PEMERINTAH DAERAH DAERAH ISTIMEWA YOGYAKARTA

DINAS KOMUNIKASI DAN INFORMATIKA

Jalan Brigjen Katamso, Yogyakarta 55152; Telepon (0274) 373444; Faksimile (0274) 374496
Pos-el diskominfo@jogjaprov.go.id; Laman diskominfo.jogjaprov.go.id

4 Februari 2025

Nomor : B/500.12.13/1516/D11
Sifat : Biasa
Lampiran : 1 (satu) berkas
Hal : Standar Teknis dan Pengujian Keamanan Aplikasi SPBE Pemda DIY

Yth. Paniradya Pati, Sekretaris DPRD, Inspektur, Kepala Satuan Pol.PP, Kepala Badan, Kepala Dinas, Kepala Biro, Direktur RS, Kepala Unit Pelaksana Teknis, Kepala Satuan Pendidikan
di Lingkungan Pemda DIY

Diberitahukan dengan hormat, bahwa dalam periode Januari s.d. Desember 2024, Dinas Komunikasi dan Informatika DIY telah menindaklanjuti 191 insiden kerentanan aplikasi/kerentanan sistem. Adapun *trend* insiden berupa:

1. *web defacement*, yaitu peretas berhasil mengubah tampilan sebuah situs web misalnya menampilkan promosi judi online sebanyak 51 kasus;
2. *Cross Site Scripting* (XSS), yaitu pelaku menyisipkan kode berbahaya ke dalam halaman web misalnya memunculkan *pop up* sebanyak 35 kasus;
3. *security misconfiguration*, yaitu ketika pengaturan keamanan aplikasi atau server tidak diatur dengan benar sehingga membuka celah kerentanan untuk penyerang sebanyak 25 kasus;
4. *sensitive data exposure*, kebocoran data sensitif seperti *credential* akun, dan data pribadi sebanyak 18 kasus; dan
5. SQL *Injection*, yaitu pelaku memasukkan perintah berbahaya ke dalam kolom input (seperti formulir *login* atau pencarian) di sebuah situs web sebanyak 18 kasus;

Kerentanan tersebut selain menyebabkan gangguan layanan, juga berpotensi penyerang dapat mengambil alih akses, menghapus atau mengubah data sehingga

menurunkan reputasi instansi dan tingkat kepercayaan pengguna. Kerentanan tersebut pada umumnya disebabkan karena:

1. adanya kerentanan dan kesalahan konfigurasi sistem yang dimanfaatkan penyerang untuk masuk ke dalam sistem;
2. penggunaan bahasa pemrograman dan *library* yang tidak diperbaharui;
3. penggunaan *framework* yang tidak standar atau masih *native*;
4. pengaturan hak akses pengguna dan penggunaan *password* yang tidak memenuhi prinsip keamanan;
5. pembangunan aplikasi belum menggunakan prinsip *secure by design*;
6. kurangnya kesadaran dan pemahaman keamanan informasi di kalangan pengembang, pengelola sistem ataupun pengguna;
7. proses pengembangan dan pengujian aplikasi yang tidak memadai; dan
8. kelalaian dalam pembaruan dan pemeliharaan sistem.

Berdasarkan Peraturan Gubernur DIY Nomor 67 Tahun 2022 tentang Sistem Pemerintahan Berbasis Elektronik dan Peraturan Badan Siber dan Sandi Negara Nomor 4 Tahun 2021 tentang Pedoman Manajemen Keamanan Informasi Sistem Pemerintahan Berbasis Elektronik dan Standar Teknis dan Prosedur Keamanan Sistem Pemerintahan Berbasis Elektronik, dimohon dalam pelaksanaan pembangunan, pengembangan dan/atau pemeliharaan aplikasi/sistem informasi untuk memperhatikan hal-hal berikut:

