Register : Jurnal Ilmiah Teknologi Sistem Informasi 5 (2) 2019 128-139
ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

Tersedia *online* di www.journal.unipdu.ac.id
Unipdu
Terakreditasi S2 – SK No. 34/E/KPT/2018
Halaman jurnal di www.journal.unipdu.ac.id/index.php/register

Manajemen *jpeg/exif file fingerprint* dengan algoritma *Brute Force string matching* dan *Hash Function* SHA256

Management of jpeg/exif file fingerprint with Brute Force string matching algorithm and Hash Function SHA256

Rachmad Fitriyanto [a], Anton Yudhana [b], Sunardi Sunardi [c]

[a] *Teknik Informatika, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*
[b,c] *Teknik Elektro, Universitas Ahmad Dahlan, Yogyakarta, Indonesia*

email: [a] fitriyanto7477@gmail.com, [b] yudhana@ee.uad.ac.id, [c] sunardi@mti.uad.ac.id

INFO ARTIKEL

Sejarah artikel:
Menerima 15 Mei 2019
Revisi 6 Agustus 2019
Diterima 12 September 2019
Online 12 September 2019

Kata kunci:
Brute Force string matching
file fingerprint
exif
jpeg
SHA256

Keywords:
Brute Force string matching
file fingerprint
exif
jpeg
SHA256

Style APA dalam menyitasi artikel ini:
Fitriyanto, R., Yudhana, A., & Sunardi, S. (2019). Manajemen jpeg/exif file fingerprint dengan algoritma Brute Force string matching dan Hash Function SHA256. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 5(2), 128-139.

ABSTRAK

Metode pengamanan berkas gambar jpeg/exif saat ini hanya mencakup aspek pencegahan, belum pada aspek deteksi integritas data. *Digital Signature Algorithm* (DSA) adalah metode kriptografi yang digunakan untuk memverifikasi integritas data menggunakan *hash value*. SHA256 merupakan *hash function* yang menghasilkan 256-bit *hash value* yang berfungsi sebagai *file fingerprint*. Penelitian ini bertujuan untuk menyusun *file fingerprint* dari berkas jpeg/exif menggunakan SHA256 dan algoritma *Brute Force string matching* untuk verifikasi integritas berkas jpeg/exif. Penelitian dilakukan dalam lima tahap. Tahap pertama adalah identifikasi struktur berkas jpeg/exif. Tahap kedua adalah akuisisi konten segmen. Tahap ketiga penghitungan *hash value*. Tahap keempat adalah eksperimen modifikasi berkas jpeg/exif. Tahap kelima adalah pemilihan elemen dan penyusunan *file fingerprint*. Hasil penelitian menunjukkan sebuah *jpeg/exif file fingerprint* tersusun atas tiga *hash value*. *SOI (Start of Image) segment hash value* digunakan untuk mendeteksi terjadinya modifikasi berkas dalam bentuk perubahan tipe berkas dan penambahan objek pada konten gambar. *Hash value* segmen APP1 digunakan untuk mendeteksi modifikasi pada metadata berkas. *Hash value* segmen SOF0 digunakan untuk mendeteksi gambar yang dimodifikasi dengan teknik *recoloring*, *resizing*, dan *cropping*.

ABSTRACT

The method of securing jpeg/exif image files currently has covered only the prevention aspect instead of the data integrity detection aspect. Digital Signature Algorithm is a cryptographic method used to verify the data integrity using hash value. SHA256 is a hash function that produces a 256-bit hash value functioning as a fingerprint file. This study aimed at compiling fingerprint files from jpeg/exif files using SHA256 and Brute Force string matching algorithm to verify the integrity of jpeg/exif files. The research was conducted in five steps. The first step was identifying the jpeg/exif file structure. The second step was the acquisition of the segment content. The third step was calculating the hash value. The fourth step was the jpeg/exif file modification experiment. The fifth step was the selection of elements and compilation of fingerprint files. The obtained results showed a jpeg/exif fingerprint file which was compiled in three hash values. The hash value of SOI segment was used to detect the occurrence of file modification in the form of file type changing and object addition on the image content. The hash value of APP1 segment was used to detect the metadata file modification. The hash value of SOF0 segment was used to detect the images modified by recoloring, resizing, and cropping techniques.

ISSN 2502-3357 (*Online*)
R. Fitriyanto dkk./Register 5 (2) 2019 128-139 ISSN 2503-0477 (*Print*)

1. Pendahuluan

Jpeg/exif merupakan format berkas untuk gambar yang dihasilkan dari pemakaian kamera digital seperti pada *smartphone*. Penggunaan berkas jpeg/exif semakin meningkat seiring dengan perkembangan perangkat keras pada *smartphone* dan popularitas media sosial. Peningkatan jumlah pemakaian berkas jpeg/exif harus diimbangi dengan keamanan informasi. Metode pengamanan informasi untuk berkas gambar saat ini hanya mencakup pada aspek pencegahan dari tiga aspek keamanan informasi yaitu pencegahan, deteksi, dan respons (Park, Ruighaver, Maynard, & Ahmad, 2011). Penggunaan *password* menjadi opsi pengamanan yang mudah digunakan, namun memiliki kelemahan karena dapat diatasi dengan aplikasi yang tersedia di internet.

Metode pengamanan lainnya ditemui dalam penelitian Wijayanto, Riadi, dan Prayudi (2016) yang memanfaatkan metadata pada berkas jpeg/exif untuk pencegahan pencurian hak cipta. Metadata dari berkas gambar diakuisisi dan dienkripsi untuk dipindahkan ke bagian *End of File* (EOF) (Wijayanto, Riadi, & Prayudi, 2016). Pengamanan berkas gambar menggunakan metadata juga ditemui dalam penelitian Gangwar dan Pathania (2018) yang membandingkan dua buah metadata untuk mengetahui orisinalitas dari kedua gambar. Pemanfaatan tanda air (*watermark*) untuk keamanan informasi ditemui dalam penelitian Sukarno (2013). *Watermark* dalam penelitian Sukarno (2013) tersebut didesain untuk mendeteksi tindakan pemalsuan kepemilikan dokumen digital. Penelitian lain yang dilakukan oleh Madhu, Holi, dan Murthy (2016) membahas beberapa metode pengamanan berkas gambar. Hasil penelitian menyebutkan metode-metode enkripsi memiliki performa yang baik untuk mencegah serangan terhadap berkas gambar.

Keempat penelitian yang telah disebutkan menunjukkan metode untuk mendeteksi keutuhan data pada berkas gambar belum banyak dikembangkan. Parameter keutuhan data menjadi indikator keotentikan informasi bagi penerima pesan untuk mengetahui keutuhan informasi yang diterimanya. Proses identifikasi keutuhan data dapat dilakukan dengan membandingkan objek-objek yang merepresentasikan berkas tersebut. Refialy, Sediyono, dan Setiawan (2015) dalam penelitiannya menggunakan *Digital Signature Algorithm* (DSA) untuk mendeteksi perubahan yang terjadi pada sebuah dokumen berekstensi pdf. DSA menggunakan tiga elemen utama untuk memenuhi kebutuhan tiga parameter keamanan informasi (Bansal, Sharma, & Mishra, 2017). *Hash value* untuk memverifikasi keutuhan data, *asymmetric key pair* untuk memverifikasi kepemilikan informasi, dan *digital certificate* untuk acuan status pengiriman informasi.

