Message Digest dalam bentuk *QR Code* Sebagai Tanda Tangan Digital

Ripandy Adha - 13507115
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
if17115@students.if.itb.ac.id

Abstrak*—Kriptografi telah banyak digunakan untuk berbagai keperluan keamanan informasi. Salah satu bentuk keamanan informasi tersebut adalah untuk verifikasi keaslian suatu pesan, apakah pesan tersebut masih memiliki konten yang sama dengan yang aslinya, ataukah sudah dilakukan perubahan terhadap isi pesan sehingga pesan menjadi tidak asli lagi. Verifikasi tersebut dapat dilakukan dengan cara membuat sebuah pesan menjadi suatu *Message Digest* yang kemudian dapat ditambahkan kedalam pesan sebagai sebuah tanda tangan digital. *Message Digest* dapat dibangkitkan dengan algoritma *hash*, salah satunya adalah algoritma *SHA*. Dalam makalah ini, algoritma *SHA* yang digunakan adalah algoritma *SHA1*, dan algoritma untuk pembangkitan tanda-tangan digital yang digunakan adalah algoritma *RSA*. Seiring dengan berkembangnya teknologi, bentuk dari kriptografi pun semakin berkembang. Saat ini kriptografi terbaru dan cukup populer adalah kriptografi visual. Salah satu contoh kriptografi visual tersebut adalah dengan menggunakan *QR Code*. *QR Code* adalah kriptografi visual dua dimensi yang dapat dibangkitkan dari sebuah karakter, kalimat, URL, atau lainnya. Pada umumnya *QR Code* dibaca dengan menggunakan kamera pada handphone, dan ditranslasikan kedalam bentuk asalnya. Dalam makalah ini penulis akan mencoba melakukan studi terhadap penggunaan *QR Code* yang dibangkitkan dari tanda-tangan digital suatu pesan dengan algoritma penanda-tangan yang digunakan adalah algoritma *RSA*. *QR Code* ini kemudian dapat ditambahkan kedalam pesan sebagai *QR Code Digital Signature*. *QR Code Digital Signature* ini adalah *QR Code* yang dibangkitkan dari *Message Digest* dari isi pesan yang digabungkan dengan *Digital Signature* dari algoritma *RSA*. Pembacaan *Digital Signature* dengan *QR Code* ini nantinya dapat menggunakan kamera handphone.

Kata Kunci—Algoritma *Hash*, Algoritma *SHA1*, *Digital Signature*, Kriptografi, *Message Digest*, *QR Code*.

I. PENDAHULUAN

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. *Cryptography* berasal dari bahasa yunani. *Crypto* berarti *hidden / secret* (tersembunyi / rahasia) dan *Graphy* berarti *writing* (tulisan), sehingga kriptografi secara harfiah adalah *secret writing* (tulisan rahasia). Ilmu kriptografi diterapkan untuk keperluan pengiriman pesan agar tidak dapat dibaca dan dimengerti oleh pihak yang tidak berkepentingan.

Ilmu kriptografi secara umum dapat dikategorikan menjadi dua, yaitu kriptografi kunci simetri, dan kriptografi kunci nirsimetri. Pada kriptografi kunci simetri, kunci yang digunakan untuk melakukan enkripsi sama dengan kunci yang digunakan untuk melakukan dekripsi. Algoritma kriptografinya disebut dengan algoritma simetri. Sementara itu, pada kriptografi kunci asimetri, kunci yang digunakan untuk melakukan enkripsi tidak sama dengan kunci yang digunakan untuk melakukan dekripsi. Kunci untuk enkripsi diketahui secara publik, dan disebut dengan *public key*, sementara kunci untuk dekripsi hanya diketahui oleh pihak yang berkepentingan dan bersifat privat, dan disebut dengan *private key*. Kriptografi ini disebut juga dengan kriptografi kunci-publik.

Ilmu kriptografi semakin banyak digunakan seiring dengan berkembangnya ilmu pengetahuan dan teknologi. Saat ini banyak sekali terjadinya penyelundupan atau pemalsuan pesan digital oleh orang-orang yang tidak bertanggung-jawab. Disini, aspek keamanan yang disediakan kriptografi yang digunakan adalah untuk otentikasi (*authentication*), keaslian pesan (*data integrity*), sekaligus anti-penyangkalan (*non-rupdiation*). Teknik yang digunakan untuk menerapkan ketiga aspek keamanan tersebut adalah dengan membubuhkan tanda-tangan digital ke dalam pesan tersebut. Seperti halnya tanda-tangan biasa, tanda-tangan digital digunakan sebagai tanda pengenal dari pengirim pesan. Tanda pengenal disini bukan dimaksudkan sebagai siapa pemilik tanda-tangan tersebut, melainkan untuk penanda bahwa isi pesan adalah sesuai dengan yang pesan asli yang dikirimkan.

