Algoritma *QR Code Digital Signature* dengan Memanfaatkan *Fingerprint*

Candy Olivia Mawalim (13513031)
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10 Bandung 40132, Indonesia
13513031@std.stei.itb.ac.id

***Abstract* — Makalah ini membahas tentang teknik-teknik yang digunakan untuk membentuk tanda tangan digital dengan memanfaatkan sidik jari (*fingerprint*) seseorang. Bentuk tanda tangan digital yang dihasilkan adalah dalam Quick Response Code (QR Code). Dalam makalah ini dijelaskan pula metode mengekstraksi informasi melalui hasil sidik jari dari suatu *fingerprint scanner* dan membentuk tanda tangan digital dari pesan, masukan kunci publik dan informasi sidik jari yang diterima. Dengan modifikasi algoritma tanda tangan digital ini, diharapkan tingkat keabsahan pesan dapat semakin meningkat karena sidik jari setiap orang yang berbeda.**

Keywords — sidik jari (fingerprint); QR Code; tanda tangan digital; kunci publik

I. PENDAHULUAN

Dewasa ini, perkembangan teknologi informasi dan komunikasi di dunia semakin pesat. Hal ini dibuktikan dengan informasi yang semakin mudah kita peroleh. Dengan hanya memasukan suatu kata kunci tertentu pada mesin pencari seperti *google*, kita langsung dapat menemukan informasi mengenai kata kunci tersebut. Informasi yang semakin mudah untuk diperoleh ini tentu memberikan dampak positif bagi berkembangnya ilmu pengetahuan. Akan tetapi, selain memberikan dampak positif, kemudahan perolehan informasi ini mengakibatkan beberapa dampak negatif, misalnya informasi yang diberitakan mungkin adalah berita yang tidak benar (*hoax*), sumber informasi yang tidak jelas, dan penyalahgunaan hak akses terhadap suatu informasi.

Hal-hal tersebut menjadi sorotan dalam perkembangan ilmu kriptografi, khususnya yang berkaitan dengan keamanan informasi. Keamanan menjadi hal yang sangat penting untuk diperhatikan saat ini. Informasi penting yang tidak aman dan tidak terjaga privasinya memungkinkan pihak pemilik informasi dirugikan. Salah satu bidang dalam ilmu kriptografi yang akan dibahas dalam makalah ini adalah mengenai tanda tangan digital.

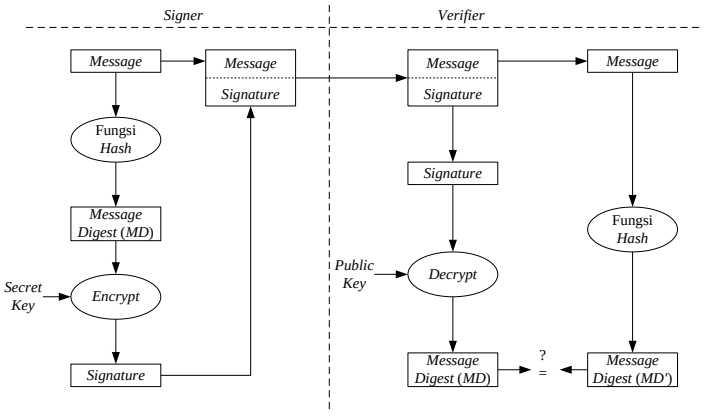
Tanda tangan digital diciptakan untuk mengamankan, memastikan keaslian dan mencegah penyangkalan seseorang terhadap informasi tertentu. Banyak algoritma yang sudah dikembangkan untuk tanda tangan digital ini. Dalam makalah ini, penulis hendak mengembangkan algoritma tanda tangan digital yang sudah ada dengan penambahan atribut berupa sidik jari pemilik pesan. Tanda tangan tersebut kemudian digambarkan dalam bentuk *Quick Response Code* (QR Code).