1. sebelum dilakukan pengembangan aplikasi, PIC OPD dan pengembang aplikasi wajib bergabung ke Discord channel Pemda DIY pada alamat: https://s.id/discord-diskominfo-diy untuk melakukan koordinasi awal terkait pemenuhan kesesuaian rencana pengembangan sistem, pengujian keamanan dan *deployment* aplikasi;
2. pada pembangunan aplikasi baru, pengembangan maupun pemeliharaan aplikasi/sistem informasi wajib mengikuti standar keamanan aplikasi sebagaimana terlampir.
3. pengembangan wajib dilakukan menggunakan sistem repositori kode sumber https://git.jogjaprov.go.id;
4. arsitektur pengembangan wajib menggunakan 3-tier-architecture, yaitu *front-end (FE), back-end (BE),* dan *database* yang dijalankan pada environment terpisah;

5. pembangunan aplikasi *front-end (FE)* wajib menggunakan model *compiled based framework* seperti *React JS, Vue JS, dan Flutter*;
6. pengembangan aplikasi *back-end (BE)* dapat menggunakan *framework* seperti *Laravel, CodeIgniter (CI), Node JS,* dan *Go*;
7. database yang digunakan pada aplikasi wajib menggunakan *MySQL* atau *PostgreSQL LTS (Long Term Support)*;
8. pengembangan *website* profil instansi wajib menggunakan *template* yang disediakan Dinas Komunikasi dan Informatika DIY;
9. *tools* dan *framework* yang digunakan wajib versi *Long Term Support (LTS)*;
10. tidak diperkenankan menggunakan *plain text query* pada aplikasi *Back-end (BE)* untuk melakukan pemrosesan *database*. *Query* ke dalam database wajib dibangun menggunakan *Object Relational Mapper (ORM)* pada masing-masing *framework* yang digunakan, atau *tools third party* yang mendukung seperti *Sequelize*;
11. tidak diperkenankan melakukan pembangunan/pengembangan tanpa menggunakan *framework* yang direkomendasikan untuk menjaga *sustainability sistem*;
12. memastikan seluruh *input field* sudah disanitasi dan hanya menerima tipe dan *value input* yang diizinkan;
13. menyediakan fitur perubahan *password* yang diintegrasikan dengan sistem pembuatan *password* Dinas Komunikasi dan Informatika DIY.
14. memastikan penerapan konfigurasi yang aman untuk menghindari kebocoran data pribadi (*PII leaks*) seperti penerapan *robots.txt*, batasan hak akses, *data masking*, enkripsi data pribadi, dll;
15. aplikasi dapat beroperasi pada *environment production* jika sudah melakukan pengujian UAT (*User Acceptance Test*) dari sisi pemilik dan lolos pengujian keamanan aplikasi oleh Dinas Komunikasi dan Informatika DIY. Pengujian keamanan aplikasi meliputi:
    a. *vulnerability assessment*;
    b. pengecekan kesesuaian dengan standar keamanan aplikasi;
16. apabila masih ditemukan kerentanan atau ketidaksesuaian dengan standar keamanan, tim pengembang aplikasi wajib melakukan perbaikan atau penyesuaian;

17. merespon notifikasi keamanan yang dikirimkan oleh Dinas Komunikasi dan Informatika DIY melalui persuratan elektronik atau nara hubung instansi.

Dinas Komunikasi dan Informatika DIY akan melakukan patroli keamanan siber secara berkala untuk memastikan aplikasi masih memenuhi persyaratan keamanan aplikasi di atas. Selanjutnya kami akan melakukan tindakan pengamanan yang dianggap perlu apabila terjadi insiden keamanan siber, termasuk memutuskan akses aplikasi (*take down*) dari jaringan, sesuai dengan kebijakan layanan *data center* https://peladen.jogjaprov.go.id/static/kebijakan_layanan.pdf.

Demikian, atas perhatian dan kerjasamanya diucapkan terima kasih.

Kepala Dinas Komunikasi dan Informatika,

Hari Edi Tri Wahyu Nugroho, S.I.P., M.Si.