Hash value memiliki peran penting di dalam DSA, dikarenakan menjadi satu-satunya elemen yang merepresentasikan informasi yang dikirim. Konsep *message digest* sebagai inti dari sebuah pesan atau data melekat dalam *hash value* (Roussev, 2009). Penggunaan *hash value* sebagai indikator keutuhan data telah banyak ditemui seperti dalam verifikasi keutuhan berkas *installer* dari *developer* perangkat lunak dan juga dalam pencarian barang bukti digital dalam lingkup forensik digital (Roussev, 2011). Komparasi dua *hash value* menjadi metode yang efektif untuk memverifikasi keutuhan sebuah informasi. Metode komparasi tersebut menggunakan algoritma *string matching* yang membandingkan elemen dari pola yang dicari dengan data yang diuji.

Uraian beberapa penelitian terdahulu tersebut menjadi acuan pelaksanaan penelitian ini. Tujuan yang ingin dicapai adalah untuk menyusun *file fingerprint* dari sebuah berkas jpeg/exif. *File fingerprint* yang terbentuk digunakan untuk memverifikasi keutuhan data pada berkas jpeg/exif dalam DSA.

2. *State of the Art*

Tabel 1. Segmen dan *segment marker* berkas jpeg/exif

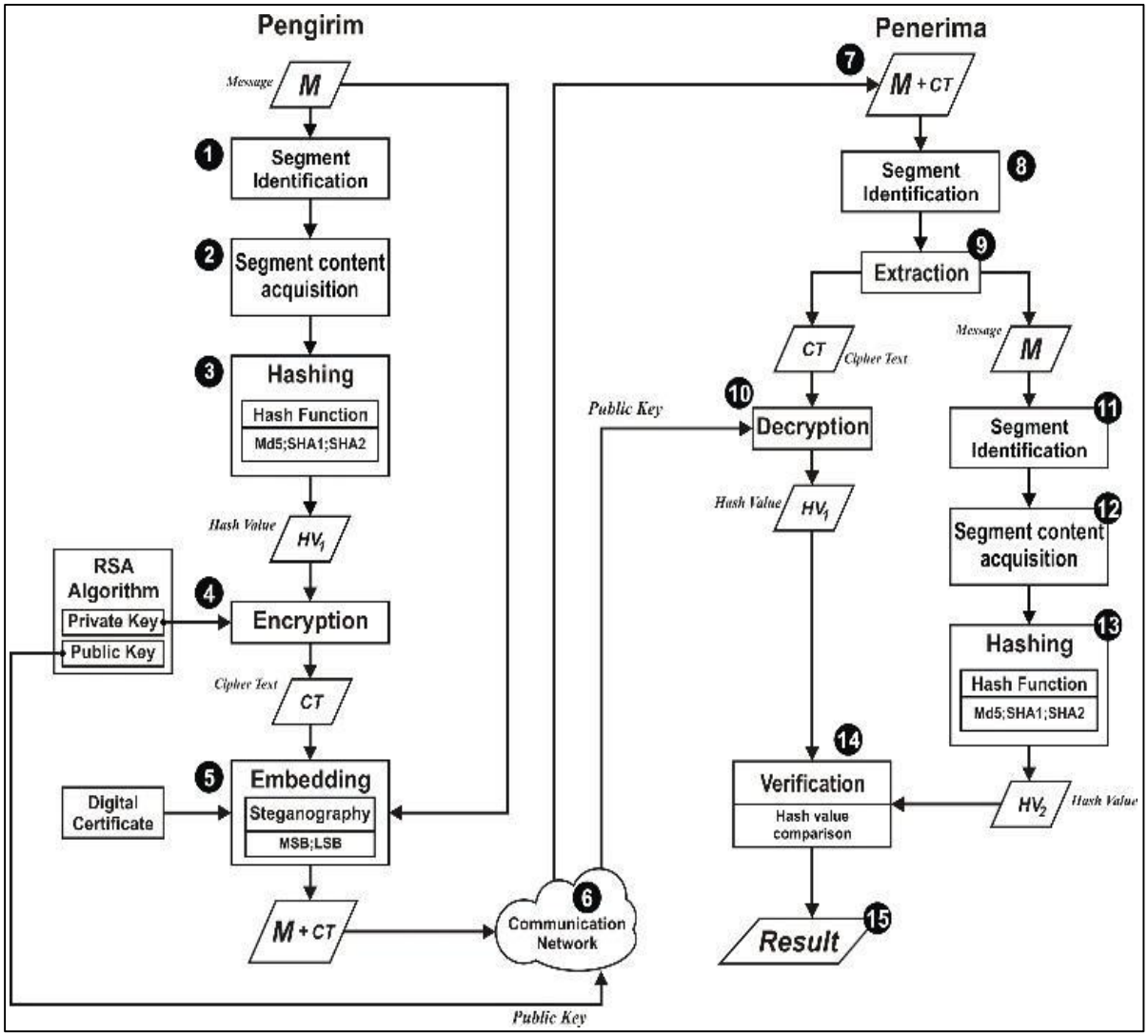
Segmen	*Segment Marker*
SOI (*Start of Image*)	ffd8
APP1 (*Application-1*)	ffe1
DQT (*Define Quantization Table*)	ffdb
SOF0 (*Start of Frame-0*)	ffc0
DHT (*Define Huffman Table*)	ffc4
SOS (*Start of Scan*)	ffda

Berkas jpeg/exif tersusun dari beberapa bagian yang disebut dengan segmen (Wijayanto, Prabowo, & Harsadi, 2018). Setiap segmen berfungsi sebagai kontainer data yang digunakan oleh komputer untuk

ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

menampilkan gambar. Komputer mengakses setiap segmen dengan mencari identitas unik dari segmen yang disebut dengan *segment marker* (Orozco, González, Villalba, & Hernández-Castro, 2015). Tabel 1 menunjukkan segmen yang penyusun berkas jpeg/exif dan *segment marker*. *Segment marker* pada Tabel 1 tersusun dalam format heksadesimal. *Segment marker* berfungsi sebagai pengenal (*signature*) untuk mengidentifikasi segmen, dengan membandingkan *segment marker* dengan bit-bit data pada berkas jpeg/exif.

Message digest merupakan instisari dari informasi yang digunakan untuk keamanan komunikasi pada pengamanan dokumen seperti berkas pdf dan pada berkas *installer*. *Message digest* disusun dalam bentuk *hash value* menggunakan *hash function* (Shaker & Jumaa, 2017). Karakteristik *message digest* sebagai representasi sebuah informasi digunakan di dalam DSA untuk memverifikasi keutuhan data yang sampai ke penerima. *Message digest* dalam DSA berbentuk *hash value* yang digunakan pada sisi pengirim dan penerima informasi. Gambar 1 menunjukkan proses-proses yang berlangsung di kedua sisi tersebut.