II. LANDASAN TEORI

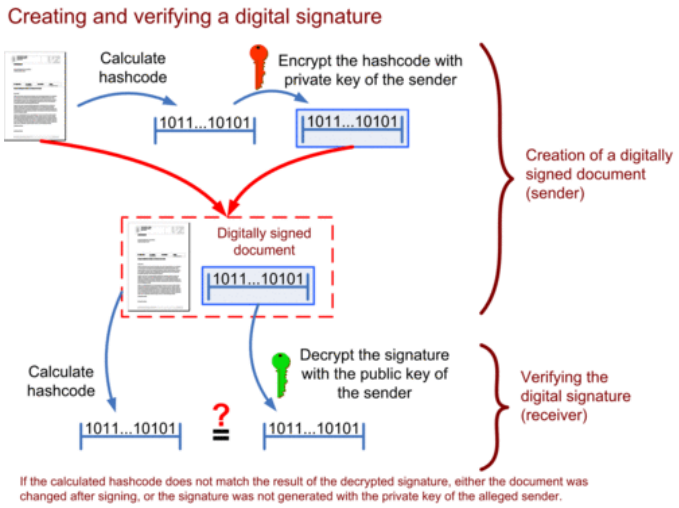
A. *Tanda Tangan Digital*

Keamanan yang disediakan oleh bidang ilmu kriptografi menyangkut 4 aspek, yaitu kerahasiaan pesan, otentikasi, keaslian pesan dan anti penyangkalan. Dengan adanya tanda tangan digital aspek otentikasi, keaslian pesan dan anti penyangkalan terhadap pesan tersebut akan dapat terselesaikan. Tanda tangan digital mengandung nilai kriptografis yang bergantung pada isi pesan dan kunci. Setiap dokumen memiliki tanda tangan digital yang berbeda karena isi dokumen berbeda satu dengan yang lain.

Ada beberapa algoritma yang dikembangkan untuk tanda tangan digital, antara lain dengan menggunakan kriptografi simetri, kriptografi kunci-publik, dan dengan fungsi *hash.* Algoritma yang akan digunakan dalam pembangkitan tanda tangan digital di makalah ini adalah dengan menggunakan kriptografi kunci-publik dan fungsi hash. Secara umum, langkah-langkah dalam pembangkitan tanda tangan digital dengan metode ini dapat digambarkan dalam gambar berikut.



Gambar 1. *Flow Chart* Pembentukan dan Pemeriksaan Keaslian Tanda Tangan

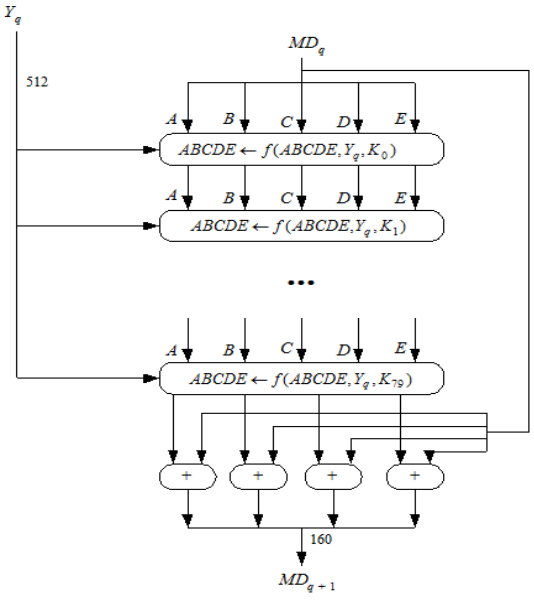


Gambar 2. Langkah Pembangkitan Hingga Verifikasi Tanda Tangan Digital

B. *Algoritma SHA 1*

SHA adalah fungsi *hash* satu-arah yang dibuat oleh *NIST* dan digunakan bersama *DSS* (*Digital Signature Standard*). Algoritma *SHA* menerima masukan berupa pesan dengan ukuran maksimum 2^{64} bit (2.147.483.648 *gigabyte*) dan menghasilkan *message digest* yang panjangnya 160 bit, lebih panjang dari *message digest* yang dihasilkan oleh *MD5*. Langkah-langkah pemuatan *message digest* dengan SHA-1

1. Penambahan bit-bit pengganjal (*padding bits*)
2. Penambahan nilai panjang pesan semula
3. Inisialisasi penyangga (*buffer*) MD
4. Pengolahan pesan dalam blok berukuran 512 bit



Gambar 3. Alur Fungsi SHA-1

C. *QR Code*

QR Code (*Quick Response* Code) merupakan bentuk evolusi kode batang dari satu dimensi menjadi dua dimensi. Penggunaan kode QR sudah sangat lazim di Jepang. Akan tetapi, untuk penggunaannya dalam pembuatan tanda tangan digital masih belum banyak dikembangkan. Keunggulan kode QR dibanding kode batang yang telah ada terlebih dahulu adalah kemampuan penyimpanan informasi kode QR jauh lebih besar dari pada kode batang biasa. Berikut adalah contoh kode QR.

