p-ISSN : 2406-7733
e-ISSN : 2597-9922

PENGEMBANGAN APLIKASI ANDROID UNTUK *VULNERABILITY ASSESMENT* MENGGUNAKAN API OWASP ZAP

Aulia Putri[1], Sigit Auliana [2], Gelard Untirtha Pratama [3]
Program Studi Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Bina Bangsa
Jl Raya Raya Serang-Jakarta KM. 03 No. 1.B Kota Serang Banten.
E-mail: *auliaputr018@gmail.com[1], pasigit@gmail.com[2], tirthagelard@gmail.com[3]

Abstrak - *Vulnerability Assesment* adalah langkah penting dalam keamanan siber yang bertujuan untuk mengidentifikasi, mengevaluasi, dan memperbaiki kerentanan situs *web*. Studi ini berfokus pada penerapan analisis kerentanan untuk mengidentifikasi potensi kerentanan pada situs *web* yang dapat dieksploitasi oleh penyerang. Metodologi yang digunakan meliputi pemindaian otomatis dengan alat keamanan siber dan analisis manual untuk menilai risiko yang terkait dengan setiap kerentanan yang ditemukan. Hasil dari penelitian ini menunjukkan bahwa banyak *website* yang masih rentan terhadap serangan seperti *SQL injection*, *cross-site scripting* (XSS), dan konfigurasi yang tidak aman. Hasil penilaian kerentanan ini memberikan wawasan untuk mengembangkan strategi mitigasi yang efektif dan perbaikan yang direkomendasikan untuk meningkatkan keamanan situs *web*. Selain itu, penelitian ini juga menyoroti pentingnya mewaspadai ancaman siber yang terus berkembang dan secara teratur memperbarui sistem keamanan untuk melindungi data dan informasi dari akses yang tidak sah. Dengan demikian, analisis kerentanan tidak hanya bertindak sebagai tindakan pencegahan tetapi juga sebagai alat untuk meningkatkan kepercayaan pengguna terhadap situs *web*. Dalam menghadapi serangan siber yang terus berkembang, termasuk munculnya berbagai jenis serangan baru yang semakin kompleks dan sulit dideteksi, para peneliti mencoba merancang sebuah aplikasi inovatif untuk menyederhanakan proses evaluasi kerentanan. Dengan menggunakan teknologi terkini, pengguna dapat secara proaktif mengidentifikasi potensi kerentanan pada sistem mereka sebelum dieksploitasi oleh pihak yang tidak bertanggung jawab. Aplikasi ini tidak hanya mendeteksi kerentanan, tetapi juga memberikan analisis mendalam dan saran tindakan yang dapat diambil untuk meningkatkan keamanan sistem. Oleh karena itu, aplikasi ini diharapkan dapat menjadi alat yang komprehensif dalam membantu individu, kelompok, dan perusahaan dalam melindungi aset digital mereka dari ancaman siber yang terus berkembang.

Kata Kunci: API OWASP ZAP, Keamanan Siber, Serangan Siber, *Vulnerability Assesment*, *Website*

I. PENDAHULUAN

Keamanan siber di Indonesia akhir-akhir ini menjadi isu yang cukup hangat. Data BSSN (Badan Siber dan Sandi Negara) menunjukkan bahwa serangan siber di Indonesia masih marak terjadi dalam tiga tahun terakhir, meskipun menunjukkan tren penurunan anomali serangan siber. Pada tahun 2021 jumlah anomali serangan siber di Indonesia sebanyak 1.637.973.022 anomali. Pada tahun 2022 jumlah anomali serangan siber di Indonesia mengalami penurunan menjadi 976.429.996 anomali. Pada tahun 2023 jumlah anomali serangan siber di Indonesia menjadi 403.990.813 anomali. Sementara itu, trend positif juga ditunjukkan dari laporan insiden serangan web defacement dalam tiga tahun terakhir. Dimana di tahun 2021 terjadi 5.940 kasus *web defacement*, di tahun 2022 terjadi 2.348 kasus, dan di tahun 2023 terjadi 189 kasus *web defacement* (Negara, 2022) (Direktorat Operasi Keamanan Siber & Negara, 2022) (Negara, 2023).

Maraknya *Web Deface* ini disebabkan karena lemahnya sistem keamanan sehingga menjadi target serangan, dengan celah seperti SQL *Injection*, *Remote Code Execution* (RCE), XSS untuk mengakses *server* secara ilegal dan mengubah isi situs. Meski menunjukkan *trend* positif dalam 3 tahun terakhir, namun jumlah CVE *(Common Vulnerabilities exposures)* mengalami peningkatan dalam 3 tahun terakhir. Dimana di tahun 2021 terdapat 20.161 laporan CVE, di 2022 terdapat 25.059 laporan, dan di 2023 mencapai 28.961 laporan (Negara, 2022) (Direktorat Operasi Keamanan Siber & Negara, 2022) (Negara, 2023). Grafik ini menggambarkan peningkatan kerentanan sistem di Indonesia, sehingga perlu dilakukan penilaian kerentanan secara berkala untuk menguji situs web untuk mencari celah dan menghindari serangan di masa depan. Selain itu, faktor lain yang berkontribusi terhadap meningkatnya frekuensi serangan siber adalah deteksi yang rendah. Penelitian menunjukkan bahwa ketidakmampuan untuk mendeteksi kerentanan berpotensi menyebabkan kerugian akibat serangan siber (Alserhani & Aljared, 2023). Penelitian lain juga menunjukkan bahwa tidak adanya penemuan kerentanan secara dini meningkatkan risiko dimanfaatkan atau menjadi korban serangan siber. Dan organisasi yang tidak melakukan penilaian kerentanan secara teratur cenderung menjadi target utama serangan siber (Ozkan-okay *et al*., 2023). Sehingga dapat disimpulkan bahwa

p-ISSN : 2406-7733
e-ISSN : 2597-9922

ketidakmampuan atau keterlambatan dalam deteksi kerentanan dapat berakibat terjadinya serangan *cyber* yang menyebabkan kerugian.

Common Vulnerability Exposure (CVE) merupakan sistem yang dirancang untuk mengidentifikasi dan mendokumentasikan kerentanan keamanan *cyber* di jaringan dan sistem komputer, setiap CVE memiliki fitur unik yang memungkinkan para profesional untuk secara aman mendeteksi dan merespons kerentanan tertentu dalam jaringan atau sistem. Tujuan dari CVE adalah untuk meningkatkan kesadaran dan berbagi informasi mengenai celah keamanan serta potensi dampak yang dapat ditimbulkan oleh celah tersebut (Tasmiha Khan, 2024). CVE membantu dalam mengidentifikasi kerentanan dengan cepat dan menentukan tingkat keparahan dari setiap kerentanan yang teridentifikasi. Hal ini melindungi bisnis dari kerentanan yang dapat disebabkan oleh standar dan kode keamanan yang berbeda pada berbagai platform. Angka ini menunjukkan bertambahnya kerentanan pada sistem di seluruh dunia, sehingga perlu adanya *vulnerability assessment* secara berkala untuk menguji celah yang ada pada website guna mencegah terjadinya serangan di kemudian hari.