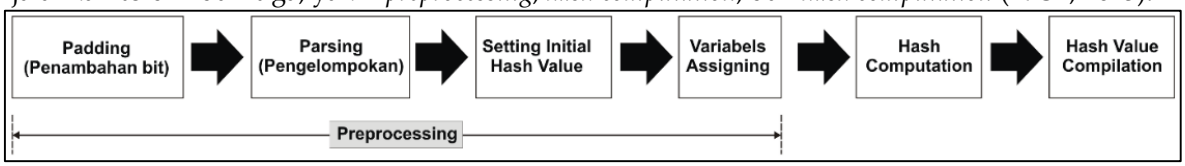
Gambar 1. Proses dalam *Digital Signature Algorithm* (DSA)

Tahap pertama sampai ketiga menunjukkan proses penyusunan *fingerprint data*. Tahap keempat adalah proses enkripsi *fingerprint data* menggunakan *asymmetric key pair*. Tahap kelima adalah proses penyisipan hasil enkripsi ke dalam berkas informasi yang dikirim (Bansal, Sharma, & Mishra, 2017). DSA memenuhi parameter keamanan dengan cara membandingkan dua *fingerprint data*. Hal ini membuat *fingerprint data* menjadi elemen utama di dalam DSA.

Secure Hash Algorithm (SHA) merupakan varian *hash function* yang dikembangkan oleh *National Institute of Standards and Technology* (NIST, 2015). SHA memiliki beberapa varian, yaitu SHA0, SHA1 dengan *hash function* MD4 dan MD5, SHA2 dengan *hash function* SHA256, SHA384, dan SHA512 (Jain, Jones, & Joshi, 2017). Varian SHA dan *hash function* dibedakan berdasarkan metode penyusunan dan

ISSN 2502-3357 (*Online*)
R. Fitriyanto dkk./Register 5 (2) 2019 128-139 ISSN 2503-0477 (*Print*)

ukuran *hash value* yang dihasilkan. Karakteristik utama yang membuat SHA digunakan dalam pengamanan informasi adalah *one way encryption* dan ukuran *hash value*. *One way encryption* membuat *hash value* yang dihasilkan tidak dapat diterjemahkan (*decrypt*) kembali ke bentuk asal. Ukuran *hash value* yang dihasilkan selalu tetap sesuai varian *hash function* untuk berbagai ukuran input (Sumagita & Riadi, 2018). Gambar 2 menunjukkan tahapan penyusunan *hash value* dengan SHA256. Proses yang dijalankan terdiri dari tiga, yakni *preprocessing*, *hash computation*, dan *hash computation* (NIST, 2015).

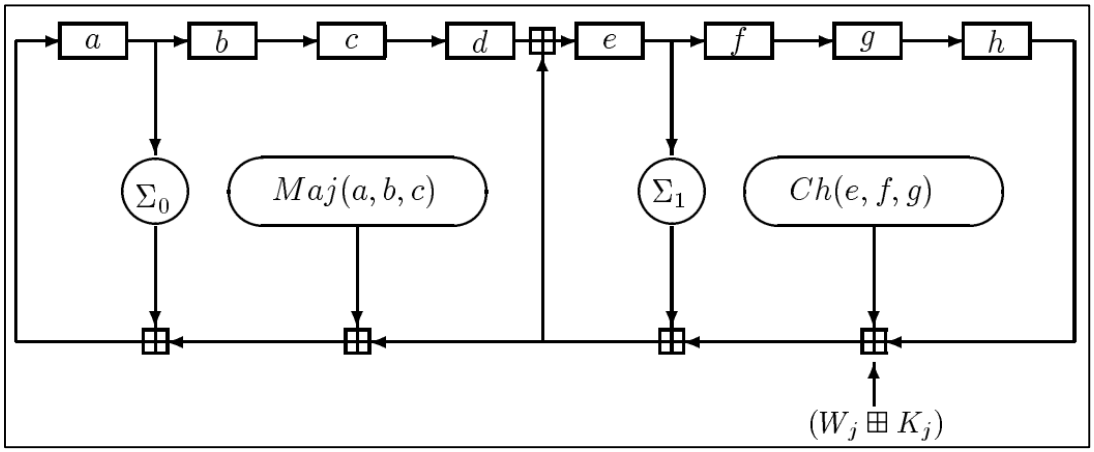


Gambar 2. Tahapan pada SHA256

Proses *padding* merupakan proses penambahan bit data pada informasi sehingga bit informasi memiliki panjang bit kelipatan 256. *Parsing* merupakan proses pengelompokan bit data hasil proses sebelumnya menjadi beberapa blok bit yang memiliki ukuran 512 bit. *Setting initial hash value* merupakan proses inisiasi delapan nilai *hash* dengan panjang 32 bit. Kedelapan nilai *hash* tersebut kemudian disimpan ke dalam delapan variabel seperti yang dirumuskan pada Tabel 2.

Tabel 2. Nilai *hash* untuk delapan register

Buffer	*Hash Value*	Variabel
$H_{(0)}{}^{0}$	6a09e667	a
$H_{(1)}{}^{0}$	bb67ae85	b
$H_{(2)}{}^{0}$	3c6ef372	c
$H_{(3)}{}^{0}$	a54ff53a	d
$H_{(4)}{}^{0}$	510e527f	e
$H_{(5)}{}^{0}$	9b05688c	f
$H_{(6)}{}^{0}$	1f83d9ab	g
$H_{(7)}{}^{0}$	5be0cd19	h



Gambar 3. Alur penghitungan nilai *hash* pada SHA256

Tahap *hash computation* adalah penghitungan nilai *hash* dari input yang digunakan. Gambar 3 berisikan proses penghitungan yang dilakukan secara berurutan sebanyak jumlah blok data yang terbentuk dari proses *parsing*. Proses penghitungan pada Gambar 3 dilakukan pada delapan register, dimulai dari register a menggunakan enam fungsi logika seperti yang ditunjukkan pada Persamaan 1 sampai Persamaan 6.

$$Ch(e,f,g) = (e \wedge f) \oplus (\neg e \wedge g) \quad (1)$$

$$Maj(a,b,c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c) \quad (2)$$

$$\sum_0(a) = S^2(a) \oplus S^{13}(a) \oplus S^{22}(a) \quad (3)$$

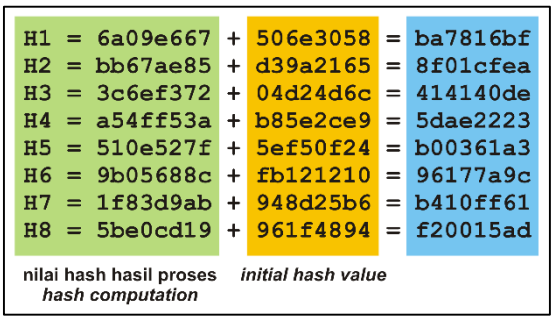
$$\sum_1(x) = S^6(a) \oplus S^{11}(a) \oplus S^{25}(a) \quad (4)$$

$$\sigma_0(a) = S^7(a) \oplus S^{18}(a) \oplus R^3(a) \quad (5)$$

$$\sigma_1(a) = S^{17}(a) \oplus S^{19}(a) \oplus R^{10}(a) \quad (6)$$

R. Fitriyanto dkk./Register 5 (2) 2019 128-139
ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

Hasil dari tahap *hash computation* adalah delapan buah nilai *hash* yang tersimpan di masing-masing variabel. Kedelapan nilai tersebut diproses pada tahap terakhir yang terdiri dari dua langkah. Langkah pertama adalah penjumlahan setiap nilai di dalam register dengan *initial hash value*. Gambar 4 menunjukkan penjumlahan pada langkah pertama untuk input "abc". Langkah kedua adalah menggabungkan kedelapan nilai dari langkah pertama menjadi satu *string* seperti yang ditunjukkan pada Gambar 5.



