Implementasi dan Perbandingan Kecepatan Dari Fungsi Beberapa Fungsi Hash Populer

Dibi Khairurrazi Budiarsyah, 13509013
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13509013@std.stei.itb.ac.id

Abstraksi — Kriptografi merupakan ilmu dan seni mengubah suatu pesan agar tidak dapat dikenali oleh orang lain selain pembuat dan penerima pesan. Ilmu kriptografi telah digunakan dari zaman dahulu dan membantu suatu pihak untuk menjaga agar pesannya tidak dapat dibaca orang lain. Salah satu contohnya adalah enigma cipher yang digunakan oleh Nazi pada perang dunia kedua. Terdapat beberapa jenis kriptografi, salah satunya adalah hash kriptografis, yakni fungsi hash yang memiliki keamanan tambahan dan biasanya digunakan untuk mengautentikasi dan menguji integritas data. Beberapa contoh dari fungsi hash kriptografis ini adalah MD5, SHA1, SHA512, Whirlpool, Tiger, dll. Pada makalah ini, pengujian akan difokuskan pada hash kriptografis. Fungsi hash yang akan diimplementasi adalah MD5, SHA512, haval, RIPEMD160, tiger, dan Whirlpool. Tujuan dari makalah ini adalah untuk mengetahui fungsi hash mana yang paling cepat untuk memproses sebuah pesan berukuran x byte dan file-file lain seperti gambar, video, ataupun file executable. Implementasi dilakukan dengan menggunakan bahasa c# dan pengujian dilakukan pada laptop Asus N43SL.

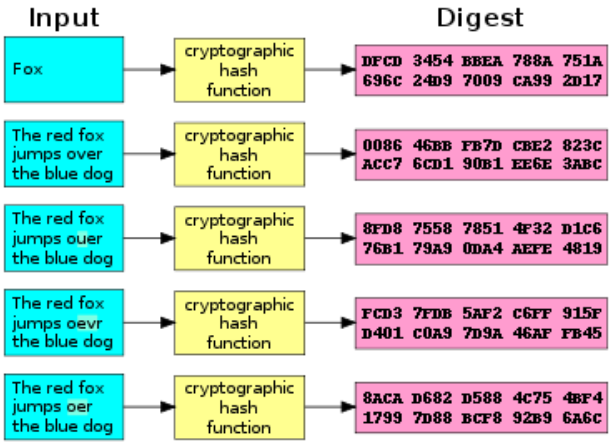
Kata kunci — Hash, MD5, SHA512, Whirlpool.

I. PENDAHULUAN

Seiring dengan berkembangnya teknologi di dunia ini, arus penyebaran informasi menjadi lebih cepat dan mudah. Salah satu teknologi yang mendukung penyebaran informasi ini adalah internet. Dengan menggunakan internet, setiap orang dapat saling berbagi informasi dan data. Namun terkadang, data yang terdapat di internet tidak asli atau sudah mengalami perubahan. Selain itu, terdapat permasalahan lain yakni permasalahan otentikasi.

Untuk mengatasi permasalahan tersebut, dibutuhkan sebuah mekanisme tertentu untuk memeriksa apakah sebuah file yang beredar di internet merupakan file asli dan belum mengalami perubahan. Fungsi hash kriptografis dapat digunakan untuk menjadi solusi dari permasalahan ini.

Fungsi hash kriptografis merupakan salah satu jenis fungsi hash yang diberikan fungsi keamanan tambahan, dimana sebuah algoritma dijalankan pada sebuah file dan mengembalikan sebuah nilai hash dengan panjang string yang tetap. Dengan demikian, kita dapat membandingkan nilai hash yang didapat dari dua file yang kita ingin uji. Apabila nilai hash dari kedua file berbeda, maka dapat dipastikan file tersebut sudah mengalami modifikasi. Contohnya dapat dilihat pada gambar dibawah ini.



Gambar 1. Hasil dari fungsi hash terhadap suatu pesan dan modifikasi pada pesan tersebut
Sumber : http://www.wikipedia.org

Pada gambar diatas, dapat dilihat apabila terdapat perubahan sedikit saja pada pesan atau *input*, maka nilai *message digest* yang dihasilkan akan jauh berbeda. Hal ini disebut juga dengan *avalanche effect*.

Beberapa contoh pengaplikasian dari fungsi hash kriptografis ini adalah tanda tangan digital dan pengecekan file. Beberapa situs yang memberikan layanan pengunduhan file menyediakan nilai hash dari file tersebut lengkap dengan algoritma yang digunakan (misalnya md5, dll). Hal ini bertujuan untuk memberikan user suatu sarana untuk menguji apakah file yang ia unduh dari suatu website adalah benar file yang diinginkan dan bukan file

hasil modifikasi atau file lain yang tidak diinginkan.

Sudah banyak fungsi hash kriptografis yang beredar di internet, diantaranya yang populer adalah HAVAL, GOST, SHA, MD, Whirlpool, dll. Dari sekian banyak algoritma hash kriptografis yang ada, algoritma manakah yang menghasilkan nilai hash secara cepat untuk sebuah pesan atau file yang sama? Makalah ini dibuat untuk menjawab pertanyaan tersebut.