Vulnerability Assessment (VA) merupakan proses pemindaian sistem untuk menemukan kerentanan dan celah yang ada pada sistem suatu website. Kerentanan ini memberikan penyerang *backdoor* untuk menyerang sistem korban. *Vulnerability Assessment* adalah langkah penting dalam keamanan cyber untuk melindungi data dan aset organisasi dari ancaman yang terus berkembang. Tujuan utama *Vulnerability Assesment* yaitu untuk mengidentifikasi kerentanan sistem, memahami risiko yang terkait dengan kerentanan tersebut, dan memberikan mitigasi untuk mengurangi potensi. Manfaatnya meliputi peningkatan keamanan sistem, pemahaman risiko yang lebih baik, dan kemampuan untuk menghentikan masalah sebelum terjadi (Kurniawan & Christianto, 2024).

Serangan *cyber* merupakan ancaman serius di era digital saat ini, dengan potensi untuk merusak data, mengganggu operasi bisnis, dan membahayakan infrastruktur kritis (Yoheswari, n.d.). Berbagai jenis serangan *cyber* yang umum terjadi seperti *security vulnerabilities* antara lain *phishing, ransomware, malware, DDoS (Distributed Denial of Service), man-in-the-middle (MITM), zero-day, injeksi SQL*, dan *social engineering* (Hapsari & Pambayun, 2023). Kerentanan atau *Vulnerability* merupakan kelemahan atau cacat pada suatu sistem, aplikasi, atau jaringan yang dapat dimanfaatkan oleh pihak yang tidak berkepentingan untuk mendapatkan akses yang tidak sah atau merusak data atau sistem. Kerentanan keamanan dapat terjadi karena berbagai alasan, termasuk kesalahan pengkodean, konfigurasi sistem yang salah, dan kelemahan dalam desain sistem (Christina Sari *et al*., 2024).

Beberapa penelitian terdahulu telah dilakukan untuk menciptakan alat dan metodologi untuk menganalisis kerentanan pada website, termasuk OWASP ZAP, Burp Suite, dan sistem lain yang serupa. Alat-alat ini sering kali dirancang untuk menemukan dan menyelidiki kelemahan keamanan pada website dengan akurasi tinggi. Namun, karena sebagian besar solusi ini dirancang untuk platform desktop atau server, penggunaannya memerlukan perangkat tertentu dan tidak selalu memenuhi kebutuhan pengguna dengan mobilitas tinggi. Meskipun OWASP ZAP telah menyediakan API untuk otomatisasi, penerapannya terbatas pada skenario yang membutuhkan perangkat keras tambahan atau konfigurasi teknis yang ekstensif. Oleh karena itu, penelitian ini untuk mengatasi batasan tersebut dengan mengembangkan solusi berbasis mobile yang lebih mudah beradaptasi dan dapat diakses secara langsung melalui smartphone Android.

Penelitian ini bertujuan untuk mempermudah masyarakat umum dalam melakukan penilaian kerentanan dengan menawarkan solusi berbasis mobile yang mudah digunakan dan diakses. Dengan aplikasi yang dirancang untuk perangkat Android, penelitian ini untuk meningkatkan kesadaran akan pentingnya melakukan evaluasi ketahanan secara teratur untuk memastikan keamanan data dan sistem. Selain itu, penelitian ini mendukung upaya deteksi dini kerentanan pada website, yang dimaksudkan untuk membantu mencegah serangan siber yang berpotensi merusak. Penciptaan solusi ini dimaksudkan untuk memungkinkan pengguna, terutama mereka yang tidak memiliki keterampilan teknis yang luas, untuk lebih proaktif dalam memastikan keamanan sistem mereka.

Manfaat dari penelitian ini sangat penting dalam aspek keamanan siber, terutama dalam upaya mengurangi frekuensi dan dampak serangan siber yang semakin meningkat. Dengan melakukan deteksi dini terhadap kerentanan yang ada dalam sistem, penelitian ini memberikan solusi yang dapat diakses dan digunakan oleh siapa saja, tanpa memandang latar belakang teknis mereka. Pendekatan yang sederhana dan mudah dipahami ini memungkinkan individu maupun organisasi, termasuk yang memiliki sumber daya terbatas, untuk secara proaktif mengidentifikasi potensi ancaman sebelum dapat dieksploitasi oleh pihak yang tidak bertanggung jawab.

Dengan kemampuan untuk mendeteksi kerentanan lebih awal, penelitian ini berpotensi mengurangi kerugian finansial dan reputasi yang sering kali dialami akibat serangan siber. Kerugian ini tidak hanya berdampak pada aspek keuangan, tetapi juga dapat merusak kepercayaan pelanggan dan reputasi organisasi di pasar. Oleh karena itu, dengan menyediakan alat dan metode deteksi yang

p-ISSN : 2406-7733
e-ISSN : 2597-9922

efektif, penelitian ini berkontribusi pada penguatan pertahanan siber secara keseluruhan.

Selain itu, penelitian ini juga berfungsi sebagai sarana edukasi bagi masyarakat tentang pentingnya keamanan siber. Dengan meningkatkan kesadaran akan kerentanan yang ada dan cara-cara untuk mendeteksinya, diharapkan lebih banyak orang akan mengambil langkah-langkah preventif untuk melindungi data dan sistem mereka. Secara keseluruhan, manfaat dari penelitian ini tidak hanya terbatas pada pengurangan serangan siber, tetapi juga menciptakan lingkungan digital yang lebih aman dan terlindungi bagi semua pihak.

