Proses Otentikasi Gambar dan Video menggunakan *Robust Digital Signature (RDS)*

Eriek Rahman Syah Putra – NIM : 13503032

Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13032@students.if.itb.ac.id

Abstrak

Teknik otentikasi multimedia dibutuhkan untuk memastikan kepercayaan (*trustworthiness*) dari data-data multimedia. Tanda tangan *digital* bersifat nirpenyangkalan, di mana *message digest* yang terenkripsi diekstraksi dari data. Dengan adanya beberapa manipulasi yang dilakukan pada sebuah media, maka tanda tangan *digital* tradisional tidak dapat secara langsung diaplikasikan. Oleh karena itu, dibutuhkan sebuah teknik tanda tangan *digital* yang mampu menangani hal ini, yaitu dengan *Robust Digital Signature* (RDS). Manipulasi terhadap gambar dapat dipertimbangkan dari 2 sisi, yaitu dari sisi metode dan tujuan. Dari sisi metode, manipulasi dapat berbentuk kompresi, pengubahan format, *shifting*, perbesaran, pemotongan, *filtering*, penggantian, dan sebagainya. Sedangkan dari sisi tujuan, manipulasi dapat berbentuk transformasi atau serangan (*attack*). Dengan adanya manipulasi ini, maka didesainlah sebuah otentikator yang dapat mendeteksi pengubahan format, kompresi normal, dan *JPEG lossy compression*, tetapi otentikasi ini membuktikan hasil negatif terhadap manipulasi *replacement* karena manipulasi tersebut sering digunakan untuk penyerangan. Teknik otentikasi yang diaplikasikan pada gambar di atas nantinya juga dipakai untuk otentikasi video MPEG.

Pada prinsipnya sistem otentikator ini menggunakan kunci privat dan kunci publik. Kunci privat digunakan untuk mengenkripsi kode fitur pada proses *Signature Generator*. Sedangkan kunci publik digunakan untuk mendekripsi *digital signature* pada proses otentikasi. Pada proses otentikasi ini, hasil dekripsi dari *digital signature* yaitu berupa kode fitur yang nantinya akan dibandingkan dengan kode fiitur yang dihasilkan dari data gambar aslinya. Bila menunjukkan kesamaan, maka dua data gambar dinyatakan otentik, begitu juga sebaliknya. Untuk data video, proses otentikasi hampir sama dengan data gambar, perbedaannya yaitu *raw data* yang akan dikenakan operasi hash yaitu berupa GOP (*Group of Picture*)

Di dalam makalah ini, akan dibahas seputar *review* mengenai sistem JPEG dan sistem umum dari teknik otentikasi yang digunakan untuk mengatasi manipulasi. Selain itu, juga dibahas mengenai bagaimana mengendalikan beberapa parameter untuk penggunaan praktis yang konteksnya berbeda, dengan disertai contoh sederhana yang dapat merepresentasikannya. Kemudian juga diberikan sebuah analisis performansi dan desain dari *Robust Digital Signature* untuk mengotentikasi gambar dan video.

***Kata kunci**: message digest, Robust Digital Signature, JPEG lossy compression, Signature Generator*

1. Pendahuluan

Perkembangan teknik otentikasi gambar yang *robust* menjadi sorotan utama yang sangat penting. Apabila kita menganggap sebuah gambar *digital* hanyalah sebuah *bitstream* biasa di mana tidak ada modifikasi yang dilakukan, maka tidak ada perbedaan bila kita bandingkan antara otentikasi gambar dengan otentikasi pesan lain dalam bentuk teks. Dua metode sudah diusulkan untuk memperoleh keaslian dari gambar *digital*, yaitu menggunakan kamera

digital untuk membubuhi sebuah tanda tangan *digital*, atau meng-*embed* sebuah kode rahasia pada gambar. Metode pertama menggunakan sebuah tanda tangan *digital* terenkripsi yang dihasilkan dari perangkat peng-*capture* seperti kamera *digital*. Pesan yang dienkripsikan ini disebut dengan "*signature*" (arti harfiah: tanda tangan) dan pesan ini memberikan sebuah cara untuk memastikan bahwa *signature* ini tidak dapat dipalsukan.

Signature kemudian dibawa bersamaan dengan file gambar. Proses otentikasi dari gambar tersebut membutuhkan kunci publik yang tergabung di dalamnya untuk mendekripsikan *signature*. Gambar yang diterima akan dilakukan proses *hash* dan dibandingkan dengan kode *signature*-nya. Jika hasilnya sama, maka gambar yang diterima tersebut otentik. Metode kedua meng-*embed* sebuah "*watermark*" ke dalam gambar. *Watermark* yang lemah biasanya dapat dirusak setelah adanya manipulasi. Otentikasi ditentukan dengan memeriksa *watermark* yang diekstraksi dari gambar yang diterima. Kedua metode di atas memiliki kelemahan yang signifikan.

Otentikasi tidak akan dipertahankan kecuali bila setiap piksel dari gambar tidak ada yang diubah. Tetapi, karena adanya kompresi yang merugikan seperti JPEG sering kali dapat kita terima atau bahkan kita inginkan dalam rangka mendapatkan file yang tidak terlalu besar, maka dengan demikian dibutuhkan sebuah metode otentikasi yang mampu mengidentifikasi kompresi yang merugikan dari adanya beberapa manipulasi.

Manipulasi terhadap gambar dapat dipertimbangkan dari 2 sisi, yaitu dari sisi metode dan tujuan. Dari sisi metode, manipulasi dapat berbentuk kompresi, pengubahan format, *shifting*, perbesaran, pemotongan, *filtering*, penggantian, dan sebagainya. Sedangkan dari sisi tujuan, manipulasi dapat berbentuk transformasi atau serangan (*attack*). Terdapat dua macam transformasi yaitu sebagai berikut:

1. Transformasi format dan kompresi yang *lossless*.
   Dengan mengacuhkan *noise* yang disebabkan oleh pembatasan presisi selama komputasi, nilai-nilai pikselnya tidak berubah setelah dilakukannya manipulasi ini. Oleh karena itu, manipulasi ini tidak akan dibahas di dalam makalah ini.

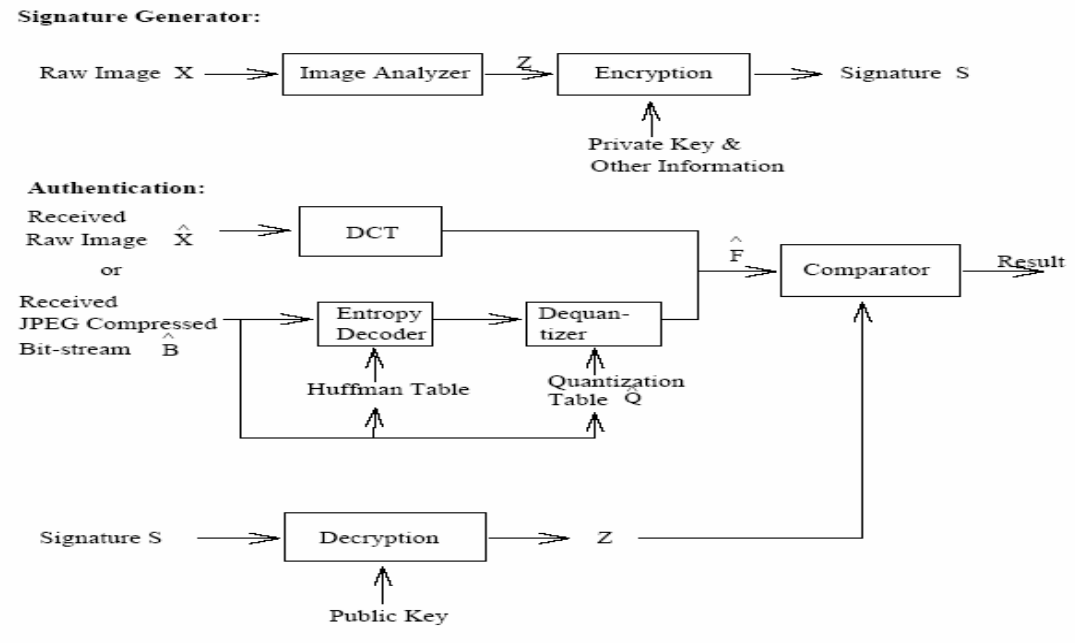
2. Transformasi aplikasi khusus.
   Beberapa aplikasi mungkin saja membutuhkan kompresi yang *lossy* untuk memenuhi kebutuhan *resource* dari aspek *bandwidth* atau tempat penyimpanan yang digunakan. Beberapa aplikasi juga butuh meningkatkan kualitas gambar, memotongnya, mengubah ukuran, atau melakukan beberapa operasi lainnya. Aspek umum dari manipulasi ini adalah bahwa siapa saja dapat mengubah nilai piksel yang dapat mengakibatkan perbedaan distorsi visual dari gambar aslinya. Biasanya kebanyakan operasi semacam ini mencoba untuk meminimalisasi distorsi visual.

