada suatu email seperti berikut,



Gambar 2Contoh Letak Digital Signature pada email

Untuk proses otentikasi suatu pesan, cara-cara yang dilakukan ditunjukkan pada diagram berikut,



Gambar 3Proses Autentikasi Pesan Dengan Digital Signature

Langkah pertama yang dilakukan adalah melakukan hash pada pesan aslinya. Kemudian, nilai hash dari pesan asli nantinya akan dibandingkan dengan nilai hash hasil dekripsi digital signature yang diletakkan dalam pesan. Langkah kedua, adalah melakukan dekripsi dari digital signature yang ditambahkan ke dalam pesan. Kali ini, digunakan kunci publik untuk mendekripsi tanda tangan digital ini. Setelah dilakukan proses dekripsi, maka didapatlah nilai hash asli dari pesan. Kemudian, kedua nilai hash ini dibandingkan, jika sama, maka pesan yang diterima adalah otentik. Jika tidak, maka telah terjadi sesuatu pada pesan yang diterima.

Terdapat tiga kemungkinan jika hasil otentikasi tanda tangan digital tidak valid, yaitu :

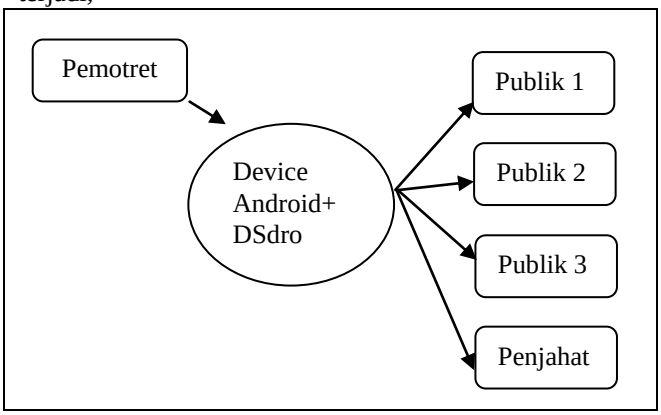
- Digital signature nilainya telah diubah, sehingga ketika didekripsi, didapat nilai hash yang berbeda/bukan aslinya.
- Jika pesan telah diubah setelah ditambahkan digital signature, maka nilai hash pada langkah pertama akan berbeda dengan nilai hash hasil dekripsi digital signature.
- Kemungkinan kesalahan ketiga adalah kunci publik yang digunakan untuk dekripsi dan kunci private yang digunakan untuk enkripsi tidak bersesuaian.

Ketidakvalidan hasil otentikasi tidak semuanya dikarenakan digital signature/isi pesan telah diubah. Namun, terkadang orang yang melakukan autentikasi salah menggunakan kunci publik pasangan dari kunci privat yang digunakan untuk enkripsi. Kesalahan ini dapat ditangani dengan cara menggunakan sertifikat yang berisi informasi dari kunci publik dari seseorang. Setiap kunci publik yang dimiliki seseorang memiliki sertifikat sendiri-sendiri. Sertifikat inilah yang dikirim juga bersama dengan pesan tersebut dengan cara yang lebih aman.

III. TEKNIK DIGITAL SIGNATURE YANG DITERAPKAN UNTUK FOTOGRAFI PADA DEVICE ANDROID

Teknik digital signature yang diterapkan mirip dengan teknik yang telah dijelaskan pada bagian sebelumnya, namun dengan sedikit beberapa perubahan. Perbedaan