Lampiran Surat Kepala Dinas Komunikasi dan Informatika
Nomor : B/500.12.13/1516/D11
Tanggal : 4 Februari 2025

CHECKLIST PENGUJIAN KEAMANAN APLIKASI BERBASIS WEB

DINAS KOMUNIKASI DAN INFORMATIKA

DAERAH ISTIMEWA YOGYAKARTA

A. Informasi Umum

Nama Aplikasi	
Alamat (URL)	
Instansi	
PIC Aplikasi	
Pengembang	
Discord channel	
Jenis Aplikasi	Web Dinas / Sistem Informasi
Sifat	Publik / Internal Pemda DIY / Internal Perangkat Daerah

B. Penggunaan Teknologi

		Versi	Status
Bahasa Pemrograman /Framework Frontend			
Bahasa Pemrograman /Framework Backend			
Database			
Web Server			
Operating System			

C. Spesifikasi Server

CPU	
Memory	
Storage	

D. Pengujian Keamanan

No	Kriteria Keamanan	Acuan/Penjelasan	Hasil Pengujian	Keterangan/Tangkapan Layar
AUTENTIKASI				
1.	Menggunakan manajemen kata sandi untuk proses autentikasi.	Aplikasi meminta kata sandi/password pada saat login aplikasi untuk proses autentikasi pemilik akun	☐	

2.	Menerapkan verifikasi kata sandi pada sisi server.	Aplikasi melakukan verikasi kata sandi pada sisi server, salah satunya dapat menggunakan fungsi hash	☐	
3.	Mengatur jumlah karakter, kombinasi jenis karakter, dan masa berlaku dari kata sandi.	Aplikasi menerapkan konfigurasi pada manajemen kata sandi terkait jumlah karakter, kombinasi jenis karakter, dan masa berlaku dari kata sandi	☐	
4.	Mengatur jumlah maksimum kesalahan dalam pemasukan kata sandi.	Aplikasi membatasi percobaan login pada aplikasi	☐	
5.	Mengatur mekanisme pemulihan kata sandi.	Aplikasi memiliki fungsi untuk pergantian password	☐	
6.	Menjaga kerahasiaan kata sandi yang disimpan melalui mekanisme kriptografi.	Aplikasi menerapkan fungsi hash untuk keamanan password yang tersimpan pada database aplikasi	☐	
7.	Menggunakan jalur komunikasi yang diamanankan untuk proses autentikasi	Aplikasi menerapkan protokol TLS untuk keamanan jalur komunikasi	☐	
MANAJEMEN SESI				
1.	Menggunakan pengendalian sesi untuk proses manajemen sesi.	aplikasi menggunakan pengendalian sesi untuk mengatur interaksi antara user ke server	☐	
2.	Menggunakan pengendalian sesi yang disediakan oleh kerangka kerja aplikasi/3rd party yang mendukung.		☐	
3.	Mengatur pembuatan dan keacakan token sesi	Aplikasi harus memastikan token sesi yang diproduksi acak,	☐	

	yang dihasilkan oleh pengendali sesi.	hal ini agar tidak dapat dilakukan serangan bruteforce untuk masuk sebagai orang lain		
4.	Mengatur kondisi dan jangka waktu habis sesi.	Aplikasi harus memastikan pemberlakuan batas waktu masa hidup sesi	☐	
5.	Validasi dan pencantuman session id.		☐	
6.	Pelindungan terhadap lokasi dan pengiriman token untuk sesi terautentikasi.	Aplikasi harus memastikan sesi yang dikirim melalui jalur yang aman	☐	
7.	Pelindungan terhadap duplikasi dan mekanisme persetujuan pengguna.		☐	
AKSES KONTROL				
1.	Menetapkan otorisasi pengguna untuk membatasi kontrol akses.	Aplikasi menetapkan kontrol akses pada setiap role pengguna (super admin, admin, user). Kontrol akses yang dimaksud adalah otorisasi pada fitur/fungsi/data aplikasi sesuai role yang pengguna aplikasi	☐	
2.	Mengatur peringatan terhadap bahaya serangan otomatis apabila terjadi akses yang bersamaan atau akses yang terus-menerus pada fungsi.	Aplikasi memberikan peringatan bila akun pengguna melakukan akses pada device yang berbeda pada waktu yang sama dan akses yang terus menerus. Hal ini berkaitan dengan manajemen sesi yang diatur pada setiap pengguna.	☐	