Gambar 4. Penjumlahan hasil *hash computation* dengan *initial hash value*

```
ba7816bf8f01cfea414140de5dae2223
b00361a396177a9cb410ff61f20015ad
```

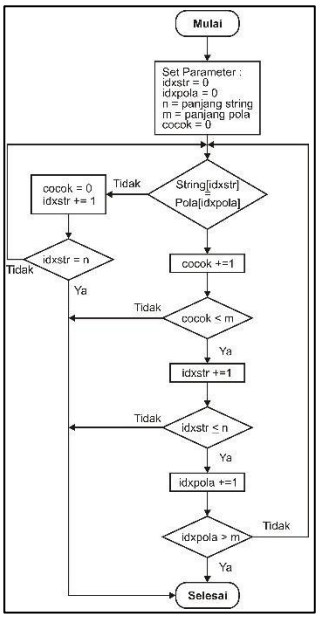
Gambar 5. Hasil akhir SHA256 untuk input "abc"

Metode penggunaan *hash function* dilakukan dengan input berupa seluruh bit data dari informasi. Ukuran berkas informasi akan mempengaruhi lamanya waktu penyusunan *message digest*. Penelitian ini bertujuan untuk mengembangkan metode penyusunan *message digest* sebagai *file fingerprint* dengan memilih bagian-bagian tertentu dari berkas.

3. Metode Penelitian

Penelitian dilakukan dalam lima tahap seperti diilustrasikan pada Gambar 6. Tahap pertama adalah identifikasi struktur berkas jpeg/exif. Tahap ini bertujuan untuk mengetahui bagian-bagian penyusun dari berkas jpeg/exif. *Output* dari tahap ini adalah informasi tentang data yang tersimpan dari setiap bagian dan data yang dapat digunakan untuk mengidentifikasi posisi setiap bagian di dalam berkas.

Gambar 6. Tahapan penelitian



Gambar 7. *Flowchart* algoritma *Brute Force string matching*

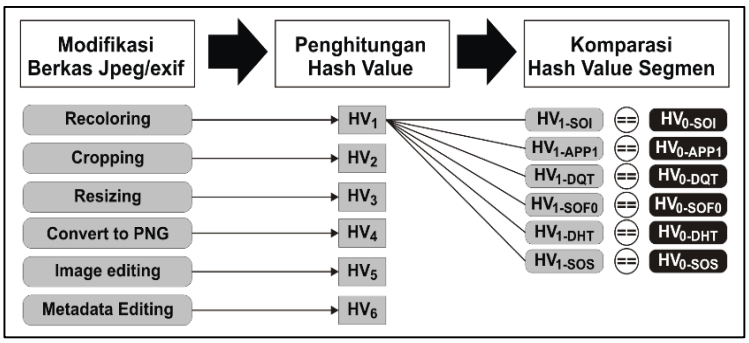
ISSN 2502-3357 (*Online*)
R. Fitriyanto dkk./Register 5 (2) 2019 128-139 ISSN 2503-0477 (*Print*)

Berkas jpeg/exif yang digunakan berasal dari pemakaian kamera pada *smartphone* Asus Z00UD dan Xiaomi Mi4i. Jumlah berkas jpeg/exif dari setiap tipe *smartphone* yang digunakan sebanyak 10 berkas. Mode pengambilan gambar pada *smartphone* Asus Z00UD dan Xiaomi Mi4i menggunakan *auto mode*. Lokasi pengambilan gambar dikombinasikan untuk pengambilan di ruangan tertutup (*indoor*) dan ruangan terbuka (*outdoor*). Proses identifikasi dimulai dengan melakukan konversi bit data berkas jpeg/exif ke dalam bentuk *string*. Pengidentifikasian lokasi dilakukan menggunakan algoritma *Brute Force string matching*. Gambar 7 menunjukkan *flowchart* dari algoritma tersebut.

Algoritma *Brute Force string matching* terdiri atas tahap *preprocessing* dan komparasi karakter. Tahap *preprocessing* berisikan proses inisiasi variabel yang terdiri dari lima variabel yang memiliki tipe data integer. Variabel $\boldsymbol{idxstr}$ menunjukkan posisi indeks dari karakter *string* yang diuji. Variabel $\boldsymbol{idxpola}$ adalah nilai indeks dari karakter pada pola yang dicari. Variabel $\boldsymbol{m}$ adalah panjang dari pola dan variabel $\boldsymbol{n}$ adalah panjang dari *string*. Variabel cocok menunjukkan banyaknya komparasi yang bernilai *TRUE*. Nilai maksimum dari variabel cocok sama dengan nilai variabel $\boldsymbol{n}$. Algoritma *Brute Force string matching* bekerja dengan membandingkan karakter paling kiri dari pola yang dicari dengan karakter dari hasil konversi data jpeg/exif (Jiji & Mahalakshmi, 2018). Pola yang digunakan dalam pencarian adalah data *segment marker* dari setiap segmen seperti yang telah ditunjukkan pada Tabel 1. Algoritma *Brute Force string matching* berhenti melakukan pencarian pola apabila terpenuhi dua kondisi. Kondisi pertama apabila pola yang dicari telah ditemukan. Hal ini ditandai apabila nilai variabel cocok sama dengan nilai variabel $\boldsymbol{m}$ (panjang pola). Kondisi kedua apabila karakter dari data jpeg/exif telah mencapai karakter terakhir. Hal ini ditandai dengan nilai variabel $\boldsymbol{idxstr}$ yang memiliki nilai sama dengan variabel $\boldsymbol{n}$ (panjang *string*). Jika pola ditemukan, nilai variabel $\boldsymbol{idxstr}$ terakhir digunakan sebagai nilai indeks lokasi segmen dari *segment marker* yang dicari.

Indeks lokasi segmen yang dicari ditunjukkan oleh nilai variabel $\boldsymbol{idxstring}$ dikurangi panjang pola ($\boldsymbol{idxstring - m}$). Indeks lokasi segmen yang ditemukan digunakan sebagai parameter untuk duplikasi konten pada tahap kedua. Tahap duplikasi konten bertujuan untuk mengumpulkan konten dari segmen yang akan digunakan sebagai input pada tahap ketiga. Proses duplikasi menggunakan dua parameter, yaitu indeks lokasi segmen dan panjang segmen. Berdasarkan *output* dari tahap pertama, lokasi awal sebuah segmen memiliki fungsi sebagai penanda dari awal segmen tersebut dan sebagai akhir dari segmen yang lainnya. Panjang segmen diukur dari selisih dua indeks lokasi segmen yang berdekatan. *Output* dari tahap kedua kemudian diproses menggunakan *hash function* SHA256 pada tahap ketiga.

Pemilihan komponen *file fingerprint* pada tahap keempat bertujuan untuk menentukan *hash value* yang dapat mengidentifikasikan terjadinya perubahan pada struktur jpeg/exif. Modifikasi berkas jpeg/exif dilakukan pada tahap ini dengan enam jenis perlakuan seperti yang diilustrasikan pada Gambar 8.



Gambar 8. Alur modifikasi berkas dan perbandingan *hash value*

Proses modifikasi berkas jpeg/exif dilakukan menggunakan aplikasi *Image Viewer*, *File Conversion*, dan *Hex Editor*. Modifikasi *recoloring*, *cropping*, *resizing*, dan *image editing* dilakukan menggunakan aplikasi ACDSee Pro.8. Modifikasi *recoloring* dilakukan dengan mengubah warna gambar menjadi *grayscale*. Modifikasi *cropping* dilakukan dengan mengambil bagian tertentu dari gambar dengan ukuran yang bervariasi. Modifikasi *resizing* dilakukan dengan mengubah persentase ukuran gambar menjadi lebih kecil. Dalam penelitian, *resizing* gambar dilakukan sebesar 50% lebih kecil dari ukuran awal. Modifikasi *convert to png* dilakukan dengan mengubah tipe berkas dari jpeg menjadi png menggunakan aplikasi *FormatFactory*. Modifikasi *image editing* dilakukan dengan menambahkan tulisan

ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

atau objek gambar pada gambar. Modifikasi metadata *editing* dilakukan dengan aplikasi *Hex Editor Neo* dengan mengubah data model *smartphone* pada bagian metadata berkas jpeg/exif.

Sebelum berkas gambar dimodifikasi, dilakukan penghitungan *hash value* dari berkas asli (HV0). *Hash value* dari berkas asli digunakan dalam komparasi *hash value* dari berkas yang telah dimodifikasi (HV1, HV2, HV3, HV4, HV5, HV5, HV6). Setiap bentuk modifikasi akan disusun *hash value* setiap segmennya. Komparasi *hash value* dilakukan untuk setiap segmen yang sama. *Hash value* yang mengalami perubahan disetiap bentuk modifikasi akan digunakan sebagai penyusun *file fingerprint* pada tahap selanjutnya. Tahap kelima, penyusunan *file fingerprint* adalah proses penggabungan beberapa *hash value* yang telah ditentukan dari tahap keempat.

4. Hasil dan Pembahasan

Hasil dari tahap pertama penelitian, identifikasi lokasi segmen berkas jpeg/exif dari *smartphone* Asus Z00UD ditunjukkan pada Tabel 3.

Tabel 3. Lokasi segmen berkas jpeg/exif dari *smartphone* Asus Z00UD

No.	Nama *File*	Ukuran *File*	SOI	APP1	DQT	SOF0	DHT	SOS
1	P_20180723_141211	2,117 KB	0	4	26400	26844	26726	27630
2	P_20180731_134049	1,915 KB	0	4	26368	61656	26694	27598
3	P_20180823_124724	2,488 KB	0	4	26378	26664	26704	27610
4	P_20180905_085850	2,457 KB	0	4	25350	25636	25676	26580
5	P_20190110_100735	1,977 KB	0	4	26318	26602	26642	27548
6	P_20190324_100013	1,601 KB	0	4	34212	25544	25584	26490
7	P_20190324_100040	1,564 KB	0	4	25378	25662	25704	26608
8	P_20190324_114023	2,315 KB	0	4	25278	25769	25809	26713
9	P_20190324_115302	1,495 KB	0	4	25348	25789	25792	26713
10	P_20190324_121005	2,415 KB	0	4	25405	25663	25564	26580

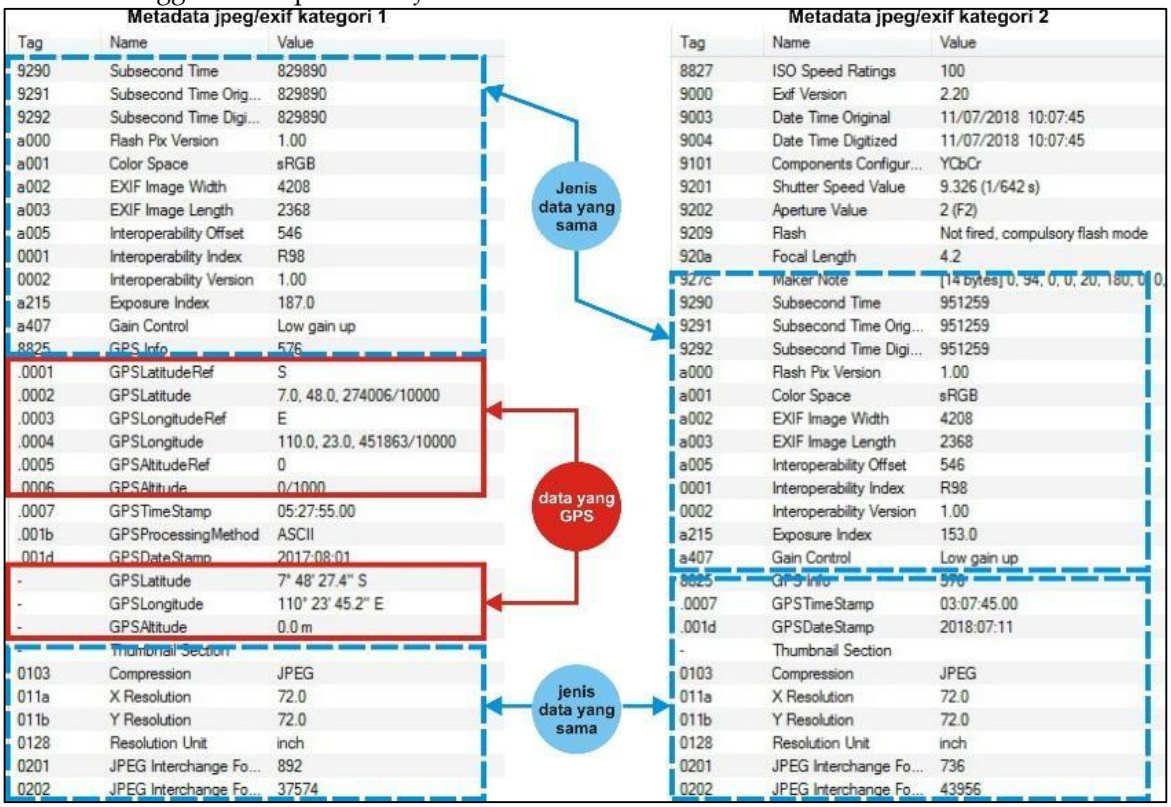
Indeks lokasi segmen SOI dan APP1 pada seluruh berkas jpeg/exif pada Tabel 3 memiliki nilai yang sama. Segmen SOI berlokasi di indeks ke-4. Hal ini menunjukkan segmen SOI berada di awal dari data berkas. Segmen APP1 berada di indeks ke-4. Informasi lain berdasarkan indeks lokasi APP1 adalah segmen SOI memiliki konten sepanjang 4 bit, dimulai dari bit ke-0 sampai ke-3. Keempat bit tersebut berisikan *segment marker* berupa segmen SOI yaitu ffd8. Indeks lokasi segmen DQT, SOF0, DHT dan SOS pada berkas jpeg/exif dari *smartphone* Asus Z00UD memiliki nilai yang berbeda-beda. Perbedaan lokasi segmen yang ditemukan muncul setelah segmen APP1. Hal ini menunjukkan data pada segmen APP1 dari semua file jpeg/exif memiliki kuantitas yang berbeda. Segmen APP1 pada berkas ke-1 memiliki ukuran yang lebih panjang (26400) daripada berkas ke-2 (26368). Segmen SOF0 pada berkas ke-1 berada di lokasi yang paling jauh dibandingkan segmen SOF0 pada sembilan berkas lainnya. Hal ini menunjukkan segmen DQT yang berada di depan segmen SOF0 memiliki data yang terpanjang dibandingkan segmen serupa pada berkas lainnya. Hasil berbeda ditunjukkan pada identifikasi lokasi segmen untuk berkas jpeg/exif dari *smartphone* Xiaomi Mi4i pada Tabel 4.