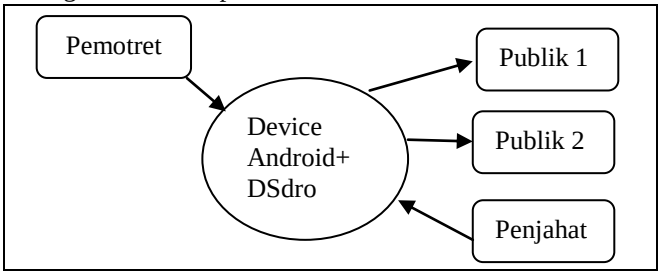
utamanya adalah pada tujuan dari digital signature ini digunakan untuk fotografi. Digital signature secara umum bertujuan agar penerima pesan dapat membuktikkan bahwa pesan yang diterimanya adalah asli dari sang pengirim. Namun, dalam bidang fotografi, digital signature yang dimaksudkan adalah untuk membuktikkan bahwa foto yang diambil adalah hasil karya dari sang pemotret sendiri. Dalam hal ini, dapat dianalisis siapakah pengirim pesan sebenarnya, pesan tersebut itu apa, dan siapakah yang akan mengotentikasi pesan tersebut. Disinilah terlihat perbedaan dari teknik yang akan diterapkan dengan teknik digital signature pada umumnya. Siapakah pengirim pesan sebenarnya adalah sang pemotret sendiri. Pesan tersebut adalah foto yang telah ditandatangani secara digital. Untuk siapakah pesan tersebut dikirim? Pesan tersebut dikirim untuk semua orang yang akan melihat hasil fotografi tersebut. Kemudian, apakah setiap orang tersebut harus memeriksa foto tersebut benar merupakan hasil karya pemotret itu atau tidak. Jawabannya, mereka bisa melakukan otentikasi sendiri ataupun sang pemotret dapat membuktikkan dirinya adalah pemotret yang asli. Diagram berikut menunjukkan kemungkinan aliran informasi yang dapat terjadi,



Gambar 4 Diagram menunjukkan aliran informasi

Setelah melewati device Android dan DSdro, foto telah disisipi dengan digital signature dari pemotret dan juga informasi dari device android sebagai device yang mengambil foto tersebut. Kemudian, foto tersebut sampai ke publik.

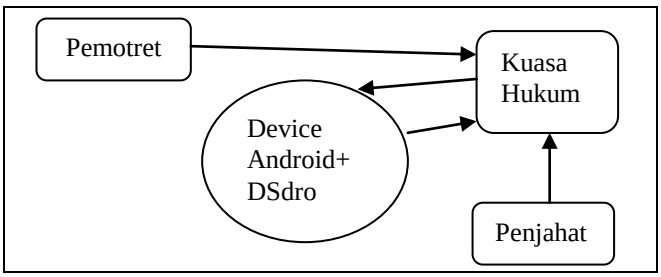
Setelah itu, si penjahat ingin mengakui bahwa foto tersebut miliknya. Maka, pemotret aslinya akan bertindak dengan cara membuktikkan bahwa foto tersebut memang dia yang memotret dengan device dia sendiri dan mengumumkan ke publik hasil otentikasi tersebut.



Gambar 5Skema otentikasi

Pemotret menggunakan aplikasi yang ada pada *device*-nya untuk mengotentikasi foto yang telah dia ambil. Pemotret memiliki kunci publik yang digunakan untuk mendekripsi digital signature. Variabel lain yang minimal diperlukan untuk proses atutentikasi adalah nama dari pemotret. Selain itu, diperlukan juga informasi khusus yang mencirikan device android tersebut, namun program sendiri yang mengambil dari sistem androidnya. Sang penjahat juga harus melakukan proses otentikasi yang sama. Dia harus memasukkan nama nya ke dalam aplikasi yang tidak harus ada di device android sang pemotret, atau di device nya sendiri. Kunci publik dapat menggunakan kunci publik yang sama dengan sang pemotret. Kemudian, pemotret dan penjahat sama-sama melakukan otentikasi dan hasilnya akan dipublikasikan. Ternyata, hasil otentikasi dari penjahat tidak valid namun sang pemotret berhasil lolos dari otentikasi. Artinya, penjahat tidak dapat memalsukan kepemilikan foto orang lain karena penjahat tidak memiliki informasi seperti kunci privat pemotret dan juga informasi device yang digunakan untuk memotret.