Dalam penerapannya, pemakaian tanda-tangan digital memiliki bakuan yang dikenal sebagai *Digital Signature Standard* (*DSS*). *DSS* sendiri terdiri dari dua komponen, yaitu algoritma tanda-tangan digital yang disebut *Digital Signature Algorithm* (*DSA*) dan fungsi *hash* standard yang disebut *Secure Hash Algorithm* (*SHA*). *DSA* sendiri termasuk kedalam algoritma kriptografi kunci publik. Tetapi, *DSA* tidak dapat digunakan untuk melakukan enkripsi, melainkan dispesifikasikan khusus untuk tanda-tangan digital. *DSA* memiliki dua fungsi utama, yaitu pembentukan tanda-tangan, dan verifikasi / pemeriksaan keabsahan tanda-tangan.

II. TANDA-TANGAN DIGITAL

Sejak zaman dahulu, tanda-tangan sudah digunakan untuk otentikasi dokumen cetak. Tanda-tangan memiliki karakteristik berupa bukti yang otentik, tidak dapat dilupakan, tidak dapat dipindah untuk digunakan ulang, tidak dapat disangkal, dan menjaga suatu dokumen agar tidak dapat diubah. Fungsi tanda tangan ini juga diterapkan untuk tujuan otentikasi pada data digital seperti pesan atau dokumen elektronik. Tanda-tangan untuk dokumen digital ini kemudian disebut dengan tanda-tangan digital atau *digital signature*.

Tanda-tangan digital bukanlah berupa tulisan tanda-tangan yang di-digitisasi, seperti dengan cara *scaning*, atau dengan membuat tanda-tangan dengan *software* editor grafis. Berbeda dengan tanda-tangan pada dokumen cetak yang selalu sama tanpa tergantung dengan isi dokumennya, tanda-tangan digital adalah nilai kriptografis yang bergantung pada isi pesan dan kunci enkripsi. Apabila isi suatu dokumen berbeda, maka berbeda pula tanda-tangan digitalnya.

Untuk menandatangani pesan terdapat dua cara, yaitu dengan enkripsi pesan atau menggunakan fungsi *hash* dan kriptografi kunci-publik. Untuk penandatanganan dengan cara mengenkripsi pesan, dapat digunakan kriptografi simetri. Pesan yang dienkripsi dengan algoritma simetri sudah memberikan solusi untuk otentikasi pengirim dan keaslian pesan, karena kunci simetri hanya diketahui oleh pengirim dan penerima pesan. Akan tetapi, cara ini tidak menyediakan mekanisme untuk anti-penyangkalan. Untuk menangani masalah penyangkalan ini, diperlukan pihak ketiga yang dipercaya oleh pengirim dan penerima. Pihak ketiga ini disebut penengah (arbitrase).

Cara lainnya yaitu dengan kriptografi kunci-publik. Jika dengan enkripsi biasa, penandatanganan hanya untuk menjaga kerahasiaan (*secrecy*), yaitu pesan dienkripsi dengan kunci publik penerima dan didekripsi dengan kunci privat penerima. Tetapi cara ini tidak memberikan sarana otentikasi karena kunci publik diketahui oleh banyak orang. Akan tetapi terdapat metode khusus untuk enkripsi tanda-tangan, yaitu pesan dienkripsi dengan kunci privat pengirim dan didekripsi dengan kunci publik pengirim. Dengan cara ini maka kerahasiaan pesan dan otentikasi dapat dicapai sekaligus.

Proses menandatangani pesan (oleh pengirim) :

$$S = E_{SK}(M) \qquad (1)$$

dan

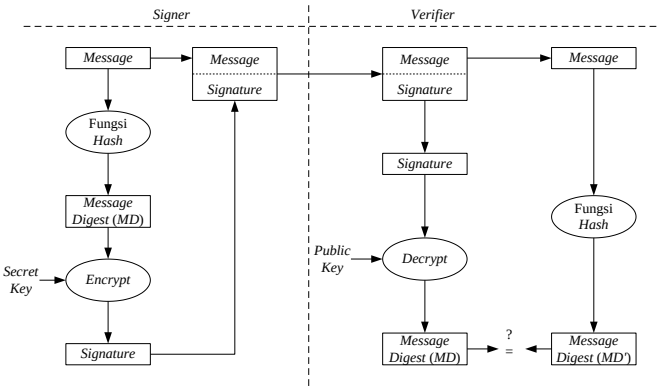
Proses membuktikan otentikasi pesan (oleh penerima) :

$$M = D_{PK}(S) \qquad (2)$$

SK = Secret Key (kunci privat pengirim)
PK = Public Key (kunci publik pengirim)
E = fungsi enkripsi
D = fungsi dekripsi
M = pesan semula
S = signature (hasil enkripsi pesan)

Dengan algoritma kunci-publik ini, penandatanganan pesan tidak lagi membutuhkan pihak penengah (arbitrase).

Selain digunakan untuk kedua fungsi sebelumnya, yaitu kerahasiaan atau kerahasiaan dan otentikasi, seringkali terdapat kasus dimana kerahasiaan pesan tidak perlu dilakukan, tetapi otentikasi pesan saja yang diperlukan. Untuk kasus seperti ini dapat digunakan Algoritma kriptografi kunci-publik dan fungsi *hash*.