II. TINJAUAN PUSTAKA

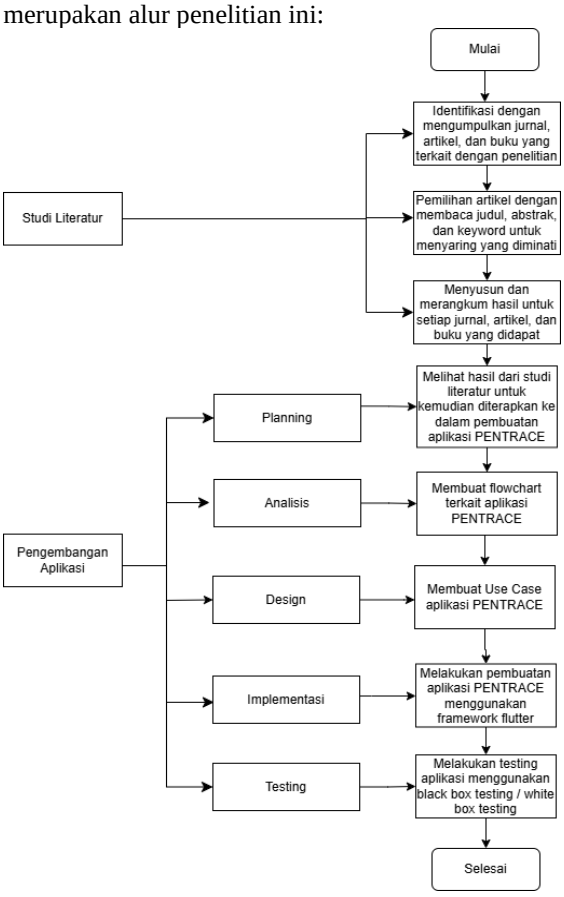
OWASP ZAP digunakan untuk mengevaluasi keamanan pada situs web labscrape.my.id. Hasilnya menunjukkan bahwa ada berbagai celah yang dapat dieksploitasi dengan tingkat keparahan yang berbeda-beda, termasuk *Low* dan *Medium*. Kerentanan yang ditemukan, seperti tidak adanya Token Anti-SCRF, dengan tingkat keparahan sedang. Kerentanan ini dapat menyebabkan serangan *Cross Site Request Forgery* (CSRF) karena kurangnya mekanisme perlindungan token keamanan, yang memungkinkan penyerang untuk mengirimkan formulir secara tidak sah (Hasibuan & Handoko, 2023). Pengembangan platform keamanan online berbasis API dengan penekanan pada keamanan API oleh OWASP. Berdasarkan hasil uji coba, platform ini dapat mendukung hingga 1000 pengguna secara bersamaan dengan tingkat keberhasilan 96,35%. Platform ini dibangun di atas CTF (*Capture The Flag*), yang mencoba mengedukasi melalui virtualisasi kontainer dengan Docker (Idris *et al*., 2022). Penggunaan OWASP ZAP untuk penilaian keamanan Sistem Informasi Akademik Universitas Pancasila. Tujuan dari penelitian ini adalah untuk mendeteksi dan menilai kerentanan keamanan dengan menggunakan standar OWASP Top-10. Temuan pengujian mengungkapkan 19 kerentanan dalam sistem, empat di antaranya ada di OWASP Top-10: Kontrol Akses Rusak, Kesalahan Konfigurasi Keamanan, Komponen yang Rentan dan Usang, dan Kegagalan Integritas Perangkat Lunak dan Data. Berdasarkan temuan ini, tingkat keamanan sistem tergolong sedang, sehingga membutuhkan peningkatan lebih lanjut dari para insinyur sistem.(Kusuma, 2022)

III. METODE PENELITIAN

Beberapa alat penilaian kerentanan, seperti OWASP ZAP, telah dibuat dan biasanya beroperasi pada perangkat desktop atau server. Penelitian sebelumnya berkonsentrasi pada pembuatan teknologi berbasis desktop, dengan sedikit penekanan pada alternatif seluler. Beberapa penelitian telah menunjukkan potensi integrasi API OWASP ZAP, namun implementasinya pada *platform* Android belum mendapat banyak perhatian. Penelitian ini memanfaatkan OWASP ZAP API untuk membuat aplikasi Android yang dirancang untuk memenuhi permintaan pengguna dengan mobilitas tinggi sekaligus mendukung fitur tambahan seperti pemindaian otomatis dan tampilan hasil. Salah satu keunggulannya adalah kemampuannya untuk menjalankan pemeriksaan kerentanan secara langsung di ponsel pintar Android, sehingga tidak memerlukan perangkat tambahan dan memberikan kebebasan yang lebih besar kepada konsumen. Hasil pemindaian disajikan dalam bentuk grafik interaktif, yang membantu pengguna memahami temuannya. Selain itu, perangkat lunak ini menawarkan penilaian kerentanan pada API saat ini seperti REST API dan GraphQL, yang umum digunakan dalam pengembangan aplikasi berbasis cloud. Desain aplikasi juga berisi modul untuk otentikasi OWASP ZAP API dan visualisasi data menggunakan pustaka grafik terbaru, sehingga menghasilkan solusi yang kompleks dan mudah digunakan.

A. Metode Penelitian

Pada bagian metode penelitian penulis menggunakan dua metode untuk melakukan penelitian terkait topik Pengembangan Aplikasi Android untuk *Vulnerability Assesment* Menggunakan API OWASP ZAP. Berikut merupakan alur penelitian ini:

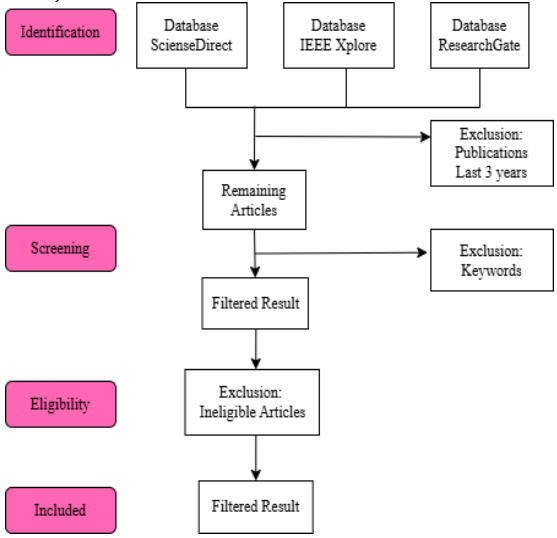


Gambar 1. Alur Penelitian

p-ISSN : 2406-7733
e-ISSN : 2597-9922

1) Studi Literatur

Pada metode pengumpulan data ini untuk mengidentifikasi, mengevaluasi, dan menafsirkan semua penelitian yang berkaitan dengan suatu masalah atau bidang penelitian (Valencia et al., 2022).



Gambar 2. Tahapan Studi Literatur

Perencanaan yang matang adalah langkah awal yang penting saat melakukan penelitian. Peneliti harus menentukan *research question* (RQ) yang berbeda. Pertanyaan ini akan menjadi panduan untuk menemukan informasi yang relevan. Setelah RQ ditentukan, peneliti melanjutkan ke tahap implementasi serta membuat kriteria inklusi dan eksklusi untuk memilih berbagai sumber informasi yang relevan (Ritonga, 2021). Hal ini termasuk mengidentifikasi kata kunci yang terkait dengan RQ dan cara mencarinya di berbagai *database* yang dapat diakses. Setelah itu, prosedur identifikasi dimulai. Pada tahap ini, peneliti mengumpulkan artikel dan sumber yang relevan berdasarkan kriteria yang telah ditetapkan sebelumnya. Peneliti mencari data dari berbagai sumber, termasuk publikasi ilmiah, buku, dan artikel yang mungkin mengandung informasi yang berguna.