Teknik otentikasi lainnya yaitu melibatkan pihak penengah yang mempunyai kuasa hukum lebih tinggi, yang ditunjukkan seperti diagram berikut :

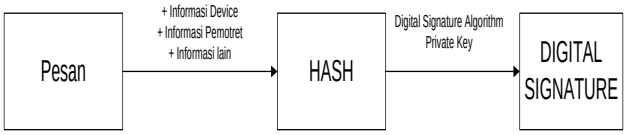


Gambar 6Teknik otentikasi lain yang melibatkan kuasa hukum

Kuasa hukum menerima bukti dari pemotret maupun penjahat berupa kunci publik dan identifikasi device yang digunakan. Identifikasi device ini dapat digenerate oleh aplikasi. Kemudian kuasa hukum melakukan otentikasi atas kedua bukti tersebut (bisa dilakukan di device android lain) untuk menentukan siapa yang benar. Dengan informasi dari pemotret dan penjahat, dapat ditentukan siapa yang sebenarnya telah memotret foto tersebut.

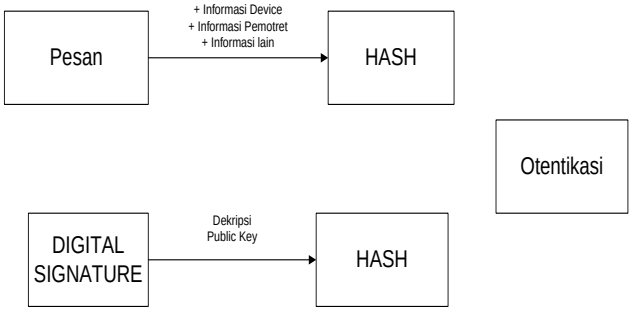
Informasi-informasi yang digunakan untuk membuat digital signature/mengecek keaslian foto dapat ditambahkan agar membuat variabel lebih banyak sehingga lebih sulit lagi untuk melakukan tindakan kriminal pada foto tersebut. Informasi ini misalnya, gambar tanda tangan asli pemotret, tanggal lahir, ataupun pin/password. Kerahasiaan dan keaslian foto dapat terjaga lebih aman dengan tambahan-tambahan variabel tersebut.

Untuk lebih jelasnya, diagram berikut akan menjelaskan proses pembuatan digital signature yang menerapkan teknik ini.



Gambar 7Proses Pembuatan Digital Signature

Kemudian, proses otentikasi ditunjukkan pada diagram sebagai berikut,



Gambar 8 Proses Otentikasi

IV. IMPLEMENTASI PADA APLIKASI DSDRO

DSDro adalah aplikasi yang dibuat penulis untuk melakukan pemotretan sekaligus membuat dan memasukkan digital signature ke dalam foto hasil pemotretan tersebut. DSDro kepanjangan dari Digital Signature For Android. Pada menu utama, ada tiga pilihan menu, yaitu :

1. Take Photo
2. Authenticate
3. Setting

Menu Take Photo akan melakukan proses pengambilan gambar dan secara otomatis memasukkan digital signature ke dalamnya. Untuk menu Authenticate akan melakukan proses otentikasi dengan cara memilih file foto yang akan diotentikasi pada file browser dan secara otomatis mengeluarkan pesan apakah foto tersebut asli berasal dari device tersebut atau tidak. Menu setting adalah menu pengaturan yang didalamnya ada sub menu generate kunci, pilihan untuk menambahkan digital signature secara otomatis atau tidak, menambahkan nama, dan menambahkan gambar tanda tangan.