II. HASH

A. Definisi Fungsi Hash dan Fungsi Hash Kriptografis

Fungsi hash merupakan algoritma yang digunakan untuk memetakan kumpulan data yang memiliki variabel dengan panjang berbeda menjadi sebuah kumpulan data yang panjang variabelnya tetap. Nilai yang dikembalikan dari sebuah proses hash disebut dengan *hash value*, *hash codes*, *checksum*, atau cukup hash. Pada makalah ini, fungsi hash yang digunakan adalah fungsi hash kriptografis.

Fungsi hash kriptografis merupakan salah satu jenis fungsi hash yang diberikan fungsi keamanan tambahan, dimana sebuah algoritma dijalankan pada sebuah file dan mengembalikan sebuah nilai hash dengan panjang string yang tetap. Perubahan pada data akan menghasilkan fungsi hash yang baru. Dengan demikian, untuk menguji keaslian data kita dapat membandingkan nilai hash yang didapat dari dua buah file.

Fungsi hash kriptografis yang ideal memiliki empat sifat utama. yakni :

- Mudah untuk menghitung nilai hash dari pesan apapun
- Nilai hash tidak dapat digunakan untuk merekontruksi pesan
- Nilai hash berubah apabila terdapat perubahan sekecil apapun pada pesan
- Tidak ada pesan berbeda yang memiliki nilai hash yang sama

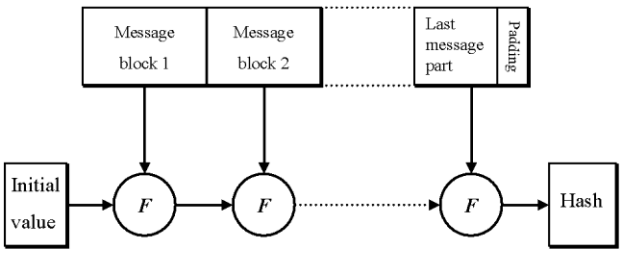
Fungsi hash kriptografis harus dapat menangkal serangan. minimal ada tiga yang dipenuhi yakni :

- *Pre-image resistance*
  Apabila terdapat suatu nilai hash h, maka akan sulit untuk menemukan sebuah pesan dimana pesan tersebut memenuhi fungsi h=hash(m). dengan kata lain, mengetahui nilai dari hash tidak akan memberikan pesan yang membentuk nilai hash tersebut. Fungsi hash haruslah sebuah fungsi yang satu arah, fungsi yang tidak memiliki sifat ini akan mudah diserang dengan *pre-image attack*.
- *Second pre-image resistance*
  Apabila terdapat sebuah pesan m1, akan sulit untuk menemukan sebuah pesan m2 dimana m1 ≠ m2 dan hash(m1) = hash(m2). Sifat ini terkadang dinamakan *weak collision resistance*, fungsi yang tidak memiliki sifat ini akan rentan terhadap *second pre-image attack*.
- *Collision resistance*
  Akan sulit untuk menemukan dua pesan yang berbeda memiliki nilai hash yang sama. Sifat ini disebut juga *strong collision resistance*.

Fungsi hash ini telah diterapkan pada banyak aplikasi yang membutuhkan keamanan informasi, diantaranya adalah :

- Pengecekan integritas pesan atau file
  Apabila terdapat perubahan pada pesan atau file, maka nilai hash yang didapat dengan menggunakan algoritma yang sama akan berbeda. Oleh karena itu, dengan membandingkan kedua nilai hash yang didapat kita akan mengetahui apakah file tersebut sudah dimodifikasi atau tidak.
- Pengecekan *password*
  Pada saat *login*, *password* akan diubah menjadi sebuah nilai hash dan kemudian dibandingkan dengan nilai hash yang ada di *database*. Oleh karena itu apabila *password* hilang atau terlupa, kita tidak dapat mengembalikan *password* yang lama tetapi mendapakan *password* yang baru.
- *File identifier*
  Message digest dapat juga digunakan untuk mengidentifikasi sebuah file. Beberapa *source code management system* seperti *Git*, *Mercurial*, dan *Monotone* menggunakan *sha1sum* untuk mengidentifikasi file.
- *Pseudorandom generation* dan *key derivation*
  Fungsi hash dapat pula digunakan untuk menghasilkan nilai *pseudorandom bit*. Selain itu, fungsi hash dapat digunakan untuk menurunkan sebuah nilai kunci atau *password* dari sebuah kunci atau *password* tunggal.

Damgard dan Merkle memberikan pengaruh yang besar terhadap desain dari fungsi hash kriptografis dengan mengajukan *compression function*. Fungsi kompresi ini mengambil sebuah input yang panjangnya tetap kemudian mengompresinya menjadi sebuah output yang lebih pendek. Dengan demikian sebuah fungsi hash dapat didefinisikan dengan menggunakan *compression function* ini secara berulang-ulang seperti pada gambar dibawah.