Serangan atau manipulasi yang merugikan, dapat mengubah sebuah gambar menjadi gambar baru yang mempunyai penampakan visual berbeda. Salah satu contohnya yaitu dengan mengganti beberapa bagian dari gambar dengan konten yang berbeda. Hal ini sangat sulit bagi otentikator untuk mengetahui adanya manipulasi tersebut. Pendekatan praktisnya yaitu dengan mendesain sebuah otentikator yang menerima transformasi format, kompresi yang *lossless* dan kompresi JPEG yang merugikan.

Otentikator menolak adanya manipulasi *replacement* karena manipulasi jenis ini sering kali digunakan untuk serangan (*attack*). Otentikator yang dibahas di dalam makalah ini tidak ditujukan untuk menerima maupun menolak metode manipulasi yang dilakukan oleh aplikasi. Tetapi beberapa manipulasi dapat dengan jelas dispesifikasikan oleh user seperti *shifting*, *cropping*, atau penambahan intensitas. Teknik otentikasi ini diperluas dan diaplikasikan untuk mengotentikasi video MPEG.

2. Sistem Otentikasi Gambar

Metode otentikasi yang dirancang ditunjukkan pada Gambar 1 di bawah. Metodenya menggunakan sebuah konsep yang sama dengan yang terdapat pada metode tanda tangan *digital* yang dikemukakan oleh Friedman [3], namun teknik mereka tidak mendukung kompresi yang *lossy*. Sebuah tanda tangan dan sebuah gambar dihasilkan pada saat yang bersamaan. Tanda tangan (*signature*) adalah bentuk terenkripsi dari kode-kode



Gambar 1 Signature Generator & Proses Otentikasi

fitur atau hasil *hash* dari gambar. Ketika user mengotentikasi gambar yang diterimanya, dia harus mendekripsi *signature*-nya dan membandingkannya dengan hasil *hash* dari gambarnya untuk dibandingkan. Bila nilainya sama, maka dapat dinyatakan bahwa gambar tersebut otentik. Perbedaan utama antara metode ini dengan metode Friedman yaitu metode ini menggunakan *property invariance* pada kompresi *lossy* JPEG sebagai kode fitur yang handal daripada menggunakan hasil *hash* dari gambar.

2.1 Invarian dari gambar sebelum dan sesudah pengkompresian JPEG

Dari proses kompresi JPEG, kita mendapatkan bahwa beberapa perubahan kuantitatif dan *properties* pembuat perkiraan dapat diekstraksi. Dua tahap di dalam proses ini mengurangi bit-bit yang dibutuhkan untuk merepresentasikan gambar, yaitu :

a. kuantisasi dan *rounding* koefisien DCT
b. pengkodean entropi

Tahap kedua merupakan operasi yang *lossless*. Sedangkan tahap pertama adalah operasi yang *lossy* yang mengubah nilai piksel namun menyimpan karakteristik visual penting dari gambar. Oleh karena itu, jika kode fitur yang handal digunakan untuk otentikasi, pastinya dapat bertahan dari tahap ini. Teorema berikut memberikan basis teknikal untuk menghasilkan kode fitur handal.

Teorema 1

Anggap F_p dan F_q adalah vektor koefisien DCT dari 2 blok *image* X berukuran 8x8 yang tidak *overlapping*, dan Q adalah tabel kuantisasi dari *JPEG lossy compression*. $\forall v \in [1,..,64]$ dan p,q $\in [1,..,\wp]$, di mana $\wp$ adalah jumlah total blok.

Didefinisikan $\Delta F_{p,q} \equiv F_p - F_q$ dan $\Delta \tilde{F}_{p,q} \equiv \tilde{F}_p - \tilde{F}_q$ di mana $\tilde{F}_p$ didefinisikan sebagai $\tilde{F}_p(v) \equiv$ *Integer*

$Round\left(\frac{F_p(v)}{F_q(v)}\right).Q(v)$. Dengan demikian, *property* berikut harus bernilai benar:

- jika $\Delta F_{p,q}(v) > 0$, maka $\Delta\tilde{\tilde{F}}_{p,q}(v) \geq 0$,
- jika $\Delta F_{p,q}(v) < 0$, maka $\Delta\tilde{\tilde{F}}_{p,q}(v) \leq 0$,
- selain itu, $\Delta F_{p,q}(v) < 0$, maka $\Delta\tilde{\tilde{F}}_{p,q}(v) = 0$.

Oleh karena semua matriks koefisien DCT dibagi dengan tabel kuantisasi yang sama, maka hubungan antara dua koefisien DCT dari posisi koordinat yang sama tidak akan berubah setelah proses kuantisasi. Pengecualiannya hanya berupa "lebih dari" atau "kurang dari" dapat menjadi "sama dengan" hasil dari efek *rounding* kuantisasi. Teorema di atas mengasumsikan bahwa tabel kuantisasi yang sama digunakan untuk memproses gambar secara keseluruhan. Teorema 1 tidak memperhatikan berapa banyak iterasi rekompresi yang dilakukan dan macam tabel kuantisasi apa yang digunakan.

Untuk implementasi yang praktis, tabel kuantisasi dapat diekstraksi dari file yang dikompresi atau diestimasi dari koefisien DCT file yang didekompresi, dengan catatan bahwa Teorema 1 hanya memperlihatkan tanda perbedaan koefisien. Teorema berikut memperluasnya dengan memberikan nilai yang berbeda dan dengan resolusi yang beragam.

Teorema 2

Dengan menggunakan parameter yang didefinisikan pada Teorema 1, dan diasumsikan sebuah ambang batas tetap $k \in \Re$. $\forall v$, didefinisikan $\tilde{k}_v \equiv$ *Integer* $Round\left(\frac{k}{Q(v)}\right)$. Dengan demikian,

- jika $\Delta F_{p,q}(v) > k$:

  $\Delta\tilde{\tilde{F}}_{p,q}(v) \geq \tilde{k}_v.Q(v)$, untuk $\frac{k}{Q(v)} \in Z$,

  $\Delta\tilde{\tilde{F}}_{p,q}(v) \geq (\tilde{k}_v - 1).Q(v)$, untuk yang lain.

- jika $\Delta F_{p,q}(v) < k$:

  $\Delta\tilde{\tilde{F}}_{p,q}(v) \leq \tilde{k}_v.Q(v)$, untuk $\frac{k}{Q(v)} \in Z$,

  $\Delta\tilde{\tilde{F}}_{p,q}(v) \leq (\tilde{k}_v + 1).Q(v)$, untuk yang lain.

- jika $\Delta F_{p,q}(v) = k$:

  $\Delta\tilde{\tilde{F}}_{p,q}(v) = \tilde{k}_v.Q(v)$, untuk $\frac{k}{Q(v)} \in Z$,

  $\Delta\tilde{\tilde{F}}_{p,q}(v) = (\tilde{k}_v$ atau $\tilde{k}_v \pm 1).Q(v)$, untuk yang lain.

Di dalam Teorema 2, *k* adalah sebuah ambang batas yang digunakan untuk membatasi perbedaan dua koefisien DCT dari posisi yang sama di dalam dua blok terpisah dari sebuah gambar. Sebaliknya, Teorema 1 hanya menjelaskan *property invariant* dari tanda $\Delta F_{p,q}$. Kita dapat menganggap Teorema 1 sebagai kasus khusus dari Teorema 2 (dengan *k* yang diset menjadi 0). Beberapa *k* yang berbeda dapat digunakan untuk sistem otentikasi *single* pada level yang berbeda. Berdasarkan Teorema 2, dapat diprediksi hubungan yang berbeda antar koefisien setelah proses kompresi.

Seperti digambarkan pada Gambar 2 di bawah ini, dengan menggunakan Teorema 1 dan Teorema 2, kita dapat mengekstraksi kode fitur Z dari hubungan antara dua koefisien DCT dari posisi yang sama di dalam dua blok yang terpisah. Kode fitur ini kemudian dienkripsi sebagai sebuah *signature*. Di dalam proses otentikasi, seorang *user* harus menghitung koefisien DCT dari gambar, dan membandingkannya dengan fitur-fitur yang didekripsi dari *digital signature* S. gambar ini dikatakan otentik apabila semua hubungan koefisien DCT memenuhi kriteria yang diprediksikan oleh fitur-fitur gambar aslinya.