Menu Take Photo diimplementasikan dengan kelas activity TakePhoto.java. Di dalam kelas ini akan memanggil default aplikasi kamera yang disediakan di android. Proses pemanggilan aplikasi kamera ditunjukkan pada kode di bawah ini,

```
  File file = new File( _path );
  Uri outputFileUri = Uri.fromFile( file );

  Intent intent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE
);
  intent.putExtra(MediaStore.EXTRA_OUTPUT,
outputFileUri );
  startActivityForResult( intent, 0 );
```

Setelah, foto diambil, akan dipanggil method OnPhotoTaken() yang langsung akan membuat digital signature dari foto tersebut.

```
protected void onPhotoTaken()
    {
        _taken = true;

        // add digital signature here
        ds di = null;
        byte[] append = null;
        try {
        di = new ds(_path);
        byte[] sig = di.generateDS();
        append = ds.appendByte(di.isi, sig);

        FileManager.WriteFile(_path, append);
        } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        }
        BitmapFactory.Options options = new
BitmapFactory.Options();
        options.inSampleSize = 4;
        Bitmap bitmap =
BitmapFactory.decodeFile( _path, options );
        _image.setImageBitmap(bitmap);
        Toast.makeText(this, " Image has been
saved ", Toast.LENGTH_LONG).show();
    }
```

GenerateDS adalah method dari kelas ds.java yang didalamnya terdapat implementasi fungsi hash yang digunakan, fungsi RSA, dan generate key.

Fungsi hash yang digunakan mengimplementasikan algoritma SHA-256. Untuk method generate key, panjang kunci RSA yang digunakan adalah 2048 bit. Untuk keperluan hash, RSA, dan key generator kelas mengimport beberapa package yaitu

```
import java.security.KeyFactory;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.PrivateKey;
import java.security.PublicKey;
import
java.security.spec.InvalidKeySpecException;
import java.security.spec.RSAPrivateKeySpec;
import java.security.spec.RSAPublicKeySpec;

import javax.crypto.Cipher;
```

Untuk masalah kunci yang akan digunakan, diperlukan objek KeyFactory, KeyPair, KeyPairGenerator, PrivateKey, PublicKey, RSAPrivateKeySpec, dan RSAPublicKeySpec. Generator key ini akan selalu membuat kunci secara random. Method keygenerator ditunjukkan oleh kode dibawah ini,

```
// RSA key generator
public static void keyGenerator() throws
NoSuchAlgorithmException,
InvalidKeySpecException, IOException {
        KeyPairGenerator kpg =
KeyPairGenerator.getInstance("RSA");
        kpg.initialize(2048);
        KeyPair kp = kpg.genKeyPair();
        KeyFactory fact =
KeyFactory.getInstance("RSA");
        RSAPublicKeySpec pub =
fact.getKeySpec(kp.getPublic(),
          RSAPublicKeySpec.class);
        RSAPrivateKeySpec priv =
fact.getKeySpec(kp.getPrivate(),
```

```
            RSAPrivateKeySpec.class);

        FileManager.saveKeyToFile("public.key",
pub.getModulus(),
            pub.getPublicExponent());
        FileManager.saveKeyToFile("private.key",
priv.getModulus(),
            priv.getPrivateExponent());
}
```

Pada method saveKeyToFile digunakan writeObjectOutput, sehingga kunci ditulis dalam binary dan tidak bisa dibaca menggunakan file teks editor.

Sedangkan untuk keperluan hash digunakan objek MessageDigest dan Cipher.

Generate digital signature method sebagai berikut,

```
public byte[] generateDS() throws Exception {
        String Nama = nama;
        Build b = new Build();
        String fingerprint = b.FINGERPRINT;
        String add = Nama+fingerprint;
        byte[] result = appendByte(isi,
add.getBytes());
        byte[] hash = doDigest(result);

        return rsaEncrypt(hash);
}
```

Informasi tambahan yang ditambahkan setelah byte akhir isi file sebelum di-hash adalah nama pemotret dan fingerprint dari android device tersebut. Fingerprint ini sebenarnya berbentuk format :

```
$(BRAND)/$(PRODUCT)/$(DEVICE)/$(BOARD):
$(VERSION.RELEASE)/$(ID)/$(VERSION.INCREMENTAL)
:$(TYPE)/$(TAGS)
```

Fingerprint ini dipilih sebagai salah satu informasi tambahan karena setiap device android pasti memiliki fingerprint yang berbeda-beda.Setelah digital signature digenerate, digital signature ini ditambahkan di akhir/end of file jpeg sehingga file gambar masih dapat dibuka dengan aplikasi pembuka gambar.