Gambar 2. Struktur pengulangan Damgard dan Merkle untuk fungsi hash
Sumber : http://www.rsa.com/rsalabs/node.asp?id=2176

B. Jenis-jenis Fungsi Hash Kriptografis

Terdapat banyak fungsi hash kriptografis yang ada. Beberapa yang populer diantaranya adalah :

1) SHA

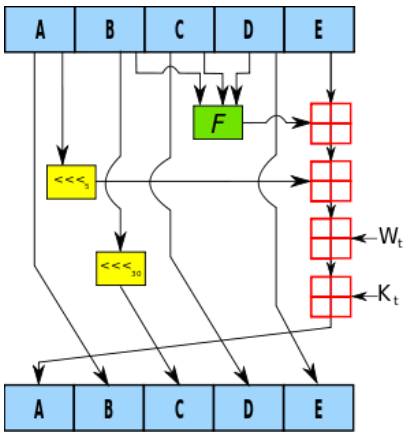
SHA (*Secure Hash Algoritm*) merupakan salah satu jenis fungsi hash kriptografis yang dipublikasikan oleh *National Institute of Standards and Technology* (NIST). Terdapat beberapa jenis dari SHA antara lain SHA-0, SHA-1, SHA-2, dan SHA-3.

a) SHA-0

SHA-0 merupakan SHA versi awal yang diterbitkan pada tahun 1993. Namun, tidak lama kemudian SHA ditarik dari peredaran dikarenakan adanya cacat produk yang signifikan dan kemudian digantikan dengan SHA-1.

b) SHA-1

SHA-1 adalah pengembangan dari SHA-0 dimana SHA-1 memperbaiki kelemahan yang ada di SHA-0. SHA-1 merupakan fungsi hash yang paling populer dibandingkan dengan fungsi hash SHA lainnya. SHA-1 memproduksi 160bit *digest* berdasarkan prinsip yang sama dengan algoritma MD4 dan MD5 namun dengan design yang berbeda.



Gambar 3. Struktur hashing pada SHA-1
Sumber : http://www.wikipedia.org

c) SHA-2

Merupakan keluarga dari dua fungsi hash yang mirip namun memiliki ukuran blok yang berbeda yakni SHA-256 dan SHA-512.

i) SHA-256

Merupakan varian dari SHA-2 yang menggunakan 32-bit *word*.

ii) SHA-384

Merupakan varian dari SHA-2 yang merupakan *truncated version* dari SHA-512.

iii) SHA-512

Merupakan varian dari SHA-2 yang menggunakan 64-bit *word*.

d) SHA-3

Merupakan salah satu fungsi hash kriptografis yang dulunya bernama Keccak. SHA-3 mensupport panjang hash sama seperti SHA-2 namun memiliki struktur internal yang jauh berbeda dibandingkan fungsi hash SHA lainnya.

2) MD

MD merupakan salah satu fungsi hash kriptografis yang populer di dunia ini. Fungsi hash ini dikembangkan oleh Ronald Rivest. Terdapat beberapa versi dari MD yakni MD2, MD4, MD5, dan MD6.

a) MD2

Salah satu varian MD yang optimal digunakan pada komputer 8 bit. *Message digest* dari MD2 direpresentasikan dengan angka heksadesimal berukuran 32bit.

b) MD5

Salah satu algoritma hash kriptografis yang populer digunakan. MD5 menghasilkan message digest berukuran 128 bit. Peneliti telah membuktikan bahwa MD5 tidak terlindungi dari *collision attack*, dengan demikian MD5 tidak cocok untuk digunakan pada *SSL certificate* dan *digital signature*. MD5 biasanya digunakan untuk pengecekan file yang didownload di internet. Pemilik situs biasanya mencantumkan nilai hash MD5 yang dihasilkan dari file yang ia letakkan pada situsnya, dengan demikian seseorang yang mengunduh file dari situs tersebut dapat mengecek keaslian dari file tersebut dengan cara membandingkan nilai hash MD5 dari file tersebut.

3) HAVAL

Haval merupakan sebuah algoritma hash satu arah yang menyediakan lima belas tingkat

keamanan. Haval didesain pada tahun 1992 oleh Yuliang Zheng, Josef Pieprzyk, dan Jennifer Seberry. Haval kemudian direvisi pada tahun 1997. Haval dapat memproduksi hash dengan panjang 128 bit, 160 bit, 192 bit, 224 bit, dan 256 bit. Selain itu haval juga meminta penggunanya untuk memasukkan jumlah ronde (3, 4, atau 5) yang digunakan untuk menghasilkan nilai hash. Fungsi hash pada haval menimbulkan *avalanche effect* dimana apabila terdapat perubahan sekecil apapun akan mengakibatkan berubahnya nilai hash menjadi nilai yang sangat berbeda dari sebelumnya. Hasil riset dari Xiaoyun Wang menunjukkan bahwa haval memiliki beberapa kelemahan. Haval yang digunakan pada pengujian kali ini menggunakan 5 ronde dan keluaran 256 bit.