Gambar 1. Skema Penandatanganan Digital

Keontetikan pesan dapat diketahui dengan cara membandingkan *message digest* (*MD*) pesan. Apabila pesan yang diterima sudah berubah, maka *MD* yang dihasilkan dari fungsi *hash* akan berbeda dengan *MD'* semula. Hal ini menunjukkan bahwa pesan tidak asli lagi. Selain itu, apabila pesan tidak berasal dari orang yang sebenarnya, maka *message digest* (*MD*) yang dihasilkan dari enkripsi pesan asli dengan fungsi *hash* berbeda dengan *MD'* yang dihasilkan pada proses verifikasi. Hal ini terjadi karena kunci publik yang digunakan oleh penerima tidak berkoresponden dengan kunci privat pengirim. Setelah *MD = MD'*, ini berarti pesan yang diterima adalah pesan yang asli (*message authentication*) dan dikirimkan oleh orang yang sebenarnya (*user authentication*).

Dua algoritma *signature* yang digunakan secara luas adalah *RSA* dan *ElGamal*. Dalam makalah ini, akan digunakan algoritma *RSA* sebagai pembangkit tanda-tangan digital.

A. Tanda-tangan dengan algoritma RSA

Langkah-langkah pemberian tanda-tangan :

1. Pengirim menghitung nilai *hash* dari pesan M yang akan dikirim,dan misalkan nilai *hash* yang dihasilkan adalah *h*.

2. Pengirim mengenkripsi *h* dengan kunci privatnya menggunakan persamaan enkripsi *RSA* :

$$S = h^{SK} \bmod n \qquad (3)$$

S = *signature* (hasil enkripsi)
SK = *Secret Key* (kunci privat pengirim)
n = *pq* (modulus)
p dan q adalah dua buah bilangan prima

3. Pengirim mentransmisikan *M* + *S* ke penerima.

Langkah-langkah verifikasi tanda-tangan :

1. Penerima menghitung nilai *hash* dari pesan *M* yang dikirim, misalkan nilai *hash* dari *M* adalah h'.

2. Penerima melakukan dekripsi terhadap tanda-tangan *S* dengan kunci publik si pengirim menggunakan persamaan dekripsi *RSA* :

$$h = S^{PK} \bmod n \qquad (4)$$

S = *signature* (hasil enkripsi)
PK = *Public Key* (kunci publik pengirim)
n = *pq* (modulus)
p dan q adalah dua buah bilangan prima

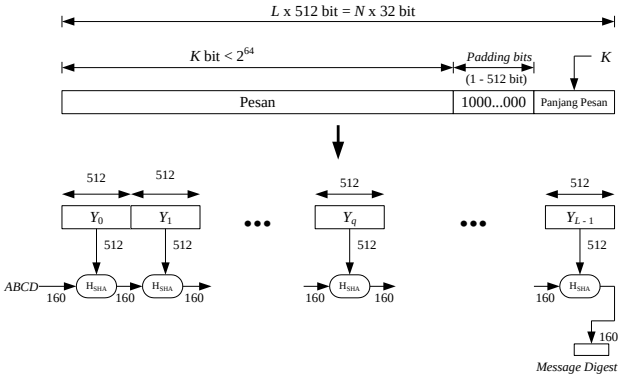
3. Penerima membandingkan *h* dengan *h'*. Jika *h* = *h'* maka tanda-tangan digital adalah otentik. Jika tidak sama, maka tanda-tangan tidak otentik, sehingga dapat dianggap bahwa pesan sudah tidak asli lagi atau pengirim bukan orang yang seharusnya.

B. Secure Hash Algorithm (SHA)

Secure Hash Algorithm (SHA) adalah fungsi *hash* satu-arah yang dibuat oleh *The National Institute of Standard and Technology* (*NIST*) yang digunakan bersama *DSS*. *SHA* didasarkan pada algoritma *MD4*. Algoritma *SHA* menerima masukan berupa pesan dengan ukuran maksimum 2^{64} bit (2.147.483.648 *gigabytes*) dan menghasilkan *message digest* yang panjangnya 160 bit. *Message digest* yang dihasilkan dari algoritma *SHA* ini lebih panjang dari *message digest* yang dihasilkan oleh algoritma *MD5*. Beberapa varian *SHA* : SHA-0, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512. SHA-0 sering dikenali dengan *SHA* saja.

Langkah-langkah pemuatan *message-digest* dengan *SHA*-1 adalah :