Proses otentikasi merupakan kebalikan dari fungsi generateDS hanya saja kunci yang digunakan adalah kunci publik. Setelah didapat dua hash dibandingkan secara hex stringnya.

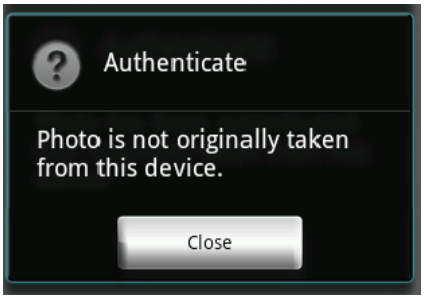
Untuk struktur modul yang digunakan dan tampilan aplikasi disertakan dalam lampiran.

V. ANALISIS HASIL IMPLEMENTASI

Pengujian dilakukan dengan melakukan pemotretan di dua device android yang berbeda dengan menggunakan aplikasi DSDro. File gambar dari device 1 dinamakan gambar1.JPG dan dari device 2 dinamakan gambar2.JPG. Ketika keduanya dicoba langsung diotentikasi di masing-masing device, didapat hasil yang sama seperti ditunjukkan pada gambar di bawah ini,

Gambar 9Otentikasi kedua foto berhasil

Namun, ketika kedua foto dipindahkan yaitu gambar1.JPG dipindahkan ke device 2 dan gambar2.JPG dipindahkan ke device 1, maka ketika diotentikasi hasil yang didapat menunjukkan foto tersebut tidak diambil pada device ini. Hasil ini ditunjukkan pada gambar berikut,



Gambar 10 Otentikasi yang tidak valid

Pengujian dilanjutkan dengan mencoba mengubah beberapa byte di akhir file yang merupkan bagian dari digital signature dan mencoba untuk mengotentikasi file tersebut. Namun, hasil yang didapat sama dengan pengujian sebelumnya yaitu, foto bukan berasal dari device tersebut.

Pengujian dilakukan lagi dengan menggenerate kunci baru di menu setting. Generate kunci baru ini secara otomatis akan menyimpan ke dalam file eksternal yang ada di folder "/sdcard/DsDroImages/". Dengan adanya pasangan kunci baru ini digunakan untuk mengotentikasi file sebelumnya, hasil yang didapat foto bukan berasal dari device tersebut. Hal ini menunjukkan pengubahan kunci menimbulkan kegagalan dalam proses otentikasi.

Proses pengujian cukup sederhana dan hasil yang didapat sesuai dengan teknik yang dideskripsikan sebelumnya. Program berjalan lancar, baik dari proses pengambilan gambar, pembuatan digital signature, penambahan digital signature ke dalam gambar, proses otentikasi, dan generate kunci.

Pengimplementasi package kriptografi untuk algoritma SHA-256 dan RSA cukup efektif dan mangkus karena program tidak berjalan terlalu berat. Bilangan yang digunakan pada RSA adalah BigInteger yang melibatkan bilangan-bilangan yang besar dengan kunci 2048 bit.

VI. KESIMPULAN

Beberapa fitur dari DSdro masih ada yang belum diimplementasikan, seperti penambahan gambar tanda tangan dan otentikasi pada device lain. Namun, implementasi sudah cukup menggambarkan teknik digital signature yang dimaksud untuk dunia fotografi.

Proses pembuatan digital signature dan otentikasi sudah akurat dengan pengujian dari foto yang diambil di kamera 1 dan diotentikasi pada kamera 2 terbukti bukan berasal dari kamera 2.

Teknik digital signature ini masih perlu dikembangkan lagi karena kekurangan dan kelemahannya masih perlu dianalisis lebih lanjut.