3.	Mengatur antarmuka pada sisi administrator.	Antarmuka sisi administrator atau Admin UI (User Interface) dirancang untuk administrator mengelola dan mengontrol sistem, aplikasi, atau situs web. Antarmuka ini berbeda dari antarmuka pengguna biasa dan menyediakan fitur-fitur canggih dan kontrol yang diperlukan untuk mengawasi dan mengkonfigurasi berbagai aspek sistem.	☐	
4.	Mengatur verifikasi kebenaran token ketika mengakses data dan informasi yang dikecualikan.	Aplikasi menerapkan fitur otorisasi menggunakan token (saat masuk/login ke aplikasi atau saat akan mengakses informasi dikecualikan) Informasi yang dikecualikan dapat mengacu pada UU KIP atau aplikasi telah menentukan data apa saja yang masuk dalam Informasi Dikecualikan	☐	
VALIDASI INPUT				
1.	Menerapkan fungsi validasi input pada sisi server.	Memverifikasi data dari pengguna sebelum diproses lebih lanjut, untuk memastikan sesuai format, tipe, dan kriteria validitas. Mengurangi risiko injeksi SQL, XSS, dan manipulasi data lainnya.	☐	
2.	Menerapkan mekanisme penolakan input jika terjadi kesalahan validasi.	Otomatis menolak data yang tidak memenuhi kriteria validasi untuk mencegah serangan SQL injection, XSS, dan manipulasi data lainnya.	☐	

3.	Memastikan runtime environment aplikasi tidak rentan terhadap serangan validasi input.	Mengamankan lingkungan runtime dari serangan seperti injeksi SQL dan XSS untuk melindungi aplikasi.	☐	
4.	Melakukan validasi positif pada seluruh input. (public access)	Memastikan semua data memenuhi kriteria valid dan format sebelum diproses untuk mencegah serangan SQL injection, XSS, dan manipulasi data.	☐	
5.	Melakukan filter terhadap data yang tidak dipercaya.	Mencegah data berbahaya masuk ke sistem untuk menghindari injeksi SQL, XSS, dan serangan berbasis data lainnya.	☐	
6.	Menggunakan fitur kode dinamis.	Mengeksekusi kode secara dinamis untuk fleksibilitas aplikasi, dengan manajemen risiko keamanan yang ketat.	☐	
7.	Melakukan pelindungan terhadap akses yang mengandung konten skrip.	Mengamankan aplikasi dari serangan XSS dengan kebijakan kontrol sumber-sumber.	☐	
8.	Melakukan pelindungan dari serangan injeksi berbasis data.	Mencegah injeksi SQL dan manipulasi data dengan parameterized queries dan validasi input.	☐	
KRIPTOGRAFI				
1.	Menggunakan algoritma kriptografi, modul kriptografi, protokol kriptografi, dan kunci kriptografi.	**Algoritma Kriptografi**: serangkaian instruksi atau prosedur matematis yang digunakan untuk mengenkripsi (mengubah pesan asli menjadi bentuk yang tidak terbaca) dan mendekripsi	☐	

		(mengembalikan pesan terenkripsi ke bentuk aslinya) pesan. Algoritma kriptografi dapat berupa simetris atau asimetris. **Modul Kriptografi**: perangkat lunak atau perangkat keras yang mengimplementasikan algoritma kriptografi. Modul ini biasanya digunakan untuk melakukan operasi enkripsi dan dekripsi, serta untuk mengelola kunci kriptografi. **Protokol Kriptografi**: serangkaian aturan dan prosedur yang digunakan dalam komunikasi untuk memastikan keamanan data. Protokol ini mencakup cara untuk mengamankan pengiriman pesan, autentikasi pengguna, dan manajemen kunci, di antara hal lainnya. Contoh protokol kriptografi termasuk HTTPS untuk komunikasi web yang aman, SSH untuk akses jarak jauh yang aman, dan TLS/SSL untuk keamanan data dalam pengiriman email dan aplikasi lainnya. **Kunci Kriptografi**: nilai rahasia yang digunakan dalam algoritma kriptografi untuk mengenkripsi dan mendekripsi pesan.		
2	Melakukan autentikasi data yang dienkripsi.	Autentikasi data yang dienkripsi dapat menggunakan modul/	☐	