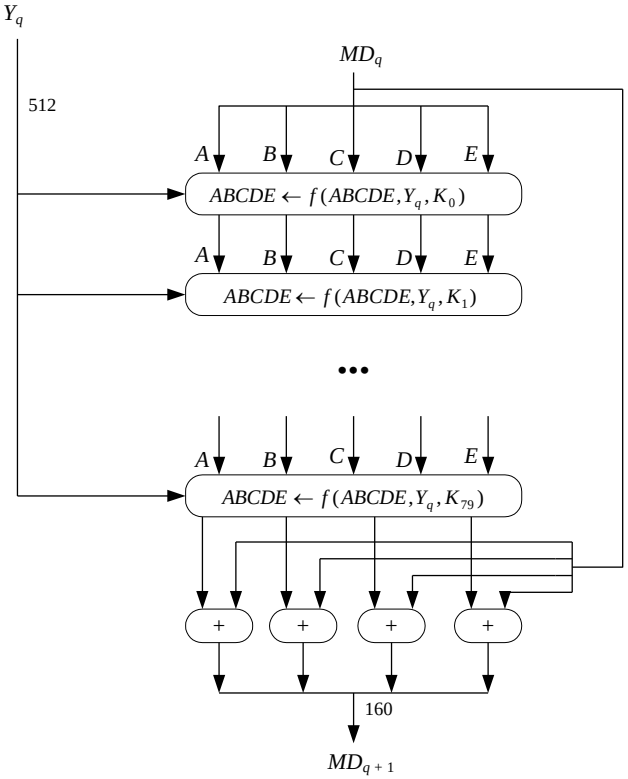
1. Penambahan bit-bit pengganjal (*padding bits*).
2. Penambahan nilai panjang pesan semula.
3. Inisialisasi penyangga (*buffer*) *MD*.
4. Pengolahan pesan dalam blok berukuran 512 bit.



Gambar 2. Skema Pembuatan *Message Digest* dengan *SHA*-1

SHA-1 membutuhkan 5 buah penyangga (*buffer*) yang masing-masing panjangnya 32 bit. Total panjang penyangga adalah 5 x 32 bit = 160 bit. Kelima penyangga *MD* ini diberi nama *A*, *B*, *C*, *D*, dan *E*. Setiap penyangga diinisialisasi dengan nilai-nilai (dalam notasi HEX) sebagai berikut :

$$A = 67452301 \qquad (5)$$
$$B = EFCDAB89 \qquad (6)$$
$$C = 98BADCFE \qquad (7)$$
$$D = 10325476 \qquad (8)$$
$$E = C3D2E1F0 \qquad (9)$$



Gambar 3. Skema Pengolahan blok 512-bit (Proses H_{SHA})

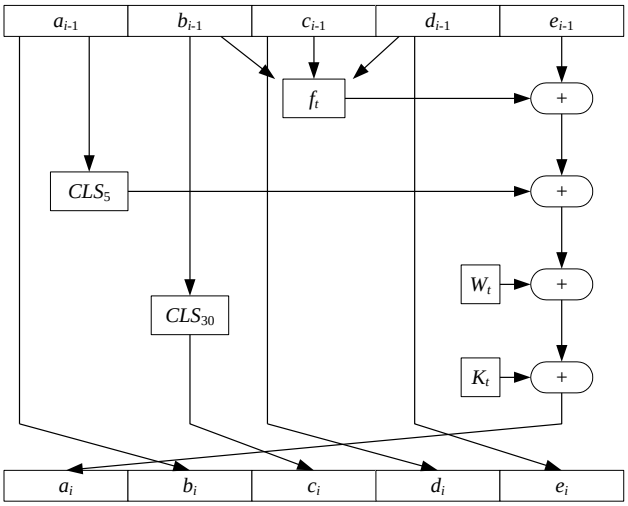
Proses H_{SHA} terdiri dari 80 buah putaran. Masing-masing putaran menggunakan bilangan penambah K_t yaitu:

Putaran $0 \le t \le 19$, K_t = 5A827999 (10)

Putaran $20 \le t \le 39$, K_t = 6ED9EBA1 (11)

Putaran $40 \le t \le 59$, K_t = 8F1BBCDC (12)

Putaran $80 \le t \le 79$, K_t = CA62C1D6 (13)



Gambar 4. Operasi Dasar pada Setiap Putaran

Putaran	$f_t(b, c, d)$
0 .. 19	$(b \wedge c) \vee (\neg b \wedge d)$
20 .. 39	$b \oplus c \oplus d$
40 .. 59	$(b \wedge c) \vee (b \wedge d) \vee (c \wedge d)$
60 .. 79	$b \oplus c \oplus d$

Tabel 1. Fungsi Logika f_t pada setiap putaran

Nilai W_1 sampai W_{16} berasal dari 16 *word* pada blok yang sedang diproses, sedangkan nilai W_t berikutnya didapakan dari persamaan :

$$W_t = W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3} \quad (14)$$

III. QR CODE

QR Code adalah sejenis simbologi 2-D (dua dimensi) yang dikembangkan oleh Denso Wave dan dirilis pada tahun 1994 dengan target utama untuk menjadi simbol yang mudah diinterpretasi oleh perangkat scanner. *QR Code* (*2D code*) menyimpan informasi di kedua arah vertikal dan horizontal, dimana *bar code* hanya menyimpan data di satu arah. *QR Code* menyimpan jauh lebih banyak informasi dibandingkan *bar code*.