Tabel 4. Lokasi segmen berkas jpeg/exif dari *smartphone* Xiaomi Mi4i

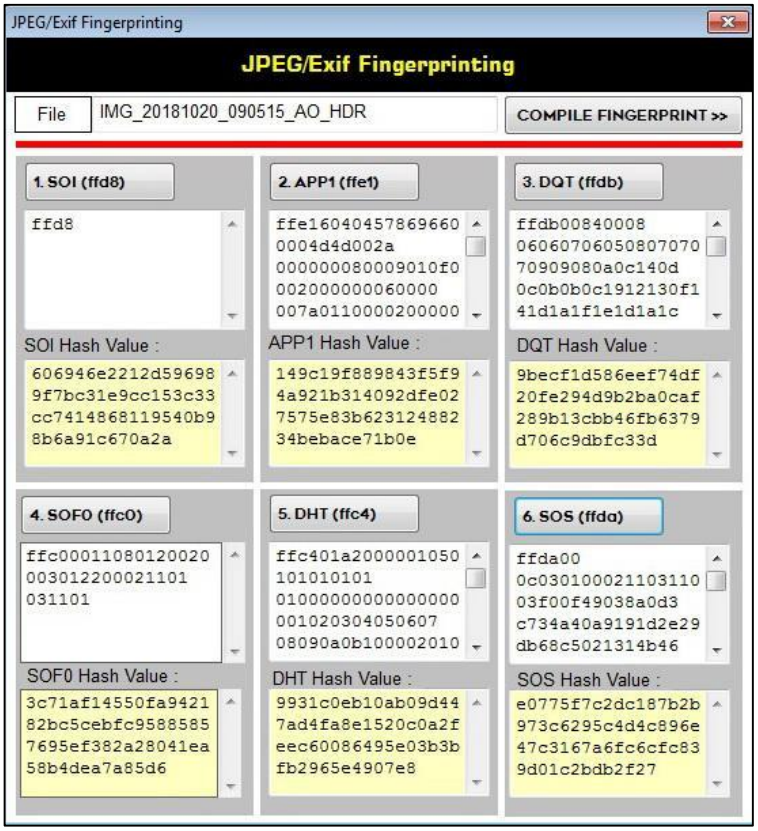
No.	Nama *File*	Ukuran *File*	SOI	APP1	DQT	SOF0	DHT	SOS
1	IMG_20170801_122807	1,895 KB	0	4	1924	2210	2250	3154
2	IMG_20180711_100746_BURST1	4,997 KB	0	4	1592	1878	1918	2824
3	IMG_20181013_044032_AO_HDR	2,997 KB	0	4	1592	1878	1918	2824
4	IMG_20181013_085450_AO_HDR	3,523 KB	0	4	1592	1878	1918	2824
5	IMG_20181014_083846_AO_HDR	2,716 KB	0	4	1592	1878	1918	2824
6	IMG_20181014_154345_BURST1	3,075 KB	0	4	1592	1878	1918	2824
7	IMG_20181020_090515_AO_HDR	3,417 KB	0	4	1924	2210	2250	3154
8	IMG_20190415_080646	3,123 KB	0	4	1924	2210	2250	3154
9	IMG_20190415_082151	3,009 KB	0	4	1924	2210	2250	3154
10	IMG_20190415_084850	3,323 KB	0	4	1924	2210	2250	3154

Hasil dari Tabel 4 menunjukkan dua kategori. Kategori pertama ditunjukkan pada file ke-1 yaitu 7, 8, 9 dan 10. Kategori kedua ditunjukkan pada file ke-2 yaitu 3, 4, 5 dan 6. Lokasi segmen untuk setiap *file* pada kategori yang sama memiliki nilai yang sama seperti lokasi segmen DQT hanya terdiri dari dua nilai, yakni 1924 dan 1592. Lokasi SOF0 hanya ada dua: 2210 dan 1918. Lokasi DHT ada di indeks

2250 dan 1918 serta segmen SOS berlokasi di 3154 dan 2824. Perbandingan lokasi segmen antarkategori menunjukkan perbedaan. Perbedaan lokasi segmen ini terdapat pada segmen DQT, SOF0, DHT, dan SOS. Hal ini menunjukkan titik awal perbedaan keempat segmen tersebut disebabkan oleh panjang segmen APP1 setiap *file*. Gambar 9 menunjukkan perbandingan metadata dari dua kategori hasil pencarian menggunakan aplikasi *ExifPro Photo Browser*.



Gambar 9. Perbandingan metadata dua mode pengambilan gambar



Gambar 10. Tampilan antarmuka penyusunan *hash value* jpeg/exif

ISSN 2502-3357 (*Online*)
R. Fitriyanto dkk./Register 5 (2) 2019 128-139 ISSN 2503-0477 (*Print*)

Perbedaan empat lokasi segmen setelah APP1 pada Tabel 4 disebabkan adanya tambahan data GPS yang ditunjukkan pada Gambar 9. Fitur lain yang membuat hasil pencarian pada Tabel 4 memiliki dua kategori adalah fitur pengambilan gambar yang dimiliki oleh aplikasi *default* (*auto mode*) pada *smartphone* Xiaomi Mi4i. Hal ini tidak ditemukan pada *smartphone* Asus Z00UD yang memberlakukan pengaturan yang sama untuk setiap pengambilan gambar.

Pada tahap kedua penelitian, duplikasi konten setiap berkas dilakukan berdasarkan dua parameter, yakni titik awal segmen dan panjang segmen. Titik awal ditunjukkan dengan nilai indeks segmen pada Tabel 3 dan Tabel 4. Panjang segmen dihitung dari selisih antara titik awal segmen kedua dengan segmen pertama seperti yang dirumuskan pada Persamaan 7.

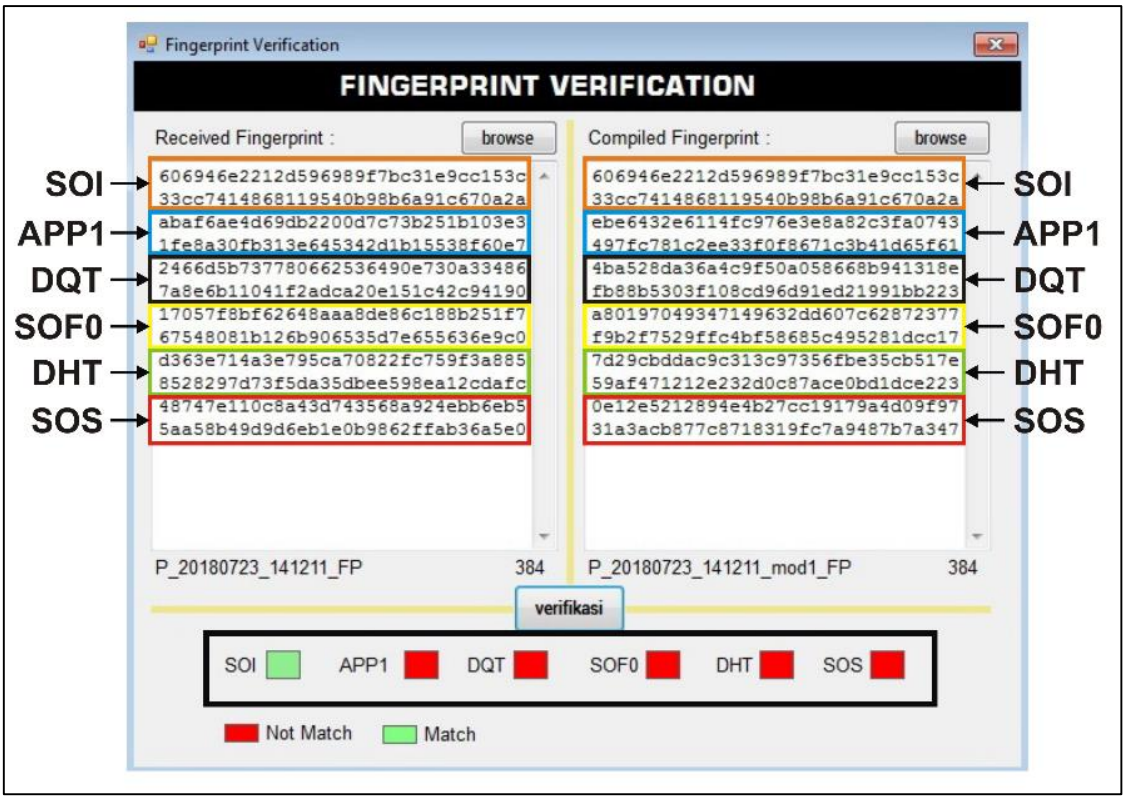
$$P_{segmen(n)} = Index(n) - Index(m) \quad (7)$$

Konten yang diperoleh digunakan sebagai input pada *hash function* SHA256. Gambar 10 menunjukkan contoh tampilan antarmuka aplikasi untuk menyusun *hash value* dari setiap segmen yang dikembangkan dengan aplikasi Visual Studio 2010 dengan bahasa pemrograman VB.net.

Proses duplikasi membutuhkan waktu yang linear dengan ukuran konten segmen. Segmen SOI memiliki konten paling sedikit sebesar 4-bit. Konten terbesar terdapat pada segmen SOS yang berisikan data gambar dari berkas jpeg/exif. Lamanya waktu duplikasi konten menjadi pertimbangan dalam pemilihan dan penghitungan *hash value* pada tahapan selanjutnya.

(a) (b)

Gambar 11. Contoh komparasi berkas gambar (a) Gambar orisinal; (b) Modifikasi *recoloring (grayscale)*



Gambar 12. Perbandingan *hash value* dari gambar asli dan hasil *recoloring*

ISSN 2502-3357 (*Online*)
R. Fitriyanto dkk./Register 5 (2) 2019 128-139 ISSN 2503-0477 (*Print*)

Hasil dari tahapan penelitian yang ketiga, penghitungan *hash value* ditunjukkan pada Gambar 11. Enam *hash value* pada dapat digunakan sebagai penyusun *file fingerprint*. Pemilihan bagian yang tepat dilakukan berdasarkan dua hal, kemampuan segmen untuk mendeteksi perubahan pada berkas dan lama penyusunan *hash value*. Pendeteksian perubahan *hash value* dilakukan dengan membandingkan dua *hash value* dari segmen yang sama dari dua berkas jpeg/exif yang berbeda. Gambar 11 menunjukkan salah satu contoh hasil modifikasi *recoloring* seperti yang telah dideskripsikan pada bagian metodologi. Perbandingan *hash value* antara berkas orisinal dengan berkas hasil modifikasi *recoloring* ditunjukkan pada aplikasi yang telah dikembangkan seperti pada Gambar 12.

Bagian sebelah kiri pada Gambar 12 menunjukkan *hash value* dari enam segmen pada berkas orisinal. Bagian sebelah kanan menunjukkan *hash value* dari enam segmen pada berkas hasil modifikasi *recoloring*. Proses perbandingan dilakukan menggunakan algoritma *Brute Force string matching* dengan parameter input dua *hash value* dari segmen yang sama. Hasil perbandingan per segmen terdiri dari dua kondisi, *match* apabila *hash value* dari kedua segmen bernilai sama dan *not match* apabila tidak sama. Perbandingan yang dicontohkan pada Gambar 12 menunjukkan hanya *hash value* berupa segmen SOI yang berstatus *match*. Empat segmen lain mengalami perubahan konten saat terjadi modifikasi *recoloring*. Segmen-segmen yang mengalami perubahan untuk setiap bentuk modifikasi gambar ditunjukkan pada Tabel 5.