4) RIPEMD

Ripemd (*RACE Integrity Primitives Evaluation Message Digest*) merupakan sebuah algoritma hash 160 bit yang dikembangkan di Leuven, Belgia oleh Hans Dobbertin, Antoon Bosselaersm dan Bart Preneel. Ripemd pertama kali dipublikasikan pada tahun 1996. Ripemd juga tersedia dalam versi 128, 256, dan 320 bit.

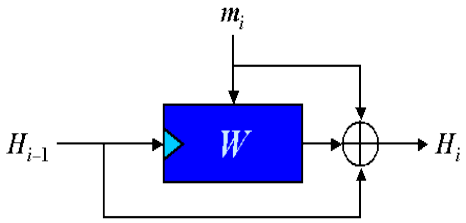
5) GOST

GOST merupakan salah satu algoritma hash kriptografis 256 bit. Fungsi hash ini didasari oleh GOST *block cipher*. Terdapat tiga proses utama yakni *Key Generation*, *Encryption Transformation*, dan *Mixing Transformation*. Kriptanalis menemukan kelemahan pada GOST. Peneliti berhasil melakukan *collision attack*, *preimage attack*, dll.

6) Whirlpool

Whirlpool merupakan sebuah algoritma hash kriptografis yang didesain oleh Vincent Rijmen dan Paulo S. L. M. Barreto yang memproses pesan dengan panjang kurang dari 2^{256} bit. *Whirlpool* memiliki tiga versi yakni *whirlpool-0*, *whirlpool-T*, dan *whirlpool*.

Whirlpool menggunakan struktur pengulangan Damgard dan Merkle serta skema hashing Miyaguchi-preneel 512bit berbentuk *block cipher* (W).



Gambar 4. Skema *hashing* dari Miyaguchi-Preneel
Sumber :
http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html

block cipher W terdiri dari 8 x 8 *state matrix* S dalam byte, dengan total 512 bit. Proses enkripsi dilakukan dengan mengupdate state menggunakan empat *round function* dalam sepuluh ronde. Keempat round function ini adalah *SubBytes* (SB), *ShiftColumns* (SC), *MixRows* (MR) dan *AddRoundKey* (RK). Dalam tiap ronde, *state* S diupdate dimana nilai S didapat dari $S = AK \cdot MR \cdot SC \cdot SB(S)$.

7) Tiger

Tiger merupakan salah satu fungsi hash kriptografis yang populer. *Tiger* didesain oleh Ross Anderson dan Eli Biham pada tahun 1995. Algoritma ini efisien digunakan pada platform 64 bit. Ukuran hash dari *tiger* adalah 192 bit. Tiger memiliki dua varian yakni tiger 1 dan tiger 2. Tiger yang digunakan untuk pengujian pada makalah ini adalah tiger 1.

Belum ada laporan serangan yang berhasil terhadap tiger dengan 24 ronde namun John Kelsey dan Stefan Lucks berhasil menemukan kelemahan pada tiger dengan 16 ronde. Tiger dengan 16 *round* lemah terhadap *collision attack*.

III. IMPLEMENTASI DAN PENGUJIAN

A. Algoritma Hash Kriptografis yang Digunakan

Pada pengujian kecepatan ini, algoritma yang digunakan adalah SHA1, SHA512, *Whirlpool*, *Tiger* 1, Haval dengan 5 ronde dan keluaran 256 bit, dan MD5. Implementasi dilakukan dalam bahasa c# dengan menggunakan *Microsoft Visual Studio*. Pengujian dilakukan pada laptop Asus N43SL.

B. Pengujian Dengan Menggunakan Pesan

Pengujian pertama dilakukan dengan menggunakan pesan yang dituliskan pada *textbox*. Pengujian dilakukan tiga kali dengan menggunakan panjang pesan yang berbeda.

- Pesan pertama (P1) menggunakan kata "kriptografi"
- Pesan kedua (P2) menggunakan kalimat "menggunakan beberapa fungsi hash kriptografis seperti sha, tiger, haval, dll. untuk diuji kecepatannya"
- Pesan ketiga (P3) menggunakan sebuah artikel berita yang ada di situs detik.com.

Hasil pengujian kecepatan terhadap pesan dan hasil hash dapat dilihat pada dua tabel dibawah ini