DAFTAR PUSTAKA

[1] http://www.informatika.org/~rinaldi/Kriptografi/kriptografi.htm
Tanggal akses : 8 Mei 2011

[2] http://developer.android.com/
Tanggal akses : 9 Mei 2011

[3] http://www.developer.com/java/ent/article.php/3092771/How-Digital-Signatures-Work-Digitally-Signing-Messages.htm
Tanggal akses : 27 April 2011 Waktu : 05.00

[4] http://www.javamex.com/tutorials/cryptography/rsa_encryption_2.shtml
Tanggal akses : 9 Mei 2011 Waktu : 02.00

PERNYATAAN

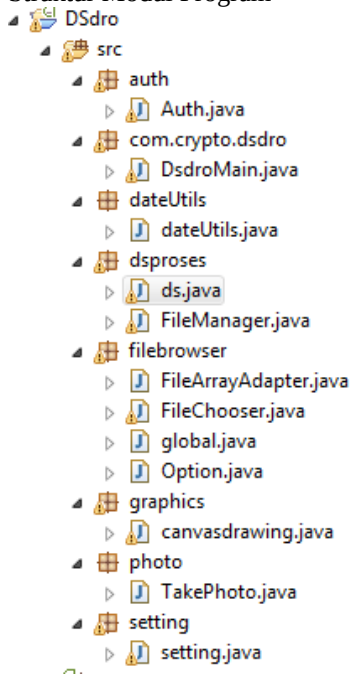
Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Mei 2011

Danang Tri Massandy
13508051

LAMPIRAN

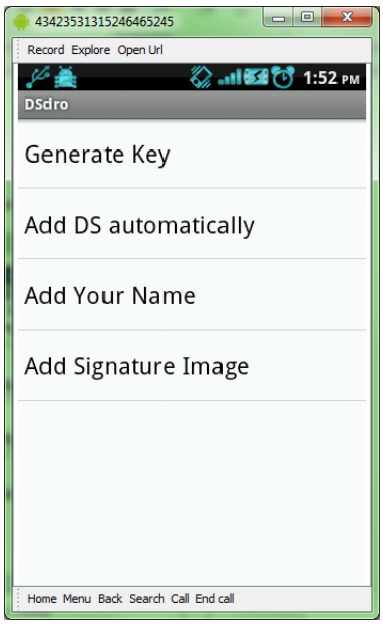
1. Struktur Modul Program



Gambar 11Struktur Modul Program

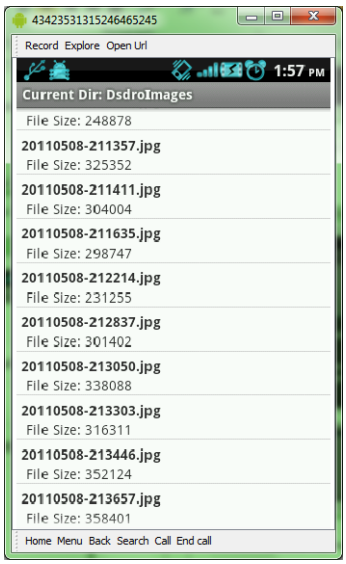
2. Tampilan Program

Gambar 12Menu Utama

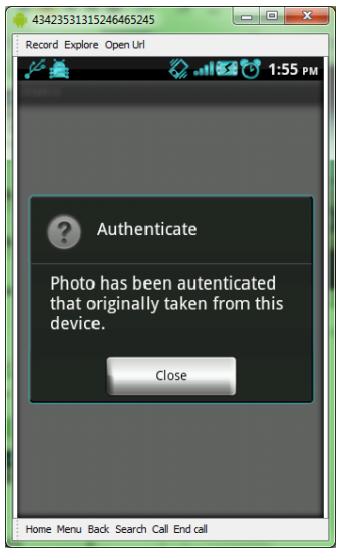


Gambar 13 Setting

Gambar 14Pemotretan Foto



Gambar 15Proses pemilihan foto untuk otentikasi



Gambar 16Proses Autentikasi berhasil