A. Bar Code to 2D Code

Bar code telah menjadi cukup populer dikarenakan kecepatan pembacaan, ketepatan, dan karakteristik fungsionalitasnya yang superior. Seiring dengan semakin populernya bar *code*, dan pemakaiannya sudah dikenali secara umum, banyak yang semakin menginginkan *code* yang mampu menyimpan lebih banyak informasi, tipe karakter yang lebih beragam, dan dapat dicetak dalam daerah yang lebih kecil. Untuk melakukan hal tersebut, berbagai usaha telah dilakukan untuk meningkatkan banyaknya informasi yang dapat disimpan dalam *bar code*, seperti menambahkan jumlah digit *bar code* atau menumpuk beberapa *bar code* sekaligus. Akan tetapi, usaha ini menimbulkan masalah seperti memperbesar area *bar code*, memperumit operasi pembacaan, dan meningkatkan biaya pencetakan. *2D code* diperkenalkan untuk menjawab kebutuhan dan masalah tersebut. *2D code* pun semakin berkembang dari metode *bar code* menumpuk, hingga metode matrix yang memiliki pertambahan kepadatan informasi.

B.Pengenalan QR Code

Versi Simbol dari *QR Code* bervariasi dari Versi 1 hingga Versi 40. Setiap versi memiliki konfigurasi modul atau jumlah modul yang berbeda. Yang disebut dengan modul adalah titik hitam-putih yang membentuk *QR Code*. "Konfigurasi Modul" merujuk kepada jumlah modul yang terdapat dalam sebuah simbol, diawali versi 1 (21 x 21 modul) hingga versi 40 (177 x 177 modul). Setiap versi yang lebih tinggi jumlah modul bertambah 4 di setiap sisi. Masing-masing versi simbol *QR Code* memiliki kapasitas data maksimum sesuai dengan jumlah data, tipe karakter, dan level koreksi kesalahan. Dengan kata lain, seiring dengan semakin besarnya jumlah data, semakin banyak modul yang diperlukan untuk membentuk *QR Code*, sehingga menghasilkan *QR Code* yang lebih besar.

QR Code memiliki kapabilitas koreksi kesalahan untuk mengembalikan data jika kode mengalami kerusakan atau kotor. Terdapat empat tingkat koreksi yang dapat digunakan dan dipilih oleh pengguna disesuaikan dengan lingkungan operasi. Dengan meningkatkan tingkat koreksi dapat meningkatkan kapabilitas penanganan kesalahan, tetapi juga meningkatkan jumlah ukuran data pada *QR Code*. Pertimbangan dalam menentukan tingkat koreksi kesalahan melibatkan beberapa faktor seperti lingkungan operasi dan ukuran dari *QR Code*.

QR Code Error Correction Capability*	
Level L	Approx.7%
Level M	Approx. 15%
Level Q	Approx. 25%
Level H	Approx. 30%

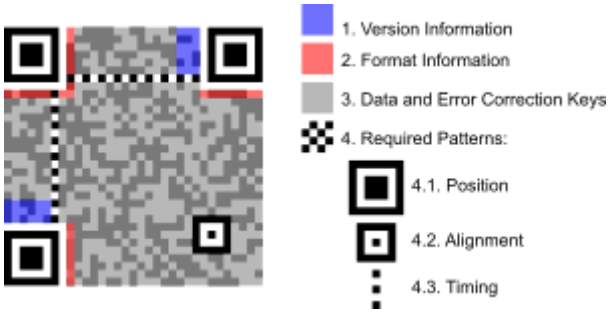
Tabel 2. Tingkat Kapabilitas Koreksi Kesalahan *QR Code*

C.Fitur QR Code

QR Code menyediakan fitur-fitur berikut dibandingkan

dengan *bar code* konvensional :

- Kapasitas *encoding* data yang tinggi.
- Ukuran *printout* yang kecil.
- Kapabilitas terhadap huruf Kanji dan Katakana (jepang).
- Tahan terhadap debu atau kerusakan.
- Dapat dibaca dari segala arah 360^{o}.
- Struktur yang dapat dipecah.



Gambar 5. Struktur *QR Code*

D.Standardisasi QR Code

Karena *2D Code* dimaksudkan untuk digunakan secara luas, hal yang paling penting bahwa spesifikasi *QR Code* untuk didefinisikan secara jelas dan diperkenalkan secara publik. *QR Code* distandardisasikan sebagai berikut :

QR Code Standardization	
October, 1997	Approved as AIM International (Automatic Identification Manufacturers International) standard (ISS - QR Code)
March, 1998	Approved as JEIDA (Japanese Electronic Industry Development Association) standard (JEIDA-55)
January, 1999	Approved as JIS (Japanese Industrial Standards) standard (JIS X 0510)
June, 2000	Approved as ISO international standard (ISO/IEC18004)
November, 2004	Micro QR Code is Approved as JIS (Japanese Industrial Standards) standard (JIS X 0510)

Tabel 3. Standardisasi *QR Code*

Symbol size	21 × 21 - 177 × 177 modules (size grows by 4 modules/side)	
Type & Amount of Data (Mixed use is possible.)	Numeric	Max. 7,089 characters
	Alphanumeric	Max. 4,296 characters
	8-bit bytes (binary)	Max. 2,953 characters
	Kanji	Max. 1,817 characters
Error correction (data restoration)	Level L	Approx. 7% of codewords can be restored.
	Level M	Approx. 15% of codewords can be restored.
	Level Q	Approx. 25% of codewords can be restored.
	Level H	Approx. 30% of codewords can be restored.
Structured append	Max. 16 symbols (printing in a narrow area etc.)	