Tabel 1. Hasil *message digest* dari sebuah pesan

P	Algoritma	*Message Digest*
P1	SHA512	FDD45BE4B47A2105A04DE71D1080FD03186B04E518975CE2188D2E31CA1207E4F8FE5D5A3B9B47CE9039E1B8D80B9A8DB111534FBEF2DC35319E0F2655DA0C49
	MD5	0B8B3F943DEDE1501DEC2D478984F08D
	Whirlpool	6EAE8BBDC8054AB683ED95B4D405750BED15BE44CC12D77135141A42D7758337B8E64618F73BF50E5F2F6DC5A716517C4175C943F19311AB4B65D00CFD09682F
	Haval	595034222D89E6F3D347BE2878B2CFCFF7B08A7D65DEB24A7B2B170FE3C45FCF
	RIPEMD160	134BB544D7DC0DF139D60D7A85238988709AC2E2
	Tiger	E6427A8A59FE6FC6E8466919015EC22017C6F9B1B2F5C710
	SHA-1	FB79FF9F4C0FDF257536101E29D79443A345033C
P2	SHA512	E539FAF0FE30AD5711BDEF43347129CF7240D23E216AFEEDF2DCC5FB538206E9C260DB90B24395A4CBDFDB33FC6622B554B84994CA0D8A9A6C46C15D908C1341
	MD5	7B5AD6503D7CFC12FB233A9C67C5A5D6
	Whirlpool	EB0FDFE233DEC8576E03AB22CD5E4F868ED706666307861613252E346CA76B324317454D20A83B49201108557D758AAE64A6A7DB8340072FC5B5A548617CBB2F
	Haval	A097CF515BD011BDCAD836E1E1BF6A374042C0DE159E23EC30E7CA066FA62288
	RIPEMD160	B6D5819E2C37D1BC77B96149FC012E1A65293538
	Tiger	A6B3A780EF7A1FCCF76D503063EF0B881D2A9D1FEBC1BAA2
	SHA-1	BCA5CDA6AB738CE6B30B22C7BFF91D33FB4523C0
P3	SHA512	EC0456E5BE5C8E23BFA2498442A0AA5564ABD1A149BAB4220FD15FBD08254B9922FF17A284A2600C2D1F04DA6733B2DA918C3EFAD07FE345C715693C7997BAE4
	MD5	24F1C929A35E96775EDFB16FE69F0386
	Whirlpool	23C60B6F4DAD986BEF42CD7AF8100CC18EED18084070413E89EF185C3529238C2EF108DECF128E59622B57E02C5B7BBD6BD6650A6717405D0F1318B9F4703530
	Haval	E707EBF26EAF7EC1138714D8A6963549D0A9CA941AC19F33A60D1992038CD3BF
	RIPEMD160	1D84D7239C10CF1EA77A59583266D5547134E46B
	Tiger	09504FDE0AAE2E3BAA93EBCD7EA2195585BA0D6578D093F7
	SHA-1	84D4C594D1A592298D60CE9707B0D7DD9ECBF9D6

Tabel 2. kecepatan hash tiap algoritma dari sebuah pesan dalam satuan milidetik(ms)

	Sha512	H	W	T	Ripemd	MD5	Sha1
P1	1	24	14	15	5	2	2
P2	3	30	8	29	8	3	3
P3	2	32	12	35	4	4	1

Dari tabel diatas dapat dilihat bahwa ukuran pesan tidak menjadi masalah untuk SHA512, whirlpool (W) lebih cocok digunakan untuk pesan berukuran sedang, Haval (H) dan tiger (T) menghasilkan nilai hash jauh lebih lambat dibandingkan fungsi hash yang lain. MD5 cukup cepat dalam melakukan proses hashingnya. RIPEMD160 dan SHA1 lebih cocok digunakan untuk pesan berukuran besar.

C. Pengujian Dengan Menggunakan File Gambar

Pengujian selanjutnya dilakukan dengan menggunakan beberapa gambar. Seluruh gambar yang digunakan dalam pengujian memiliki ekstensi JPEG. Gambar 1 (G1) memiliki ukuran 32 KB, Gambar 2 (G2) memiliki ukuran 970KB, dan gambar 3 (G3) memiliki ukuran 1,6MB.

Tabel 3. Hasil *message digest* dari sebuah gambar

G	Algoritma	*Message Digest*
G1	SHA512	49A43189B27051F4DDC6D9611A9A9404C4EE703878BB31F6CEBBD7D30874B6EA7FF86D5D0CB438D556674E4915FC42F8D6C88BFB5798547C44657B05B06FC745
	MD5	B71DE5CAB267347C41456D0093D8205F
	Whirlpool	7E6EF04850230A3052473F0F7197D7A22B2A6D77EA064BF9C51815AB4BA6E51A4373B352265711E4199B6644811E9A390B75478E7DDC96762586DFE8C9FE34A2

