EVALUASI KEAMANAN FITUR UNGGAH BERKAS PADA SITUS WEB DAN IMPLEMENTASI MODUL UNGGAH AMAN BERBASIS VALIDASI BERLAPIS UNTUK MENCEGAH SERANGAN BERKAS BERBAHAYA

Aolia Ikhwanudin[1], Tubagus Toifur[2], Agianto Syamhalim[3], Muhamad Yusuf [4], Ibnu Mas'ud[5], Ferdi Ahyana Yusri [6], Muhamad Jihad Ardiansyah[7]
[1,2,3,4] Informatika, Institut Teknologi Tangerang Selatan
Jl. Raya Serpong No.Kav.9, Lengkong Karya, Serpong Utara, South Tangerang City, Banten
[5,6,7] Informatika, Universitas Islam Negeri Sultan Maulana Hasanuddin Banten
Jl. Syekh Moh. Nawawi Albantani, Kemanisan, Kec. Curug, Kota Serang, Banten
aikwanudin@itts.ac.id

ABSTRAK

Fitur unggah berkas merupakan komponen penting dalam aplikasi web modern, namun sering menjadi titik masuk serangan ketika validasi sisi server tidak diterapkan secara memadai. Permasalahan utama yang dihadapi adalah masih banyaknya aplikasi web yang mengizinkan unggahan berkas tanpa pembatasan ekstensi, tipe MIME, dan isi berkas, sehingga memungkinkan berkas berbahaya dieksekusi di server. Penelitian ini bertujuan untuk mengevaluasi keamanan fitur unggah berkas yang rentan serta mengukur efektivitas modul unggah aman berbasis validasi berlapis. Metode yang digunakan adalah pendekatan eksperimental dengan melakukan pengujian pra-uji dan pasca-uji pada prototipe aplikasi web. Modul aman menerapkan pembatasan ukuran berkas, daftar putih ekstensi, validasi MIME sisi server, pemeriksaan tanda tangan berkas (magic bytes), penamaan acak, penyimpanan aman di luar webroot, hashing SHA-256, serta pencatatan audit. Pengujian dilakukan menggunakan enam berkas uji yang terdiri dari tiga berkas berbahaya dan tiga berkas valid. Hasil pengujian menunjukkan bahwa pada tahap pra-uji seluruh berkas berhasil diunggah, sedangkan pada tahap pasca-uji seluruh berkas berbahaya berhasil ditolak dan berkas valid tetap diterima. Temuan ini membuktikan bahwa validasi berlapis secara signifikan meningkatkan keamanan fitur unggah berkas.

Kata kunci : *keamanan web, unggahan berkas tanpa batasan, validasi sisi server, MIME & tanda tangan, pencatatan audit.*

1. PENDAHULUAN

Perkembangan aplikasi web yang semakin kompleks telah mendorong kebutuhan akan fitur interaktif sebagai bagian dari layanan digital, salah satunya adalah fitur unggah berkas. Fitur ini secara luas digunakan untuk mendukung berbagai kebutuhan pengguna, seperti pengiriman dokumen, unggah gambar, pertukaran data, dan penyimpanan berkas online. Fitur unggah berkas merupakan komponen penting dalam meningkatkan fleksibilitas dan fungsionalitas aplikasi web modern di sektor pendidikan, pemerintahan, dan industri.

Namun, meskipun memiliki manfaat tersebut, fitur unggah berkas juga dikenal sebagai salah satu permukaan serangan yang paling sering dieksploitasi oleh penyerang. Dalam aplikasi web modern, mekanisme unggah konten yang dihasilkan pengguna umumnya diproses di sisi server atau ditampilkan kepada pengguna lain dan administrator layanan. Kondisi ini membuka peluang bagi penyisipan berkas berbahaya yang dirancang untuk mengeksploitasi kerentanan dalam aplikasi klien dan server. Eksploitasi yang berhasil dapat memiliki implikasi serius bagi keamanan dan privasi aplikasi web serta penggunanya [11].

Serangan terhadap fitur unggah berkas umumnya terjadi akibat proses validasi berkas yang lemah, seperti tidak adanya pembatasan jenis berkas, batasan ukuran berkas, atau pemeriksaan konten berkas yang komprehensif. Kelemahan keamanan ini memungkinkan penyerang mengunggah berkas berbahaya yang mengandung kode berbahaya yang berpotensi dieksekusi langsung oleh server. Ketika berkas tersebut berhasil dieksekusi, penyerang dapat melakukan berbagai aktivitas berbahaya, seperti menanam web shell, memodifikasi sistem, merusak aplikasi web, dan menyebarkan malware ke pengguna lain [1]. Malware yang disisipkan melalui mekanisme unggahan berkas berbahaya dapat merusak sistem serta mengakses, memodifikasi, atau mencuri data pada target [2].