Setelah mengumpulkan bahan pustaka, para peneliti memasuki tahap penyaringan (*screening*). Dalam proses ini, peneliti mengevaluasi dan memilih artikel yang memenuhi kriteria relevansi dan kualitas. Hal ini penting untuk memastikan bahwa hanya sumber-sumber yang paling berkualitas yang akan digunakan dalam analisis mendalam (Firdha *et al*., 2021). Selanjutnya, pada tahap eligibilitas, peneliti mengevaluasi setiap artikel yang telah dikumpulkan, memastikan bahwa artikel-artikel tersebut memenuhi standar kualitas dan relevansi yang ditetapkan untuk topik penelitian (Subahan *et al.*, 2021). Terakhir, pada tahap inklusi, peneliti akan memasukkan artikel yang telah memenuhi semua kriteria tersebut untuk dianalisis lebih lanjut. Di sini, peneliti mengintegrasikan dan melaporkan hasil yang diperoleh dari literatur yang dipilih, untuk memberikan gambaran yang lebih jelas dan mendalam tentang topik yang mereka teliti (Putra & Huda, 2021).

2) Pengembangan Aplikasi

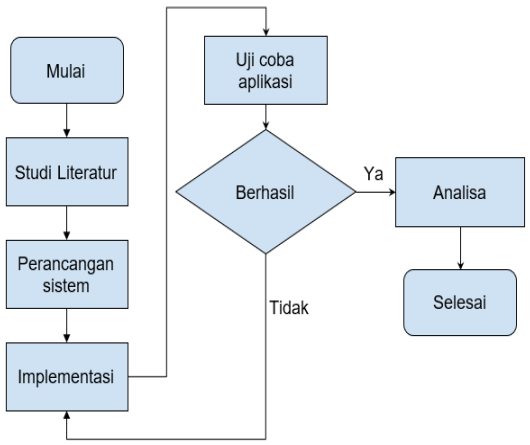
Dalam pengembangan aplikasi ini, penulis menggunakan *framework* Flutter dengan bahasa pemrograman Dart. Untuk *backend* aplikasi, penulis memanfaatkan OWASP ZAP yang dihosting pada server lokal. Konektivitas antara *frontend* dan *backend* dilakukan melalui API yang disediakan oleh OWASP ZAP, dengan sistem *tunneling* ke internet agar API dapat diakses secara publik. Selain itu, untuk mekanisme login dan registrasi akun, penulis menggunakan Google Firebase sebagai database. Berikut adalah gambaran alur pengembangan aplikasi Pentrace:

a. *Planning*

Pada tahap perencanaan ini, penulis menganalisis hasil dari studi literatur yang telah dilakukan untuk mengidentifikasi dan pendekatan yang relevan. Temuan-temuan tersebut akan diterapkan dalam pengembangan aplikasi PENTRACE, sehingga dapat memastikan bahwa aplikasi ini tidak hanya memenuhi kebutuhan pengguna, tetapi juga mengikuti standar dan metodologi yang telah terbukti efektif.

b. Analisis

Pada tahap analisis ini, penulis membuat *flowchart* yang menggambarkan alur kerja aplikasi PENTRACE. *Flowchart* ini bertujuan untuk memvisualisasikan proses-proses utama dalam aplikasi, mulai dari interaksi pengguna hingga pengolahan data di backend. Dengan membuat *flowchart*, penulis dapat mengidentifikasi langkah-langkah yang diperlukan, memudahkan pemahaman alur aplikasi, serta memastikan bahwa semua fungsi yang diinginkan telah dipertimbangkan.



Gambar 3. Alur Pengembangan Aplikasi

c. *Design*

Pada tahap perancangan ini, penulis membuat *user interface* untuk aplikasi PENTRACE. Ini bertujuan untuk mengilustrasikan interaksi antara pengguna dan sistem, serta mendefinisikan berbagai skenario penggunaan yang mungkin terjadi. Dengan membuat *user interface*, penulis dapat

p-ISSN : 2406-7733
e-ISSN : 2597-9922

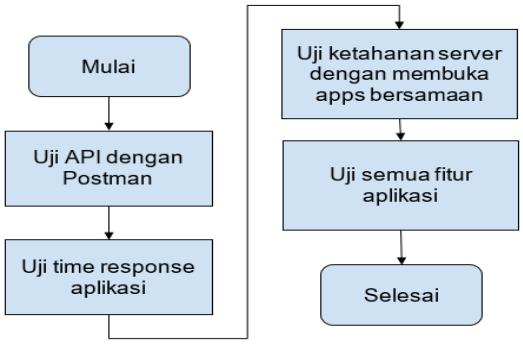
mengidentifikasi gambaran aplikasi yang akan dibuat.

d. Implementasi

Pada tahap implementasi ini, penulis mulai membangun aplikasi PENTRACE menggunakan *framework* Flutter. Flutter dipilih karena kemampuannya untuk membangun aplikasi yang responsif dan berkinerja tinggi di berbagai platform, termasuk iOS dan Android, dengan satu basis kode.

e. *Testing*

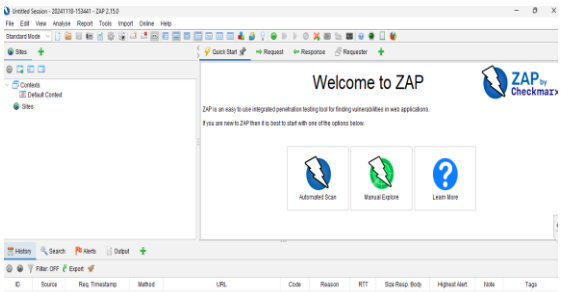
Pada tahap pengujian ini, penulis melakukan pengujian aplikasi PENTRACE menggunakan metode *black box testing* atau *white box testing*. *Black box testing* mengevaluasi fungsionalitas aplikasi dari perspektif pengguna, memastikan semua fitur berfungsi sesuai spesifikasi. Sementara itu, *white box testing* menganalisis bagian internal aplikasi, termasuk logika dan alur kontrol, untuk mengidentifikasi bug dan memastikan semua jalur kode diuji. Metode ini bertujuan untuk memberikan gambaran menyeluruh tentang kinerja dan keandalan aplikasi PENTRACE sebelum diluncurkan kepada pengguna.



Gambar 4. Alur Pengujian Aplikasi

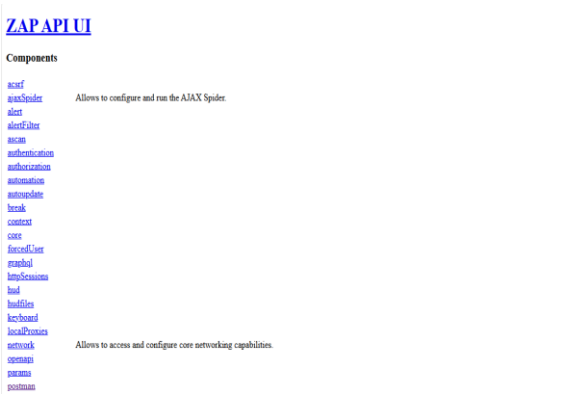
B. Analisis Sistem

OWASP ZAP digunakan untuk melakukan *Penetration Testing* atau memeriksa kerentanan pada website. Terdapat dua fungsi utama dalam OWASP ZAP, yaitu *automated scan* dan *manual scan*.