	Haval	A11CFCBC2450E7756E73CC9AE8F408E04B00FE0BFF00E4BC884317C4EE6BA5C8
	RIPEMD160	B105A1E2219E00CC802842F2DE79E4ABDCF2E63B
	Tiger	076AF0DEB77A9027B9CAAADA65BB4561744FCCE007986800
	SHA-1	DBCDBC26C91105BE309B90D484690CC6C316B60D
G2	SHA512	B591F058F18B48CB4B4CA5965CC8FE76D905C1DF3B30D2454B43BA317BB6323D574CACE2992E52FA67457E04112FBB073E4F68D76EB9E9334D0D72E22E3E4BD5
	MD5	029324F334D5D4B4B18EC55CF4A32ABF
	Whirlpool	D23273AA56D5B5012710477CC41F3B96B0E60D611B59DF3AF43CD286C17DC7A411F4BE23C141F211340F4EFFCBD1B7EE0F5CF18B55381645B3889CAD4DFFCAA3
	Haval	D6B9674A543BF95AABC779FD38B842977BA4242565FA73A9849C75D23DD80686
	RIPEMD160	6CF7C71BBCF6D4D1E329AF7E589BCA8786883244
	Tiger	5908731CD57CF7641F7A9F1030A9AE616716F478FE0BE46C
	SHA-1	DFC7A8A918838E688E446E09CE1413A461666FA0
G3	SHA512	7DDBD44AB210AAD524F407706E09618ECD0F131C501224DE89A78FFFD90B773FBB186095D818003774A9733AFE0920F82AA3B3C062AAA450A1CE67E8F6562CC4F
	MD5	E2CA66A0BB02E49F9E1DFBFA33BC4170
	Whirlpool	80A4B36F787D530652BD2C96E6DA97AE8F2EF5A11488C3D9E7755911423807E22D953AE4C8292A0D6FD3FDCB5EBC0A7AC065176FDE2E5BD2F59C54AFF972FFF3
	Haval	9159D4601182CA7A3AF674D5AF03344C301B5EEF9AC3230366C70513FCBD3BF1
	RIPEMD160	D5917E727A9652A4CD9C5DB59BF3324CABB7445D
	Tiger	F87D340F9F8A97ABD04A113D565DD97F6CF12D749FBB2731
	SHA-1	E86B5EB2EF2918E3F9B174A51CB13097B25CD4E3

Tabel 4. kecepatan hash tiap algoritma dari sebuah gambar dalam satuan milidetik(ms)

	Sha512	H	W	T	Ripemd	MD5	Sha1
G1	2	32	15	1	9	4	1
G2	27	79	57	21	22	4	9
G3	55	165	143	47	31	15	23

Untuk melakukan hashing pada file gambar, MD5 merupakan fungsi hash kriptografis yang paling baik dibandingkan yang lain. Tiger dan SHA1 sangat cepat untuk file yang berukuran kecil dan cukup cepat untuk file yang lebih besar. Haval dan Whirlpool bekerja lebih lambat dibandingkan yang lain. SHA512 dan RIPEMD160 memiliki kecepatan yang cukup bagus untuk melakukan hashing pada file gambar.

D. Pengujian Dengan Menggunakan File Video

Pengujian yang ketiga menggunakan file video. Terdapat dua file video yang digunakan dan keduanya berekstensi AVI. Video pertama (V1) memiliki ukuran file sebesar 13MB dan Video kedua (V2) memiliki ukuran file sebesar 210 MB. Hasil pengujian kecepatan pembuatan message digest dan hasil message digest dari file video dapat dilihat pada tabel dibawah ini.

Tabel 5. Hasil *message digest* dari sebuah file video

V	Algoritma	*Message Digest*
V1	SHA512	5A49E258C80C2F0882ED228C69A1E97CC1B2070DA8B3BC1C3CC565FDA2B43E08A05A49BB45EBCC4791D2142AABB343ADD9183F71A4C9500A93F74A09EC359C79
	MD5	F00DD8510570F032E0785EA98997D8FA
	Whirlpool	67377F3B5DE098863F694FE3F05A7F3B77943F332F322369862D57AC17ADCA9D2BCD8DFA5FD5480A6A06CC5FBA1D17B442D7EB4F95E777ACF2C6C8D3B933FB64
	Haval	9D6925843AED0D02310222A5D15448350853BA3287BAD76A9B6FE43BAD27D191
	RIPEMD160	58131C8CD24966DC27CDC49343BC5FC3085731DB
	Tiger	E769CC558FCC5E907EF6EA21D41602328CC58E7E55796397
	SHA-1	E28054B7BB0699C62BDE2D5E7195A3D0EBF5C5C0
V2	SHA512	9419C4BB2AED31113EDC33AFE83F87684F354FFCA6EA2D6E7FB7D1C744C688F95B81DB6D7390E7EBE659168888C9B76013AB20F22566DEC874FC2BA569A610A5
	MD5	638F24D296C571E862D0E94B765F8E1E

	Whirlpool	DCB06169D0BB44264289B8F003A1D8AC124CAFDD229BDBB914F0C56706A7D0FDC2AEF4B7982E3C2173FED7165BB6FC6FEDD7220EA3D244496A6C8333B72D4F55
	Haval	F692FAE5EAAD574A42AB20FD5EAC846A127F179E122EA67BAEAB20D3EBD56F7D
	RIPEMD160	3B963753A7AFA365BED8EA2C350AF70553A4A94B
	Tiger	EC59B9EF3D3EDD40E482FD5E0F3B4DBAD14B2794B28670BA
	SHA-1	03E488E6C2C1E0A00F56E71E090C875B55890D6F

Tabel 6. kecepatan hash tiap algoritma dari sebuah video dalam satuan milidetik(ms)

	Sha 512	H	W	T	Ripemd	MD5	Sha 1
V 1	322	1074	1055	378	273	75	89
V 2	6077	17051	17352	6047	4240	943	1151

Untuk file video yang tentunya memiliki ukuran yang besar, MD5 dan SHA1 menjadi pilihan yang tepat dimana mereka berhasil menghasilkan message digest jauh lebih cepat dibandingkan yang lainnya. Untuk ukuran keluaran yang sama, SHA512 lebih baik dibandingkan dengan Whirlpool.

