



Rancang Bangun Sistem Identifikasi Integritas File Digital Menggunakan Metode Hash Untuk Keperluan Forensik IT

Luthfi Rahman^{1*}, Rizaldy Khair²

^{1,2}Program Studi Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia

Email: ¹*luthfirahman191@gmail.com, ²rizaldykhair@umsu.ac.id

Abstrak

Perkembangan kejahatan siber yang semakin masif menyebabkan file digital rentan terhadap manipulasi dan pemalsuan tanpa meninggalkan jejak yang terlihat secara kasat mata. Dalam ranah forensik IT, keaslian dan integritas file digital merupakan syarat mutlak agar barang bukti dapat dipertanggungjawabkan secara hukum. Penelitian ini bertujuan membangun sistem berbasis web yang mampu mengidentifikasi integritas file digital secara otomatis menggunakan metode hash dengan mengintegrasikan tiga algoritma sekaligus, yaitu MD5, SHA-1, dan SHA-256. Metodologi penelitian menggunakan pendekatan kuantitatif eksperimen komputasional dengan model pengembangan sistem Waterfall. Sistem dibangun menggunakan bahasa pemrograman Python, framework Flask, dan basis data MySQL. Hasil penelitian berupa sistem yang mampu menghitung nilai hash dari file digital menggunakan ketiga algoritma secara simultan, menyimpan nilai hash sebagai referensi ke dalam basis data, membandingkan nilai hash secara otomatis, menampilkan status integritas file, serta menghasilkan laporan hasil verifikasi yang terdokumentasi. Pengujian menggunakan metode Black Box Testing menunjukkan bahwa seluruh fungsi sistem berjalan sesuai dengan spesifikasi kebutuhan fungsional yang telah ditetapkan, dengan tingkat keberhasilan deteksi modifikasi file mencapai 100% pada seluruh skenario pengujian. Sistem ini diharapkan menjadi solusi teknis yang efektif bagi investigator digital dalam memverifikasi keaslian file digital untuk keperluan forensik IT.

Kata Kunci: integritas file digital, metode hash, forensik digital, MD5, SHA-1, SHA-256, sistem berbasis web, verifikasi integritas.

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah membawa perubahan mendasar pada cara manusia menyimpan, mengelola, dan mendistribusikan informasi. Hampir seluruh data kini tersimpan dalam bentuk file digital, mulai dari dokumen penting, rekam medis, bukti transaksi perbankan, hingga rekaman multimedia. Namun demikian, kemudahan manipulasi file digital tanpa jejak yang tampak oleh mata menjadi tantangan serius dalam dunia keamanan informasi dan penegakan hukum digital (Casey, 2011).

Kejahatan siber yang terus berkembang semakin memperparah ancaman terhadap integritas file digital. Berdasarkan laporan Badan Siber dan Sandi Negara (BSSN, 2023), jumlah insiden siber di Indonesia terus meningkat secara signifikan, dengan manipulasi dan pemalsuan dokumen digital sebagai salah satu modus yang paling sering ditemukan dalam proses investigasi forensik. Kondisi ini menuntut tersedianya alat bantu teknis yang andal untuk memverifikasi keaslian file digital secara ilmiah dan terdokumentasi.

Dalam konteks forensik digital, integritas barang bukti merupakan prasyarat utama agar dapat dipertimbangkan di hadapan pengadilan (Nance dan Horsman, 2021). Prinsip *chain of custody* mengharuskan setiap perubahan yang terjadi pada barang bukti digital terdokumentasi dengan baik. Apabila integritas sebuah file tidak dapat diverifikasi, nilai pembuktiannya akan diragukan bahkan ditolak (Carrier, 2005). Sementara itu, investigator digital saat ini masih sering melakukan verifikasi secara manual yang rentan terhadap kesalahan manusia (*human error*) dan tidak tersistematis.

Salah satu metode yang telah terbukti efektif dalam memverifikasi integritas file digital adalah fungsi hash kriptografi. Fungsi hash menghasilkan nilai ringkasan (*digest*) yang unik dari sebuah masukan data dengan panjang tetap. Sifat deterministik dan sensitif terhadap perubahan sekecil apapun (*avalanche*

effect) menjadikan fungsi hash sebagai alat yang andal dalam mendeteksi modifikasi file (Stallings, 2022). Algoritma seperti MD5 (Rivest, 1992), SHA-1 (NIST, 2012), dan SHA-256 (NIST, 2015) telah digunakan secara luas dalam praktik keamanan informasi dan forensik digital.

Kajian literatur terdahulu menunjukkan bahwa penelitian terkait penerapan fungsi hash dalam forensik digital telah banyak dilakukan. Riadi dkk. (2021) membuktikan efektivitas MD5 dalam verifikasi barang bukti digital, namun hanya menggunakan satu algoritma. Prayudi dan Ashari (2022) mengembangkan sistem verifikasi berbasis web menggunakan SHA-256 dan MD5, tetapi belum menyediakan fitur laporan verifikasi yang terstruktur. Handoko dkk. (2022) membandingkan performa MD5, SHA-1, dan SHA-256 secara teoritis, namun belum membangun sistem terintegrasi yang siap pakai. Pratama dkk. (2023) mengevaluasi kombinasi ketiga algoritma tersebut dan menyimpulkan bahwa penggunaannya secara bersamaan menghasilkan verifikasi yang lebih komprehensif dibandingkan satu algoritma saja.