Selain itu, sistem deteksi malware yang hanya mengandalkan identifikasi tipe berkas juga rentan terhadap serangan berkas polyglot, yaitu berkas yang memiliki struktur valid untuk lebih dari satu format. Dalam kondisi ini, proses analisis dapat menjadi tidak lengkap, seperti pada kasus berkas polyglot JPG+JAR yang hanya dikenali sebagai JPG sehingga muatan berbahaya terlewatkan [14]. Bahkan aplikasi web yang telah menerapkan pemeriksaan berbasis spoofing tetap berpotensi disusupi oleh berkas polyglot, yang dapat memicu serangan seperti Cross-Site Scripting (XSS) dan Remote Code Execution (RCE) [12].

Penelitian ini bertujuan untuk menganalisis tingkat kerentanan fitur unggah berkas pada aplikasi web yang tidak dilengkapi mekanisme validasi yang memadai, serta mengevaluasi efektivitas penerapan modul unggah aman berbasis validasi berlapis dalam

mendeteksi dan mencegah unggahan berkas berbahaya yang berpotensi dieksekusi oleh server. Selain itu, penelitian ini bertujuan untuk membandingkan kondisi sistem sebelum dan sesudah penerapan mekanisme pengamanan guna menilai peningkatan keamanan yang dihasilkan tanpa mengganggu proses unggah berkas yang sah.

Untuk mencapai tujuan tersebut, penelitian ini dilakukan dengan melakukan analisis dan pengujian terhadap fitur unggah berkas menggunakan pendekatan eksperimental. Penelitian ini mencakup pengujian terhadap kondisi sistem tanpa validasi serta setelah penerapan mekanisme pengamanan. Pendekatan ini didasarkan pada temuan penelitian sebelumnya yang menunjukkan bahwa kerentanan unggahan berkas tanpa batasan dapat menyebabkan eksekusi kode jarak jauh ketika aplikasi hanya mengandalkan pemeriksaan ekstensi berkas [4]. Selain itu, studi pemetaan dan evaluasi pengujian keamanan menekankan pentingnya penerapan kombinasi kontrol keamanan dan pengujian sistematis pada aplikasi web [7]–[12], serta penggunaan validasi metadata dan tipe MIME untuk membedakan berkas sah dan berkas berbahaya [5].

2. TINJAUAN PUSTAKA

2.1. Unrestricted Berkas Upload

Fitur unggah berkas pada aplikasi web dapat menjadi celah keamanan yang serius apabila tidak dilengkapi dengan mekanisme validasi yang memadai. Serangan yang memanfaatkan kelemahan ini dikenal sebagai Unrestricted Berkas Upload, yaitu kondisi ketika aplikasi web mengizinkan pengguna mengunggah berkas tanpa pembatasan yang efektif terhadap jenis, ukuran, maupun konten berkas. Kerentanan ini berpotensi mengancam aspek kerahasiaan, integritas, dan ketersediaan sistem.

Aplikasi web yang hanya mengandalkan pemeriksaan ekstensi berkas sangat rentan terhadap penyalahgunaan, karena ekstensi dapat dengan mudah dimanipulasi. Kondisi tersebut memungkinkan berkas berbahaya masuk ke dalam sistem dan berpotensi dieksekusi oleh server, sehingga membuka peluang terjadinya pengambilalihan sistem dan penyalahgunaan sumber daya.

2.2. Web Shell

Web shell merupakan salah satu bentuk muatan berbahaya yang sering digunakan dalam serangan unggah berkas. Web shell memungkinkan penyerang menjalankan perintah pada server melalui antarmuka berbasis web. Dengan adanya web shell, penyerang dapat mengakses, memodifikasi, maupun menghapus berkas dan direktori, serta mempertahankan akses jangka panjang ke dalam sistem.

Meskipun ukurannya relatif kecil, web shell memiliki kemampuan yang sangat fleksibel dan berbahaya. Keberhasilan unggah web shell menunjukkan bahwa mekanisme validasi unggah berkas pada aplikasi web tidak diterapkan secara optimal, sehingga sistem menjadi rentan terhadap serangan lanjutan.

2.3. Validasi Tipe Berkas

Validasi tipe berkas merupakan salah satu langkah penting dalam mengamankan fitur unggah berkas. Ekstensi berkas yang tertera pada nama berkas tidak selalu mencerminkan struktur dan isi berkas yang sebenarnya. Penyerang dapat memanipulasi ekstensi untuk menyamarkan berkas berbahaya agar terlihat seperti berkas yang sah.

Oleh karena itu, diperlukan mekanisme validasi tambahan yang mampu memeriksa karakteristik internal berkas, seperti tipe MIME dan struktur berkas. Dengan melakukan validasi terhadap konten berkas, sistem dapat mengurangi risiko berkas berbahaya lolos melalui mekanisme unggah berkas.