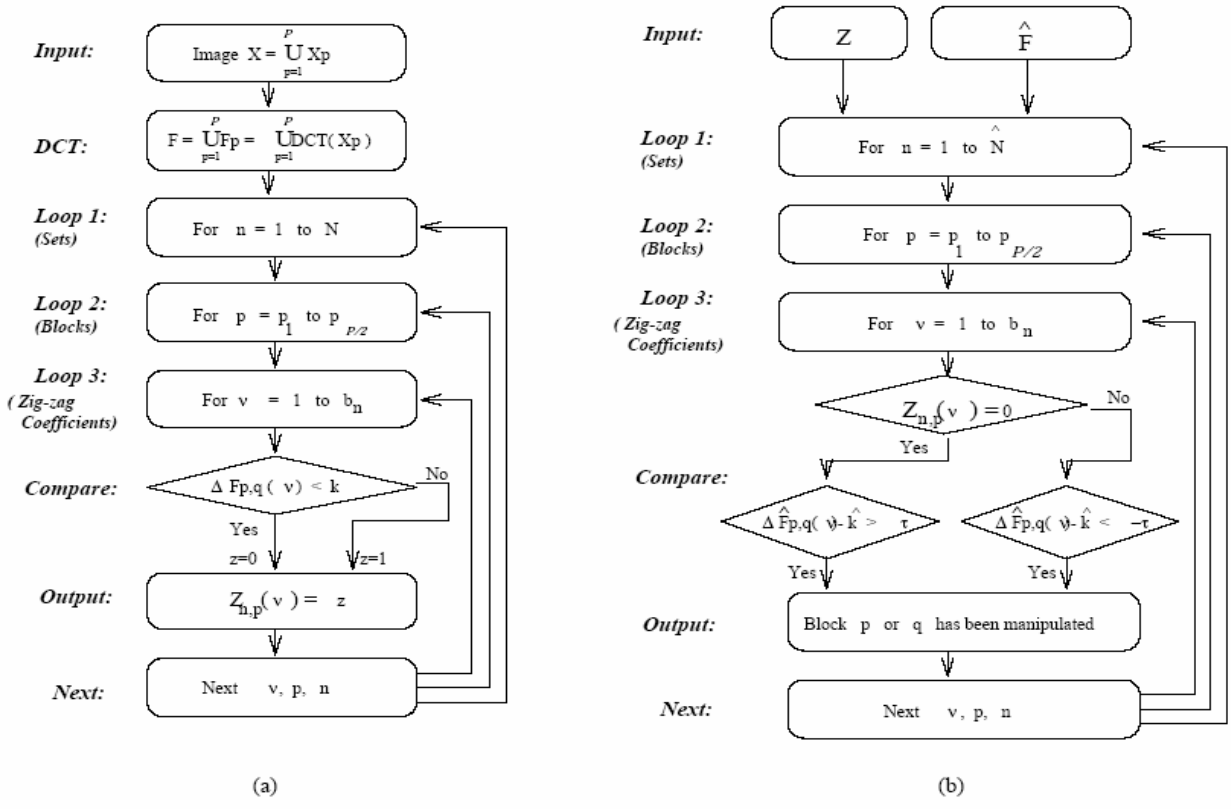
2.2 Ekstraksi Fitur pada Image Analyzer

Gambar 2.(a) merupakan diagram alir dari proses ekstraksi fitur. Pertama kali, gambar *digital* X dikirim ke dalam *image analyzer*. Tiap 8x8 blok dari gambar ini kemudian ditransformasikan ke dalam koefisien DCT. Terdapat tiga buah loop untuk menghasilkan kode fitur:

- Loop 1: menghasilkan N himpunan kode fitur $Z_{n,p}$, n = 1 sampai N. Tiap himpunan menggunakan *k* dan b_n yang berbeda, di mana *k* didefinisikan di dalam Teorema 2, b_n adalah jumlah koefisien DCT yang dibandingkan pada tiap pasangan bloknya.
- Loop 2: mengiterasi semua pasangan blok yang mungkin, $p = p_1$ sampai $p\frac{\wp}{2}$, di mana $\wp$ adalah jumlah total blok pada gambar.
- Loop 3: mengiterasi tiap b_n pasangan koefisien yang terpilih.

Pada Loop 1, N himpunan kode fitur di-*generate*. Untuk tiap himpunan, parameter b_n merepresentasikan berapa banyak bit yang dihasilkan pada tiap blok. Parameter *k* merepresentasikan ambang batas presisi yang digunakan pada Teorema 2. Himpunan pertama, *k* = 0, menjaga tanda $\Delta F_{p,q}$. Dari himpunan kedua sampai terakhir, *k* di-*set* untuk menjaga besar $\Delta F_{p,q}$ dengan akurasi yang meningkat.

Pada Loop 2, dibentuk blok-blok DCT ke dalam pasangan-pasangan blok. Seperti didefinisikan pada Teorema 2, perbedaan koefisien DCT antara blok p dan blok q dihitung. Misalkan diberikan satu set blok $P_p = \{p_1,p_2,..,\ p\frac{\wp}{2}\}$ dan himpunan blok yang lain $P_q = \{q1,q2,..,\ q\frac{\wp}{2}\}$. Sebagai contoh, P_p adalah semua blok yang genap, $\{0,2,4,..,\ \wp-1\}$, dan P_q adalah semua blok yang ganjil, $\{1,3,5,..\ \wp-2\}$. Formasi semua blok di dalam sebuah gambar ke



Gambar 2 (a) Ekstraksi Fitur (b) Authentication: Comparator

dalam pasangan blok dapat didasarkan pada fungsi pemetaan sembarang.

Pada Loop 3, untuk tiap blok, dibandingkan b_n nilai-nilai yang terpilih (di*index* dalam urutan zigzag) pada domain DCT. Nilai DC dan AC digunakan dan pertama-tama, perbedaan nilai DC pada blok p dan q, $\Delta F_{p,q}(1)$, digunakan untuk perbandingan. Bila nilai ini lebih kecil dari *k*, maka bit kode fitur z = 0 ditambahkan pada kode fitur sebelumnya. Sebaliknya, jika nilai ini lebih besar atau sama dengan *k*, maka z = 1. Setelah itu, perbedaan pada nilai AC yang terpilih dibandingkan dengan *k*. Hanya b_n-1 perbedaan AC yang digunakan di dalam proses ini. Setelah Loop 1, Loop2, dan Loop 3 dilakukan, kode fitur Z dari gambar ini di-*generate*. Biasanya, b_n posisi terpilih ditempatkan pada frequency *band* yang rendah dan menengah.

2.3 Proses Otentikasi

Gambar 1 juga mendeskripsikan mengenai proses otentikasi. Proses ini terdiri tiga bagian. Pertama, gambar yang diterima, X atau B, harus ditransformasi ke dalam domain DCT, F. Proses ini melibatkan transformasi DCT blok per blok jika sebuah gambar dasar/asal, X, digunakan. Jika yang digunakan adalah gambar JPEG yang dikompresi, B, sebuah *parser* harus digunakan untuk merekonstruksi Tabel Huffman dan Tabel Kuantisasi, Q. *Signature* S harus didekripsi untuk merekonstruksi kode fitur Z. Setelah F dan Z dihasilkan, maka F dan Z dikirim pada *Authentication Comparator* dengan tujuan untuk menentukan apakah gambar ini sudah dimanipulasi atau tidak.

Authentication Comparator ditunjukkan pada Gambar 2(b), di mana juga terdapat 3 loop sama seperti pada *Image Analyzer*. Pada Loop 1, jumlah loop n, bernilai berbeda dengan yang digunakan pada *Image Analyzer*, yaitu lebih sedikit loop yang digunakan. Sedangkan Loop 2 dan Loop 3 sama seperti yang ada di *Image Analyzer*. Di dalam loop-loop ini, harus dibandingkan hubungan koefisien DCT yang didapatkan dari gambar asal dan gambar yang diterima.