Kebaruan ilmiah penelitian ini terletak pada pengintegrasian ketiga algoritma hash (MD5, SHA-1, dan SHA-256) secara simultan dalam satu platform berbasis web yang dilengkapi fitur perbandingan otomatis, manajemen basis data referensi hash, dan *generate* laporan verifikasi yang terdokumentasi untuk keperluan forensik IT. Penelitian ini mengisi kesenjangan yang ada pada penelitian-penelitian sebelumnya yang belum mengintegrasikan ketiga aspek tersebut dalam satu sistem yang siap dioperasikan oleh investigator digital.

Berdasarkan latar belakang tersebut, penelitian ini merumuskan masalah sebagai berikut: (1) Bagaimana membangun sistem berbasis web yang mampu mengidentifikasi integritas file digital secara otomatis menggunakan metode hash untuk keperluan forensik IT? (2) Bagaimana mengintegrasikan algoritma MD5, SHA-1, dan SHA-256 secara bersamaan dalam satu sistem? (3) Bagaimana mengukur tingkat keberhasilan sistem dalam mendeteksi perubahan pada file digital?

METODE

Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen komputasional, berfokus pada pengukuran dan evaluasi performa algoritma hash dalam mengidentifikasi integritas file digital. Jenis penelitian adalah kuantitatif eksplanatori yang bertujuan menjelaskan bagaimana metode hash mendeteksi perubahan pada file digital melalui pengujian sistem secara sistematis. Model pengembangan sistem yang digunakan adalah Waterfall (Pressman, 2010), yang terdiri dari lima tahapan berurutan: analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan.

Landasan Teori

Fungsi Hash

Fungsi hash adalah fungsi matematis yang memetakan data masukan dengan ukuran sembarang menjadi keluaran berupa nilai tetap (*fixed-length output*) yang disebut sebagai nilai hash, *digest*, atau *checksum* (Stallings, 2022). Fungsi hash bekerja secara satu arah (*one-way function*), artinya nilai hash yang dihasilkan tidak dapat dikembalikan ke bentuk data aslinya. Secara matematis, fungsi hash dinotasikan sebagai:

$$H(M) = h$$

di mana H adalah fungsi hash yang digunakan, M adalah data masukan berupa file digital dengan ukuran sembarang, dan h adalah nilai hash (*digest*) yang dihasilkan dengan panjang tetap.

Fungsi hash yang baik memiliki empat karakteristik utama: (a) Deterministik, yaitu data masukan yang sama selalu menghasilkan nilai hash yang sama; (b) Avalanche Effect, yaitu perubahan sekecil apapun pada data masukan menghasilkan nilai hash yang sangat berbeda; (c) Collision Resistant, yaitu sangat sulit menemukan dua data masukan berbeda yang menghasilkan nilai hash sama; dan (d) One-Way, yaitu nilai hash tidak dapat digunakan untuk merekonstruksi data aslinya (Paar dan Pelzl, 2010).

Algoritma MD5

MD5 (*Message Digest Algorithm 5*) dikembangkan oleh Ronald Rivest pada tahun 1991 dan menghasilkan nilai hash sepanjang 128 bit (32 karakter heksadesimal). Algoritma ini memproses data dalam blok 512 bit melalui serangkaian operasi logika dan aritmatika (Rivest, 1992). Meskipun MD5 telah dinyatakan tidak aman untuk keperluan kriptografi karena rentan terhadap serangan *collision* (Wang dan Yu, 2005), algoritma ini tetap digunakan dalam forensik digital sebagai salah satu mekanisme verifikasi integritas file.

Algoritma SHA-1

SHA-1 (*Secure Hash Algorithm 1*) dikembangkan oleh NSA dan diterbitkan oleh NIST pada tahun 1995, menghasilkan nilai hash sepanjang 160 bit (40 karakter heksadesimal). SHA-1 menawarkan tingkat keamanan yang lebih tinggi dibandingkan MD5. Namun, pada tahun 2017, Stevens dkk. (2017) berhasil membuktikan adanya kelemahan *collision* pada SHA-1 melalui serangan yang dikenal sebagai SHAttered.

Meskipun demikian, SHA-1 masih relevan digunakan dalam konteks forensik digital untuk keperluan verifikasi integritas data (NIST, 2012).

Algoritma SHA-256

SHA-256 merupakan bagian dari keluarga algoritma SHA-2 yang dikembangkan oleh NSA dan diterbitkan oleh NIST pada tahun 2001, menghasilkan nilai hash sepanjang 256 bit (64 karakter heksadesimal). Hingga saat ini, SHA-256 dianggap sebagai algoritma hash yang paling aman di antara ketiga algoritma yang dibahas, karena ketahanannya terhadap serangan *collision* maupun *preimage attack* (NIST, 2015). SHA-256 banyak digunakan dalam berbagai aplikasi keamanan modern, termasuk teknologi blockchain, sertifikat digital, dan forensik IT.

Tabel 1. Perbandingan Algoritma MD5, SHA-1, dan SHA-256

No	Karakteristik	MD5	SHA-1	SHA-256
1	Panjang nilai hash	128 bit	160 bit	256 bit
2	Karakter heksadesimal	32 karakter	40 karakter	64 karakter
3	Ukuran blok	512 bit	512 bit	512 bit
4	Tahun diterbitkan	1991	1995	2001
5	Diterbitkan oleh	Ronald Rivest	NSA/NIST	NSA/NIST
6	Tingkat keamanan	Rendah	Sedang	Tinggi
7	Kerentanan Collision	Ya	Ya (2017)	Tidak
8	Kecepatan komputasi	Sangat cepat	Cepat	Sedang
9	Cocok untuk forensik	Ya	Ya	Ya

Forensik Digital

Forensik digital (*digital forensics*) adalah cabang ilmu forensik yang berfokus pada proses identifikasi, pengumpulan, pengamanan, analisis, dan penyajian bukti digital dari perangkat elektronik untuk keperluan investigasi hukum (Casey, 2011). Dalam forensik digital, integritas barang bukti harus dapat dibuktikan melalui prinsip-prinsip: autentisitas, integritas, *chain of custody*, dan reproduisibilitas (Nance dan Horsman, 2021).

Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan melalui observasi dan studi literatur untuk memastikan sistem yang dikembangkan dapat menjawab permasalahan utama. Ditemukan empat permasalahan utama: (a) belum tersedianya sistem terstandarisasi untuk verifikasi integritas file digital secara otomatis; (b) sulitnya membuktikan bahwa file bukti tidak mengalami perubahan; (c) belum adanya sistem berbasis web yang mengintegrasikan tiga algoritma hash secara bersamaan; (d) kurangnya dokumentasi hasil verifikasi yang terstruktur dan sistematis.

Berdasarkan analisis tersebut, ditetapkan kebutuhan fungsional sistem meliputi: (1) menerima unggahan file digital dari pengguna; (2) menghitung nilai hash menggunakan tiga algoritma secara simultan; (3) menyimpan nilai hash referensi ke dalam basis data; (4) membandingkan nilai hash secara otomatis; (5) menampilkan status integritas file; (6) menghasilkan laporan hasil verifikasi; dan (7) menyediakan antarmuka berbasis web yang mudah digunakan.

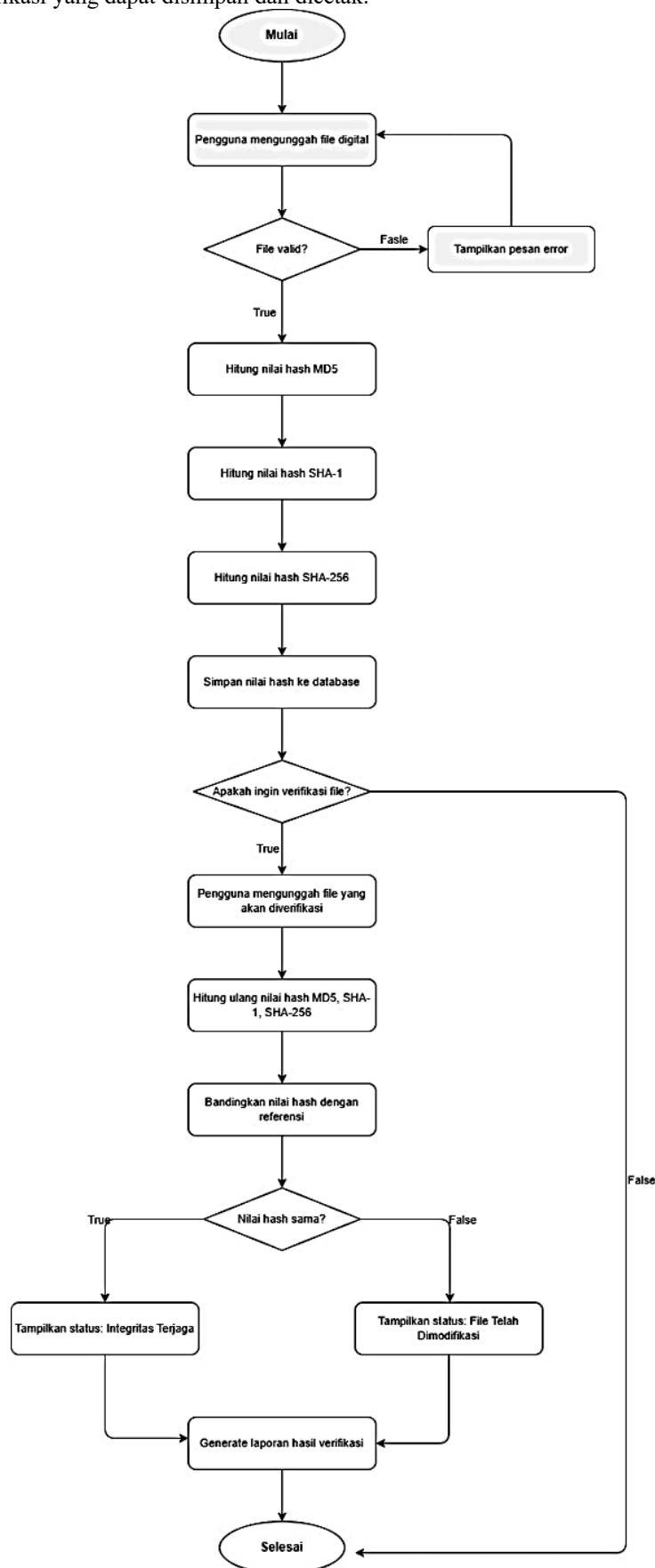
Perancangan Sistem

Flowchart Sistem

Alur kerja sistem dimulai ketika pengguna mengunggah file digital melalui antarmuka berbasis web. Sistem melakukan validasi terhadap jenis dan format file. Apabila file tidak valid, sistem menampilkan pesan *error*. Apabila valid, sistem menghitung nilai hash MD5, SHA-1, dan SHA-256 secara berurutan, kemudian menyimpan ketiga nilai hash tersebut beserta metadata file ke dalam basis data sebagai referensi.

Untuk proses verifikasi, pengguna mengunggah kembali file yang ingin diperiksa. Sistem menghitung ulang nilai hash dan membandingkannya dengan nilai hash referensi. Apabila seluruh nilai hash identik, sistem menampilkan status "Integritas Terjaga". Apabila terdapat perbedaan, sistem

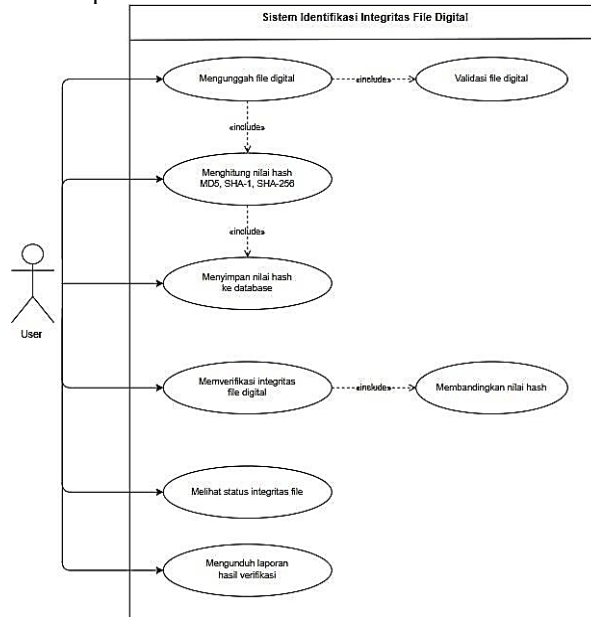
menampilkan status "File Telah Dimodifikasi". Pada tahap akhir, sistem secara otomatis menghasilkan laporan hasil verifikasi yang dapat disimpan dan dicetak.



Gambar 1. Flowchart Sistem Identifikasi Integritas File Digital

Diagram Use Case

Sistem memiliki satu aktor utama yaitu User (investigator digital) yang dapat melakukan delapan interaksi utama: mengunggah file digital, validasi file (otomatis via *include*), menghitung nilai hash MD5/SHA-1/SHA-256 (otomatis via *include*), menyimpan nilai hash ke basis data (otomatis via *include*), memverifikasi integritas file digital, membandingkan nilai hash (otomatis via *include*), melihat status integritas file, dan mengunduh laporan hasil verifikasi.



Gambar 2. Diagram Use Case Sistem Identifikasi Integritas File Digital

Activity Diagram

Activity Diagram menggambarkan alur aktivitas yang terjadi antara pengguna dan sistem secara paralel melalui dua *swimlane*. Aktivitas dimulai ketika pengguna membuka sistem melalui browser, dilanjutkan dengan pengunggahan file, validasi oleh sistem, penghitungan hash, penyimpanan ke basis data, keputusan verifikasi, perbandingan hash, penampilan status, dan *generate* laporan.



Gambar 3. Activity Diagram Sistem Identifikasi Integritas File Digital

Implementasi Sistem

Implementasi sistem dilakukan berdasarkan rancangan yang telah ditetapkan menggunakan spesifikasi perangkat sebagaimana tercantum pada Tabel 2 dan Tabel 3.

Tabel 2. Spesifikasi Perangkat Keras

No	Komponen	Spesifikasi
1	Prosesor	Intel Core i5
2	RAM	8 GB
3	Penyimpanan	281 GB
4	Sistem Operasi	Windows 11

Tabel 3. Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Versi
1	Python	3.13.1
2	Flask	3.1.3
3	MySQL	3.3.0
4	Google Chrome	151.0.7922.108
5	Visual Studio Code	1.13.2

Basis data sistem terdiri dari dua tabel utama. Tabel files menyimpan data referensi hash dengan kolom: id_file, nama_file, ekstensi, ukuran_file, tipe_file, lokasi_file, md5_hash, sha1_hash, sha256_hash, dan tanggal_upload. Tabel verifications menyimpan riwayat verifikasi dengan kolom: id_verifikasi, id_file, md5_hash, sha1_hash, sha256_hash, status, waktu_verifikasi, dan keterangan.

Metode Pengujian

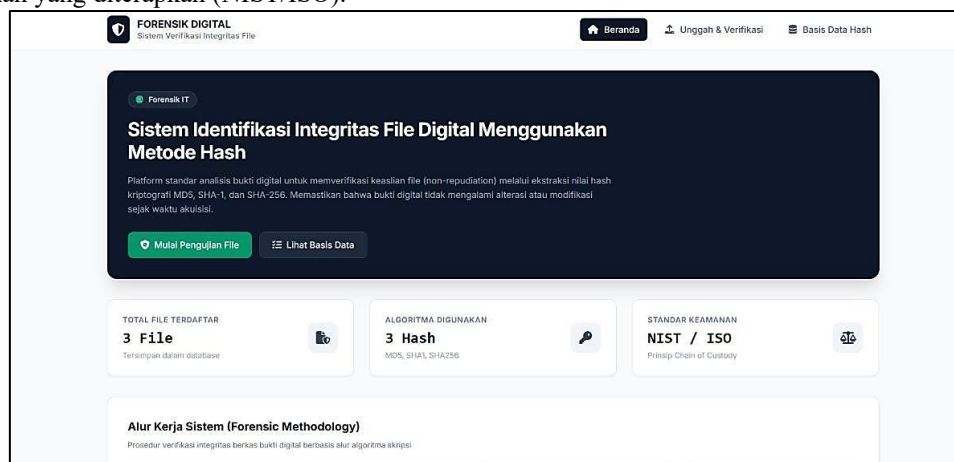
Pengujian sistem dilakukan menggunakan dua pendekatan:

1. Black Box Testing, yaitu metode pengujian yang berfokus pada fungsionalitas sistem dari sudut pandang pengguna tanpa memperhatikan struktur kode internal. Pengujian ini dilakukan dengan menguji setiap fungsi sistem berdasarkan kebutuhan fungsional yang telah ditetapkan (Pressman, 2010).
2. Pengujian Akurasi Algoritma Hash, dilakukan dengan dua skenario: (a) Skenario file tidak dimodifikasi — file asli diunggah dua kali untuk memastikan sistem menghasilkan nilai hash yang identik; dan (b) Skenario file dimodifikasi — file asli diubah kemudian diunggah kembali untuk memastikan sistem mendeteksi perubahan. Pengujian dilakukan terhadap tiga jenis file: dokumen PDF, gambar JPEG, dan video MP4.

HASIL DAN PEMBAHASAN

Implementasi Antarmuka Sistem

Sistem berhasil dibangun dengan empat halaman utama yang saling terintegrasi. Halaman Beranda menampilkan informasi umum tentang sistem beserta fitur-fitur utama yang tersedia, meliputi total file terdaftar dalam basis data, algoritma hash yang digunakan (MD5, SHA-1, SHA-256), dan standar keamanan yang diterapkan (NIST/ISO).



Gambar 4. Halaman Beranda Sistem

Halaman Unggah File menyediakan fitur *drag and drop* untuk mengunggah file digital beserta pilihan mode operasi: (a) Registrasi File Asli Baru untuk menghitung dan menyimpan hash referensi, dan (b) Verifikasi Integritas File untuk membandingkan hash file yang diunggah dengan data referensi dalam basis data.

FORENSIK DIGITAL
Sistem Verifikasi Integritas File

Beranda Unggah & Verifikasi Basis Data Hash

Modul Unggah & Pemrosesan Hash
Pilih mode operasi: Menyimpan hash file asli baru atau Memverifikasi file yang dicurigai.

1. PILIH MODE OPERASI SISTEM

Registrasi File Asli Baru
Menghitung MD5, SHA1, SHA256 & menyimpan nilai hash ke database sebagai baseline.

Verifikasi Integritas File
Mengunggah ulang file uji untuk dibandingkan dengan data asli di database.

2. PILIH FILE BUKTI DIGITAL

Pilih file dari komputer Anda
Mendukung segala ekstensi file (Dokumen, Gambar, Executable, Archive) hingga 64MB

Choose File No file chosen

Proses & Hitung Hash Digital

Gambar 5. Halaman Unggah File Sistem

Halaman Hasil Verifikasi menampilkan matriks perbandingan *hashing* forensik yang memuat: metadata file asli dari basis data, metadata file hasil pengujian, nilai hash dari ketiga algoritma (MD5, SHA-1, SHA-256) beserta statusnya (IDENTIK atau BERBEDA), dan status integritas keseluruhan file.

FORENSIK DIGITAL
Sistem Verifikasi Integritas File

Beranda Unggah & Verifikasi Basis Data Hash

INTEGRITAS TERJAGA (FILE VALID)
Seluruh nilai hash (MD5, SHA-1, SHA-256) antara file uji dan data registrasi awal persis sama. Berkas terbukti otentik secara forensik.

Matriks Perbandingan Hashing Forensik Waktu Uji: 2026-08-10 06:19:28

METADATA FILE ASLI (DATABASE)
Nama File: KHS_sem_2.pdf
Ukuran: 81457 Bytes
Tgl Registrasi: 2026-08-04 08:24:17

METADATA FILE UJI (PENGUJIAN)
Nama File: KHS_sem_2.pdf
Ukuran: 81457 Bytes
Status Komparasi: MATCH

ALGORITMA MD5 (128-bit)
Asli: f26532550771c67f4e6a6477a65da87
Pengujian: f26532550771c67f4e6a6477a65da87 IDENTIK

ALGORITMA SHA-1 (160-bit)
Asli: 87a481a8398262928e4650e62f98bdafee6fc9
Pengujian: 87a481a8398262928e4650e62f98bdafee6fc9 IDENTIK

ALGORITMA SHA-256 (256-bit) IDENTIK

Gambar 6. Halaman Hasil Verifikasi Sistem

Halaman Basis Data Hash menampilkan seluruh daftar file yang pernah dihitung nilai hash-nya, lengkap dengan informasi nama file, ukuran, nilai MD5, SHA-1, SHA-256, dan tanggal penyimpanan.

FORENSIK DIGITAL
Sistem Verifikasi Integritas File

Beranda Unggah & Verifikasi Basis Data Hash

Basis Data Nilai Hash Terregistrasi
Daftar checksum digital terverifikasi sebagai referensi baseline forensik.

+ Tambah File Baru

ID	NAMA FILE	UKURAN	MD5	SHA-1	SHA-256	TANGGAL SIMPAN
#4	KHS_sem_2.pdf	79.55 KB	f26532550...	87a481a83...	02e7c921d6b...	2026-08-04 08:24
#1	Proposal_Seri_Kewirausahaan_Luthfi.docx	553.35 KB	b700c17a2a...	88b390e407...	e68486f7eb3...	2026-08-04 08:00
#2	Makalah_Kewirausahaan_Azmi.docx	641.97 KB	a58b37a5c...	6bf54fc237...	1a4088f0bce...	2026-08-04 07:57

Gambar 7. Halaman Basis Data Hash

Hasil Pengujian Black Box Testing

Pengujian Black Box Testing dilakukan terhadap sepuluh fungsi utama sistem. Seluruh fungsi yang diuji menunjukkan output yang sesuai dengan ekspektasi, sehingga seluruh pengujian berstatus Berhasil.

Tabel 4. Hasil Pengujian Black Box Testing

No	Fungsi yang Diuji	Skenario Uji	Input	Output yang Diharapkan	Output Sistem	Status
1	Unggah File Digital	Mengunggah file valid	File PDF 2.4 MB	File berhasil diunggah dan diproses	File berhasil diunggah	✓ Berhasil
2	Validasi File	Mengunggah file tidak valid	File tanpa ekstensi	Sistem menampilkan pesan <i>error</i>	Sistem menampilkan pesan <i>invalid</i>	✓ Berhasil
3	Hitung Hash MD5	Mengunggah file dokumen	File DOCX	Nilai hash MD5 32 karakter tampil	Menampilkan nilai hash MD5 32 karakter	✓ Berhasil
4	Hitung Hash SHA-1	Mengunggah file gambar	File JPG	Nilai hash SHA-1 40 karakter tampil	Menampilkan nilai hash SHA-1 40 karakter	✓ Berhasil
5	Hitung Hash SHA-256	Mengunggah file video	File MP4	Nilai hash SHA-256 64 karakter tampil	Menampilkan nilai hash SHA-256 64 karakter	✓ Berhasil
6	Simpan Hash ke Basis Data	Menyimpan hash referensi	File valid	Nilai hash tersimpan ke basis data	Nilai hash tersimpan	✓ Berhasil
7	Verifikasi File Tidak Berubah	Mengunggah file asli untuk verifikasi	File asli	Status "Integritas Terjaga" tampil	Menampilkan status file valid	✓ Berhasil
8	Verifikasi File Berubah	Mengunggah file yang dimodifikasi	File yang diubah	Status "File Telah Dimodifikasi" tampil	Menampilkan status file <i>invalid</i>	✓ Berhasil
9	Bandingkan Nilai Hash	Membandingkan hash referensi dan verifikasi	Dua file berbeda	Sistem mendeteksi perbedaan hash	Sistem mendeteksi perbedaan hash	✓ Berhasil
10	Cetak Laporan	Mencetak laporan verifikasi	—	Laporan berhasil dicetak	Laporan berhasil dicetak	✓ Berhasil

Hasil pengujian Black Box Testing menunjukkan tingkat keberhasilan 100% pada seluruh fungsi yang diuji. Hal ini membuktikan bahwa sistem yang dibangun memenuhi seluruh kebutuhan fungsional yang telah ditetapkan.

Hasil Pengujian Akurasi Algoritma Hash

Skenario 1: File Tidak Dimodifikasi

Pengujian skenario pertama dilakukan dengan mengunggah file asli dua kali untuk memastikan sistem menghasilkan nilai hash yang identik dan menampilkan status "Integritas Terjaga".

Tabel 5. Hasil Pengujian Akurasi — File Tidak Dimodifikasi

No	Nama File	Jenis	Ukuran	MD5	SHA-1	SHA-256	Status
1	KHS_Sem 2	PDF	79.55 KB	f265325550...77a61da87	87a401d...4f6e6f4c9	02e9c921dd...a3b75	Terjaga
2	Pas Foto 3x4	JPEG	51.99 KB	4ae609002b...d36465eeb	e4c125c7...d44d66d	83712f8a...411b	Terjaga
3	Record UAS	MP4	65.601 MB	fe9bb55aed...a9dd0d1d6	b32cec7a...f1997d236	6ab0156e...73d5	Terjaga

Seluruh file yang diunggah dua kali tanpa modifikasi menghasilkan nilai hash yang identik pada ketiga algoritma. Hasil ini membuktikan sifat deterministik dari fungsi hash, di mana data masukan yang sama selalu menghasilkan nilai hash yang sama, sesuai dengan teori yang dikemukakan oleh Stallings (2022).

Skenario 2: File Telah Dimodifikasi

Pengujian skenario kedua dilakukan dengan mengubah file asli kemudian mengunggahnya kembali untuk memastikan sistem mendeteksi perubahan dan menampilkan status "File Telah Dimodifikasi".

Tabel 6. Hasil Pengujian Akurasi — File Dimodifikasi

No	Nama File	Jenis	Perubahan	MD5	SHA-1	SHA-256	Status
1	KHS_Sem 1	PDF	Teks diubah	24ae9bfc...91e53cd7	2d1b91af...7feae705	0cd1274...7a4cf90	Dimodifikasi
2	Pas Foto Serti	JPEG	Piksel diubah	345efb06...f5102b	6b4132fc...323510ac8e	ee02a620...b967f81e	Dimodifikasi
3	Record UTS	MP4	Frame dihapus	0c046c42...4bf2e282	b32cec7a...f1997d236	6ab0156e...73d548	Dimodifikasi

Seluruh file yang telah mengalami modifikasi berhasil dideteksi oleh sistem pada ketiga algoritma hash. Perubahan sekecil apapun — baik perubahan teks pada dokumen, piksel pada gambar, maupun penghapusan *frame* pada video — menghasilkan nilai hash yang berbeda secara signifikan. Fenomena ini membuktikan sifat *avalanche effect* yang merupakan karakteristik fundamental dari fungsi hash kriptografi (Paar dan Pelzl, 2010).

Pembahasan

Hasil penelitian ini menunjukkan bahwa sistem yang dibangun berhasil memenuhi seluruh tujuan penelitian yang telah ditetapkan. Beberapa temuan penting yang dapat diidentifikasi adalah sebagai berikut:

1. Pengintegrasian tiga algoritma hash secara simultan terbukti menghasilkan verifikasi yang lebih komprehensif dibandingkan penggunaan satu algoritma saja. Apabila satu algoritma memberikan hasil yang meragukan — misalnya karena kerentanan *collision* pada MD5 atau SHA-1 — maka hasil dari algoritma lain, terutama SHA-256, dapat dijadikan pembanding yang lebih kuat. Temuan ini selaras dengan hasil penelitian Pratama dkk. (2023) yang menyimpulkan bahwa kombinasi tiga algoritma menghasilkan identifikasi integritas file yang lebih akurat.
2. Sifat *avalanche effect* yang terverifikasi dalam pengujian membuktikan bahwa bahkan perubahan satu piksel pada gambar JPEG atau penghapusan sebagian *frame* video MP4 sudah cukup untuk menghasilkan nilai hash yang sama sekali berbeda. Hal ini konsisten dengan karakteristik teoritis fungsi hash yang dikemukakan oleh Stallings (2022), yang menjamin tidak ada modifikasi file yang dapat luput dari deteksi sistem.
3. Fitur *generate* laporan verifikasi yang terdokumentasi merupakan kontribusi praktis yang membedakan penelitian ini dari penelitian-penelitian sebelumnya. Laporan tersebut memuat seluruh metadata pemeriksaan, nilai hash ketiga algoritma, dan status integritas file, sehingga dapat digunakan sebagai dokumentasi teknis pendukung yang dapat dipertanggungjawabkan secara ilmiah dalam proses forensik IT.
4. Antarmuka berbasis web yang responsif memungkinkan sistem diakses oleh investigator digital tanpa memerlukan instalasi perangkat lunak khusus, meningkatkan aksesibilitas dan kemudahan penggunaan (*usability*) sebagaimana dipersyaratkan dalam kebutuhan non-fungsional sistem.

Apabila dibandingkan dengan penelitian terdahulu, posisi penelitian ini dapat dilihat pada Tabel 7 berikut.

Tabel 7. Perbandingan dengan Penelitian Terdahulu

Peneliti	Tahun	Algoritma	Sistem Berbasis Web	Laporan Verifikasi	Terintegrasi Forensik IT
Riadi dkk.	2021	MD5 saja	Tidak	Tidak	Ya
Prayudi & Ashari	2022	MD5, SHA-256	Ya	Tidak	Sebagian
Handoko dkk.	2022	MD5, SHA-1, SHA-256	Tidak	Tidak	Tidak
Pratama dkk.	2023	MD5, SHA-1, SHA-256	Ya	Tidak	Sebagian
Penelitian ini	2026	MD5, SHA-1, SHA-256	Ya	Ya	Ya

Tabel 7 memperlihatkan bahwa penelitian ini merupakan yang pertama mengintegrasikan ketiga aspek utama, penggunaan tiga algoritma sekaligus, platform berbasis web, dan fitur laporan verifikasi terdokumentasi dalam satu sistem yang dirancang khusus untuk keperluan forensik IT. Hal ini menegaskan kebaruan ilmiah penelitian ini dibandingkan dengan penelitian-penelitian terdahulu di bidang yang sama.

KESIMPULAN

Berdasarkan hasil penelitian, implementasi, dan pengujian sistem yang telah dilakukan, dapat ditarik tiga kesimpulan utama. Pertama, sistem identifikasi integritas file digital berbasis web berhasil dibangun menggunakan bahasa pemrograman Python dengan framework Flask dan basis data MySQL. Sistem mampu mengidentifikasi integritas file digital secara otomatis melalui fitur unggah file, penghitungan nilai hash, penyimpanan referensi hash ke basis data, perbandingan hash otomatis, penampilan status integritas file, dan *generate* laporan verifikasi terdokumentasi yang dapat digunakan sebagai alat bantu teknis dalam proses investigasi forensik IT. Kedua, sistem berhasil mengintegrasikan tiga algoritma hash (MD5, SHA-1, dan SHA-256) secara simultan dalam satu platform berbasis web. Pengintegrasian ketiga algoritma menghasilkan verifikasi integritas file yang lebih komprehensif dan dapat dipertanggungjawabkan secara ilmiah, karena setiap algoritma memiliki karakteristik dan tingkat keamanan yang berbeda-beda sehingga saling melengkapi. Ketiga, hasil pengujian Black Box Testing menunjukkan tingkat keberhasilan 100% pada seluruh fungsi yang diuji. Pengujian akurasi membuktikan bahwa ketiga algoritma hash mampu mendeteksi perubahan sekecil apapun pada file digital melalui sifat *avalanche effect*, baik pada file dokumen PDF, gambar JPEG, maupun video MP4 yang telah dimodifikasi.