2.4. Serangan Berkas Polyglot

Selain serangan berbasis pemalsuan ekstensi, terdapat serangan yang lebih kompleks, yaitu serangan berkas polyglot. Berkas polyglot dirancang memiliki struktur yang valid untuk lebih dari satu format sekaligus, sehingga dapat melewati berbagai mekanisme validasi yang umum digunakan.

Berkas jenis ini dapat dikenali sebagai berkas yang sah oleh sistem, namun tetap mengandung muatan berbahaya yang dapat dieksekusi dalam konteks tertentu. Serangan polyglot berpotensi dimanfaatkan untuk menyisipkan skrip berbahaya dan menembus kebijakan keamanan unggah berkas, yang selanjutnya dapat mengarah pada serangan seperti Cross-Site Scripting dan Remote Code Execution.

3. METODE

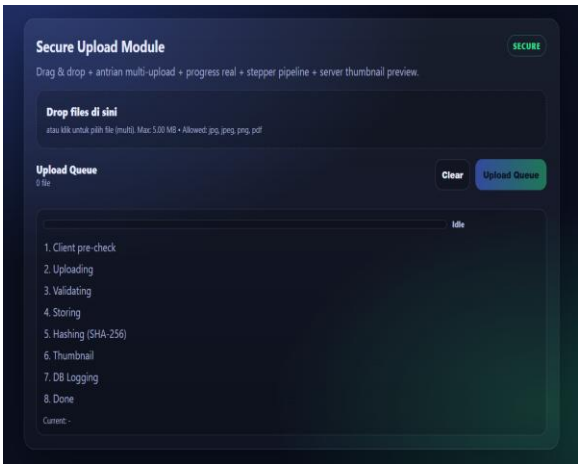
Metode penelitian ini dikembangkan untuk mengevaluasi keamanan fitur unggah berkas pada aplikasi web melalui pengujian sebelum dan setelah implementasi mekanisme validasi. Penelitian ini menggunakan pendekatan eksperimental pada prototipe situs web yang memiliki dua kondisi: (1) modul rentan tanpa validasi untuk menangkap risiko awal, dan (2) modul aman yang menerapkan kontrol validasi berlapis. Seri penelitian meliputi pengujian pra-implementasi, implementasi validasi, dan pengujian pasca-implementasi, seperti yang dijelaskan dalam deskripsi berikut:

3.1. Desain Pengujian

Penelitian ini menggunakan pendekatan eksperimental dengan menciptakan prototipe situs web yang terdiri dari dua kondisi: (1) modul rentan tanpa validasi dan (2) modul aman terbaru yang menerapkan validasi berlapis dan peningkatan fitur antarmuka (drag-and-drop, antrian unggah multi, progres real-time, dan pipa stepper). Kedua modul menerima masukan berkas dari pengguna melalui halaman unggah dan memprosesnya sesuai dengan mekanisme yang diterapkan. Lingkungan pengujian dilakukan pada server pengujian lokal/terisolasi agar tidak

mempengaruhi sistem produksi. Konfigurasi lingkungan yang digunakan dalam studi ini adalah Windows 11 dengan XAMPP (Apache 2.4.x, PHP 8.2.x, MariaDB 10.4.x) dan browser Google Chrome. Direktori penyimpanan unggahan ditempatkan di luar direktori webroot (misalnya, storage/uploads) dengan izin akses terbatas, dan ekstensi GD diaktifkan untuk tujuan pratinjau (thumbnail). Model ancaman mengasumsikan bahwa penyerang dapat mengakses titik akhir unggahan dan mencoba mengunggah berkas berbahaya; fokus penelitian terbatas pada kontrol validasi dan penyimpanan berkas di sisi server, bukan pada aspek otentikasi/otorisasi pengguna.

Jenis berkas yang digunakan dalam pengujian terdiri dari enam berkas sampel, yaitu tiga berkas berbahaya (index.php, IPV6.pptx, dan shell.jpg yang disamarkan sebagai gambar) dan tiga berkas valid (download.jpeg, images.jpeg, dan cv_ferdi_ahyana_yusri.pdf). Pemilihan berkas gambar sebagai salah satu sampel didasarkan pada fakta bahwa malware dapat masuk ke sistem melalui berbagai media, termasuk berkas gambar [3]. Enam sampel dipilih untuk mewakili pola serangan umum, seperti penyisipan skrip berbahaya, penggunaan berkas di luar daftar putih, dan muatan yang disamarkan, sekaligus mewakili berkas yang sering diunggah oleh pengguna.