Gambar 4. Contoh Kode QR

III. PEMBAHASAN

A. *Rancangan Algoritma*

Secara garis besar, ada empat langkah dalam pembangkitan tanda tangan digital yang berbentuk kode QR ini. Adapun keempat langkah tersebut adalah sebagai berikut.

1) Ekstraksi informasi sidik jari yang sudah dihasilkan dari *fingerprint scanner*. Informasi yang diperoleh (dalam bentuk *byte*) kemudian diubah ke dalam bentuk *hexadecimal*.
2) Penentuan kunci publik oleh pemilik pesan.
3) Pembangkitan tanda tangan digital dengan menggunakan algoritma SHA-1.
4) Tanda tangan digital tersebut kemudian diubah ke dalam bentuk kode QR.

Berikut adalah *pseudocode* untuk langkah-langkah di atas.

```
//TAHAP 1 - FingerprintReader
INPUT  : file gambar sidik jari yang
diperoleh dari fingerprint scanner
PROSES :
 1. Pembacaan file gambar sidik jari
 2. Konversi gambar ke bentuk kumpulan
    byte
 3. Konversi byte ke hexadecimal
OUTPUT : informasi sidik jari dalam
bentuk hexadecimal

//TAHAP 2 - PublicKeyScanner
INPUT  : kunci publik dari pengguna,
output tahap 1
PROSES :
 1. Menerima kunci publik dari pengguna
 2. Kunci publik disisipkan dalam
```

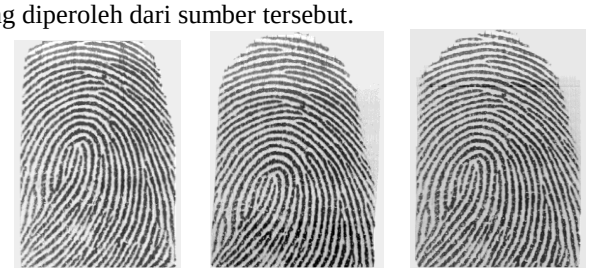
```
    informasi sidik jari yang diperoleh
    pada tahap 1
OUTPUT : informasi sidik jari yang sudah
disisipkan dengan kunci public

//TAHAP 3 – GenerateDigitalSignature
INPUT  : output tahap 2, pesan
PROSES :
1. Mengombinasikan output yang diperoleh
   di tahap 2 dengan pesan
2. Menggunakan fungsi SHA-1 untuk
   menghasilkan tanda tangan digital
3. Pengubahan hasil SHA-1 ke bentuk
   Hexadecimal
OUTPUT : tanda tangan digital dalam
bentuk hexadecimal

//TAHAP 4 – CreateQRCode
Input  : tanda tangan digital dalam
bentuk hexadecimal
PROSES :
Konversi tanda tangan digital dalam
bentuk hexadecimal ke QRCode
OUTPUT : tanda tangan digital dalam
bentuk kode QR
```

Untuk memverifikasi keabsahan tanda tangan digital tersebut, langkah-langkah yang dilakukan adalah sebagai berikut.

1) Mengekstraksi pesan yang terkandung pada kode QR
2) Memasukan kunci publik dari kode QR
3) Menghitung *checksum* dari kombinasi pesan dan kunci publik dan kemudian membandingkannya dengan tanda tangan digital yang diperoleh dari kode QR. Jika *checksum* dari kombinasi pesan dan kunci publik sesuai dengan tanda tangan digital yang diperoleh dari kode QR maka pesan itu terverifikasi keasliannya.

B. Implementasi Algoritma

Dalam pengembangan algoritma ini, penulis menggunakan bahasa pemrograman Java berdasarkan rancangan algoritma yang dijabarkan pada bagian sebelumnya. Karena dalam tahapan implementasi ini belum tesedia perangkat keras untuk menerima masukan sidik jari, penulis menggunakan data uji sidik jari dari website: *http://www.advancedsourcecode.com/fingerprintdatabase.asp.* Berikut adalah contoh data sidik jari yang diperoleh dari sumber tersebut.