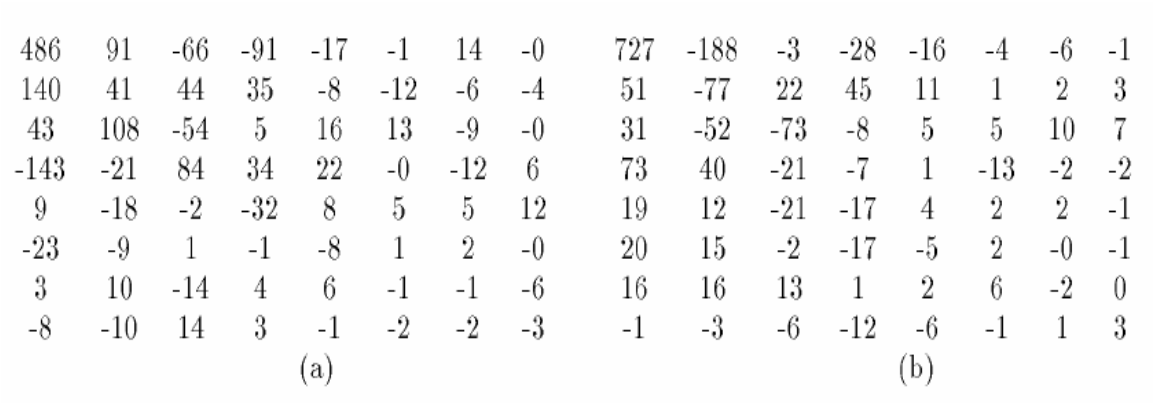
Dari Teorema 2, dapat didefinisikan:

- $\tilde{k} = \begin{cases} -\tilde{k}_v.Q(v), & \frac{k}{Q(v)} \text{ adalah sebuah } \textit{integer}, \\ -(\tilde{k}_v+1).Q(v), & \frac{k}{Q(v)} \text{ bukan integer dan } Z_n(v)=0, \\ -(\tilde{k}_v-1).Q(v), & \frac{k}{Q(v)} \text{ bukan integer dan } Z_n(v)=1, \end{cases}$

$\tilde{k}$ adalah fungsi *v*, *p*, dan *n*. Gambar 2(b) menunjukkan, jika $Z_n(v) = 0$, yaitu $\Delta F_{p,q}(v) < k$, maka $\Delta \hat{F}_{p,q}(v) - k \le 0$ harus dipenuhi. Oleh karena itu, jika $\Delta \hat{F}_{p,q}(v) - k \le 0$, dapat diketahui bahwa beberapa parameter blok p atau q pasti sudah dimodifikasi. Hasil yang sama dapat diperoleh di dalam kasus $\Delta F_{p,q}(v) \ge k$

2.4 Enkripsi, Dekripsi, dan Panjang Signature

Kode-kode fitur dienkripsi dengan sebuah kunci privat rahasia dengan Algoritma Kunci Publik. Seperti yang dijelaskan pada bab sebelumnya, panjang l_f dari kode fitur ditentukan dengan bit-bit perbandingan $\frac{\wp}{2}.\left(\sum_{n=1}^{N} b_n\right)$, *seed* dari fungsi pemetaan pasangan blok dan posisi DCT yang terpilih, dan nilai rata-rata DCT. Sebagai contoh, asumsikan ukuran gambar 320x240 = 76800 (*bytes*). 10 bit dari kode fitur digunakan untuk tiap pasangan blok, misalnya, N=1 dan b=10. Diasumsikan panjang *seed* 2 *bytes*, dan rata-rata 6 koefisien DCT di-*record*, kemudian panjang kode fitur, l_f, akan menjadi $\frac{40x30}{2}.10.\frac{1}{8} + 2 + 2 + 6$ = 760 (*bytes*). Panjang *signature* dapat lebih jauh dikurangi dengan reduksi keefektifan otentikator.



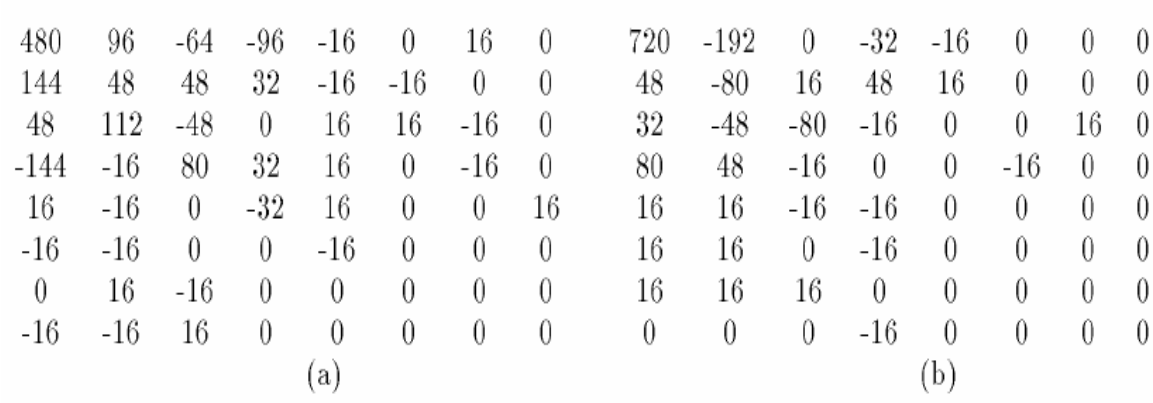
486	91	-66	-91	-17	-1	14	-0
140	41	44	35	-8	-12	-6	-4
43	108	-54	5	16	13	-9	-0
-143	-21	84	34	22	-0	-12	6
9	-18	-2	-32	8	5	5	12
-23	-9	1	-1	-8	1	2	-0
3	10	-14	4	6	-1	-1	-6
-8	-10	14	3	-1	-2	-2	-3

(a)

727	-188	-3	-28	-16	-4	-6	-1
51	-77	22	45	11	1	2	3
31	-52	-73	-8	5	5	10	7
73	40	-21	-7	1	-13	-2	-2
19	12	-21	-17	4	2	2	-1
20	15	-2	-17	-5	2	-0	-1
16	16	13	1	2	6	-2	0
-1	-3	-6	-12	-6	-1	1	3

(b)

Gambar 3 Dua blok koefisien DCT untuk area 16x8 yang dipotong dari gambar "Lenna" (daerah mata kanan)



480	96	-64	-96	-16	0	16	0
144	48	48	32	-16	-16	0	0
48	112	-48	0	16	16	-16	0
-144	-16	80	32	16	0	-16	0
16	-16	0	-32	16	0	0	16
-16	-16	0	0	-16	0	0	0
0	16	-16	0	0	0	0	0
-16	-16	16	0	0	0	0	0

(a)

720	-192	0	-32	-16	0	0	0
48	-80	16	48	16	0	0	0
32	-48	-80	-16	0	0	16	0
80	48	-16	0	0	-16	0	0
16	16	-16	-16	0	0	0	0
16	16	0	-16	0	0	0	0
16	16	16	0	0	0	0	0
0	0	0	-16	0	0	0	0

(b)

Gambar 4 Koefisien DCT pada Gambar 3 yang dikuantisasi dengan sebuah matriks yang seragam

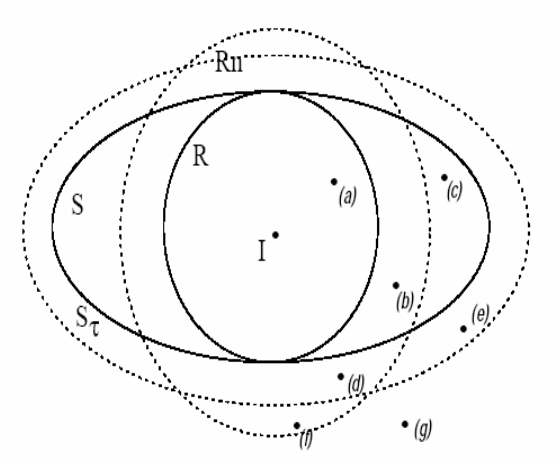
Algoritma Kunci Publik digunakan sehingga *user* manapun dapat dengan mudah mengakses sebuah kunci publik untuk mendekripsikan *signature*. Algoritma kunci publik yang paling terkenal yaitu RSA. Panjang kunci RSA bervariasi tetapi pada umumnya panjang yang digunakan yaitu 512 bit, di mana ukuran blok *message* harus lebih kecil dari panjang kunci. Jika kode fitur dibagi ke dalam blok B-bit, maka akan dibutuhkan $\left\lceil l_f .8.\frac{1}{B} \right\rceil$ perhitungan RSA. Diasumsikan panjang *output* tiap RSA yaitu l_f, maka panjang *signature* akan menjadi $\left\lceil l_f .8.\frac{1}{B} \right\rceil .l_f$ bit. Sebagai contoh, pada contoh sebelumnya, bila B = 510 dan l_r = 511 digunakan, maka algoritma RSA harus dijalankan 12 kali dan panjang urutan zig zag dari dua blok dibandingkan. Dalam kasus ini, panjang kode fitur Z menjadi 10 bit (b_1 = 10), $\Delta F_{1,2}(1) = -241 < 0$. Oleh karena itu, bit pertama dari kode fitur Z adalah 0. Koefisien kedua di dalam urutan zig zag yaitu: $F_1(2)$ = 91 dan $F_2(2)$ = -188 masing-masing. Oleh karena $\Delta F_{1,2}(2) = 279 > 0$, maka bit kedua dari fitur kedua adalah 1. Setelah dilakukan 10 iterasi, kode fitur Z-nya yaitu: 0111100110.