		libarary yang telah diimplementasikan, misalnya memastikan bahwa penggunaan algoritma kriptografi hash function SHA-1 sesuai dengan hasil outputnya maupun ketika pengecekan kesamaan.		
3	Menerapkan manajemen kunci kriptografi.	Manajemen kunci kriptografi diantaranya memastikan bahwa kunci kriptografi dibuat, disimpan, dikontrol dan dapat dihancurkan dengan aman. Misalkan penggunaan kriptografi asimetris untuk SSH, dipastikan public key dan private key dibuat oleh user terotorisasi, disimpan di tempat yang aman, diketahui oleh pemilik kunci saja dan dapat dihancurkan dengan aman (penggunaan time based).	☐	
4	Membuat angka acak yang menggunakan generator angka acak kriptografi.	Generator angka acak bertujuan agar kunci kriptografi maupun algoritma kriptografi tidak dapat ditebak dengan mudah oleh penyerang. Ada beberapa teknik dalam menghasilkan angka acak kriptografi, salah satunya menggunakan **Entropi**: gerakan mouse, aktivitas jaringan, interupsi perangkat keras, waktu sistem, dan sebagainya.	☐	
ERROR DAN PENCATATAN LOG				
1.	Mengatur konten pesan yang	Aplikasi membatasi informasi yang	☐	

	ditampilkan ketika terjadi kesalahan.	ditampilkan ketika terjadi kesalahan		
2.	Menggunakan metode penanganan eror untuk mencegah kesalahan terprediksi dan tidak terduga serta menangani seluruh pengecualian yang tidak ditangani.	aplikasi harus menerapkan error handling untuk membantu mendeteksi dan menangani kesalahan yang tidak terduga	☐	
3.	Tidak mencantumkan informasi yang dikecualikan dalam pencatatan log.	Event logs can contain sensitive data and personally identifiable information. Appropriate privacy protection measures should be taken (see 5.34).	☐	
4.	Mengatur cakupan log yang dicatat untuk mendukung upaya penyelidikan ketika terjadi insiden.	aplikasi harus menyimpan log terkait informasi tidak terbatas pada : - user ID - aktivitas sistem - tanggal, waktu dan detail even yang relevan - identitas device, lokasi - network address dan protokol	☐	
5.	Mengatur perlindungan log aplikasi dari akses dan modifikasi yang tidak sah.	Aplikasi harus memprotect log agar tidak dapat diakses oleh pihak yang tidak berhak untuk menghapus maupun memanipulasi (modifikasi) log tersebut	☐	
6.	Melakukan enkripsi pada data yang disimpan untuk mencegah injeksi log.	sql injection? klo iya ketika aplikasi rentan terhadap sql injection, maka penyerang dapat menyerang database sehingga dapat mengambil database tersebut, ketika database tidak di	☐	

		enkripsi, maka penyerang dapat dengan mudah melihat isi informasi database tersebut		
7.	Melakukan sinkronisasi sumber waktu sesuai dengan zona waktu dan waktu yang benar.	keseluruhan aplikasi harus menggunakan sinkronisasi waktu yang sama untuk memudahkan proses analisis dan mengkorelasikan ketika terjadi insiden	☐	
PROTEKSI DATA				
1.	Melakukan identifikasi dan penyimpanan salinan informasi yang dikecualikan.		☐	
2.	Melakukan pelindungan dari akses yang tidak sah terhadap informasi yang dikecualikan yang disimpan sementara dalam aplikasi.		☐	
3.	Melakukan pertukaran, penghapusan, dan audit informasi yang dikecualikan.		☐	
4.	Melakukan penentuan jumlah parameter.		☐	
5.	Memastikan data tersimpan dengan aman.		☐	
6.	Menentukan metode untuk menghapus dan mengekspor data sesuai permintaan pengguna.		☐	