Gambar 5. Contoh Data Uji Sidik Jari

Tahapan selanjutnya setelah file gambar sidik jari diperoleh, sidik jari tersebut kemudian diolah menjadi informasi dalam bentuk hexadecimal.

```
…12021211D1D18140815201F21
1F211D18130A151F20211F221B
0F141408171E211B0E0C0F0705
0C181F201B1410060E121C201D
15070509060D1A1B1E1E0F0403
0506050B101A191E1F201E180D
08100B121C1F1D211E1C130…
```

Gambar 6. Konversi sidik jari ke hexadecimal

Berikutnya, pemilik pesan memasukkan kunci publik. Kunci publik ini kemudian diubah ke dalam bentuk hexadecimal lalu disisipkan pada informasi hexadecimal yang diperoleh di tahap sebelumnya (karena digit hexadecimal yang terbentuk sangat banyak, penulis hanya memasukkan potongan hexadecimal yang terbentuk).

Pemilik pesan kemudian memasukkan pesan yang berupa file teks untuk membangkitkan tanda tangan digital. Berikut adalah contoh uji coba yang dilakukan.

(a) kunci publik : Kriptografi → 6b726970746f6772616669

(b) informasi hexadecimal yang terbentuk :

```
6B726970746F67726166695262
4201D1B1713100E0E0F10100D0
E11100E101216191E222324211
B1A21202022325232323221E191
2101110111100E0F0F0F1112100
F100F13171C232523211C130C0
D0E0E100201B1410060E121C…
```

(c) file teks yang berisi pesan (message.txt)

SHA adalah fungsi *hash* satu-arah yang dibuat oleh *NIST* dan digunakan bersama *DSS* (*Digital Signature Standard*).

(d) Tanda tangan digital yang terbentuk:
ee521306aef18013a21340dd418f13186e6e60e3

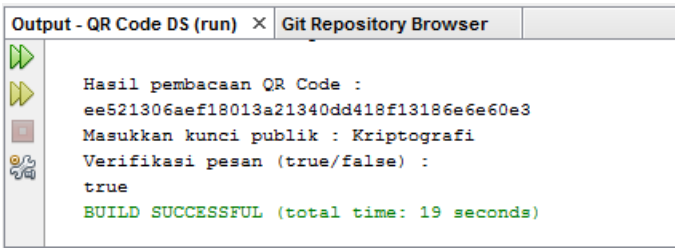
Setelah tanda tangan digital terbentuk dalam bentuk hexadecimal, tanda tangan ini kemudian dikonversi ke dalam bentuk kode QR. Library yang digunakan dalam pembentukan kode QR adalah zxing-javase.jar dan zxing-core-3.2.1.jar (sumber: *https://github.com/zxing/zxing).* Hasil QR code yang diperoleh adalah sebagai berikut.

Gambar 7. Kode QR yang terbentuk dari keluaran langkah sebelumnya

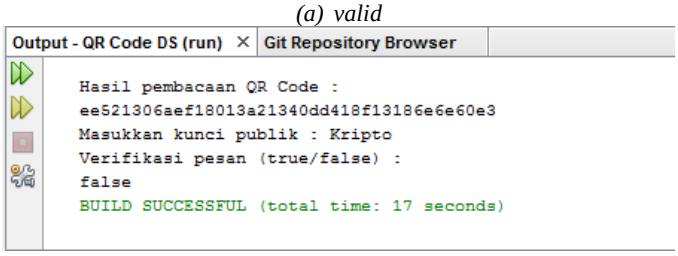
Setelah terbentuk kode QR di atas, cara memverifikasinya adalah dengan membaca pesan yang terkandung dalam kode QR. Kemudian, masukkan kunci publik oleh penerima pesan. Lalu validasinya dengan perhitungan *checksum*.

Potongan kode program untuk melakukan verifikasi:

```
public boolean verifyChecksum(String file,
String testChecksum) throws
NoSuchAlgorithmException, IOException
    {
        MessageDigest sha1 =
MessageDigest.getInstance("SHA1");
        FileInputStream fis = new
FileInputStream(file);

        byte[] data = new byte[1024];
        int read = 0;
        while ((read = fis.read(data)) != -1)
{
            sha1.update(data, 0, read);
        };

        byte[] hashBytes = sha1.digest();

        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < hashBytes.length;
i++) {

sb.append(Integer.toString((hashBytes[i] &
0xff) + 0x100, 16).substring(1));
        }

        String fileHash = sb.toString();

        return fileHash.equals(testChecksum);
    }
```