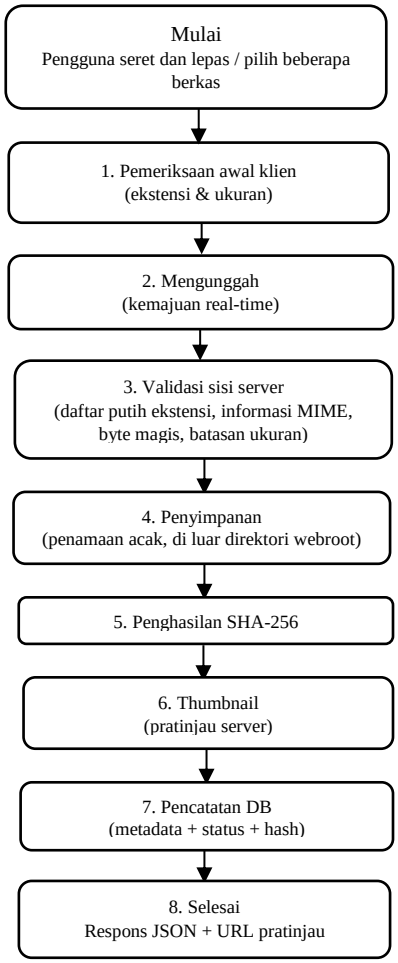


Gambar 1. Antarmuka Modul Unggah Aman dengan drag-and-drop, antrian unggah multi, kemajuan real-time, dan pipa stepper.

3.2. Tahap Pra-Uji

Pengujian awal dilakukan pada modul yang rentan untuk mengidentifikasi celah keamanan dalam kondisi awal. Pada tahap ini, sistem menerima unggahan tanpa batasan ekstensi, tanpa verifikasi tipe MIME, dan tanpa batasan ukuran berkas, sehingga perilaku sistem saat menerima berkas berbahaya dan valid dapat diamati. Langkah-langkah meliputi:

a. Mengunggah enam jenis berkas uji menggunakan modul yang rentan (upload_vulnerable.php).
b. Mengamati apakah berkas diterima atau ditolak oleh sistem.
c. Memeriksa folder penyimpanan untuk memastikan bahwa berkas berbahaya disimpan.
d. Mencatat hasil pengujian dalam basis data upload_log untuk dokumentasi.



Gambar 2. Alur pemrosesan berlapis dan validasi dalam Modul Unggah Aman (pemeriksaan awal klien, validasi server, penyimpanan, hashing SHA-256, thumbnail, dan pencatatan database).

3.3. Implementasi Mekanisme Validasi

Pada tahap kedua, mekanisme validasi diterapkan pada modul aman untuk mencegah berkas berbahaya diunggah ke server. Validasi yang diterapkan mengikuti prinsip pertahanan berlapis, yang menggabungkan beberapa kontrol sehingga kegagalan satu kontrol tidak langsung menyebabkan berkas berbahaya lolos. Validasi yang diterapkan meliputi:

a. Pemeriksaan Pra-Unggah Klien (Sisi Klien)
Sebelum mengunggah, browser memeriksa ekstensi dan ukuran berkas sehingga berkas yang jelas tidak kompatibel dapat ditolak langsung di sisi pengguna (mempercepat umpan balik), tetapi keputusan akhir tetap dibuat di server.
b. Validasi Sisi Server (Kontrol Utama)
Server menerapkan daftar putih ekstensi (jpg, jpeg, png, pdf), memverifikasi MIME menggunakan berkasinfo (finfo_berkas), dan memeriksa tanda tangan/magic bytes untuk mengurangi risiko

berkas yang disamarkan (polyglot). Dengan mengidentifikasi format berkas secara akurat , risiko konten berbahaya lolos melalui mekanisme unggah berkas dapat dikurangi secara signifikan [13].
Mekanisme validasi ini sesuai dengan praktik terbaik OWASP dan temuan penelitian terkait verifikasi MIME dan byte magic [3], [5].

c. Batasan Ukuran Berkas
Server membatasi ukuran berkas maksimum menjadi 5 MB untuk mencegah penyalahgunaan kapasitas server dan serangan denial-of-service.
d. Penyimpanan Aman dan Penamaan Acak
Berkas yang lolos validasi disimpan dengan nama acak (penamaan acak) dan ditempatkan di direktori yang tidak menjalankan skrip (disarankan di luar direktori webroot) dengan izin minimal (privilese terendah).
e. Hashing SHA-256
Setelah berkas disimpan, sistem menghitung nilai hash SHA-256 untuk memastikan integritas berkas, membantu mendeteksi perubahan, dan mendukung audit/forensik.
f. Pembuatan Thumbnail (Pratinjau Thumbnail Server)
Sistem menghasilkan thumbnail untuk berkas yang valid (gambar/PDF) menggunakan proses server-side sehingga pratinjau tidak bergantung pada manipulasi klien. Thumbnail dapat berupa pratinjau gambar atau ikon/pratinjau halaman pertama untuk PDF.
g. Pencatatan Database
Sistem mencatat metadata (nama asli, nama yang disimpan, ukuran, MIME, waktu, status, alasan penolakan jika ada), termasuk nilai hash SHA-256 dan lokasi thumbnail untuk berkas yang diterima.
h. Respons terstruktur dan umpan balik antarmuka
Server mengembalikan respons terstruktur (misalnya, JSON) yang digunakan antarmuka untuk menampilkan kemajuan, status setiap tahap (validasi, penyimpanan, hashing, thumbnail, pencatatan), dan hasil akhir per berkas dalam antrian unggahan.

Mekanisme ini dipilih dengan mengacu pada rekomendasi OWASP mengenai mitigasi unggahan berkas tanpa batasan dan prinsip pertahanan berlapis, serta penambahan kontrol integritas (hash) dan jejak audit (pencatatan) untuk meningkatkan traceability [3].

3.4. Tahap Pasca-Uji

Tahap ketiga adalah pengujian ulang (pasca-pengujian) menggunakan berkas yang sama seperti pada tahap pra-pengujian, tetapi pada sistem yang telah dilengkapi dengan validasi. Tujuan tahap ini adalah untuk mengevaluasi efektivitas mekanisme validasi dan memastikan bahwa berkas yang valid masih dapat diunggah tanpa mengganggu fungsi sistem.

Langkah-langkah pengujian meliputi:

a. Mengunggah semua berkas uji ke modul
b. aman (upload_secure.php).
c. Mengamati apakah berkas diterima atau ditolak berdasarkan aturan validasi.
d. Memverifikasi bahwa hanya berkas yang valid yang disimpan di direktori unggahan.
e. Memeriksa catatan log di database untuk melihat perubahan status dan output sistem.

Tahap ini bertujuan untuk mengevaluasi efektivitas mekanisme validasi dan membandingkan hasilnya dengan kondisi awal.

3.5. Teknik Analisis Data

Analisis dilakukan dengan membandingkan hasil pra-tes dan pasca-tes menggunakan tiga indikator:

a. Persentase berkas berbahaya yang dapat diunggah
b. Ketepatan sistem dalam mengenali berkas valid dan tidak valid
c. Penurunan risiko setelah validasi diterapkan

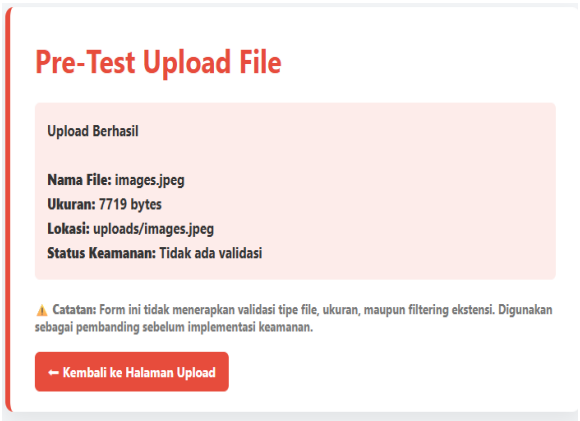
Data diperoleh dari tabel log basis data (misalnya MySQL), hasil unggahan yang ditampilkan di antarmuka, dan verifikasi isi direktori penyimpanan. Semua data dianalisis secara deskriptif dengan membandingkan hasil pra-tes dan pasca-tes untuk menunjukkan efektivitas langkah-langkah mitigasi yang diterapkan.

4. HASIL DAN PEMBAHASAN

Bagian ini menyajikan hasil uji sebelum dan setelah penerapan mekanisme validasi pada fitur unggahan berkas, serta analisis efektivitas mitigasi. Pengujian dilakukan menggunakan enam jenis berkas uji, terdiri dari tiga berkas berbahaya dan tiga berkas valid.

4.1. Hasil Uji Coba

Uji coba pra-tes dilakukan pada modul unggah tanpa validasi, di mana sistem menerima semua berkas yang diunggah tanpa memeriksa ekstensi, tipe MIME, atau ukuran berkas. Hasil menunjukkan bahwa semua berkas uji, baik yang berbahaya maupun yang valid, berhasil diunggah dan disimpan di direktori unggahan.



Gambar 3. Contoh output pra-uji pada modul yang rentan saat berkas uji diterima

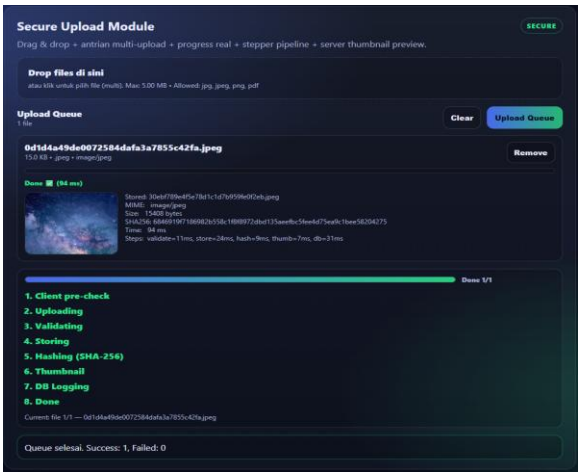
Tabel 1. Hasil pra-uji pada sistem yang rentan

Nama Berkas Asli	Mode Unggah
download.jpeg	Rentan
index.php	Rentan
IPV6.pptx	Rentan
(cv) ferdi ahyana yusri.pdf	Rentan
images.jpeg	Rentan
shell.jpg	Rentan

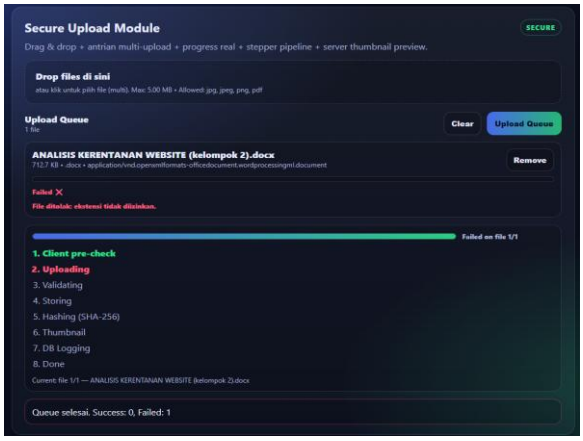
Hasil ini menunjukkan bahwa sistem rentan karena berkas berbahaya dapat masuk ke server tanpa hambatan. Kondisi ini berpotensi mengancam keamanan server (misalnya, eksekusi kode, defacement), penyalahgunaan sumber daya, atau penyisipan konten berbahaya ke dalam direktori penyimpanan.

4.2. Hasil Pasca Uji

Pada fase pasca-pengujian, sistem diuji menggunakan versi terbaru dari Modul Unggah Aman, yang menerapkan alur pemrosesan bertahap sesuai dengan stepper pada antarmuka (pemeriksaan pra-unggah klien, unggah, validasi, penyimpanan, hashing, thumbnail, dan pencatatan database). Berkas diuji dengan aturan validasi berlapis di sisi server- : daftar putih ekstensi, verifikasi MIME dan tanda tangan/byte magic, batas ukuran 5 MB, penamaan acak, penyimpanan aman (disarankan di luar direktori webroot), perhitungan hash SHA-256, pembuatan thumbnail untuk berkas yang diterima, dan pencatatan hasil proses untuk setiap berkas ke database. Pendekatan ini bertujuan untuk memastikan bahwa berkas berbahaya ditolak, sementara berkas valid tetap dapat diproses dengan umpan balik yang jelas di sisi pengguna.



Gambar 4. Hasil pasca-pengujian pada Modul Unggah Aman saat berkas valid berhasil diterima.



Gambar 5. Hasil uji coba pada Modul Unggah Aman saat berkas berbahaya ditolak oleh mekanisme validasi berdasarkan aturan ekstensi dan tipe MIME.

Tabel 2. Hasil uji coba pada sistem aman

Nama Berkas Asli	Status	Alasan
download.jpeg	Diterima	Tidak ada
index.php	Ditolak	Ekstensi tidak diizinkan
IPV6.pptx	Ditolak	Ekstensi tidak diizinkan
(cv) ferdi ahyana yusri.pdf	Diterima	Tidak ada
images.jpeg	Diterima	Tidak
shell.jpg	Ditolak	MIME tidak sesuai dengan daftar putih

Hasil pengujian pasca-tes menunjukkan bahwa semua berkas berbahaya berhasil ditolak sesuai dengan aturan validasi yang diterapkan, sementara berkas yang valid diterima dan disimpan dengan aman. Berkas *index.php* dan *IPV6.pptx* ditolak karena ekstensi mereka tidak termasuk dalam daftar putih, sementara *shell.jpg* ditolak karena ketidaksesuaian tipe MIME meskipun menggunakan ekstensi gambar. Sistem juga menghasilkan artefak audit berupa hash SHA-256, metadata penyimpanan, dan thumbnail untuk berkas yang diterima, serta mencatat semua aktivitas unggahan dalam basis data sebagai *jejak audit* yang mencakup alasan penolakan setiap berkas.

Tabel 3. Contoh metadata dan artefak audit untuk berkas valid yang diterima oleh Modul Unggah Aman.

Nama Berkas Asli	Nama Berkas yang Disimpan	MIME	Hash SHA-256 (dipotong)
images.jpeg	95842920ae123cb68f283c9326101293.jpeg	image/jpeg	bb6f81774840334ab93699e7e8523cee72a52bdcca1921b783…

Nama Berkas Asli	Nama Berkas yang Disimpan	MIME	Hash SHA-256 (dipotong)
download.jpeg	e2c671afeaad44c1b05ac7fdf3decc38.jpeg	image/jpeg	1a5de2027cc890f38664cfadcb8e62780cbacee7f144c1926e…
(cv) Ferdi Ahyana Yusri.pdf	28b1b3b1bb6f3eefe265c11abf0….pdf	application/pdf	8d896486e1ba764bf082471c9e6034b485077fc53e8cf85d4c…

Nilai hash ditampilkan sebagai ringkasan (contoh) untuk tujuan pelaporan; dalam implementasi asli, nilai hash disimpan secara lengkap (64 karakter heksadesimal). Metadata lain seperti ekstensi, ukuran, dan tipe MIME masih dicatat dalam database untuk audit dan pelacakan insiden.

4.3. Analisis Perbandingan Pra-Uji dan Pasca-Uji

Perbandingan hasil dari dua fase pengujian menunjukkan perbedaan yang signifikan dalam kemampuan sistem untuk menangani unggahan berkas berbahaya. Pada fase pra-uji, seluruh berkas berbahaya yang diuji (3/3) berhasil lolos dan disimpan di server, termasuk berkas dengan ekstensi berbahaya dan ekstensi ganda. Temuan ini menunjukkan bahwa sistem tanpa mekanisme validasi yang memadai sangat rentan terhadap serangan unggahan berkas berbahaya.

Sebaliknya, pada fase pasca-uji, seluruh berkas berbahaya yang diuji berhasil ditolak (0/3 lolos) melalui penerapan mekanisme pertahanan berlapis di sisi server. Penolakan ini terjadi karena kombinasi beberapa kontrol keamanan, yaitu penyaringan ekstensi menggunakan daftar putih, verifikasi tipe MIME dan pemeriksaan tanda tangan/byte magis untuk mendeteksi berkas yang disamarkan sebagai gambar [5], [6], serta penerapan batasan ukuran berkas maksimum sebesar 5 MB untuk mengurangi potensi penyalahgunaan sumber daya. Selain itu, penerapan penamaan berkas secara acak dan penyimpanan berkas di lokasi yang aman juga mengurangi kemungkinan serangan yang bergantung pada nama atau jalur tertentu serta mencegah eksekusi berkas di direktori unggahan. Untuk berkas yang diterima, sistem menghasilkan nilai hash SHA-256, *thumbnail* yang dibuat di sisi server, dan entri log di database. Mekanisme ini meningkatkan traceability (*audit trail*) sehingga setiap aktivitas unggah berkas dapat diverifikasi dan dilacak secara forensik.

4.4. Diskusi Keamanan dan Rekomendasi Penguatan

Validasi berlapis yang diterapkan (daftar putih ekstensi, verifikasi MIME dan byte magic, batasan ukuran, penggantian nama acak, penyimpanan aman) efektif dalam memblokir pola serangan umum dalam skenario pengujian. Pada versi terbaru, penambahan hashing SHA-256 dan pencatatan terstruktur meningkatkan aspek non-fungsional seperti auditabilitas dan integritas data. Dari perspektif pengguna, antrian unggahan multi, kemajuan real-time, dan pipa kerja bertahap membantu transparansi proses dan mengurangi kebingungan saat berkas ditolak (misalnya, karena ketidaksesuaian MIME). Namun, risiko sisa tetap ada, terutama pada berkas ambigu (polyglot) dan kerentanan dalam perpustakaan pemrosesan gambar/PDF. Sebagai lapisan tambahan, mekanisme Content Disarm and Reconstruction (CDR) dapat dipertimbangkan untuk menetralkan konten aktif dalam dokumen berisiko tinggi [9].

4.5. Batasan Penelitian dan Pekerjaan Lebih Lanjut

Studi ini memiliki beberapa batasan yang perlu dipertimbangkan. Pengujian dilakukan menggunakan sejumlah sampel berkas yang terbatas, yaitu enam berkas, dan dijalankan pada lingkungan server uji lokal, sehingga hasilnya tidak sepenuhnya mewakili kompleksitas serangan di lingkungan produksi dengan beban yang lebih tinggi dan variasi ancaman.

Untuk meningkatkan ketahanan sistem, beberapa langkah direkomendasikan, antara lain mengintegrasikan pemindaian malware menggunakan antivirus server-side (misalnya ClamAV), menerapkan pembatasan laju dan deteksi anomali untuk mencegah unggahan brute-force, serta mengisolasi proses pembuatan thumbnail dalam lingkungan sandbox guna menghindari eksekusi parser yang rentan. Selain itu, pengujian otomatis berupa pengujian unit dan integrasi pada aturan validasi juga direkomendasikan untuk menjaga konsistensi kebijakan keamanan seiring perkembangan sistem [8], [10].

5. KESIMPULAN

Studi ini menunjukkan bahwa fitur unggahan berkas dalam aplikasi web memiliki risiko keamanan tinggi jika tidak dilengkapi dengan validasi server-side. Pada tahap pra-pengujian, semua berkas berbahaya berhasil diunggah, menunjukkan sistem yang rentan. Setelah menerapkan Modul Unggah Aman dengan validasi berlapis, semua berkas berbahaya berhasil ditolak (3/3 menjadi 0/3), sementara berkas valid tetap diproses dengan benar. Modul ini juga meningkatkan integritas dan jejak audit melalui hashing SHA-256, pembangkitan thumbnail di sisi server, dan pencatatan database.

Hasil pengujian membuktikan bahwa pendekatan *pertahanan berlapis* pada fitur unggahan berkas tidak hanya efektif dalam mencegah serangan *unggahan berkas berbahaya*, tetapi juga mempertahankan fungsionalitas untuk pengguna sah dan meningkatkan kesiapan sistem untuk audit dan pelacakan insiden. Oleh karena itu, validasi sisi server yang komprehensif dan pengujian keamanan berkelanjutan direkomendasikan sebagai kontrol minimum untuk

aplikasi web.

DAFTAR PUSTAKA

[1] M. A. A. Hilmi and R. K. Yunan, "Pengujian Keamanan Fitur Upload Berkas Pada Sistem Aplikasi Web," Jurnal Informatika: Jurnal Pengembangan IT (JPIT), vol. 7, no. 1, pp. 37–42, Jan. 2022. doi: 10.30591/jpit.v7i1.3336.

[2] J. C. Chandra and M. Aldiansyah, "Rancang Bangun Sistem Deteksi Malware dalam Berkas Gambar Menggunakan Analisis Metadata dan VirusTotal," Bit (Fakultas Teknologi Informasi Universitas Budi Luhur), vol. 22, no. 1, pp. 70–76, Apr. 2025. doi: 10.36080/bit.v22i1.4009.

[3] OWASP Foundation, "Berkas Upload Cheat Sheet," 2025. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Berkas_Upload_Cheat_Sheet.html . [Accessed: Dec. 26, 2025].

[4] Y. Chen, Y. Li, Z. Pan, Y. Lu, J. Chen, and S. Ji, "URadar: Discovering Unrestricted Berkas Upload Vulnerabilities via Adaptive Dynamic Testing," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 1251–1266, 2024, doi: 10.1109/TIFS.2023.3335885 .

[5] A. Dubettier, T. Gernot, E. Giguet, and C. Rosenberger, "Berkas Type Identification Tools for Digital Investigations: A Comparative Study," Forensic Science International: Digital Investigation, vol. 46, Art. no. 301574, 2023, doi : 10.1016/j.fsidi.2023.301574 .

[6] SD Khudhur and HA Jeiad, "A Content-based Berkas Identification Dataset: Collection, Construction, and Evaluation," Karbala International Journal of Modern Science, vol. 8, no. 2, pp. 63–70, 2022, doi: 10.33640/2405-609X.3222 .

[7] M. Aydos, Ç. Aldan, E. Coşkun, and A. Soydan, "Security testing of web applications: A systematic mapping of the literature," Journal of King Saud University – Computer and Information Sciences, vol. 34, no. 9, pp. 6775–6792, 2022, doi: 10.1016/j.jksuci.2021.09.018

[8] V. Casola, A. De Benedictis, C. Mazzocca, and V. Orbinato, "Secure software development and testing: A model-based secure methodology for managing web application security," Computers & Security, vol. 137, Art. no. 103639, 2024, doi: 10.1016/j.cose.2023.103639 .

[9] R. Dubin, "Content Disarm and Reconstruction of PDF Berkass," IEEE Access, vol. 11, pp. 38399–38416, 2023, doi: 10.1109/ACCESS.2023.3267717 .

[10] H. Oz, G.S. Tuncay, A. Aris, A. Kharraz, and S. Uluagac, "Beyond the Upload Button: A 10-Year Retrospective on Security Issues Within Berkas Uploads," IEEE Communications Magazine, vol. 63, no. 2, pp. 104–110, 2025, doi: 10.1109/MCOM.001.2400253 .

[11] P. Wichmann, A. Groddecke, and H. Federrath, "BerkasUploadChecker: Detecting and sanitizing malicious berkas uploads in web applications at the request level," in Proc. 17th Int. Conf. Availability, Reliability and Security (ARES), 2022, Art. no. 66, pp. 1–10, doi: 10.1145/3538969.3538999 .

[12] H. Oz, A. Acar, A. Aris, G. S. Tuncay, A. Kharraz, and S. Uluağaç, "(In)Security of Berkas Uploads in Node.js," in Proc. ACM Web Conf. (WWW '24), 2024, pp. 1573–1584, doi: 10.1145/3589334.3645342.

[13] L. Koch, S. Oesch, A. Sadovnik, B. Weber, A. Chaulagain, M. Dixon, J. Dixon, M. Huettel, C. Watson, J. Hartman, and R. Patulski, "On the Abuse and Detection of Polyglot Berkass," in Proc. of the ACM on Web Conference (WWW '25), 2025, pp. 4810–4822, doi: 10.1145/3696410.3714814.

[14] Koch, S. Oesch, A. Chaulagain, M. Adkisson, S. Erwin, and B. Weber, "Toward the Detection of Polyglot Berkass," in Proc. 15th Workshop on Cyber Security Experimentation and Test (CSET '22), 2022, pp. 120–128, doi: 10.1145/3546096.3546106.