E. Pengujian Dengan Menggunakan File Executable

Pengujian selanjutnya dilakukan dengan menggunakan sebuah file *executable*. Terdapat tiga file executable yang digunakan yakni file pertama (E1) berukuran 11KB, file kedua (E2) berukuran 660KB, dan file terakhir (E3) berukuran 8,75MB. *Message digest* yang dihasilkan beserta kecepatan pembuatan *message digest* tersebut dapat dilihat pada tabel dibawah ini.

Tabel 7. Hasil *message digest* dari sebuah pesan

P	Algoritma	*Message Digest*
E 1	SHA512	BF2FA4BF9CDADFAF09E7499614146B8353CE34A9A5DAE816FE0C063E86311960A4E8998F5A539D7AB180B33AB125F05C1202C7FCD7423061E3623EC58E82D120
	MD5	38A9A92EAC7EA7180F59B491A072CC21
	Whirlpool	1A23E7FED63B6BFFBB55E5211DFE2DAAB70D88F5780CCA192B1F11CB3F21BDF82194939DD499F0D56998B89A8D3BF8B8568907234E3EB137C0F771819EC9ED52
	Haval	D5D94841429B44DFBD01CEBC40417F307A2C47DCFF48FC399D2806AB7859AC0C
	RIPEMD160	4F5050D48090EFCE77AEBD6DDF0A3FB078CE21EB
	Tiger	4829A1FCE7C5FAC889A801A06C4DAC918438DEE7F95FEB8C
	SHA-1	7E8F6AF8B4E64B26EFB74759E79C20370D15F4D5
E 2	SHA512	6351B5526D1F6177C09BC9A07E9DB779F5EE1D7A1A462070A78827B93537FA48DAE75C02064FE3FEAD1FBCE2DA8CCD78D541D7CDF4B06FFD186D96D1343FD275
	MD5	ECABBF3B44A86179051496E19082F9FB
	Whirlpool	B32B6459AE58F6CE41A6E31015138007C64C28BC97B2224FBBC744615C712F293685FDFDED742B1E133D71677826428E23E3DFE013B6546D3D15ABB535087589
	Haval	9886BC5218A18C39AFA1392D067D936C7A54EF9C2D1C896E0A7B6EA270ECABC7
	RIPEMD160	E8E4D4BF22999F65BD4F54D2819BF5E66C071B24
	Tiger	7D77F1F397931A56811ABE27B818D89B67DB3F7658FC04B8
	SHA-1	A524A611A54A37CB79FC3D07BFD8D733A5DB8211
E 3	SHA512	7AB3A4DA3DF231F37550D3CE25E1A82BF3DF36C638740431F54BB381C794277A98497E056AE458C5B6329A6E0FFC7C239DD1E9ED33B00261DDD5F7EEA9FAC98C
	MD5	A05EB04B11B6AD23B07B7DF5B3639FB2
	Whirlpool	E84CB552B394E8F96D6CC647AE8810DDEAD9D11D5C7D1A77C915B843777D8233BCBD50EB00281A7CECCCCAD921A33D235D80469DC3BA895001FE7F999C33FE73
	Haval	486F34589FC4C8B3349267E136874DC18930B65794BE2ACF6812EDAA6F05103F
	RIPEMD160	90C2A07288245CB96591B365914054E5DE83598A
	Tiger	3DA608313C291E2DFEF8CFC1CEB16272BC1D11D727882D1F

	SHA-1	C41D30F847728AAB780B04907AF8FE08CECE1D79

Tabel 8. kecepatan hash tiap algoritma dari sebuah pesan dalam satuan milidetik(ms)

	Sha512	H	W	T	Ripemd	MD5	Sha1
E1	1	26	10	6	4	5	1
E2	25	70	61	31	25	8	6
E3	190	720	810	296	185	51	50

Dari tabel diatas dapat kita lihat bahwa SHA1 dan MD5 berhasil menghasilkan *message digest* dalam waktu yang sangat cepat bila dibandingkan dengan yang lainnya. Haval dan *Whirlpool* tidah cocok untuk digunakan dalam menghasilkan *message digest* dari file *executable* yang berukuran besar.

F. Pengujian Terakhir Menggunakan Gambar, Pesan, Video, dan Executable yang memiliki ukuran sama

Pengujian terakhir dilakukan dengan menggunakan empat file yakni gambar, video, pesan, dan executable yang memiliki ukuran yang sama yakni 2 MB. Kecepatan pembuatan *message digest* dapat dilihat pada tabel dibawah ini.

Tabel 9. kecepatan hash tiap algoritma dari berbagai jenis file dalam satuan milidetik(ms)

	Sha512	H	W	T	Ripemd	MD5	Sha1
P	60	206	191	83	56	28	18
G	75	166	190	65	60	17	10
V	46	180	182	78	48	18	32
E	296	180	183	72	57	11	12

P merupakan file pesan, G merupakan file gambar, V merupakan file video, dan E merupakan file *executable*. Dapat dilihat dari tabel diatas untuk MD5 cocok digunakan untuk mendapatkan nilai hash dari sebuah file *executable* dan video dikarenakan kecepatannya yang tinggi. Untuk file pesan dan gambar, SHA-1 mampu mengungguli kecepatan dari MD5. SHA512 lebih baik dibandingkan *whirlpool* meskipun memiliki ukuran keluaran yang sama, namun *whirlpool* unggul dalam mendapatkan nilai hash dari file *executable* dibandingkan dengan SHA-512. Haval tidak cocok untuk digunakan apabila pengguna hanya mengincar kecepatan pembuatan message digest dikarenakan *process time*-nya yang lambat. RIPEMD160 dan tiger cukup baik untuk digunakan untuk bentuk file manapun.

IV. KESIMPULAN

Pada makalah ini telah dilakukan serangkaian pengujian terhadap beberapa fungsi hash kriptografis yang populer yakni SHA1, SHA512, Haval, Whirlpool, Tiger, RIPEMD160, dan MD5. Pengujian dilakukan dengan menggunakan beberapa jenis file dan ukuran dan diakhiri dengan membandingkan kecepatan terhadap empat jenis file yang berbeda namun memiliki ukuran yang sama. Pengujian pada file pesan menunjukkan SHA1, SHA512, dan MD5 mampu menghasilkan nilai message digest secara cepat dibandingkan fungsi hash kriptografis lainnya. Pada file gambar, SHA1, SHA512, MD5, dan Tiger mampu menghasilkan message digest dengan cepat.

Pada file video, hanya SHA1 dan MD5 yang mampu menghasilkan message digest dengan cepat dan MD5 mengungguli kecepatan dari SHA1. Dan pada file executable, MD5 dan SHA1 lagi-lagi berhasil menghasilkan message digest dengan cepat dimana SHA1 mengungguli MD5. Untuk file berbeda yang memiliki ukuran yang sama. SHA1 dan MD5 menghasilkan message digest lebih cepat dibanding yang lain disusul dengan RIPEMD160 , Tiger, dan SHA512.

Penarikan kesimpulan dilakukan dengan memberikan nilai dari semua percobaan yang telah dilakukan dimana peringkat pertama diberi nilai 6, peringkat kedua diberikan nilai 5, peringkat ketiga diberikan nilai 4, dst. Dari seluruh percobaan yang dilakukan dapat disimpulkan bahwa SHA1 merupakan fungsi hash kriptografis yang tercepat dibandingkan fungsi hash kriptografis yang disebutkan dalam makalah ini. MD5 menempati urutan kedua dilanjutkan dengan RIPEMD160.

REFERENCES

http://www.rsa.com/rsalabs/node.asp?id=2176. Waktu akses : 26 April 2013, 17.16.

http://www.larc.usp.br/~pbarreto/WhirlpoolPage.html. Waktu akses : 26 April 2013, 17.18.

http://www.unixwiz.net/techtips/iguide-crypto-hashes.html, waktu akses : 22 April 2013, 20.27.

Daniel J. Bernstein and Tanja Lange (editors). eBACS: ECRYPT Benchmarking of Cryptographic Systems. http://bench.cr.yp.to, waktu akses 22 April 2013 20.25.

http://pcsupport.about.com/od/termsc/g/cryptographic-hash-function.htm. Waktu akses : 26 April 2013, 17.20.

http://pcsupport.about.com/od/termsm/g/md5.htm. Waktu akses : 26 April 2013, 17.16.

https://www.cosic.esat.kuleuven.be/fse2009/slides/2402_1150_Schlaeffer.pdf. Waktu akses : 26 April 2013, 17.16.

Preneel, B. (1994). Cryptographic hash functions. *European Transactions on Telecommunications*, *5*(4), 431-448.

http://www.kellermansoftware.com/t-articlestrongesthash.aspx Waktu akses : 26 April 2013, 17.16.

Zheng, Y., Pieprzyk, J., & Seberry, J. (1993, January). HAVAL—a one-way hashing algorithm with variable length of output. In *Advances in Cryptology—AUSCRYPT'92* (pp. 81-104). Springer Berlin Heidelberg.

http://labs.calyptix.com/haval.php. Waktu akses : 26 April 2013, 17.41.
http://ehash.iaik.tugraz.at/wiki/GOST. Waktu akses : 6 Mei 2013, 03.19.
http://tools.ietf.org/html/rfc5831#section-5. Waktu akses : 6 Mei 2013, 03.19.
http://help.csharptest.net/CSharpTest.Net.Library~CSharpTest.Net.Crypto.WhirlpoolManaged.html. Waktu akses : 15 Mei 2013, 10.21.
http://msdn.microsoft.com/en-us/library/system.security.cryptography. waktu akses : 15 Mei 2013, 10.26.
https://code.google.com/p/classless-hasher/. Waktu akses : 15 mei 2013, 12.00

PERNYATAAN

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

Bandung, 16 Mei 2013

Dibi Khairurrazi Budiarsyah
13509013