(a) valid



(b) tidak *valid*

Gambar 8. Hasil verifikasi tanda tangan digital

C. Analisis Eksperimen

Pada bagian ini akan dibahas mengenai analisis terhadap implementasi yang telah dilakukan pada bagian sebelumnya. Algoritma QR Code Digital Signature ini memiliki keunggulan karena algoritma ini memanfaatkan sidik jari pemilik pesan. Sidik jari pemilik pesan yang dikonversi ke string hexadecimal yang panjangnya berjumlah hingga 167648 karakter. Hal ini mengakibatkan algoritma ini sulit untuk diserang dengan cara *brute force* (perlu 16^{167648}) kemungkinan solusi. Selain sidik jari, tanda tangan digital juga dipengaruhi masukan kunci publik pengguna.

Bentuk penyajian tanda tangan digital ini juga mengakibatkan tanda tangan digital sulit diubah. Perlu teknik khusus untuk mengubah kode QR. Hal ini mengakibatkan fungsi tanda tangan digital sebagai alat otentikasi, keaslian pesan dan anti penyangkalan semakin terjamin. Ada beberapa kemungkinan yang menyebabkan tanda tangan digital ini tidak valid: (1) kunci publik yang dimasukkan penerima pesan berbeda dengan kunci publik pengirim pesan, (2) isi pesan sudah tidak asli lagi (ada perubahan yang terjadi dalam pesan), (3) sidik jari yang tidak sesuai dengan pengirim pesan, (4) terjadi perubahan pada tanda tangan digital yang berbentuk kode QR.

IV. Simpulan dan Saran

Berdasarkan hasil analisis pada bagian sebelumnya, dapat disimpulkan bahwa algoritma pembangkitan tanda tangan digital ini memiliki tingkat keamanan yang tinggi. Hal ini dikarenakan dalam proses pembangkitannya digunakan sidik jari dari pemilik pesan. Karena sidik jari setiap orang di dunia berbeda, tanda tangan yang dibangun dengan sidik jari ini juga tidak mungkin sama antara satu orang dengan yang lain.

Penggunaan kunci publik dan kode QR dalam algoritma ini juga semakin meningkatkan tingkat keamanan tanda tangan digital ini. Terdapat teknik khusus untuk pembuatan dan pembacaan kode QR. Hal ini mengakibatkan pengubahan terhadap tanda tangan digital menjadi lebih sulit.

Algoritma ini masih memiliki beberapa kelemahan diantaranya biaya untuk pembangkitan tanda tangan digital lebih tinggi karena dibutuhkan *hardware* untuk dapat menerima masukan berupa sidik jari pengguna. Selain itu, pemrosesan informasi yang diperoleh dari sidik jari masih kurang efektif karena sangat bergantung dari ukuran file gambar sidik jari yang diperoleh. Oleh karena itu, untuk pengembangan selanjutnya file gambar seharusnya distandardisasikan terlebih dahulu sehingga lebih mudah untuk diolah.

Ucapan Terima Kasih

Pertama-tama, penulis bersyukur kepada Tuhan Yang Maha Esa atas berkat-Nya makalah ini dapat selesai. Penulis juga mengucapkan banyak terima kasih kepada dosen mata kuliah IF4020 Kriptografi, Dr. Ir. Rinaldi Munir, M.T., yang telah memberikan bimbingan dalam pembuatan makalah ini. Selain itu, penulis juga hendak berterima kasih kepada seluruh pihak yang telah membantu penyelesaian makalah ini yang tidak dapat disebutkan namanya satu per satu.

REFERENSI

[1] Rinaldi Munir, "Slide Kuliah Tanda Tangan Digital", 2016.

[2] Rinaldi Munir, "Slide Kuliah *Secure Hash Algorithm* (SHA)", 2016.

[3] Denso Waive, Inc. "ZXing Library for QR Code". https://github.com/zxing/zxing diakses tanggal 13 Mei 2016.

[4] Mkyong, "Java SHA Hashing Example," http://www.mkyong.com/java/java-sha-hashing-example/ diakses tanggal 13 Mei 2016.

[5] Luigi ROSA, "A List of Fingerprint Databases Available on the Web", www.advancedsourcecode.com/fingerprintdatabase.asp diakses tanggal 12 Mei 2016.

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, 13 Mei 2016

Candy Olivia Mawalim