Untuk penelitian selanjutnya, disarankan: (1) menambahkan algoritma hash terbaru seperti SHA-3 atau BLAKE2 untuk meningkatkan standar keamanan sistem; (2) mengembangkan fitur autentikasi dan manajemen pengguna (*multi-user*) agar sistem dapat digunakan oleh beberapa investigator secara bersamaan dengan hak akses yang terpisah; dan (3) mengembangkan versi aplikasi *mobile* berbasis Android atau iOS untuk meningkatkan portabilitas dan fleksibilitas penggunaan di lapangan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Rizaldy Khair, S.Kom., M.Kom. selaku dosen pembimbing atas bimbingan dan arahan dalam penelitian ini, serta kepada Program Studi Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara atas dukungan akademis yang diberikan.

DAFTAR PUSTAKA

- Badan Siber dan Sandi Negara (BSSN). (2023). *Laporan Tahunan Keamanan Siber Indonesia 2023*. Jakarta: BSSN. <https://www.bssn.go.id>
- Carrier, B. (2005). *File System Forensic Analysis*. Addison-Wesley Professional.
- Casey, E. (2011). *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet* (3rd ed.). Academic Press. <https://doi.org/10.1016/B978-0-12-374268-1.00001-9>
- Handoko, Santoso, B., dan Wibowo, A. (2022). Perbandingan performa algoritma hash MD5, SHA-1, dan SHA-256 dalam verifikasi integritas file. *Jurnal Ilmu Komputer dan Informatika*, 8(2), 45–54. <https://doi.org/10.30645/jiki.v8i2.187>
- Kang, J., dan Kim, H. (2022). Digital forensics investigation methodology applicable to social network services. *IEEE Access*, 10, 72490–72506. <https://doi.org/10.1109/ACCESS.2022.3189289>
- Nance, K., dan Horsman, G. (2021). Evidence integrity in digital forensic investigations. *Forensic Science International: Digital Investigation*, 36, 301086. <https://doi.org/10.1016/j.fsidi.2021.301086>
- National Institute of Standards and Technology (NIST). (2012). *FIPS PUB 180-4: Secure Hash Standard (SHS)*. U.S. Department of Commerce. <https://doi.org/10.6028/NIST.FIPS.180-4>
- National Institute of Standards and Technology (NIST). (2015). *FIPS PUB 180-4: Secure Hash Standard (SHS) (Revision)*. U.S. Department of Commerce. <https://doi.org/10.6028/NIST.FIPS.180-4>
- Paar, C., dan Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer. <https://doi.org/10.1007/978-3-642-04101-3>
- Pratama, R., Kurniawan, A., dan Hidayat, T. (2023). Evaluasi performa algoritma hash dalam identifikasi integritas file pada sistem forensik digital berbasis web. *Jurnal Teknologi dan Sistem Informasi*, 9(1), 12–23. <https://doi.org/10.25126/jtsi.2023.91.312>
- Pressman, R. S. (2010). *Software Engineering: A Practitioner's Approach* (7th ed.). McGraw-Hill.
- Prayudi, Y., dan Ashari, A. (2022). Pengembangan sistem verifikasi integritas file digital berbasis web menggunakan algoritma hash. *Jurnal Informatika*, 16(1), 33–42. <https://doi.org/10.26555/jifo.v16i1.a23541>

- Riadi, I., Umar, R., dan Bintoro, A. T. (2021). Analisis forensik digital pada bukti elektronik menggunakan metode hash MD5. *Jurnal Rekayasa Sistem dan Teknologi Informasi*, 5(3), 411–419. <https://doi.org/10.29207/resti.v5i3.3177>
- Rivest, R. (1992). *The MD5 Message-Digest Algorithm (RFC 1321)*. Internet Engineering Task Force. <https://www.ietf.org/rfc/rfc1321.txt>
- Stallings, W. (2022). *Cryptography and Network Security: Principles and Practice* (8th ed.). Pearson.
- Stevens, M., Bursztein, E., Karpman, P., Albertini, A., dan Markov, Y. (2017). The first collision for full SHA-1. *Advances in Cryptology – CRYPTO 2017*, Lecture Notes in Computer Science, 10401, 570–596. https://doi.org/10.1007/978-3-319-63688-7_19
- Wang, X., dan Yu, H. (2005). How to break MD5 and other hash functions. *Advances in Cryptology – EUROCRYPT 2005*, Lecture Notes in Computer Science, 3494, 19–35. https://doi.org/10.1007/11426639_2
- Yunanri, W., Budi, A., dan Harjono. (2021). Implementasi algoritma SHA-256 untuk keamanan data pada sistem informasi. *Jurnal Ilmu Komputer dan Sistem Informasi*, 9(2), 78–86. <https://doi.org/10.24912/jiksi.v9i2.11234>