Tabel 4. Spesifikasi Outline *QR Code*

Gambar 6. Contoh *QR Code* yang Berisi URL ke Halaman Utama Halaman Utama English Wikipedia

IV. *QR CODE* SEBAGAI TANDA-TANGAN DIGITAL

Ide dasar dari penandatanganan digital dengan menggunakan *QR Code* adalah dengan membangkitkan terlebih dahulu *signature* dari enkripsi algoritma *RSA*, yang kemudian dilakukan pembangkitan *QR Code* yang berasal dari *signature* tersebut ditambah dengan *message digest* dari isi pesan yang dienkripsi dengan algoritma *SHA*-1. Contoh penerapan metode ini dengan contoh pesan yang akan digunakan adalah file kripto.txt, dengan langkah-langkah penerapan adalah sebagai berikut :

file : kripto.txt

```
Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Cryptography berasal dari bahasa yunani. Crypto berarti hidden / secret (tersembunyi / rahasia) dan Graphy berarti writing (tulisan), sehingga kriptografi secara harfiah adalah secret writing (tulisan rahasia). Ilmu kriptografi diterapkan untuk keperluan pengiriman pesan agar tidak dapat dibaca dan dimengerti oleh pihak yang tidak berkepentingan.
```

A.Pembangkitan Message Digest dari Isi Pesan

Pembangkitan *message digest* dari pesan pada file diatas, dapat dilakukan dengan menggunakan aplikasi pembangkit *message digest* yang menerapkan algoritma *SHA*-1. *Message Digest* yang dihasilkan untuk file kripto.txt diatas adalah :

MD : dfbe425c80e8135d025cfb059143c6a0fd9405a9

B. Pembangkitan Signature dengan Algoritma RSA dan fungsi hash SHA-1

Pembangkitan *signature* dari pesan (file kripto.txt) diatas dapat dilakukan dengan menggunakan aplikasi pembangkit tanda-tangan digital yang menggunakan algoritma *RSA* dan fungsi *hash SHA*-1. Tanda-tangan digital yang dihasilkan terhadap file kripto.txt adalah :

signature : 12AA3-973C-569A-E2EF-4000-80-1312D-AC6A-8D8-0-E2EF-8D8-1-88B-1312D-12AA3-0-80-1312D-AC6A-973C-569A-0-1312D-10F5F-1-4000-88B-AC6A-469B-C082-0-973C-12AA3-10F5F-4000-0-1312D-C082-10F5F

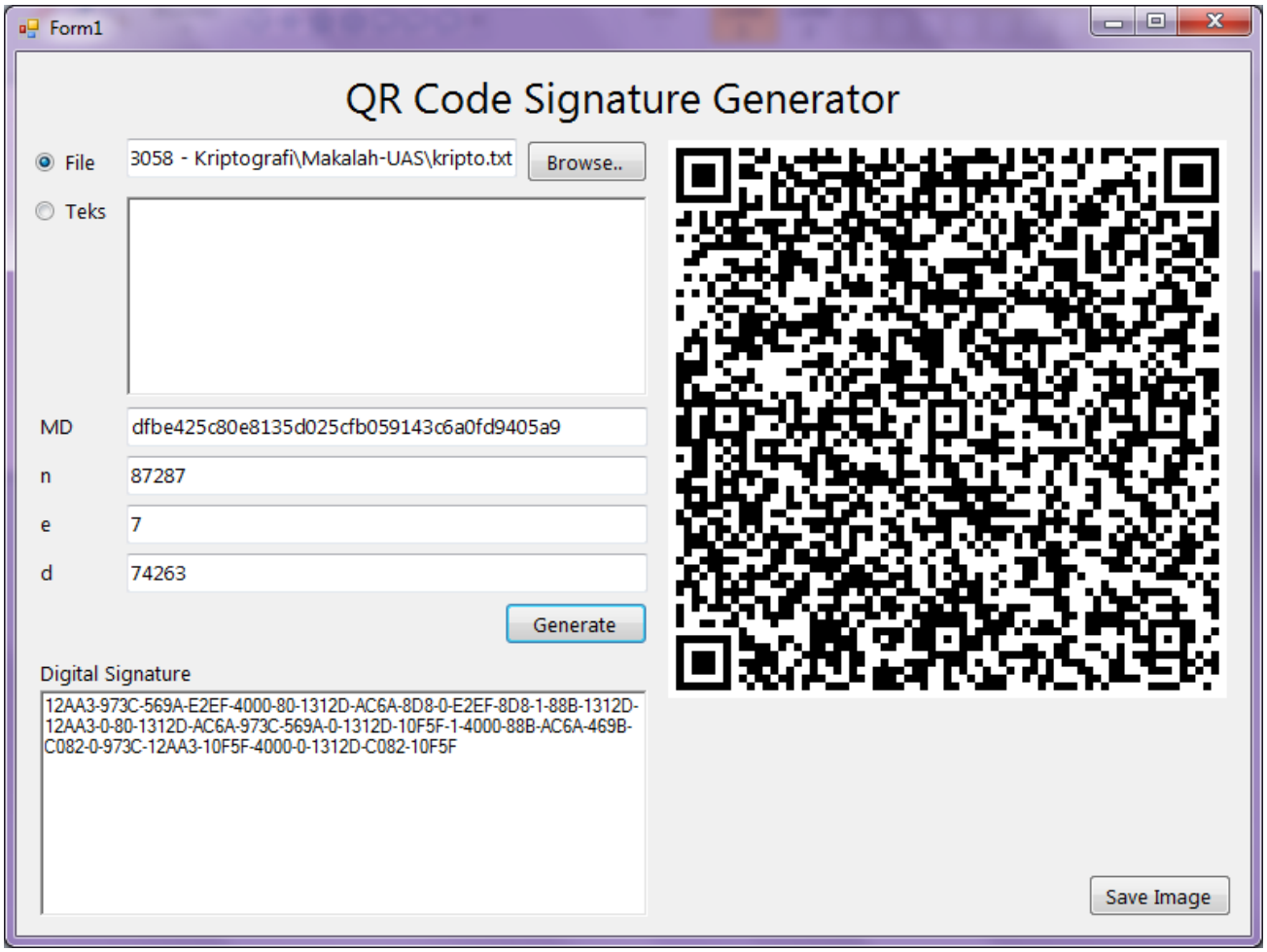
C. Pembangkitan QR Code dari Message Digest dan Tanda-Tangan Digital

Dalam pembangkitan *QR Code*, *message digest* dan tanda-tangan digital disatukan terlebih dahulu dan diformat agar pembacaan dan verifikasi menjadi lebih mudah. Format penggabungan pesan dibuat menjadi seperti berikut :

md=dfbe425c80e8135d025cfb059143c6a0fd9405a9;ds=12AA3-973C-569A-E2EF-4000-80-1312D-AC6A-8D8-0-E2EF-8D8-1-88B-1312D-12AA3-0-80-1312D-AC6A-973C-569A-0-1312D-10F5F-1-4000-88B-AC6A-469B-C082-0-973C-12AA3-10F5F-4000-0-1312D-C082-10F5F

Setelah itu, bangkitkan *QR Code* dari format tulisan komponen *message digest* dan *signature* diatas.

Gambar 7. *QR Code* hasil pembangkitan.



Gambar 8. Aplikasi Pembangkit *QR Code Signature*

D. Aplikasi Pembangkit QR Code Signature

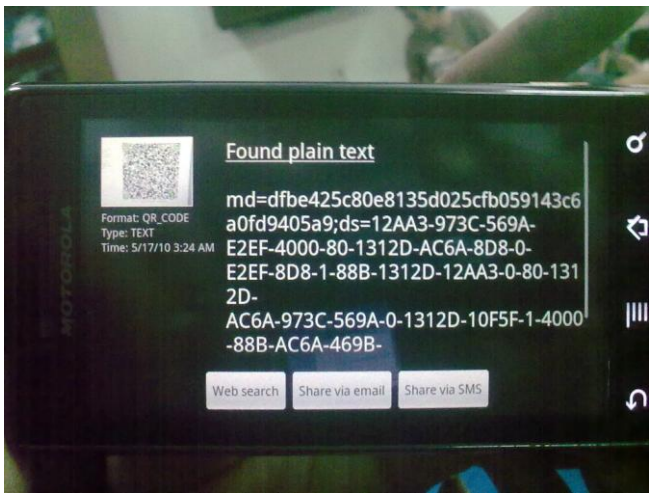
Aplikasi pembangkit *QR Code signature* pada gambar diatas adalah aplikasi yang terdiri atas pembangkit *message digest* yang telah digunakan pada tugas kecil 3 kuliah kriptografi, yang menggunakan fungsi *hash SHA*-1. Aplikasi ini juga memiliki pembangkit tanda-tangan digital yang digunakan pada tugas besar 2 kuliah kriptografi, yang menerapkan algoritma *RSA*. Selain itu, aplikasi tersebut juga memiliki pembangkit *QR Code* yang berisi data dari pesan yang telah terformat seperti yang telah dijelaskan sebelumnya. Untuk membangkitkan Barcode pada lingkungan .NET, dapat digunakan library BarcodeLib.Barcode.dll. *Source Code* pembangkit *QR Code* dalam aplikasi ini adalah sebagai berikut :

```
using System;
using System.Collections.Generic;
using System.Text;
using OnBarcode.Barcode;
using System.Drawing.Imaging;
using System.Drawing;

namespace QRGenerator
{
    class QRGen
    {
        public static Bitmap Generate(String S)
        {
            Bitmap qrcodeBitmap;
            QRCode qrcode = new QRCode();

            // Barcode data to encode
            qrcode.Data = S;
            // QR-Code data mode
            qrcode.DataMode =
QRCodeDataMode.AlphaNumeric;

            /*
              * Barcode Image Related Settings
              */
            // Unit of meature for all size related
setting in the library.
            qrcode.UOM = UnitOfMeasure.PIXEL;
            // Bar module size (X), default is 3
pixel;
            qrcode.X = 3;
            // Barcode image left, right, top,
bottom margins. Defaults are 0.
            qrcode.LeftMargin = 0;
            qrcode.RightMargin = 0;
            qrcode.TopMargin = 0;
            qrcode.BottomMargin = 0;
            // Image resolution in dpi, default is
72 dpi.
            qrcode.Resolution = 72;
            // Created barcode orientation. 4
options are: facing left, facing right, facing
bottom, and facing top
            qrcode.Rotate = Rotate.Rotate0;
            // Generate QR-Code and encode barcode
to gif format
            qrcode.ImageFormat = ImageFormat.Gif;

qrcode.drawBarcode("D:\\Pictures\\qrcode.gif");
            qrcodeBitmap = qrcode.drawBarcode();
            return qrcodeBitmap;
        }
    }
}
```