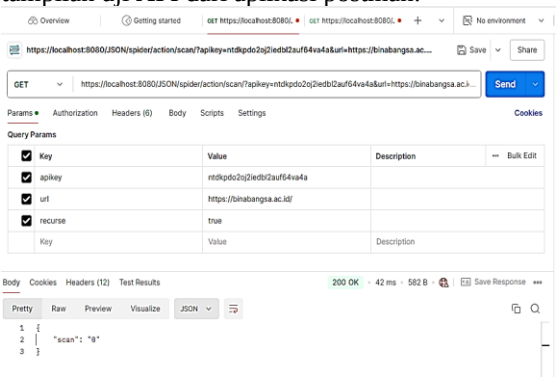
Gambar 5. OWASP ZAP

Pada bagian *automated scan*, *user* hanya perlu memasukkan url yang ingin dilakukan *penetration testing*, dan memilih beberapa parameter seperti ajax spider. Sedangkan untuk manual *explore* terdapat pilihan HUD (*Heads-Up Display*) dan beberapa parameter lain yang berkaitan dengan celah yang akan diteliti di *website* target. Di dalam OWASP ZAP juga terdapat API *gateway* untuk melakukan pengamatan dari luar aplikasi OWASP ZAP atau mengintegrasi dengan platform lainnya. Berikut adalah dokumentasi API pada OWASP ZAP:

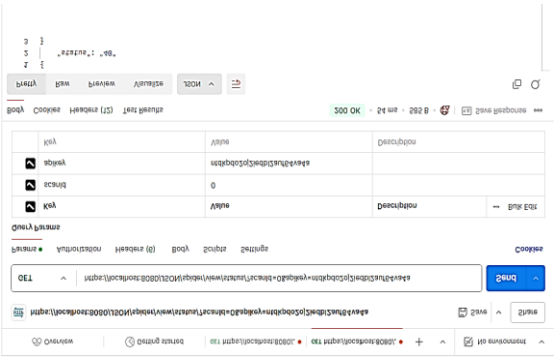


Gambar 6. *ZAP API UI*

Penulis mencoba melakukan *testing* menggunakan postman untuk mennguji respon API dalam melakukan *automated explore*. Berikut tampilan uji API dari aplikasi postman:



Gambar 7. *Scanning*



Gambar 8. *Progress Scan*

Sehingga dapat disimpulkan bahwa OWASP ZAP memiliki API yang dapat diintegrasi ke dalam *platform* lain.

p-ISSN : 2406-7733
e-ISSN : 2597-9922

C. Analisis Permasalahan/Kebutuhan

Pelaksanaan *vulnerability assesment* sering kali menghadapi sejumlah masalah, membuat prosedurnya menjadi rumit dan tidak efisien. Salah satu faktor yang paling penting adalah:

1) Kompleksitas dalam Penggunaan Alat

Banyak teknologi pendeteksi kerentanan yang tersedia membutuhkan pengetahuan teknis tingkat tinggi agar dapat berfungsi. Ini merupakan kendala yang signifikan bagi pengguna yang tidak memiliki latar belakang teknis, sehingga tidak memungkinkan bagi mereka untuk menyelesaikan pemeriksaan secara mandiri (Priyawati *et al.*, 2022).

2) Keterbatasan Mobilitas

Sebagian besar teknologi pendeteksi kerentanan masih berbasis *desktop* atau *server*, yang membutuhkan pengaturan khusus dan akses fisik ke perangkat keras tertentu. Akibatnya, penggunaannya terbatas, terutama dalam situasi yang membutuhkan pengujian cepat di beberapa lokasi.

3) Kurangnya Integrasi Teknologi Praktis

Saat ini, hanya ada beberapa *platform* ringan yang dapat digunakan secara langsung pada perangkat seluler, seperti Android; namun, solusi berbasis Android akan memungkinkan pengguna untuk melakukan *vulnerability assesment* dengan lebih mudah, fleksibel, dan realistis. Integrasi teknologi seperti ini dapat membantu menjangkau lebih banyak pengguna dengan berbagai tingkat keahlian sekaligus meningkatkan mobilitas dan efisiensi dalam proses pengujian.

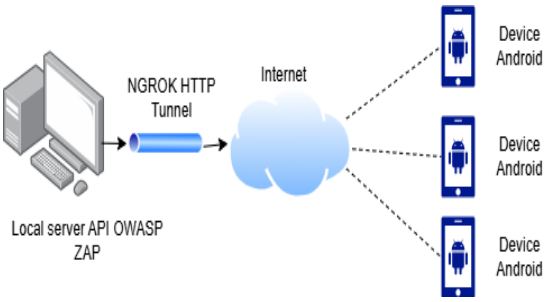
OWASP ZAP memiliki API yang dapat diintegrasi ke dalam *platform* lain, maka *platform* PENTRACE berbasis android yang dapat memudahkan dalam melakukan *penetration testing* atau *vulnerability assesment* dengan memanfaatkan API dari OWASP ZAP. Hal yang perlu disiapkan antara lain *local server* untuk menyimpan OWASP ZAP, *tunnel platform* untuk menghubungkan antara *local server* dengan internet dengan memanfaatkan *tunnel* ke arah internet, kemudian *platform* di android sebagai aplikasi di sisi *end user*.

IV. HASIL DAN PEMBAHASAN

A. Perancangan Sistem

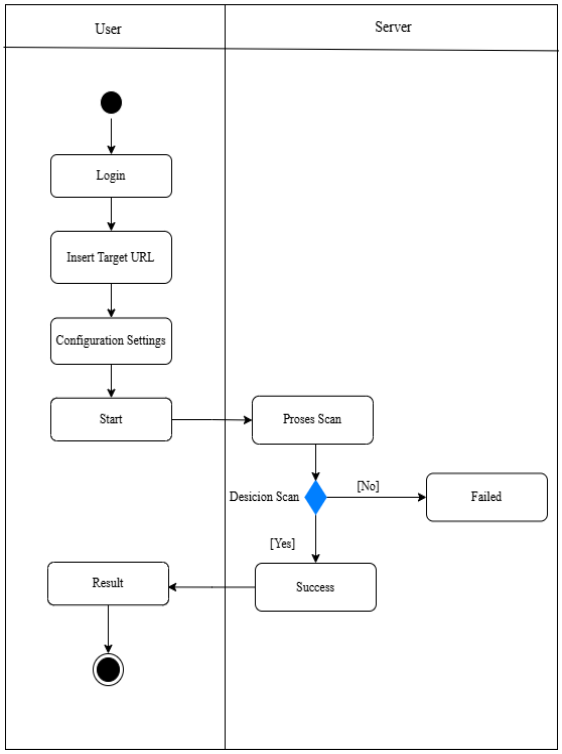
Perancangan sistem difokuskan pada pengujian keamanan situs *web* dengan OWASP ZAP dan Ngrok pada *platform* Android. Langkah ini terdiri dari identifikasi kebutuhan, pemilihan teknologi, dan perancangan arsitektur sistem. Selain itu, sebuah *activity diagram* dikembangkan untuk menunjukkan alur kerja, koneksi perangkat Android dengan *server*.

Gambar 9 menggambarkan arsitektur sistem yang dibangun untuk pengujian keamanan aplikasi berbasis Android dengan OWASP ZAP (*Zed Attack Proxy*), sebuah alat yang umum digunakan dalam pengujian aplikasi *web*. Pengujian dan analisis keamanan sistem dilakukan di server lokal yang menjalankan OWASP ZAP.



Gambar 9. Topologi Perancangan Sistem

OWASP ZAP adalah API untuk melakukan *penetration testing*, yang meliputi menemukan kerentanan pada aplikasi, memonitor aliran data, dan mensimulasikan serangan pada aplikasi yang diuji. Untuk menghubungkan *server* lokal ke perangkat smartphone Android di berbagai tempat, layanan tunneling HTTP dari Ngrok digunakan. Ngrok membuat *tunnel* HTTP yang aman yang memungkinkan akses ke *server* lokal dari internet. Dengan demikian, *server* lokal yang biasanya hanya dapat diakses di dalam jaringan lokal dapat tersedia di internet melalui alamat URL yang dibuat oleh Ngrok. Hal ini memungkinkan pengujian keamanan aplikasi pada perangkat Android tanpa memerlukan jaringan lokal yang sama, sehingga meningkatkan fleksibilitas pengujian.



Gambar 10. *Activity Diagram*

Gambar 10 menjelaskan antara pengguna (*User*) dan *server*, diagram aktivitas yang disajikan

p-ISSN : 2406-7733
e-ISSN : 2597-9922

menunjukkan alur kerja aplikasi *penetration testing* (*pentest*). Diagram ini menunjukkan bagaimana interaksi yang terjadi dari awal hingga hasil akhir pemindaian keamanan. Aktivitas tersebut dibagi menjadi dua bagian utama, yaitu sisi *User* dan Server. Di sisi Pengguna, proses dimulai dengan pengguna masuk ke dalam aplikasi untuk melakukan otentikasi. Setelah berhasil *login*, pengguna memasukkan Target URL yang merupakan alamat situs atau aplikasi yang akan dipindai. Selanjutnya, pengguna melakukan Pengaturan Konfigurasi, di mana berbagai parameter pemindaian, seperti kedalaman pemindaian dan jenis kerentanan, disesuaikan sesuai kebutuhan. Menekan tombol Start terlebih dahulu akan memulai pengguna setelah konfigurasi dan mengirimkan pemindaian ke *server*.

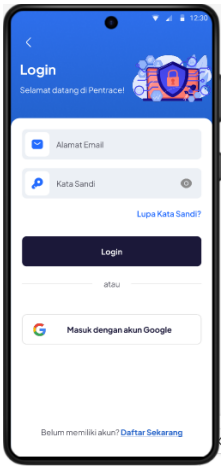
Di sisi *Server*, permintaan yang diterima dari pengguna diproses melalui tahap *Scan Process*, di mana *server* menjalankan algoritma pemindaian terhadap target yang ditentukan. Setelah pemindaian selesai, *server* melakukan *Decision Scan* untuk menentukan apakah pemindaian berhasil atau gagal. Jika pemindaian gagal, *server* akan mengembalikan status gagal kepada pengguna. Sebaliknya, jika berhasil, hasil pemindaian akan dikirim kembali ke pengguna dalam bentuk *Result*, yang berisi detail kerentanan yang ditemukan. Dimulai dengan input dari pengguna, prosedur pemindaian di *server*, hingga pengembalian hasil ke pengguna, grafik ini menunjukkan alur kerja yang disiplin. Pemisahan tanggung jawab antara pengguna dan *server* memastikan bahwa prosesnya efisien dan data yang dikirimkan dapat diproses dengan baik. Struktur ini juga memudahkan untuk mengidentifikasi kesalahan, misalnya jika pemindaian gagal, pengguna dapat segera diberitahu untuk memperbaiki konfigurasi atau target yang dimasukkan. Diagram ini mendukung pemahaman menyeluruh tentang proses aplikasi dan memastikan bahwa setiap langkah berjalan dengan cara yang terorganisir.

B. Implementasi Aplikasi

Implementasi sistem merupakan tingkat lanjut dari perancangan dimana semua komponen yang telah dirancang sebelumnya direalisasikan menjadi sebuah sistem yang dapat bekerja sebagaimana mestinya (Wiratmaka et al., 2023). Dalam hal ini, implementasi berfokus pada pembuatan sistem pengujian keamanan aplikasi Android berbasis *vulnerability assessment* yang memanfaatkan OWASP ZAP melalui API-nya. Implementasi sistem pengujian keamanan yang efektif harus mencakup alat, infrastruktur, dan skenario pengujian yang diperlukan untuk memberikan temuan yang akurat dan dapat diandalkan. Metode ini digunakan untuk memastikan keberhasilan pengujian keamanan pada aplikasi Android (Tiwari, 2021).

Gambar 11. *Splash Screen*

Gambar 11 menggambarkan layar pembuka, yang muncul saat aplikasi dijalankan. *Splash screen* adalah elemen visual yang pertama kali dilihat oleh pengguna dan memberikan kesan pertama terhadap program. Tampilan ini sering kali menyertakan identifikasi aplikasi, seperti logo, nama, atau elemen desain yang mewakili merek atau tujuan aplikasi. Dalam konteks aplikasi Pentrace, splash screen dimaksudkan untuk mengekspos pengguna pada identitas aplikasi dengan cara yang sederhana namun menarik, menghasilkan kesan pertama yang profesional dan berkesan sebelum melanjutkan ke fitur-fitur utama program.

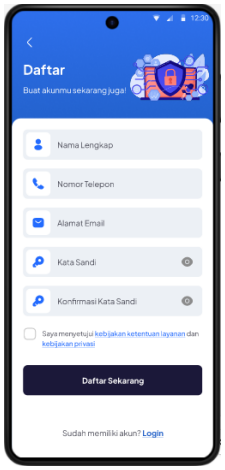


Gambar 12. Halaman Login

Gambar 12 menggambarkan layar *login*, yang dibuat dengan mengutamakan kenyamanan dan kesederhanaan akses bagi pengguna aplikasi. Halaman ini memiliki banyak fitur penting yang membantu dalam proses autentikasi, termasuk halaman *login* yang meminta pengguna untuk memasukkan alamat email dan kata sandi. Selain itu, ada tombol "Lupa Kata Sandi" untuk pengguna yang tidak dapat mengingat kata sandi mereka, serta tombol "*Login*" untuk memulai sesi pengguna. Untuk kenyamanan ekstra, situs *web* ini menyertakan opsi "Login dengan Google" yang

p-ISSN : 2406-7733
e-ISSN : 2597-9922

memungkinkan pengguna untuk dengan mudah dan aman melakukan *check-in* menggunakan akun Google mereka. Halaman login ini, dengan desain yang sederhana namun efektif, berusaha memberikan pengalaman pengguna sebaik mungkin.

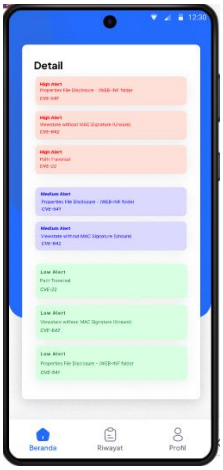


Gambar 13. Halaman Registrasi

Gambar 13 menggambarkan halaman registrasi yang ditujukan khusus untuk pengguna aplikasi baru yang belum memiliki akun Pentrace. Pada halaman registrasi, terdapat sebuah halaman formulir untuk membuat akun. Halaman ini mencakup kotak-kotak untuk memasukkan nama lengkap, nomor telepon, alamat email, dan kata sandi, serta konfirmasi kata sandi untuk menjamin bahwa kata sandi yang diberikan akurat. Desain halaman ini minimal namun informatif untuk memberikan pengalaman pendaftaran yang cepat dan mudah bagi pengguna baru. Setelah memasukkan informasi yang tepat, pengguna dapat mendaftar dan mulai menggunakan aplikasi Pentrace.

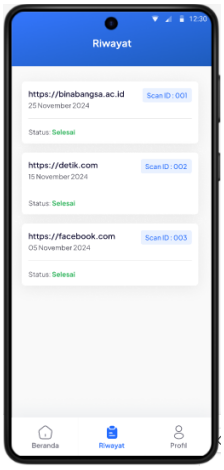
Gambar 14. Halaman Utama

Gambar 14 menggambarkan Layar utama dari aplikasi, ini ditujukan untuk memudahkan pengguna memindai situs *web* yang ingin dianalisis. Pada halaman ini, terdapat area input URL dimana pengguna dapat memasukkan alamat situs *web* yang akan dipindai. Setelah URL dimasukkan, sistem akan mulai memindai dan menampilkan elemen hasil pemindaian, yang berisi informasi tentang status atau potensi masalah pada situs yang diuji. Terdapat pula pilihan untuk memeriksa hasil pemindaian lebih lanjut, yang menawarkan ringkasan temuan yang lebih terperinci. Dengan tata letak yang jelas dan informatif, halaman utama ini bertujuan untuk memberikan pengalaman pengguna saat memeriksa keamanan situs *web* mereka.



Gambar 15. Detail Hasil

Gambar 15 menampilkan temuan-temuan dari proses pemindaian *vulnerability assesment* yang mencakup tentang kerntanan yang ditemukandan tingkat keparahannya, seperti *High Alert*, *Medium Alert*, dan *Low Alert*. Informasi ini mencakup detail nama kerentanan spesifik dan kode CVE (*Common Vulnerabilities and Exposures*), yang memungkinkan pengguna untuk mengidentifikasi dan menangani kerentanan keamanan dengan lebih baik.

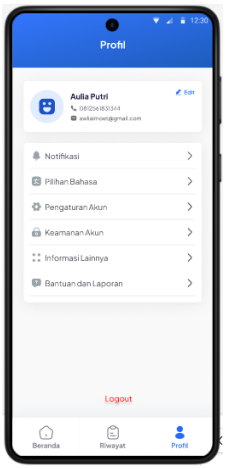


Gambar 16. Halaman Riwayat

Gambar 16 menampilkan halaman riwayat pemindaian situs yang telah dilakukan oleh penguna. Pada halaman ini, pengguna dapat dengan mudah melihat hasil pemindaian sebelumnya yang tersimpan dalam aplikasi. Fitur ini disediakan untuk

p-ISSN : 2406-7733
e-ISSN : 2597-9922

memudahkan pengguna mengakses informasi kerentanan tanpa harus melakukan pemindaian ulang. Tampilan halaman yang terstruktur memungkinkan pengguna dengan mudah untuk mencari dan mengelola riwayat pemindaian, sehingga memungkinkan proses perbaikan kerentanan yang lebih terarah dan terstruktur.



Gambar 17. Halaman Profile

Gambar 17 menampilkan halaman profil yang berisi semua informasi dan data pribadi pengguna, serta fitur-fitur pendukung tambahan yang dirancang untuk meningkatkan pengalaman pengguna. Fitur-fitur tersebut mencakup fitur notifikasi untuk memberikan informasi terbaru, pilihan bahasa yang memungkinkan pengguna memilih bahasa yang diinginkan, dan pengaturan akun yang mencakup informasi akun. Halaman ini juga mencakup opsi keamanan akun untuk mengamankan data pengguna, serta opsi bantuan dan laporan untuk membantu pengguna dalam menyelesaikan kesulitan atau memberikan umpan balik. Semua fitur ini dimaksudkan untuk mempermudah, dan memastikan keamanan pengguna dalam menggunakan aplikasi.

V. KESIMPULAN DAN SARAN

Kesimpulan

Penelitian ini menyajikan kebaruan dalam bentuk aplikasi Android yang menggabungkan API OWASP ZAP untuk penilaian kerentanan, termasuk fitur-fitur seperti pemindaian otomatis, analisis hasil interaktif, dan dukungan untuk API saat ini. Dengan solusi ini, penelitian berkontribusi secara signifikan dalam meningkatkan fleksibilitas dan efisiensi proses penilaian kerentanan, yang diharapkan dapat memberikan manfaat bagi para praktisi keamanan siber dengan mobilitas tinggi.

Terdapat beberapa kesimpulan yang dapat diambil dari desasin dan pengembangan prototipe aplikasi Pentrace.

1. Desain antarmuka aplikasi dibuat untuk memenuhi kebutuhan pengguna dengan memberikan tampilan yang intuitif, indahn dan responsif du berbagai layar perrangkat.
2. Elemen-elemen penting termasuk *splash screen*, login, *dashboard*, profil pengguna, pemindaian kerentanan otomatis, laporan kerentanan, dan lainnya telah diidentifikasi dan dibuat untuk memberikan pengalaman yang sederhana namun efektif bagi pengguna. Selain itu, prototipe ini menekankan pengalaman pengguna dengan alur navigasi yang sederhana, penggunaan komponen visual yang konsisten, dan akses cepat ke fungsionalitas utama.
3. Desain UIUX akhir memenuhi kriteria yang diperlukan untuk mempersiapkan langkah implementasi, dan telah divalidasi secara visual menggunakan simulasi beberapa perangkat untuk memastikan kesesuaian desain.

Saran

Beberapa saran untuk pengembangan lebih lanjut.

1. Lakukan pengujian *usability* dengan melibatkan calon pengguna guna mendapatkan umpan balik terhadap prototipe yang telah dirancang. Hasil pengujian ini dapat digunakan untuk menyesuaikan alur navigasi, elemen antarmuka pengguna, atau tata letak agar lebih sesuai dengan kebutuhan dan ekspektasi pengguna.
2. Menyiapkan dokumentasi desain yang lengkap, termasuk panduan warna, ukuran elemen, dan komponen antarmuka pengguna, agar proses implementasi berjalan lebih lancar. Simulasi lebih lanjut juga diperlukan untuk memastikan bahwa desain dapat menyesuaikan dengan berbagai ukuran layar perangkat, terutama yang memiliki rasio layar tidak konvensioinal.

DAFTAR PUSTAKA

Alserhani, F., & Aljared, A. (2023). Evaluating Ensemble Learning Mechanisms for Predicting Advanced Cyber Attacks. *Applied Sciences (Switzerland)*, *13*(24). https://doi.org/10.3390/app132413310

Christina Sari, N., Solichan, A., Ansor, B., Putra Ramdani, A., Zainudin Al Amin, M., Khaira, M., & Rohman Riquelme Al Ubaidah, A. (2024). Deteksi Kerentanan SQL Injection pada Website Menggunakan Vulnerability Assessment Info Artikel. *Journal of Data Insights*, *2*(1), 9–17. http://journalnew.unimus.ac.id/index.php/jodi

Direktorat Operasi Keamanan Siber, & Negara, B. S. D. S. (2022). *Laporan Tahunan Monitoring Keamanan Siber*. Direktorat Operasi Keamanan Siber Badan Siber Dan Sandi Negara. https://cloud.bssn.go.id/s/Lyw8E4LxwNiJoNw

Firdha, N., Damira, Fitri, R., Selaras, G. H., & Saputra, I. G. N. (2021). Studi Literatur Tentang Peningkatan Kompetensi Belajar Peserta Didik Melalui Kegiatan Pembelajaran Kolaboratif

p-ISSN : 2406-7733
e-ISSN : 2597-9922

Berbasis Lesson Study. *Prosiding SEMNAS BIO*, *01*, 1005–1013.

Hapsari, R. D., & Pambayun, K. G. (2023). Ancaman Cybercrime di Indonesia: Sebuah Tinjauan Pustaka Sistematis. *Jurnal Konstituen*, *5*(1), 1–17. https://doi.org/10.33701/jk.v5i1.3208

Hasibuan, A. F., & Handoko, D. (2023). Analisis Keretanan Website Dengan Aplikasi Owasp Zap. *Jurnal Ilmu Komputer Dan Sistem Informasi*, *2*(2), 257–270. https://jurnal.unity-academy.sch.id/index.php/jirsi/article/view/51

Idris, M., Syarif, I., & Winarno, I. (2022). Web Application Security Education Platform Based on OWASP API Security Project. *EMITTER International Journal of Engineering Technology*, *10*(2), 246–261. https://doi.org/10.24003/emitter.v10i2.705

Kurniawan, H., & Christianto, E. (2024). Analysis Vulnerability Website Baleomolcreative dengan Metode Penetration Testing Execution Standard & Vulnerability Assessment Pada Http Response Header Field. *Jurnal Teknologi Informasi Dan Komunikasi)*, *8*(3), 2024. https://doi.org/10.35870/jti

Kusuma, G. (2022). Implementasi Owasp Zap Untuk Pengujian Keamanan Sistem Informasi Akademik. *Jurnal Teknologi Informasi: Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, *16*(2), 178–186. https://doi.org/10.47111/jti.v16i2.3995

Negara, B. S. dan S. (2022). *Lanskap Keamanan Siber Indonesia 2022*. Badan Siber Dan Sandi Negara. https://www.bssn.go.id/monitoring-keamanan-siber/

Negara, B. S. dan S. (2023). *Laporan Keamanan Siber Indonesia (Bssn)*. https://www.bssn.go.id/monitoring-keamanan-siber/

Ozkan-okay, M., Yilmaz, A. A., Akin, E., Aslan, A., & Aktug, S. S. (2023). A Comprehensive Review of Cyber Security Vulnerabilities ,. *Electronics*, *12*(1333).

Priyawati, D., Rokhmah, S., & Utomo, I. C. (2022). Website Vulnerability Testing and Analysis of Internet Management Information System Using OWASP. *International Journal of Computer and Information System (IJCIS) Peer Reviewed-International Journal*, *3*(3), 143–147. https://doi.org/10.29040/ijcis.v3i3.90

Putra, M. Y. N., & Huda, S. N. (2021). Literature Review dengan Pendekatan Pengembangan Design Thinking untuk Sistem Informasi Studi Kasus SPP dan Beasiswa. *Automata*, 154–162.

Ritonga, M. (2021). Studi Literatur Efektivitas Model Pembelajaran Simulasi Digital Marito. *Seminar Nasional Pendidikan LPPM IKIP PGRI Bojonegoro*, 63–70.

Subahan, A., Dista, D. X., & Witarsa, R. (2021). Kajian Literatur Tentang Kebijakan Pendidikan Dasar Di Masa Pandemi Dan Dampaknya Terhadap Pembelajaran. *Jurnal Review Pendidikan Dan Pengajaran*, *4*(1), 1–9. https://doi.org/10.31004/jrpp.v4i1.1662

Tasmiha Khan, M. G. (2024). *Apa itu CVE (Common Vulnerabilities and Exposures)?* https://www.ibm.com/id-id/think/topics/cve%0A

Tiwari, S. (2021). An ensemble deep neural network model for onion-routed traffic detection to boost cloud security. *International Journal of Grid and High Performance Computing*, *13*(1), 1–17. https://doi.org/10.4018/IJGHPC.2021010101

Valencia, C., Wijaya, J. A., Meiden, C., Bisnis, I., & Kian, K. (2022). Studi Literatur: Analisis Pengaruh Laporan Arus Kas terhadap Kinerja Keuangan Menggunakan Metode Systematic Literature Review (SLR). *Jurnal Pendidikan Dan Konseling*, *4*, 7484–7496.

Wiratmaka, C. S., Al-Fajri, M., & Mustika, M. (2023). Implementasi Aplikasi Appsheet Berbasis Android Untuk Mendukung Proses Pembelajaran Di Sdn 6 Metro Utara. *Jurnal Mahasiswa Ilmu Komputer*, *4*(2), 159–167. https://doi.org/10.24127/ilmukomputer.v4i2.4187

Yoheswari, S. (n.d.). *Optimized Intrusion Detection Model For Identifying Known And Innovative Cyber Attacks Using Support Vector Machine (SVM)*. *1*(5), 401–407.