Sekarang anggap kode fitur lebih panjang dengan meng-*set* N = 4, b_1 = 10, b_2 = 6, b_3 = 3, dan b_4 = 1. Alasan menurunkan nilai b_n yaitu supaya koefisien-koefisien berfrekuensi rendah membutuhkan lebih banyak proteksi daripada koefisien-koefisien berfrekuensi tinggi. Nilai-nilai ambang batas *k*-nya yaitu 0, 128, 64, dan 32. 10 bit pertama dari kode fitur Z adalah sama seperti kasus sebelumnya. Untuk 6 bit selanjutnya, enam koefisien pertama dibandingkan lagi dengan menggunakan |*k*| = 128. Sebagai contoh, karena $\Delta F_{1,2}(1) = -241 < -128$, maka bit ke-11

= 0. $\Delta F_{1,2}(2) = 279 > 128$, maka bit ke-12 1. Dengan demikian, kode fiturnya yaitu : 01111001100100010110. Panjang Z yaitu $\sum_{n=1}^{4} b_n = 20$.

Gambar 4 menunjukkan koefisien DCT setelah kuantisasi (misalnya, $\tilde{F}_1$ dan $\tilde{F}_2$) dengan sebuah matriks 16. Hal ini untuk mensimulasikan proses kuantisasi di dalam JPEG. Dengan menggunakan Gambar 2(b), kita dapat mengotentikasi gambar yang dikompresi dengan membandingkan $\Delta\tilde{F}_{1,2}$ kode fitur Z. Sebagai contoh, $\Delta\tilde{F}_{1,2}(1) = -240 < 0$ dan $Z_1(1) = 0$, nilai ini diotentikasi menjadi benar. Proses yang sama berlanjut sampai semua kode fitur digunakan. Perlu dicatat bahwa jika tabel kuantisasi tidak diketahui oleh otentikator, maka himpunan pertama kode (dengan *k* = 0) masih dapat diverifikasi.

Sekarang kita ambil sebuah contoh manipulasi. Diasumsikan X(0,2) dan X(0,3) dimodifikasi dari 72 dan 26 ke 172 dan 126. (X dapat diperoleh dari IDCT Gambar 3). Diasumsikan kita menggunakan matriks kuantisasi yang sama. Dengan mengulang proses di atas, otentikator akan mendeteksi manipulasi karena adanya *mismatch* bit ke-4 dari kode fitur.



Gambar 5 Ilustrasi konseptual dari skenario yang 'miss', 'false alarm', dan yang lainnya

=

3 Sistem Otentikasi Video MPEG

Untuk memperluas aplikasi teknik otentikasi gambar, dua kunci pokok yang harus diperhatikan yaitu:
(1) proses mentranskode dan mengedit
(2) ukuran dari *digital* signature.
Oleh karena video *digital* jarang direkam dalam *raw format*, jadi yang menjadi bahan pertimbangan di sini terfokus pada *source* untuk otentikasi pada format MPEG-1 dan MPEG-2.

Dalam merancang sebuah sistem untuk mengotentikasi isi dari video yang terkompresi, kita harus mengetahui tipe manipulasi yang mungkin dilakukan dan dapat diaplikasikan di dalam video. Pada umumnya, 5 proses transkode dan *editing* yang dapat dilakukan yaitu:

1.*Dynamic Rate Shaping*:

Sebuah skema *rate-control real-time* dalam domain yang terkompresi. Teknik ini mengatur titik-titik kontrol dinamis untuk menurunkan koefisien DCT berfrekuensi tinggi pada tiap 8x8 blok di dalam sebuah *macroblock*, sedangkan vektor *motion*nya tidak diubah.

2.*Rate Control* tanpa *Drift Error Correction*:

Teknik ini juga dilakukan pada domain yang terkompresi. Koefisien-koefisien DCT direkuantisasi untuk memenuhi konstrain *bit-rate* yang berbeda, sedangkan vektor *motion*nya tidak diubah.

3.*Rate Control* dengan *Drift Error Correction*:

Teknik ini memperbaiki kualitas video setelah rekuantisasi koefisien DCT-nya, tetapi membutuhkan banyak komputasi. Koefisien-koefisien DCT dari residu blok *intercoded* dimodifikasi untuk memenuhi perubahan dari blok *intercoded* yang direkuantisasi. Dalam kasus ini, vektor *motion*nya tidak diubah.

Tabel 1 Property konsisten pada Situasi Proses Transkode dan Editing

	Situation 1	Situation 2	Situation 3	Situation 4	Situation 5
DCT (residual) coefficients	X (drop some coefficients)	X (requantization)		X	
Motion Vectors	X	X	X	X	
Picture Type (I,P,B)	X	X	X	X (inconsistent in boundary)	

4.*Editing* dengan *Consistent Picture Type*:

Jenis gambar seperti I, P, dan B dipastikan tidak diubah pada setiap generasi *editing*nya dan dapat digunakan dalam membuat *sequence* baru dengan memotong dan mengkopi beberapa segmen video. Batasan GOP (*Group of Picture*) pada tiap segmennya tidak diubah kecuali lokasi dekat area pemotongan. Nilai-nilai pikselnya dapat diubah untuk memperbaiki kualitas video seperti pengubahan intensitas, *filtering*, dll.

5.*Editing* atau *Transcoding* dengan *Inconsistent Picture Type*:

Pada beberapa proses, video yang dikompresi ditransformasi ke dalam *bitstream* yang tidak terkompresi yang mana nantinya diedit dan di*re-encode*. Struktur GOP dan vektor *motion* dapat diubah pada kasus ini. Jenis proses ini meliputi tranformasi format antara standar kompresi yang berbeda dan konversi jenis gambar.

Ketiga proses yang pertama digunakan untuk mengubah *bitrate*. Proses-proses tersebut dioperasikan pada domain yang terkompresi. Dari Tabel 1 dapat diketahui bahwa setelah dilakukan proses transkode, vektor *motion* dan jenis gambar dipreservasi. Satu-satunya pengubahan yaitu pada koefisien DCT *intra macroblock* dan koefisien residual DCT *non-intra macroblock*.

Memotong dan mengkopi beberapa segmen video MPEG untuk menghasilkan *sequence* video yang baru merupakan perlakuan yang umum. Hal ini dapat dilakukan dalam 2 metode, yaitu situasi proses 4 dan 5. Perbedaan di antara keduanya yaitu apakah struktur GOP dipreservasi melalui proses *editing*. Pada situasi 4, terdapat 2 macam GOP di dalam *sequence* video yang dihasilkan: GOP orisinal dan GOP buatan. Sebuah GOP orisinal berasal dari *sequence* video yang orisinal dengan strukturnya yang utuh, sedangkan GOP buatan dihasilkan dari gambar-gambar *sequence* video orisinal. Pada prakteknya, jumlah GOP buatan jauh lebih sedikit daripada GOP orisinalnya. Dari situasi ini, kita hanya akan fokus pada GOP orisinal.

Signature otentikasi video dapat dihasilkan dalam situasi yang berbeda. Kita dapat melihat bahwa untuk Situasi 1-4, struktur GOPnya tidak dimodifikasi setelah proses transkode atau *editing*. Oleh karena itu, kita dapat menghasilkan sebuah *robust digital signature* yang dapat bertahan dari manipulasi semacam ini. Kita dapat menyebut kasus ini *sebagai robust digital signature* jenis I. Sedangkan untuk Situasi 5, oleh karena struktur GOPnya sudah dirusak, hanya nilai piksel dari gambar yang akan dipreservasi. Oleh karena itu, *sequence* videonya seperti sebuah himpunan *frame* gambar, yang dapat diotentikasi dengan otentikasi gambar yang sudah dijelaskan pada bab 2. Kita dapat menyebut kasus ini sebagai *robust digital signature* jenis II.

3.1 Sintaks *Sequence* Video MPEG

Di dalam standar MPEG, tiap *sequence* video terdiri dari beberapa *Group of*

Picture (GOP) yang sekuensial. Sebuah GOP merupakan unit independen yang terdiri dari beberapa gambar. Pada MPEG-1, setiap *frame* adalah sebuah gambar. Di dalam MPEG-2, sebuah gambar dapat menjadi sebuah gambar bidang atau gambar *frame*. Terdapat beberapa potongan di dalam sebuah gambar. Sebuah potongan adalah suatu *string MacroBlocks* (MBs) bertautan dengan panjang sembarang yang berjalan dari kiri ke kanan suatu gambar. MB merupakan unit pengganti *motion* 16x16 yang terdiri dari beberapa blok 8x8. (Sebuah MB terdiri dari 6 blok dengan format krom 4:2:0, 8 blok dengan format krom 4:2:2, atau 12 blok dengan format krom 4:4:4). Setiap blok dapat berupa blok *inter-coded* atau *non-intra-coded*. Pada MPEG, sama halnya dengan JPEG, blok *intra-coded*-nya mempunyai koefisien DC sendiri yang berhubungan dengan blok sebelumnya pada jenis YCbCr yang sama, bila tidak demikian, maka blok sebelumnya merupakan blok *non-intra*, yang dimiliki oleh MB yang diloncati atau potongan lainnya. Koefisien AC dari tiap blok di dalam sebuah *macroblock* dikuantisasi dengan *quantization_step_size* seperti yang diberikan berikut ini:

$$(\kappa.Q[m][n])/(8.v),$$
$$\text{m, n} = 0,1,..,7,\ \text{m} + \text{n} \neq 0$$

di mana κ adalah *quantizer_scale* dan Q adalah matriks kuantisasi yang merupakan Intra QMatrix dari blok intra atau NonIntra QMatrix dari blok non-intra. Kedua blok tersebut dapat didefinisikan di dalam *header sequence* video jika nilainya berbeda dari nilai defaultnya. Parameter *v* sama dengan 1 untuk *sequence* video MPEG-1, atau sama dengan 2 untuk *sequence* video MPEG-2. *Quantizer_scale* κ, diset untuk sebuah potongan MB atau sebuah MB.

3.2 *Robust Digital Signature*

3.2.1 *Robust Digital Signature*: Jenis I

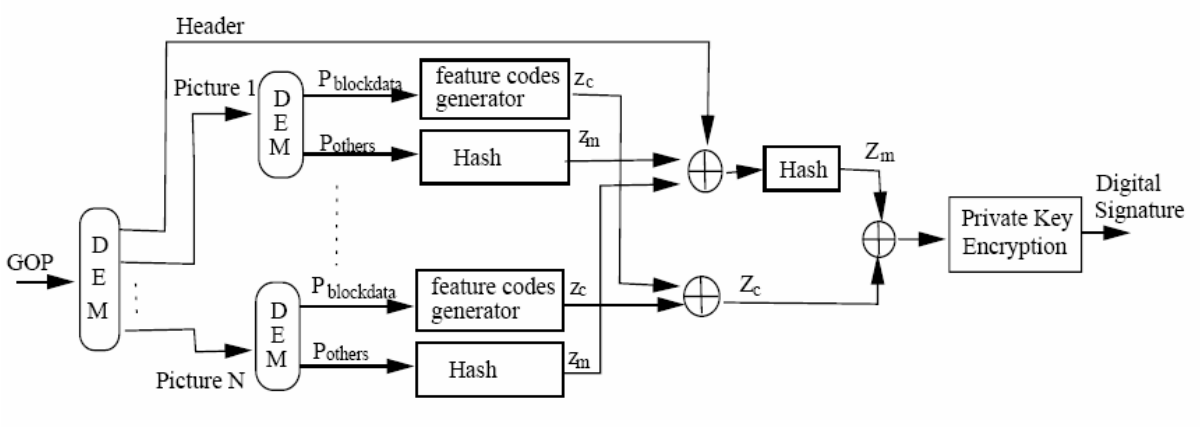
Pada bab sebelumnya, sudah dipaparkan mengenai hubungan antara pasangan koefisien, misalnya dua koefisien DCT dari posisi koordinat yang sama, pada tiap 8x8 blok sebuah gambar yang seharusnya bernilai sama setelah dilakukan proses rekuantisasi bila *quantization_step_size* yang sama diaplikasikan pada blok. Sudah dipaparkan juga bahwa perubahan nilai perbedaan dari pasangan koefisien setelah proses rekuantisasi seharusnya memiliki batasan di dalam *range* yang dispesifikasikan oleh *quantization_step_size*, yang mungkin saja bisa berbeda pada blok-bloknya. Oleh karena itu, kita dapat menyesuaikan semua blok ke dalam sebuah gambar untuk membentuk pasangan blok dan men-*generate* beberapa kode untuk merepresentasikan status hubungan koefisien-koefisiennya di dalam posisi koordinat yang dipilih. Kode-kode yang di-*generate* kemudian dienkripsi oleh kunci publik untuk membentuk sebuah *digital signature*.

Untuk men-*generate* sebuah *robust digital signature* dari situasi proses 4, dapat digunakan koefisien DCT yang terkuantisasi (intra atau non-intra) dari sebuah matriks pada tiap *macroblock*-nya untuk membentuk pasangan pembandingan. Oleh karena nilai κ sama halnya seperti *quantization_step_size* yang selalu sama pada semua blok dari sebuah *macroblock*, hubungan relatif koefisien-koefisien pada posisi blok yang berkaitan tidak berubah selama proses transkode. Oleh karena itu, sama halnya dengan proses generasi *signature* pada gambar, kita dapat menggunakannya untuk men-*generate* kode fitur. Pertama-tama, kode fitur Z_c dari sebuah *macroblock* dapat diformulasikan sebagai berikut:

$$Z_c = VLC\left(\bigcup_P \bigcup_b \operatorname{sgn}\left[f_p(b) - f_{w(p)}(b)\right]\right)$$

di mana:

- *f* merepresentasikan koefisien DCT yang terkuantisasi pada *sequence* video yang terkompresi. Variabel ini diekstraksi dari *bitstream* dan didekodekan dengan *Variable Length Coding* (VLD).



Gambar 6 Robust Digital Signature jenis I

- *p* adalah himpunan blok-blok yang terpilih di dalam sebuah *macroblock*, dan W adalah fungsi pemetaan yang memetakan setiap blok pada *p* ke dalam blok-blok yang berkaitan untuk membentuk sebuah pasangan blok. Sebagai contoh, di dalam format 4:2:0, jika kita memberi label 4 blok putih dan 2 blok berwarna sebagai Blok 1-6, maka kita dapat memilih p sebagai {1,3,5} dan himpunan q = W(p) = {2,4,6} yang membentuk tiga pasangan blok yaitu {1,2}, {3,4}, dan {5,6}. Untuk *macroblock* yang terdiri dari $\wp$ blok, maka akan terdapat $\wp!$ kombinasi.
- *b* adalah himpunan posisi koefisien DCT yang dipilih dan direpresentasikan dengan urutan zig zag seperti yang digunakan pada *sequence* video.
- Fungsi tanda didefinisikan sebagai:
  (1) *sgn(f) = 1*, jika *f > 0*
  (2) *sgn(f) = 0*, jika *f = 0*
  (3) *sgn(f) = -1*, jika *f < 0*

Fungsi tanda ini digunakan untuk merepresentasikan nilai perbedaan karena terdapat banyak nilai nol di dalam koefisien DCT dari *sequence* video yang terkompresi. Dengan demikian, kita membedakannya dari dua situasi yang berbeda seperti positif dan negatif. Hal ini berbeda dengan apa yang dilakukan pada otentikasi gambar. Oleh karena hasil pembandingan koefisien-koefisien banyak yang bernilai nol, maka metode VLC dapat dilakukan untuk mengurangi panjang kode fitur.

Selain memproteksi koefisien DCT, kita juga harus memproteksi informasi lainnya yang meliputi vektor *motion* dan kode kontrol. Hal ini dapat dilakukan dengan menambahkan kode fitur dengan nilai *hash* dari *bitstream* bekas di dalam *sequence* video. Pada tahap pertama, diasumsikan sebuah *Picture* P yang terdiri dari P_{block_data} dan P_{others}, di mana P_{block_data} meliputi kode-kode dari koefisien DCT, *quantizer_scale* pada *header* MB, dan kode kontrolnya. P_{others} meliputi semua kode yang lain di dalam P, Lalu, nilai *hash* didapatkan dengan formulasi sebagai berikut:

$$z_m = Hash(P_{others}) \quad (*)$$

di mana z_m digunakan untuk memproteksi informasi lainnya dari *Picture*.

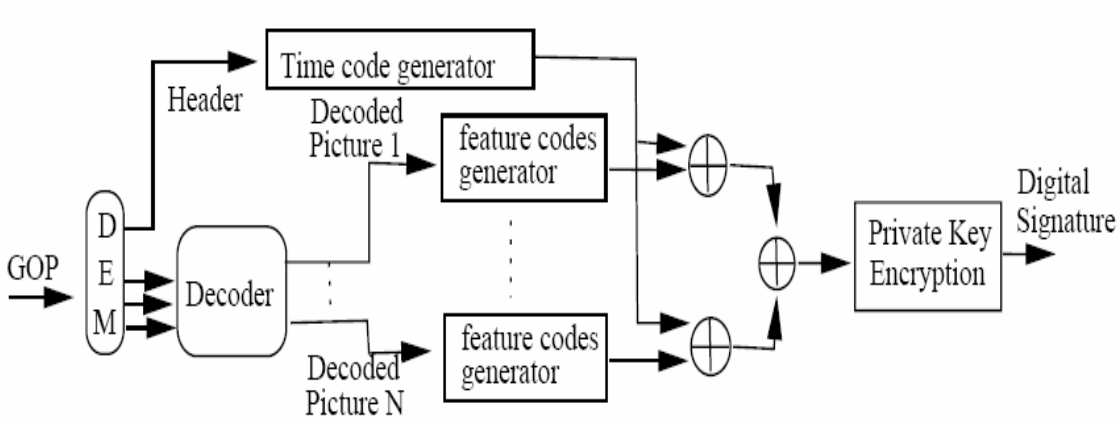
Karena GOP merupakan unit dasar yang independen dari sebuah *sequence* video di dalam *bitstream* MPEG, kita dapat mengenkripsi kode fitur dan nilai *hash* tiap gambarnya di dalam GOP untuk membentuk *digital* signature, misalnya:

$$DS = Private\ Key\ Encrypted(Z_c, Z_m)$$

di mana $Z_c = \bigcup_{Picture} VLC\left(\bigcup_{MBs} z_c\right)$ merupakan kombinasi kode fitur z_c dari semua *macroblock* di dalam GOP, dan

$$Z_m = Hash(GOP\_Header, z_{m,1}, z_{m,2}, \ldots, z_{m,N}) \quad (**)$$

di mana N merepresentasikan jumlah total *Picture* di dalam sebuah GOP. Persamaan (**) mengindikasikan bahwa daripada



Gambar 7 Robust Digital Signature jenis II

menggunakan kombinasi nilai *hash* dari tiap gambar, panjang Z_m lebih jauh dapat diperpendek dengan meng-*hash* nilai kombinasi, karena semua informasi ini ditetapkan selama proses transkode. Karena *GOP_Header* terdiri dari *time_code* yang mengacu pada gambar pertama yang di*display* setelah *header* GOP yang memiliki *temporal_reference* nol, maka sangat penting untuk menyertakannya pada *digital signature* untuk mencegah kekacauan temporal pada GOP. *Digital Signature* ini dapat ditempatkan di dalam area *user_data* dari *header* GOP. (Pada standar MPEG, *user_data* dapat di-*embed* di dalam *header sequence*, *header* GOP, atau *header* gambar).

3.2.2 *Robust Digital Signature*: Jenis II

Jenis kedua dari *Robust Digital Signature* didesain untuk melakukan proses pertahanan seperti pada Situasi 5. Karena struktur GOP, vektor *motion*, atau koefisien residual DCT dapat berubah pada situasi ini, satu-satunya *property* yang konsisten yaitu nilai piksel gambar-gambarnya. Oleh karena itu, harus di-*generate digital signature* berdasarkan nilai piksel dari tiap gambar. Dengan menggunakan metode otentikasi yang sama untuk gambar, dapat di-*generate digital signature* dari gambar. Metode ini dijelaskan sebagai berikut:

1. Merekonstruksi nilai piksel gambar dari berbagai macam tipe gambar (I, P, B)
2. Men-*generate* kode fitur dengan menggunakan prosedur yang sama seperti pada bab 2, misalnya membagi gambar ke dalam blok 8x8, membentuk pasangan blok, membandingkan koefisien DCTnya pada pasangan blok yang terbentuk, menggunakan satu bit untuk merepresentasikan tiap perbandingan.
3. Menambah kode fitur dengan kode waktu dari tiap gambar
4. Menggunakan enkripsi kunci privat untuk membentuk *digital signature*.

Diagram yang menjelaskan metode ini ditunjukkan dengan gambar di bawah.

3.3 Otentikator

3.3.1 Otentikasi *Sequence* Video Setelah *Transcoding*

Otentikator dapat diimplementasikan sebagai sebuah penambahan *general decoder*. Pada otentikator, *digital signature* diekstraksi dari *header* GOP dan didekripsi untuk mendapatkan kode fitur dan nilai *hash*nya. Untuk memeriksa otentikasi dari sebuah GOP di dalam *sequence* video, sama seperti proses dalam men-*generate signature*, tiap gambar di dalam GOP dibagi menjadi dua bagian: P_{block_data} dan P_{others}. Kemudian dari dua bagian ini dapat diotentikasi secara terpisah. Untuk mengotentikasi nilai *hash*, kita dapat memperoleh $\hat{Z}_m$ dari GOP dengan menggunakan fungsi *hash* yang sama seperti persamaan (**) dan (*). Selama bagian informasi ini utuh selama proses transkode, $\hat{Z}_m$ diharapkan bernilai sama dengan Z_m. Kalau tidak, GOP ini pastinya

sudah dimodifikasi oleh beberapa proses yang lain.

Untuk mengotentikasi kode fitur GOP, otentikator pertama-tama harus melakukan *decoding* VLC pada kode fitur untuk mendapatkan tanda hubungan dari koefisien-koefisien yang terpilih dari tiap pasangan bloknya. Dengan melakukan prosedur yang sama seperti yang dilakukan pada [4,5], kita dapat mengotentikasi apakah koefisien DCTnya sudah dimodifikasi karena:

- Pada Situasi 1, beberapa koefisien berfrekuensi tinggi di dalam sebuah blok dapat diturunkan dan di-*set* menjadi nol. Mengacu pada Teorema 1 pada bab 2.1, jika dua koefisien DCT dua-duanya bernilai sama dengan nol setelah proses transkode, maka otentikator menganggapnya otentik. Karena koefisien berfrekuensi rendah dipreservasi selama transkode, maka dapat dipastikan hubungan mereka akan sama dengan yang aslinya.

- Pada Situasi 2, koefisien DCT dapat direkuantisasi untuk memenuhi beberapa kendala *bitrate*. Oleh karena semua koefisien DCT pada posisi yang sama dari blok-blok yang berada pada MB selalu dikuantisasi dengan *quantization_step_size* yang sama, maka menurut teorema yang sama pada bab 2.1, perubahan yang mungkin dari nilai tanda perbedaan sebuah pasangan koefisien yaitu: “positif ke positif”, “positif ke nol”, “nol ke nol”, “negatif ke negatif”, dan “negatif ke nol”. Jika ditemukan hubungan koefisien-koefisien yang tidak memenuhi *rule* ini, maka dapat diklaim bahwa *sequence* video sudah dimodifikasi oleh manipulasi yang lain.

- Pada Situasi 3, koefisien DCT dari blok intra dapat direkuantisasi. Dan juga, koefisien residu DCT dari blok non-intra dapat diubah untuk mengganti kerugian *error* kuantisasi, kemudian direkuantisasi kembali. Untuk mengotentikasi blok-blok ini, dapat ditentukan beberapa nilai batas toleransi.

Bila didefinisikan $\Delta f_{p,q}(b) = f_p(b) - f_{W(p)}(b)$ di mana merupakan perbedaan koefisien-koefisien pada posisi b di dalam pasangan blok *(p, W(p))* video orisinal, dan $\Delta \hat{f}_{p,q}(b) = \hat{f}_p(b) - \hat{f}_{W(p)}(b)$ di mana merupakan perbedaan koefisien dari video yang diperiksa. Kemudian, *property-property* berikut harus dipenuhi,

- Jika $\Delta f_{p,q}(b) > 0$,
  maka $\Delta \hat{f}_{p,q}(b) \geq -\tau$ (a)
- Jika $\Delta f_{p,q}(b) = 0$,
  maka $-\tau \leq \Delta \hat{f}_{p,q}(b) \leq \tau$ (b)
- Jika $\Delta f_{p,q}(b) < 0$,
  maka $\Delta \hat{f}_{p,q}(b) \leq \tau$ (c)

di mana

- τ = 0, untuk intrablock (d)
- $\tau = 1 + \sum_i \frac{\hat{\kappa}_{ref_i} . Q_{ref_i}(b)}{\hat{\kappa} . Q_{non\,intra}(b)}$, untuk nonintrablock (e)

Pada persamaan di atas, $\hat{\kappa}$ yaitu *quantizer_scale* blok non-intra p dan q di dalam *sequence* video yang diperiksa. Himpunan *i* merepresentasikan jumlah blok acuan, misalnya i = {1} untuk blok non-intra pada gambar P pertama GOP, atau i = {1,2} pada gambar P kedua GOP. Parameter $\hat{\kappa}_{ref_i}$ dan $\hat{Q}_{ref_i}$ masing-masing adalah *quantizer_scale* dan *quantizer_matrix* dari blok acuan ke-i. Otentikator dapat memeriksa koefisien-koefisien dengan persamaan a-e di atas, bila tidak memenuhi, maka dapat dipastikan bahwa *sequence* video pasti sudah dimodifikasi dengan manipulasi tertentu.

Di sisi lain, seorang *attacker* dapat mengacaukan urutan temporal dari GOP untuk mengubah makna dari *sequence* video. Manipulasi ini dapat dideteksi dengan memeriksa kode waktu pada *header* GOP yang diproteksi di dalam *digital*

signature. Perubahan urutan temporal gambar di dalam sebuah GOP dapat dideteksi karena kode fitur dan nilai *hash digital signature* dua-duanya di-*generate* dalam urutan gambar yang ada.

3.3.2 Otentikasi *Sequence* Video Setelah *Editing*

Robust Digital Signature jenis I digunakan pada Situasi 4. Pada situasi ini, terdapat dua macam GOP yaitu GOP orisinal dan GOP buatan. GOP orisinal memiliki *digital signature* yang independen yang mana dapat diperiksa dengan metode otentikasi yang sama dengan sebelumnya. GOP buatan dihasilkan dari segmen-segmen gambar yang diambil dari *sequence* video orisinal. Mungkin tidak dapat dihasilkan GOP buatan jika kita membuat *sequence* video tidak mampu memotong GOPnya.

Pada editor video yang terkompresi, bila *digital signature source* GOP yang berkaitan disalin ke dalam *header* GOP buatan, maka gambar-gambar tersebut dapat diperiksa. Namun, otentikator tidak dapat memeriksa gambar-gambar tersebut melalui konversi jenis. Kalau tidak demikian, bika *digital signature* tidak disalin ke dalam GOP buatan, maka tidak ada indikasi akan dibutuhkannya pemeriksaan otentikasi.

Untuk situasi 5, semua nilai piksel setiap gambar dapat berubah. Tetapi, perubahan ini seperti *noise* dan biasanya berpengaruh sedikit sehingga perubahan ini tidak mengubah makna dari isi video. Perubahan nilai piksel yang seperti *noise* ini juga menyebabkan perubahan kecil pada domain DCT. Oleh karena itu, perubahan yang besar di dalam domain DCT dapat diasumsikan berasal dari manipulasi yang merugikan. Kita dapat mengotentikasi setiap gambar dengan beberapa nilai toleransi τ. Dengan mengaplikasikan persamaan a-e, bila semua pasangan koefisien memenuhi persamaan, maka dapat diklaim bahwa *sequence* video yang diperiksa otentik.

Oleh karena tidak adanya batas toleransi yang pasti, maka otentikator hanya dapat mengindikasikan beberapa area dari sebuah gambar apakah area-area tersebut sudah dimanipulasi atau belum. Hal ini dilakukan dengan mengamati hasil otentikasi gambar dengan nilai toleransi yang berbeda. Sebagai contoh, jika τ = 0, kita dapat memastikan bahwa otentikator menganggap sebagian besar blok-blok di dalam gambar sudah dimanipulasi. Tetapi, sejalan dengan meningkatnya nilai τ, kita dapat memastikan bahwa hanya beberapa area yang dimanipulasi saja yang dapat dideteksi oleh otentikator.

Kode waktu yang tergabung di dalam *digital signature* dapat digunakan untuk mendeteksi perubahan di dalam urutan temporalnya dan mengindikasikan nilai piksel di dalam gambar pada waktu yang spesifik. Selama *sequence* video diotentikasi gambar per gambar, maka otentikasinya masih dapat diperiksa.

4. Kesimpulan

Teknik otentikasi gambar pada makalah ini membedakan adanya *JPEG lossy compression* dengan manipulasi merugikan yang lainnya. Pada prakteknya, *file-file* gambar dapat dikompresi dan didekompresi beberapa kali dan masih dianggap otentik dengan gambar aslinya. Sama halnya dengan beberapa manipulasi seperti *integral value rounding*, transformasi warna, dan pemotongan. Teknik otentikasi yang diaplikasikan pada gambar di makalah ini mengijinkan adanya manipulasi JPEG *lossy compression* tetapi tidak untuk manipulasi lain yang merugikan.

Pada dasarnya, teknik otentikasi gambar dirancang sedemikian rupa supaya dengan adanya manipulasi dapat diketahui otentikasinya dengan gambar yang dikirim oleh pengirim tertentu. Dasar dari pembuatan *digital signature*nya relatif sama dengan metode otentikasi pada dokumen teks, tetapi ada beberapa hal yang membedakannya, yaitu:

1. Bila pada dokumen teks, *file* diubah menjadi *message digest* dengan menjalankan fungsi *hash*. Tetapi pada dokumen gambar, file diubah menjadi

kode fitur dengan menjalankan *image analyzer*.

2. File dokumen teks tidak diubah menjadi bentuk yang lain baik sebagai input proses ekstraksi *message digest* ataupun otentikasi, tetapi pada file gambar harus ditransformasi terlebih dahulu menjadi blok-blok koefisien DCT yang nantinya akan digunakan sebagai proses mengektraksi kode fitur dan proses otentikasi.

3. Pada proses otentikasi, file dokumen teks hanya menggunakan fungsi *hash* yang sama dengan fungsi *hash* pada ekstraksi *message digest* untuk mengubah *message* yang dikirim menjadi *message digest* yang nantinya akan dijadikan perbandingan, sedangkan pada *file* gambar menggunakan *parser* untuk merekonstruksi Tabel Huffman dan Tabel Kuantisasi dalam usahanya mengekstraksi kode fiturnya.

4. Proses otentikasi pada dokumen teks menghasilkan keputusan otentik atau tidak yang ditujukan pada dokumen secara utuh, sedangkan pada proses otentikasi dokumen gambar menghasilkan keputusan otentik atau tidak pada sebagian kecil atau blok gambar saja, karena *file* gambar rentan terhadap manipulasi yang merugikan.

Untuk file video, proses otentikasinya hampir sama dengan file gambar, perbedaannya yaitu *raw data* yang akan dikenakan operasi yaitu berupa *Group of Picture* (GOP). Otentikasi yang dilakukan meliputi otentikasi *sequence* video setelah proses transkode dan setelah proses *editing*. Pada intinya proses ekstraksi *digital signature* melibatkan *header* GOP yang kemudian didekripsi untuk mendapatkan kode fitur dan nilai *hash*nya. GOP sendiri nantinya akan dibagi menjadi dua bagian yaitu P_{block_data} dan P_{others}.

DAFTAR PUSTAKA

[1] Lal, Sunder. *"A Cryptographic Study for Digital Signature Scheme,"* 2003.

[2] Yung, Ching, *"Watermarking and Digital Signature Techniques for Multimedia Authentication and Copyright Protection,"* 2000.

[3] G.L.Friedman, *"The Trustworthy Digital Camera: Restoring Credibility to the Photographic image,"* IEEE Trans. on Consumer Electronics, Vol.39, No.4, pp.905-910, November 1993.

[4] C.Y.Lin, S.F.Chang , *"A Robust Image Authentication Method Distinguishing JPEG Compression from Malicious Manipulation,"* IEEE Trans. on Circuit and System for Video Technology, 2001.

[5] C.Y.Lin, S.F.Chang, *"A Robust Image Authentication Method Surviving JPEG Lossy Compression,"* SPIE Storage and Retrieval of Image/Video Databases, San Jose, January 1998.

[6] Munir, Rinaldi, "Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung," 2006

[7] Yung, Ching; S.F. Chang, *"Issues and Solution for Autheticating MPEG Video,"* 1999.

[8] Yung, Ching; S.F. Chang, *"Bibliography of Multimedia Authentication Research Papers,"* 1999.