Algoritma pembangkit *QR Code* tersebut menerima masukan berupa string yang dapat diperoleh dari string yang bebas. Untuk aplikasi ini, string yang diambil adalah string yang telah diformat terlebih dahulu seperti yang telah dijelaskan sebelumnya. Kemudian algoritma tersebut akan mengembalikan Bitmap berupa *QR Code* yang terbentuk dan menampilkannya pada aplikasi, seperti yang ditunjukkan pada gambar 8. Selain itu pula, gambar QR tersebut akan disimpan sebagai file qrcode.gif, dan disimpan dalam folder D:\Pictures.

E. Verifikasi Tanda-Tangan QR Code

Verifikasi pada tanda-tangan *QR Code* ini dapat dilakukan dengan cara membaca *QR Code* tersebut dengan menggunakan kamera pada *handphone*. *Handphone* dengan teknologi saat ini sebagian besar dapat membaca isi dari *QR Code*. Untuk saat ini, penulis belum sempat membuat aplikasi *handphone* untuk dapat melakukan verifikasi langsung dari *handphone*, akan tetapi verifikasi dapat dilakukan dengan *software* untuk verifikasi seperti aplikasi yang digunakan oleh penulis pada tugas besar 2 kuliah kriptografi. Langkah dalam melakukan verifikasi yang penulis lakukan adalah dengan membaca tanda-tangan *QR Code* menggunakan *handphone*, kemudian melakukan verifikasi pesan dengan *message digest* dan *digital signature* yang tersimpan di dalam *QR Code*.



Gambar 9. Hasil Pembacaan *QR Code* dengan Kamera *Handphone*

Sesuai dengan format yang telah dibentuk, dapat dibedakan bagian yang merupakan *message digest* untuk membuktikan keaslian pesan, dan tanda-tangan digital yang akan didekripsi. Dari sini, dapat dilakukan dekripsi ulang seperti normal, dengan menggunakan aplikasi *digital signature*.

Dalam makalah ini penulis hanya mampu membuat pembangkit tanda-tangan *QR Code*, akan tetapi belum dapat membuat aplikasi pendukung utamanya, yaitu aplikasi untuk verifikasi tanda-tangan *QR Code* ini. Untuk

selanjutnya, aplikasi dapat dikembangkan dengan membuat aplikasi *handphone* untuk verifikasi *QR code signature*. Karena pembaca *QR Code* umumnya terdapat pada perangkat mobile, maka pengembangan yang dapat dilakukan adalah dengan mengembangkan sebuah aplikasi mobile yang bersesuaian.

V. KESIMPULAN

Kesimpulan yang bisa diambil dari hasil studi *QR Code signature* ini, antara lain :

1. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan
2. Tanda-tangan digital dapat digunakan untuk menjaga kerahasiaan pesan (*secrecy*), otentikasi pesan, maupun keduanya.
3. Fungsi *hash SHA*-1 adalah fungsi pembangkit *message digest* yang lebih panjang dan lebih baik daripada fungsi *hash MD-5*.
4. Salah satu algoritma penandatanganan digital yang umum digunakan adalah algoritma *RSA*.
5. *QR Code* adalah simbol *2D-code* yang dapat menyimpan lebih banyak data dibandingkan *barcode* biasa.
6. *QR Code* termasuk dalam kriptografi visual.
7. *QR Code* dapat digunakan untuk menyimpan data dalam bentuk gambar dua dimensi.
8. *QR Code* dapat digunakan sebagai tanda-tangan digital.
9. Pengembangan aplikasi pemverifikasi *QR Code signature* dapat dibuat secara terpisah dari program pembangkit tanda-tangannya, dengan menggunakan aplikasi *handphone*.

REFERENCES

[1] Munir, Rinaldi, *Kriptografi*, Institut Teknologi Bandung, 2006.

[2] http://www.di-mgt.com.au/rsa_alg.html#note5

[3] http://www.faqs.org/rfcs/rfc3174.html

[4] http://www.denso-wave.com/qrcode/aboutqr-e.html

[5] http://qrcode.kaywa.com/

[6] http://www.onbarcode.com/products/net_barcode/barcodes/qrcode.html

[7] http://www.barcodelib.com/net_barcode/barcode_symbologies/qrcode.html

[8] http://www.onbarcode.com/codes/csharp_qr_code_barcode.html

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, 17 Mei 2010

Ripandy Adha - 13507115