7.	Membersihkan memori setelah tidak diperlukan.		☐	
KEAMANAN KOMUNIKASI				
1.	Menggunakan komunikasi terenkripsi.	Menerapkan mekanisme enkripsi untuk mengamankan jalur komunikasi dalam jaringan antara client dan server aplikasi. Beberapa penerapan komunikasi jaringan yang aman antara lain: secure web browsing (HTTPS), email transmission (SMTPS, POP3S, IMAPS) atau komunikasi jaringan yang aman lainnya	☐	
2.	Mengatur koneksi masuk dan keluar yang aman dan terenkripsi dari sisi pengguna.	Menerapkan sistem keamanan jaringan yang menyaring dan mengontrol lalu lintas jaringan masuk dan keluar berdasarkan aturan keamanan yang telah ditentukan.	☐	
3.	Mengatur jenis algoritma yang digunakan dan alat pengujiannya.	Menerapkan algoritma kriptografi yang kuat untuk mengamankan komunikasi jaringan dan menguji penerapan algoritma kriptografi tersebut untuk mendeteksi celah keamanan yang mungkin masih ada pada sistem/aplikasi	☐	
4.	Mengatur aktivasi dan konfigurasi sertifikat elektronik yang diterbitkan oleh penyelenggara sertifikasi elektronik.	Menerapkan prosedur aktivasi dan konfigurasi sertifikat elektronik yang aman	☐	
PENGENDALIAN KODE BERBAHAYA				
1.	Menggunakan analisis kode dalam	Proses yang bertujuan untuk mendeteksi,	☐	

	kontrol kode berbahaya.	mengidentifikasi, dan mengeliminasi potensi ancaman yang tersembunyi dalam kode program		
2.	Memastikan kode sumber aplikasi dan pustaka tidak mengandung kode berbahaya dan fungsionalitas lain yang tidak diinginkan.	Organisasi/pengembang dapat secara efektif memastikan bahwa kode sumber dan pustaka yang digunakan dalam aplikasi mereka aman dan bebas dari kode berbahaya serta fungsionalitas yang tidak diinginkan.	☐	
3.	Mengatur izin terkait fitur atau sensor terkait privasi.	Mengatur izin yang tepat untuk fitur atau sensor yang terkait dengan privasi pengguna	☐	
4.	Mengatur pelindungan integritas.	Melindungii integritas aplikasi dengan memastikan bahwa file aplikasi tidak dimodifikasi tanpa izin.	☐	
5.	Mengatur mekanisme fitur pembaruan.	Menyediakan mekanisme untuk memperbarui aplikasi Anda dengan aman dan memastikan pengguna selalu memiliki versi terbaru.	☐	
LOGIKA BISNIS				
1.	Memproses alur logika bisnis dalam urutan langkah dan waktu yang realistis.	Alur logika bisnis secara umum dapat dilihat pada dokumen perencanaan aplikasi yang membuat bisnis proses aplikasi tersebut.	☐	
2.	Memastikan logika bisnis memiliki batasan dan validasi.	Pembatasan logika bisnis dan validasi dapat dicek pada flowchart atau sejenisnya pada setiap modul maupun gambaran umum	☐	

		aplikasi, misalkan pada modul izin cuti, maka dipastikan logika bisnis berjalan sesuai bisnis proses yang disusun, misalnya hanya user yang memiliki sisa cuti saja yang dapat mengajukan cuti.		
3.	Memonitor aktivitas yang tidak biasa.	Monitoring aktivitas yang tidak biasa dapat menggunakan alat bantu pencatatan (log), misalkan terdapat catatan aktivitas user melakukan ekskalasi menjadi administrator (tindakan tidak sah).	☐	
4.	Membantu dalam kontrol antiotomatisasi.	Kontrol antiotomatisasi ini untuk mencegah beberapa serangan pada aplikasi yang menggunakan alat bantu otomatis, seperti serangan brute force attack pada satu halaman login.	☐	
5.	Memberikan peringatan ketika terjadi serangan otomatis atau aktivitas yang tidak biasa.	Peringatan adanya serangan otomatis atau aktivitas yang tidak biasa dapat menggunakan alat bantu seperti SIEM dan memastikan bahwa pencatatan log aktif pada web server.	☐	
Manajemen FILE				
1.	Mengatur jumlah file untuk setiap pengguna dan kuota ukuran file yang diunggah.	Aplikasi menetapkan batas jumlah file dan ukuran file yang dapat diupload oleh pengguna aplikasi.	☐	
2.	Melakukan validasi file sesuai dengan tipe konten yang diharapkan.	Aplikasi menerapkan kontrol jenis file (ekstensi file) yang yang dapat diupload oleh pengguna aplikasi,	☐	