Tabel 5. Perubahan *hash value* segmen untuk modifikasi berkas jpeg/exif

No	Bentuk modifikasi	SOI	APP1	DQT	SOF0	DHT	SOS
1	*Recoloring*	-	✓	✓	✓	✓	✓
2	*Metadata Modification*	-	✓	-	-	-	-
3	*Resizing*	-	✓	✓	✓	✓	✓
4	*Convert to PNG*	✓	✓	✓	✓	✓	✓
5	*Image Editing*	✓	✓	✓	✓	✓	✓
6	*Cropping*	-	✓	✓	✓	✓	✓

Gambar 13. Konten *file fingerprint* berkas jpeg/exif

Hasil perbandingan *hash value* pada Tabel 5 menunjukkan tiga kategori perubahan yang ditimbulkan akibat modifikasi berkas jpeg/exif. Kategori pertama adalah perubahan konten segmen SOI yang disebabkan oleh modifikasi dalam bentuk konversi ke-4 dan ke-5. Kategori kedua adalah perubahan konten segmen APP1 yang disebabkan oleh modifikasi metadata (modifikasi ke-2). Kategori ketiga adalah perubahan pada empat segmen DQT, SOF0, DHT, dan SOS. Hasil pada kategori ketiga menunjukkan keempat segmen tersebut selalu mengalami perubahan yang bersamaan untuk modifikasi dalam bentuk *recoloring, resizing, image editing*, dan *cropping*. Gambar 13 menunjukkan hasil penggabungan tiga *hash value* yang membentuk sebuah *file fingerprint* dari sebuah berkas jpeg/exif.

File fingerprint yang ditunjukkan pada Gambar 13 terdiri dari *hash value* berupa segmen SOI, APP1, dan SOF0. Pemilihan ketiga *hash value* segmen ini dikarenakan tiga faktor. Faktor pertama adalah ketiga *hash value* telah mampu mewakili segmen lain untuk mengidentifikasi terjadinya perubahan pada konten segmen dari enam bentuk modifikasi. Faktor kedua adalah ukuran konten segmen SOI, APP1, dan SOF0 lebih kecil dibandingkan tiga segmen lainnya. Faktor ketiga adalah lokasi segmen yang berada di bagian awal berkas seperti yang telah ditunjukkan di Tabel 3. Segmen yang berada di bagian awal struktur berkas membutuhkan proses pencarian yang lebih cepat daripada segmen yang berada di tengah atau di bagian akhir berkas, seperti DQT, DHT, dan SOS.

ISSN 2502-3357 (*Online*)
ISSN 2503-0477 (*Print*)

5. Kesimpulan

Hasil penelitian menunjukkan *file fingerprint* berkas jpeg/exif disusun dari *hash value* yang diperoleh dari tiga segmen, yaitu SOI, APP1, dan SOF0. Algoritma *Brute Force string matching* digunakan pada dua proses. Proses pertama adalah untuk identifikasi lokasi setiap segmen. Proses kedua adalah untuk membandingkan dua *hash value*. Berdasarkan susunan *hash value* penyusun *file fingerprint*, identifikasi lokasi segmen dan verifikasi *hash value* dengan algoritma *Brute Force string matching* dilakukan terhadap tiga segmen, yaitu SOI, APP1 dan DQT.

SHA256 digunakan untuk menghitung *hash value*. Proses penghitungan membutuhkan waktu yang linear dengan ukuran panjang ketiga segmen. Penelitian yang telah dilakukan baru menggunakan tipe berkas jpeg/exif sebagai objek informasi yang harus diamankan. Penelitian selanjutnya diharapkan dapat mengembangkan metode yang telah dilakukan untuk tipe-tipe berkas lain seperti audio dan video yang semakin banyak digunakan dalam komunikasi.

7. Referensi

Bansal, D., Sharma, M., & Mishra, A. (2017). Analysis of Digital Signature based Algorithm for Authentication and Privacy in Digital Data. *International Journal of Computer Applications, 161*(5), 43-45.

Gangwar, D. P., & Pathania, A. (2018). Authentication of Digital Image using Exif Metadata and Decoding Properties. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSR CSEIT), 3*(8), 335-341.

Jain, A. K., Jones, R., & Joshi, P. (2017). Survey of Cryptographic Hashing Algorithms for Message Signing. *IJCST, 8*(2), 18-22.

Jiji, N., & Mahalakshmi, T. (2018). An Efficient String Matching Algorithm for Detecting Patterns using Forward and Backward Searching Approach. *IPASJ International Journal of Computer Science (IIJCS), 6*(2), 16-26.

Madhu, B., Holi, G., & Murthy, K. S. (2016). An Overview of Image Security Techiques. *International Journal of Computer Applications, 154*(6), 37-46.

NIST, N. (2015). *FIPS 180-4 Secure Hash Standard (SHS).* Gaithersburg, Montgomery County, Maryland: National Institute of Standards and Technology. doi:http://dx.doi.org/10.6028/NIST.FIPS.180-4

Orozco, A. L., González, D. M., Villalba, L. J., & Hernández-Castro, J. (2015). Analysis of errors in exif metadata on mobile devices. *Multimedia Tools and Applications, 74*(13), 4735–4763.

Park, S., Ruighaver, A. B., Maynard, S. B., & Ahmad, A. (2011). Towards Understanding Deterrence: Information Security Managers' Perspective. *Proceedings of the International Conference on IT Convergence and Security* (pp. 21-37). Suwon: Springer.

Refialy, L., Sediyono, E., & Setiawan, A. (2015). Pengamanan Sertifikat Tanah Digital menggunakan Digital Signature SHA-512 dan RSA. *Jurnal Teknik Informatika dan Sistem Informasi (JuTISI), 1*(3), 229-234.

Roussev, V. (2009). Hashing and Data Fingerprinting in Digital Forensics. *IEEE Security & Privacy, 7*(2), 49-55.

Roussev, V. (2011). An evaluation of forensic similarity hashes. *Digital Investigation, 8*, S34-S41.

Shaker, S. H., & Jumaa, G. G. (2017). Digital Signature Based on Hash Functions. *International Journal Of Advancement In Engineering Technology, Management and Applied Science (IJAETMAS), 4*(1), 88-99.

Sukarno, A. S. (2013). Pengembangan Aplikasi Pengamanan Dokumen Digital Memanfaatkan Algoritma Advance Encryption Standard, RSA Digital Signature dan Invisible Watermarking. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) 2013* (pp. M-1 ~ M-8). Yogyakarta: Universitas Islam Indonesia.

Sumagita, M., & Riadi, I. (2018). Analysis of Secure Hash Algorithm (SHA) 512 for Encryption Process on Web Based Application. *International Journal of Cyber-Security and Digital Forensics (IJCSDF), 7*(4), 373-381.

Wijayanto, H., Prabowo, I. A., & Harsadi, P. (2018). Optimalisasi Penyusutan Exif Metadata dengan Teknik Substitusi Null Value pada Kasus Keamanan Citra Digital. *Jurnal Ilmiah SINUS, 16*(1), 1-10.

ISSN 2502-3357 (*Online*)
R. Fitriyanto dkk./Register 5 (2) 2019 128-139 ISSN 2503-0477 (*Print*)

Wijayanto, H., Riadi, I., & Prayudi, Y. (2016). Encryption EXIF Metadata for Protection Photographic Image of Copyright Piracy. *International Journal of Research in Computer and Communication Technology (IJRCCT), 5*(5), 237-243.