		sesuai dengan kebutuhan data yang diupload		
3.	Melakukan pelindungan terhadap metadata input dan metadata file.	Aplikasi mengonfigurasi perlindungan (enkripsi/hashing) metadata input dan file aplikasi. metadata: karakteristik data terutama isi, kwalitas, kondisi dan cara perolehannya)	☐	
4.	Melakukan pemindaian file yang diperoleh dari sumber yang tidak dipercaya.	Aplikasi menerapkan fitur pemindaian pada file yang diupload yang diperoleh dari sumber yang tidak dipercaya. Pemindaian dapat dilakukan dengan mengintegrasikan aplikasi ke Firewall/SIEM/IDS/IPS atau aplikasi scanner (malware/virus) opensource seperti virustotal etc.	☐	
5.	Melakukan konfigurasi server untuk mengunduh file sesuai ekstensi yang ditentukan.	Aplikasi menerapkan kontrol jenis file (ekstensi file) yang yang dapat diunduh oleh pengguna aplikasi, sesuai dengan kebutuhan data yang diunduh	☐	
KEAMANAN API DAN WEB SERVICE				
1.	Melakukan konfigurasi layanan web.	Konfigurasi layanan web untuk memastikan keamanan dan efisiensi dalam pengelolaan API.	☐	
2.	Memverifikasi uniform resource identifier API tidak menampilkan informasi yang	Memastikan URL API tidak mengekspos informasi Informasi sensitif seperti struktur direktori, nama file, parameter yang tidak	☐	

	berpotensi sebagai celah keamanan.	tervalidasi, kunci API, token sesi. atau detail implementasi sebaiknya tidak terlihat dalam URI.		
3.	Membuat keputusan otorisasi.	Mempertimbangkan akses pengguna berdasarkan identitas, peran, dan izin yang diberikan, serta memverifikasi setiap permintaan dengan token atau kredensial yang valid sebelum memberikan akses ke sumber daya yang diminta.	☐	
4.	Menampilkan metode RESTful hypertext transfer protocol apabila input pengguna dinyatakan valid.	Memastikan bahwa hanya pengguna dengan izin yang sesuai yang dapat melakukan operasi tertentu.	☐	
5.	Menggunakan validasi skema dan verifikasi sebelum menerima input.	Memastikan bahwa validasi skema telah diterapkan dan diverifikasi sebelum menerima input dari pengguna.	☐	
6.	Menggunakan metode pelindungan layanan berbasis web.	Pelindungan layanan berbasis web (web services security) melibatkan berbagai teknik dan metode untuk melindungi layanan web dari ancaman dan serangan	☐	
7.	Menerapkan kontrol antiotomatisasi.	Memastikan kontrol untuk mencegah serangan otomatis yang dapat merusak sistem	☐	
KEAMANAN KONFIGURASI				
1.	Mengonfigurasi server sesuai rekomendasi server aplikasi dan kerangka kerja	Mengoptimalkan konfigurasi server untuk keamanan dan kinerja aplikasi, dengan mengikuti panduan	☐	

	aplikasi yang digunakan.	resmi vendor dan praktik terbaik.		
2.	Mendokumentasi, menyalin konfigurasi, dan semua dependensi.	Dokumentasi dan replikasi konfigurasi dan dependensi untuk konsistensi dan manajemen infrastruktur yang efektif.	☐	
3.	Menghapus fitur, dokumentasi, sampel, dan konfigurasi yang tidak diperlukan.	Mengurangi kompleksitas dan meningkatkan keamanan dengan menghilangkan fitur dan dokumentasi yang tidak relevan.	☐	
4.	Memvalidasi integritas aset jika aset aplikasi diakses secara eksternal.	Memastikan data dan sumber daya aplikasi tetap aman dari manipulasi saat diakses dari luar.	☐	
5.	Menggunakan respons aplikasi dan konten yang aman.	Memastikan respons aplikasi dan konten terlindungi dari ancaman keamanan, serta akurat dan tepat bagi pengguna.	☐	

Pengujian ke-	Tanggal Pengujian	Hasil Pengujian	Nama dan Paraf Penguji
1			
2			
3			

KESIMPULAN HASIL PENGUJIAN: