PENGEMBANGAN APLIKASI VULNERABILITY SCANNER UNTUK MENDETEKSI CELAH KEAMANAN SIBER PADA WEBSITE

Hogy Albani Bardian, Imam Sutanto
Teknik Informatika, Universitas Esa Unggul
Jl. Harapan Indah Boulevard No.2 Pusaka Rakyat Kabupaten Bekasi, Indonesia
hogyalbani64@gmail.com

ABSTRAK

Website telah menjadi sarana utama bagi pengguna untuk mencari informasi dan melakukan berbagai aktivitas, seperti transaksi online dan penjualan. Namun, kemudahan akses ini dapat membawa risiko keamanan, di mana celah keamanan pada komponen seperti form login dan fitur upload dapat dieksploitasi oleh peretas. Untuk mengatasi masalah ini, aplikasi vulnerability scanner diperlukan sebagai solusi untuk mendeteksi dan mengevaluasi kelemahan keamanan secara otomatis. Namun, aplikasi yang ada saat ini memiliki keterbatasan, yaitu antarmuka yang rumit pada versi *open-source* dan biaya lisensi yang tinggi pada versi berbayar. Penelitian ini bertujuan untuk mengembangkan aplikasi *vulnerability scanner* berbasis web yang dapat mendeteksi celah keamanan sesuai dengan daftar OWASP Top 10 2021. Aplikasi ini bersifat *open-source* dan dilengkapi dengan fitur penilaian kerentanan menggunakan CVSS v4.0, deskripsi celah keamanan, rekomendasi perbaikan, serta laporan hasil kerentanan dalam bentuk antarmuka grafis (GUI). Metode pengembangan yang digunakan adalah metode Agile. Aplikasi ini dibangun menggunakan bahasa pemrograman Python, framework Flask, dan MySQL untuk pengelolaan basis data. Selain itu, virtualisasi menggunakan Docker agar aplikasi dapat dengan mudah diimplementasikan di berbagai lingkungan. Hasil pengujian *black box* dan pengujian scanning dari aplikasi ini menunjukkan bahwa aplikasi ini berjalan dengan baik dan dapat mendeteksi celah keamanan dalam kategori *critical*, *high*, *medium*, dan *low*.

Kata kunci : *Keamanan Website, Vulnerability Scanner, OWASP Top 10, Python*

1. PENDAHULUAN

Website atau situs web merupakan sekumpulan halaman yang mempermudah pengguna dalam mencari berbagai macam informasi, baik dalam bentuk teks, video, gambar, maupun animasi. Saat ini, masyarakat lebih memilih mengakses *website* untuk memperoleh informasi daripada harus datang langsung ke lokasi. Hal ini menunjukkan bahwa website telah menjadi bagian penting dari kehidupan sehari-hari karena dapat digunakan untuk aktivitas seperti mencari informasi, melakukan transaksi online, menjual produk, dan banyak lagi. Namun, kemudahan akses ini juga menimbulkan risiko keamanan yang menjadi tantangan bagi pengembang dalam menjaga integritas *website*. Celah keamanan dapat membuat website rentan terhadap berbagai serangan siber seperti *SQL Injection*, *Cross-Site Scripting*, dan *Sensitive Data Exposure* yang dapat dimanfaatkan oleh peretas untuk merugikan pemilik maupun pengguna *website* [1].

Pada umumnya, *website* terdiri dari berbagai macam komponen, sehingga terdapat banyak sisi yang dapat menimbulkan celah seperti pada *parameter* kolom URL, fitur *file upload*, form pencarian, dan komponen-komponen yang lainnya [2]. Untuk mengantisipasi hal ini, penggunaan aplikasi *vulnerability scanner* menjadi solusi yang penting. *Vulnerability scanner* adalah aplikasi yang dirancang untuk melakukan evaluasi terhadap kelemahan keamanan suatu website secara otomatis. Contoh aplikasi *vulnerability scanner open-source* yang umum digunakan adalah pada saat ini adalah Nuclei dan Nikto. Adapun contoh aplikasi *vulnerability scanner* yang berbayar adalah Nessus dan Acunetix [3].

Aplikasi *vulnerability scanner* pada umumnya dapat mengidentifikasi dan menganalisis berbagai komponen pada website yang rentan terhadap serangan. Selain itu, aplikasi tersebut juga dapat memberikan nilai tingkat keparahan terhadap celah yang ditemukan, sehingga dapat memungkinkan para pengembang *website* untuk mengambil langkah-langkah yang tepat untuk memperbaiki keamanan *website* mereka. Namun, seperti halnya teknologi, aplikasi *vulnerability scanner* juga memiliki kelemahan. Kebanyakan dari aplikasi *vulnerability scanner* yang berbasis *open-source* masih memiliki tampilan dalam bentuk teks atau CLI (*Command Line Interface*), sehingga sulit untuk digunakan karena harus memasukkan perintah yang cukup panjang. Adapun aplikasi *vulnerability scanner* yang berbayar seperti Nessus dan Acunetix mengharuskan pengguna untuk membayar lisensi setelah beberapa kali pemakaian dan seringkali dikenakan biaya yang cukup mahal [4]. Hal ini menunjukkan bahwa aplikasi tersebut memerlukan pembaruan agar dapat meningkatkan kemudahan dalam penggunaannya.

Berdasarkan penjelasan diatas, penelitian ini akan membangun sebuah aplikasi *vulnerability scanner* berbasis web yang dapat mengidentifikasi celah keamanan *website* berdasarkan dari daftar OWASP Top 10 2021 dengan menggunakan pendekatan berbasis *template*, sehingga dapat mengikuti perkembangan celah keamanan siber pada

website dan memungkinkan pengguna untuk menyesuaikan konfigurasi pemindaian sesuai dengan kebutuhan mereka. Aplikasi ini akan didistribusikan secara *open-source* dan akan ditambahkan fitur-fitur seperti penilaian kerentanan berdasarkan perhitungan dari CVSS (*Common Vulnerability Scoring System*) v4.0, penjelasan dari celah keamanan yang ditemukan, rekomendasi perbaikan yang bisa dilakukan, dan laporan dari hasil kerentanan yang ditemukan dalam bentuk GUI. Hasil yang diharapkan dari aplikasi ini adalah untuk membantu para pengembang *website* dalam mengidentifikasi dan menganalisis celah keamanan yang mungkin ada pada *website* mereka sehingga tidak menjadi korban serangan siber.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Pada penelitian yang berjudul "*Design and Implementation of SQL Injection Vulnerability Scanning Tool*" oleh Juanjuan Zhao dan Liu Changhua (2020), penelitian ini mengembangkan sebuah alat untuk mendeteksi celah keamanan SQL Injection pada website. Aplikasi ini menggunakan metode crawling untuk menelusuri semua URL dengan parameter pada website. Aplikasi ini dibangun menggunakan bahasa pemrograman Python dan menggunakan framework Scrapy. Aplikasi ini hanya dapat mendeteksi celah keamanan SQL Injection pada website dan framework yang digunakan adalah Scrapy [5].

Pada penelitian yang berjudul "*Development of a Browser Extension for Web Application Vulnerability Detection, Avoidance, and Secure Browsing* (VDAS)" oleh Alya Geogiana Buja, Nurul Syahirah Khairuddi, Noor Afni Deraman, Khyrina Airin Fariza Abu Samah, Mohd Nor Hajar Hasrol Jono dan Nor Aimuni MD Rashid (2021). Penelitian ini mengembangkan sebuah aplikasi vulnerability scanner berbasis extension browser Google Chrome menggunakan pendekatan Data Mining. Aplikasi ini dapat mendeteksi celah keamanan seperti Cross-site Scripting (XSS), SQL Injection (SQLi), Local File Inclusion (LFI), dan Remote Command Execution (RCE). Pada penelitian ini, aplikasi yang dikembangkan yaitu berbasis extension Google Chrome, dan metode yang digunakan menggunakan pendekatan *data mining* [6].

Pada penelitian yang berjudul "Aplikasi Deteksi Kerentanan SQL Injection dan Cross-Site Script Pada Aplikasi Berbasis Website Menggunakan Metode Waterfall" oleh Herwanto Patah dan Mustofa Zaid (2021). Penelitian ini mengembangkan aplikasi deteksi celah keamanan pada website yang dapat mendeteksi celah Cross-site Scripting (XSS) dan SQL Injection pada aplikasi berbasis website. Aplikasi ini dibangun menggunakan Bahasa pemrograman PHP. Pada penelitian ini menggunakan metode Waterfall sebagai metode pengembangan dan aplikasi ini dibangun menggunakan bahasa pemrograman PHP. Selain itu, aplikasi ini hanya dapat melakukan deteksi terhadap celah keamanan Cross-site Scripting dan SQL Injection saja [7].

Pada penelitian yang berjudul "Design of Web Vulnerability Scanner Based on Go Language" oleh Jingxia Chen, Xiuling Chen, dan Bo Yu (2021). Penelitian ini mengembangkan sebuah aplikasi vulnerability scanner menggunakan bahasa pemrograman Go, framework Vue sebagai front end, dan MySQL sebagai database. Aplikasi ini bekerja dengan memaanfaatkan plug in yang dapat mendeteksi celah keamanan pada website dan disajikan dalam tampilan GUI. Perbedaan pada penelitian ini adalah aplikasi dibangun menggunakan bahasa pemrograman Go dan Vue [8].

Pada penelitian yang berjudul "Rancang Bangun Aplikasi Analisis Standar Keamanan Website dengan Metode Scanning Vulnerability Menggunakan Module Requests Python" oleh Rizki Alam Ramdhani dan Nunu Nurdiana (2022). Penelitian ini mengembangkan sebuah aplikasi vulnerability scanner yang dapat mendeteksi bug keamanan paling dasar seperti Bruteforce, SQL Injection, Cross-site Scriptin (XSS), Directory Traversal, Local File Inclusion, dan Remote File Inclusion. Aplikasi ini dibangun menggunakan bahasa pemrograman Python dan menggunakan module Request. Pada penelitian ini, aplikasi hanya dibuat dengan tampilan CLI dan metode yang digunakan adalah vulnerability scanning, dimana aplikasi hanya mencari celah keamanannya saja dan tidak memberikan nilai tingkat keparahan terhadap celah yang ditemukan oleh aplikasi tersebut [9].

2.2. Keamanan Siber

Keamanan siber, juga dikenal sebagai *cybersecurity*, adalah serangkaian tindakan pencegahan yang bertujuan untuk melindungi sistem komputer, jaringan, dan data dari serangan di dunia maya. Hal ini mencakup penerapan langkah-langkah dan teknologi untuk mencegah dan merespons ancaman serangan siber untuk menjaga kerahasiaan, integritas, dan ketersediaan pada suatu sistem dan informasi. Keamanan siber sangat penting karena berkembangnya teknologi di perusahaan modern, serta kecanggihan serangan siber yang terus meningkat dan berkembang. Pada dasarnya, internet adalah gudang informasi yang berharga yang dapat dieksploitasi oleh para peretas jika tidak dilindungi dengan baik. Peretas dapat terus mencari kerentanan dalam sistem untuk mencuri data dan menimbulkan kerusakan pada sistem [10]. Keamanan siber memiliki 3 aspek yang biasa disebut dengan CIA Triad atau segitiga CIA. Gambar 1 menunjukkan aspek-aspek dari segitiga CIA.

Gambar 1. Aspek keamanan siber

2.3. Website

Website adalah suatu teknologi yang terdiri dari beberapa halaman-halaman yang saling terhubung yang digunakan untuk mencari informasi. *Website* menggunakan konsep *hyperlink* yang menghubungkan antara satu halaman ke halaman yang lainnya di dalam *website*. Fungsi utama website adalah sebagai media komunikasi dan sumber informasi yang dapat diakses oleh pengguna internet dari berbagai lokasi. *Website* dapat diakses oleh pengguna dari berbagai perangkat dan *platform*, seperti komputer, *handphone*, dan tablet. Hadirnya teknologi *website* dapat memungkinkan pengguna untuk melakukan berbagai macam aktivitas melalui internet dengan mudah, seperti mencari informasi, melakukan transaksi, hingga jual beli [11].

2.4. Vulnerability Scanner

Vulnerability scanner adalah aplikasi yang dirancang untuk mengidentifikasi potensi kelemahan yang secara sistematis memeriksa sistem komputer, jaringan, dan aplikasi untuk mengidentifikasi kerentanan yang dapat dieksploitasi oleh penyerang. *Vulnerability scanner* dapat memberikan penilaian serta rekomendasi yang bisa dilakukan untuk menutup celah keamanan tersebut. Tujuan dari *vulnerability scanner* adalah untuk mendeteksi celah atau kerentanan keamanan yang dapat dieksploitasi oleh penyerang yang dapat mengakses, merusak, atau mencuri data yang ada dalam sistem. *Vulnerability scanner* memainkan peran penting dalam keamanan siber, karena dapat secara proaktif menyelidiki kelemahan sistem agar tidak dapat menjadi sasaran oleh penyerang [12].

2.5. OWASP Top 10

OWASP Top 10 merupakan salah satu proyek dari OWASP yang mencakup daftar dari sepuluh celah keamanan pada aplikasi web yang paling umum dan berbahaya. Daftar dari OWASP Top 10 diperbarui setiap 3 sampai 4 tahun untuk menunjukkan perubahan dalam ancaman keamanan pada aplikasi web. OWASP Top 10 dapat membantu para *developer* hingga profesional keamanan untuk mengatasi kerentanan pada *website* yang dapat dieksploitasi oleh penyerang. OWASP Top 10 bertujuan untuk meningkatkan kesadaran akan risiko keamanan yang umum ditemukan dan memberikan panduan tentang cara memitigasi risiko keamanan tersebut. OWASP Top 10 adalah sumber daya berharga bagi para pengembang *website* untuk memahami dan mengatasi risiko keamanan paling kritis terhadap aplikasi web [13].

2.6. CVSS

CVSS (*Common Vulnerability Scoring System*) adalah suatu kerangka kerja yang digunakan untuk mengukur dan mengevaluasi tingkat kerentanan (*vulnerability*) keamanan pada perangkat lunak dan sistem, termasuk *website*. Diperkenalkan pertama kali pada tahun 2005, CVSS telah menjadi standar industri yang diakui secara luas untuk penilaian kerentanan. CVSS memberikan skor numerik berdasarkan metrik yang dinilai yang umumnya berkisar dari nilai 0 hingga 10, di mana skor yang lebih tinggi menunjukkan tingkat keparahan yang lebih tinggi. Skor yang dihasilkan oleh CVSS dapat diterjemahkan menjadi kategori tingkat keparahan, yaitu *low*, *medium*, *high*, dan *critical*. Kategori ini membantu organisasi untuk memahami sejauh mana suatu kerentanan dapat mempengaruhi suatu keamanan sistem mereka [14].

2.7. Python

Python adalah bahasa pemrograman *open-source* tingkat tinggi yang mudah dipahami dan dapat digunakan untuk berbagai bidang, termasuk keamanan siber. Python didesain dengan sintaks yang sederhana dan jelas, sehingga mudah dipelajari dan digunakan. Python dikembangkan oleh Guido van Rossum pada tahun 1991. Python memiliki banyak *library* dan *framework* yang memudahkan pengembangan berbagai jenis aplikasi, termasuk web aplikasi. Python merupakan bahasa pemrograman yang interpretatif, yang berarti kode Python dieksekusi langsung oleh interpreter tanpa dikompilasi terlebih dahulu. Kelebihan utama Python adalah kemampuannya untuk memprioritaskan keterbacaan kode, sehingga pengembang dapat dengan mudah memahami dan mengelola kode program agar lebih rapih dan efektif. Python juga mendukung beragam paradigma pemrograman seperti pemrograman berorientasi objek, pemrograman fungsional, dan pemrograman prosedural [15].

3. METODE PENELITIAN

3.1. Metode Perancangan

Metode perancangan yang digunakan dalam penelitian ini menggunakan metode *Agile*. Metode *Agile* dipilih dalam penelitian ini karena sangat cocok untuk aplikasi yang memerlukan adaptasi cepat terhadap perubahan kebutuhan dan umpan balik dari pengguna. Tahapan-tahapan dalam metode Agile yaitu

Requierement (perencanaan), *Design* (desain), *Development* (perancangan), *Testing* (pengujian), *Deploy* (peluncuran), dan *Review* (umpan balik).

a. Tahap *Requirement*
   Requirement atau perancaan merupakan tahap pertama dalam metode Agile. Pada tahap *Requirement* untuk aplikasi *vulnerability scanner* pada penelitian ini akan dilakukan pengumpulan data untuk identifikasi tujuan dari pengembangan aplikasi *vulnerability scanner* serta menentukan kebutuhan dari perancangan aplikasi ini seperti *tools* yang akan digunakan.
b. Tahap *Design*
   Design atau desain merupakan tahap kedua dari metode *Agile* setelah tahap *Requirement* telah selesai. Tahap Design pada penelitian ini akan membuat gambaran alur kerja dari aplikasi berupa diagram *UML*, serta gambaran tampilan antarmuka pada aplikasi yang dibangun.
c. Tahap *Development*
   Tahap ketiga dari metode Agile. Pada tahap ini, desain yang telah dibuat sebelumnya akan diimplementasikan ke dalam bentuk kode atau *program*. Aplikasi akan dibangun menggunakan bahasa pemrograman Python dengan *framework* Flask, yang memungkinkan script Python dijalankan di browser. *Database* akan dibangun menggunakan sistem manajemen basis data MySQL untuk menyimpan data pengguna, termasuk informasi login dan hasil pemindaian. Aplikasi ini akan dijalankan dalam lingkungan Docker, yang membantu mengurangi penggunaan memori dan memastikan konsistensi lingkungan pengembangan dan produksi.
d. Tahap *Testing*
   Tahap keempat pada metode Agile yang bertujuan untuk melakukan pengujian terhadap aplikasi yang dibuat. Pada tahap ini, *akan* dilakukan pengujian pada fitur aplikasi *vulnerability scanner* menggunakan *black-box testing* agar tidak terjadi kesalahan atau error pada saat aplikasi tersebut dijalankan, serta akan dilakukan pengujian terhadap fungsionalitas aplikasi yang dilakukan pada *website* aktif.
e. Tahap *Deployment*
   Tahap kelima dari metode Agile yang merupakan tahap peluncuran aplikasi yang sudah dibuat agar aplikasi tersebut bisa *digunakan* oleh pengguna melalui internet. Aplikasi ini akan didistribusikan secara open-source melalui platform GitHub yang disertai dengan *source code* dan tahapan-tahapan instalasinya dan di deploy menggunakan Docker agar pengguna dapat menggunakan aplikasi ini di komputer lokal mereka.
f. Tahap *Review*
   Tahap terakhir dari metode Agile yang merupakan tahap penilaian dan umpan balik dari pengguna terhadap aplikasi yang telah dibuat. Umpan balik tersebut akan digunakan sebagai revisi bagi penulis agar dapat melakukan pengembangan lebih lanjut dari aplikasi tersebut. Tahap Review pada penelitian ini dilakukan dengan cara menyebarkan kuesioner kepada pengguna. Kuesioner dilakukan menggunakan *Google Forms*, yang bertujuan untuk mengumpulkan data terkait pengalaman pengguna, tingkat kepuasan, serta aspek-aspek yang perlu diperbaiki atau ditingkatkan pada saat pengguna menggunakan aplikasi FASTT. Hasil dari tahap ini akan menjadi acuan agar aplikasi dapat lebih optimal dan sesuai dengan kebutuhan pengguna.

3.2. Teknik Pengumpulan Data

Teknik pengumpulan data pada pengembangan aplikasi ini dilakukan melalui dua metode, yaitu studi literatur dan observasi.

a. Studi Literatur
   Studi literatur merupakan kegiatan yang bertujuan untuk menyelidiki, mempelajari, dan menganalisis literatur atau sumber-sumber yang relevan dengan topik penelitian atau *kajian* tertentu. Metode studi literatur pada penelitian ini yaitu dengan mempelajari jurnal, artikel, dan dokumentasi resmi yang sudah ada sebelumnya yang diambil dari internet mengenai cara memahami berbagai jenis celah keamanan yang umum terjadi pada website, cara memahami teknik dan metode yang digunakan dalam vulnerability scanning, dam informasi tentang standar OWASP Top 10 dan kerangka kerja CVSS.
b. Observasi
   Observasi adalah metode yang digunakan untuk mengumpulkan data dan informasi dengan cara mengamati objek atau fenomena *secara* langsung. Dalam penelitian ini, observasi dilakukan terhadap beberapa aplikasi vulnerability scanner yang sudah ada di pasaran, antara lain Nikto, Nuclei, dan Nessus. Observasi dilakukan dengan cara men-*download* masing-masing dari aplikasi vulnerability scanner tersebut. Hal ini bertujuan untuk mengidentifikasi kelebihan dan kelemahan dari aplikasi yang ada yang menjadi kebutuhan pada aplikasi vulnerability scanner yang dikembangkan dalam penelitian ini. Observasi pada penelitian ini juga dilakukan perbandingan fitur antara aplikasi FASTT dengan aplikasi Nikto, Nuclei, dan Nessus dengan tujuan untuk mengidentifikasi perbedaan antara FASTT dengan aplikasi-aplikasi tersebut.

4. HASIL DAN PEMBAHASAN

4.1. Deskripsi Aplikasi

Aplikasi yang dirancang pada penelitian ini merupakan aplikasi untuk melakukan uji keamanan pada website. Aplikasi ini akan diberi nama FASTT (*Fast and Automated Security Testing Tool*). FASTT adalah aplikasi *vulnerability scanner* berbasis web yang dirancang untuk mengidentifikasi dan mengevaluasi celah keamanan pada *website* berdasarkan dari daftar OWASP Top 10 2021 dan

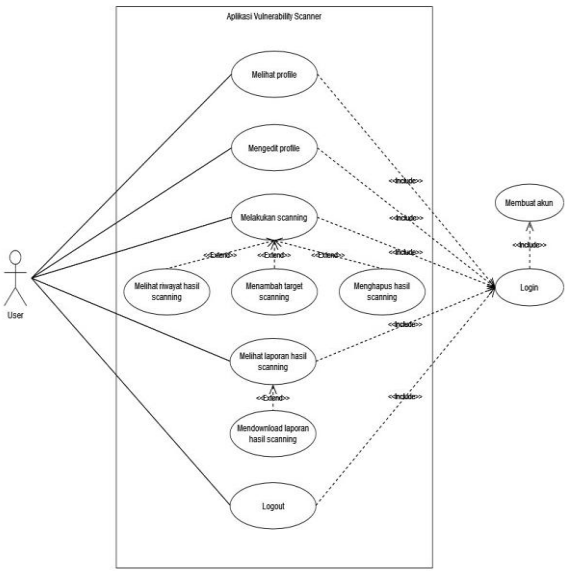
CVSS v4.0 untuk penilaian kerentanan. FASTT menggunakan metode pendekatan berbasis *template* dengan format .py yang dapat dikustomisasi sehingga dapat menyesuaikan pembaruan terhadap ancaman yang baru. FASTT dibangun dengan bahasa pemrograman Python dan *framework* Flask, sementara data hasil scanning dan informasi pengguna dikelola menggunakan database MySQL. Aplikasi ini dirancang untuk berjalan di lingkungan Docker, sehingga memungkinkan kemudahan instalasi dan kompatibilitas antar platform, baik pada Windows maupun Linux. FASTT adalah aplikasi *vulnerability scanner open-source*, di mana kode sumbernya dapat dimodifikasi dan diakses melalui GitHub. Gambar 2 menunjukkan logo dari aplikasi FASTT.

Gambar 2. Logo Aplikasi FASTT

4.2. Hasil *Design*

Hasil *Design* meliputi hasil rancangan dari *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*. Serta hasil rancangan tampilan untuk aplikasi FASTT.

4.3. Use Case Diagram

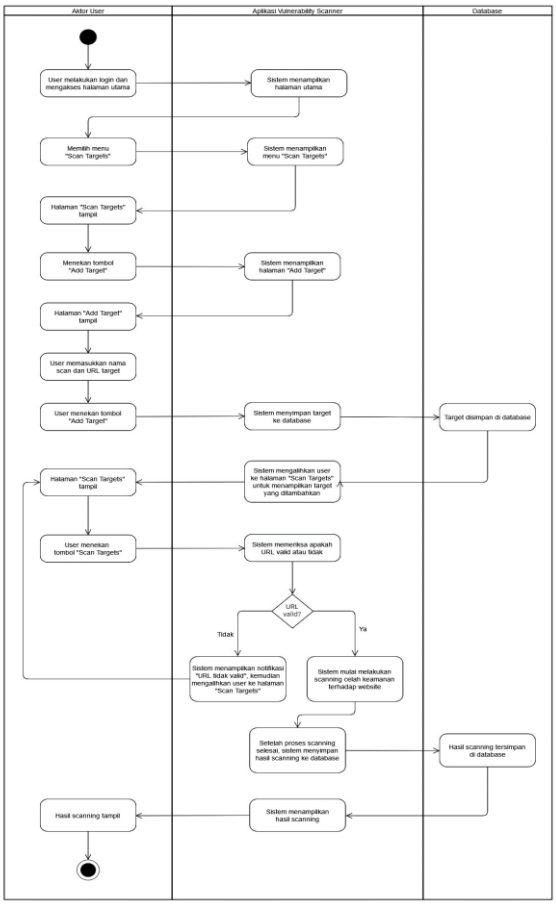


Gambar 3. Use Case Diagram

Gambar 3 menunjukkan diagram *use case* pada penelitian ini beserta interaksi antara aktor *user* dengan aplikasi *vulnerability scanner*.

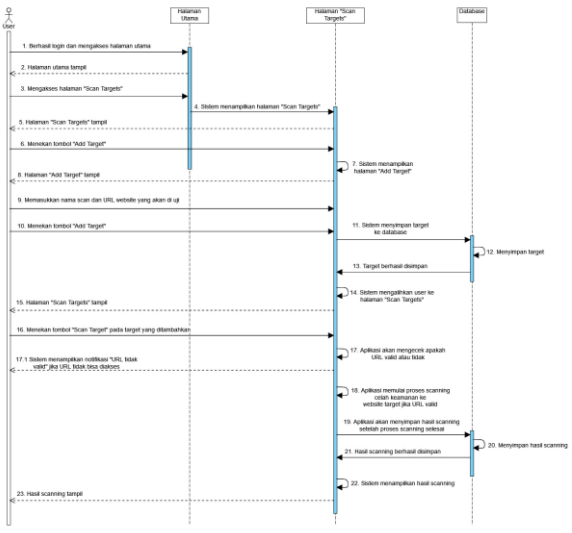
4.4. Activity Diagram

Gambar 4 menunjukkan alur pada *activity diagram* melakukan scanning celah keamanan pada website.



Gambar 4. Activity Diagram

4.5. Sequence Diagram

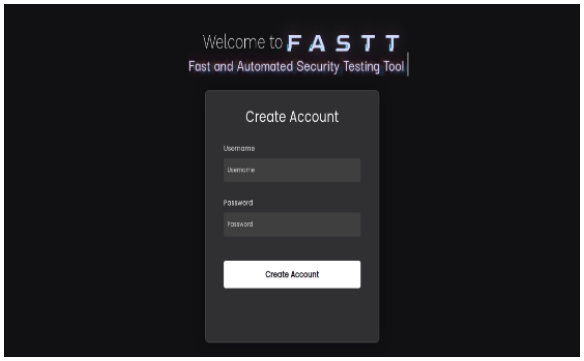


Gambar 5. Sequence Diagram

Gambar 5 menunjukkan alur pada sequence diagram melakukan *scanning* celah keamanan pada *website*.

4.6. Hasil Interface Halaman *Create Account*

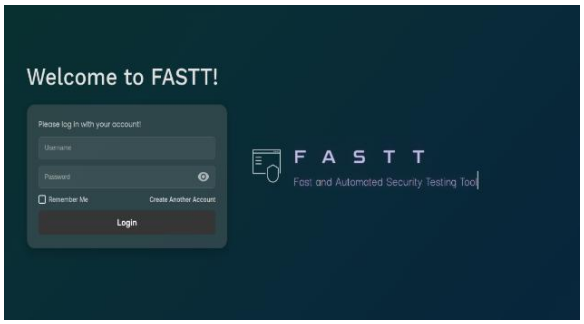
Pada halaman create account, user diharuskan untuk memasukkan kredensial berupa *username* dan *password*. Gambar 6 menunjukkan tampilan dari halaman *create account*.



Gambar 6. Tampilan Halaman Create Account

4.7. Hasil Interface Halaman *Login*

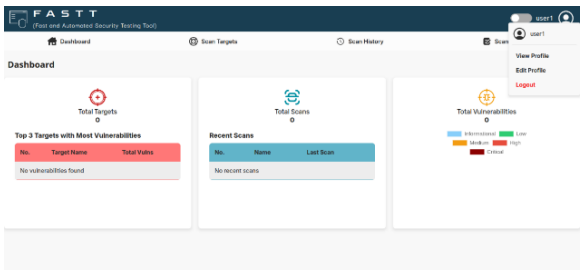
Pada halaman ini, user diharuskan untuk memasukkan username dan password yang telah dibuat, lalu dapat menekan tombol Login untuk dapat masuk ke dalam halaman *dashboard*. Gambar 7 menunjukkan tampilan dari halaman login.



Gambar 7. Tampilan Halaman Login

4.8. Hasil Interface Halaman *Dashboard*

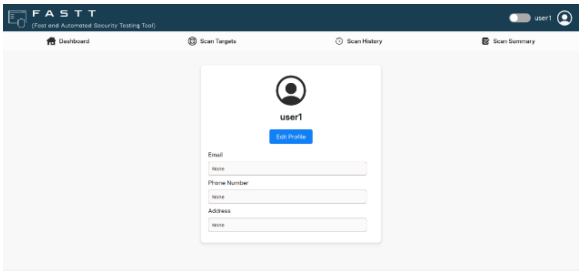
Halaman *Dashboard* merupakan halaman utama pada aplikasi FASTT. Gambar 8 menunjukkan tampilan halaman *dashboard*.



Gambar 8. Tampilan Halaman Dashboard

4.9. Hasil Interface Halaman *Profile*

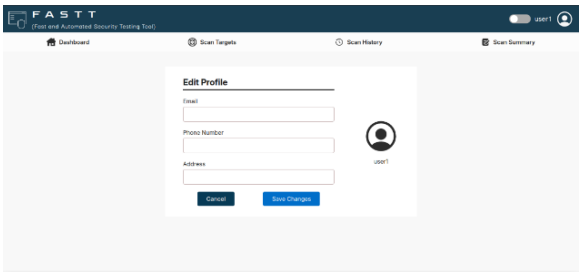
Halaman *Profile* merupakan halaman yang menunjukkan biodata *user*. Gambar 9 menunjukkan tampilan halaman *profile*.



Gambar 9. Tampilan Halaman Profile

4.10. Hasil Interface Halaman *Edit Profile*

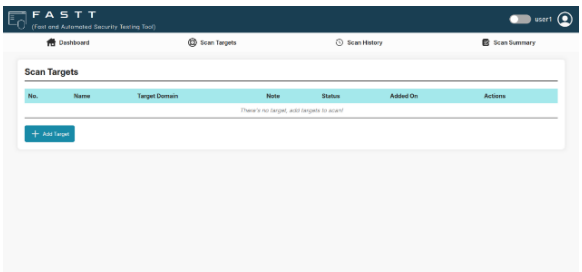
Halaman *Edit Profile* merupakan halaman untuk mengubah biodata user berupa email, nomor telepon, dan alamat. Gambar 10 menunjukkan halaman dari *Edit Profile*.



Gambar 10. Tampilan Halaman Edit Profile

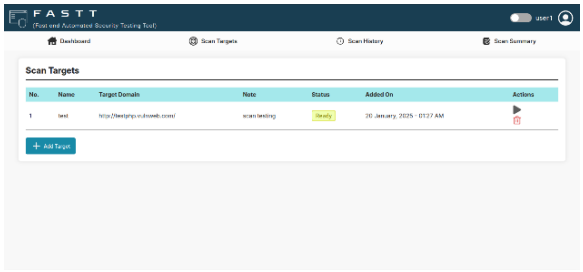
4.11. Hasil Interface Halaman *Scan Targets*

Halaman *Scan Targets* merupakan halaman dari sekumpulan target yang telah ditambahkan dalam bentuk tabel. Gambar 11 menunjukkan tampilan halaman *Scan Targets*.



Gambar 11. Tampilan Halaman Scan Targets

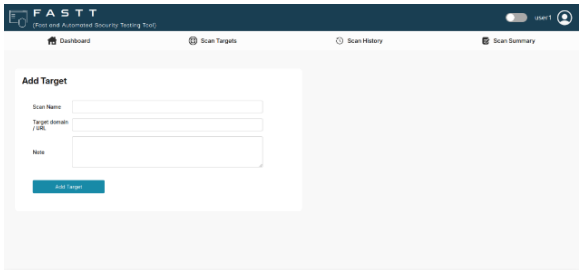
Adapun tampilan halaman *Scan Targets* ketika terdapat sebuah target yang ditambahkan ditunjukkan di Gambar 12.



Gambar 12. Tampilan Halaman Scan Targets dengan Target

4.12. Hasil Interface Halaman *Add Target*

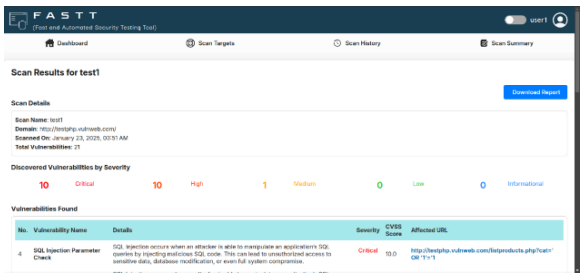
Halaman *Add Target* merupakan halaman untuk menambah target *scanning*. Gambar 13 menunjukkan tampilan halaman *Add Target.*



Gambar 13. Tampilan Halaman Add Target

4.13. Hasil Interface Halaman *Scan Results*

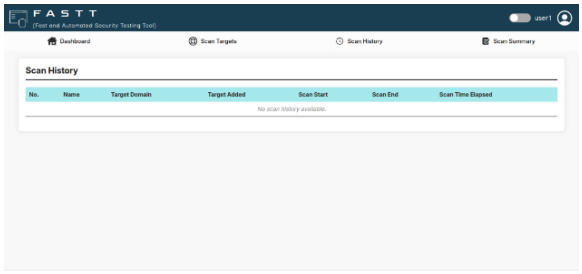
Halaman *Scan Results* merupakan halaman yang menunjukkan detail dari celah yang ditemukan. Gambar 14 menunjukkan tampilan halaman dari *Scan Results.*



Gambar 14. Tampilan Halaman Scan Results

4.14. Hasil Interface Halaman *Scan History*

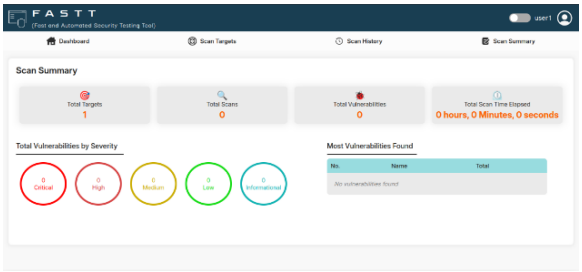
Halaman *Scan History* merupakan halaman untuk melihat riwayat dari hasil scanning yang pernah dilakukan sebelumnya yang disajikan dalam bentuk tabel. Gambar 15 menunjukkan tampilan halaman *Scan History.*



Gambar 15. Tampilan Halaman Scan History

4.15. Hasil Interface Halaman *Scan Summary*

Halaman *Scan Summary* merupakan halaman yang menunjukkan rekapan hasil scanning dari seluruh target. Gambar 16 menunjukkan tampilan halaman *Scan Summary*.



Gambar 16. Tampilan Halaman Scan Summary

4.16. Hasil Pengujian Fitur Aplikasi

Pada penelitian ini menggunakan metode *Black Box Testing* sebagai metode pengujian aplikasi. Hasil pengujian dari fitur-fitur aplikasi FASTT ditampilkan pada Tabel 1.

Tabel 1. Hasil Pengujian Fitur Aplikasi

Komponen	Komponen yang diuji	Hasil
Menu *create*	Formulir pendaftaran *username* dan *password*	Berhasil
Menu *login*	Form *login username* dan *password*	Berhasil
Menu *profile*	Halaman profil	Berhasil
Menu *edit profile*	Formulir edit profil	Berhasil
Menu menambah target *scanning*	Formulir *input* URL	Berhasil
Menu *scanning*	Tombol scan target dan proses *scanning*	Berhasil
Menu melihat riwayat hasil *scanning*	Tampilan hasil riwayat *scanning*	Berhasil
Menu menghapus hasil *scanning*	Tombol fungsi penghapusan.	Berhasil
Menu melihat laporan hasil *scanning*	Tampilan laporan	Berhasil
Menu men-*download* laporan hasil *scanning*	Tombol *download*	Berhasil

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan yang sudah dilakukan dalam penelitian ini, kesimpulan yang dapat diambil adalah penelitian ini berhasil mengembangkan aplikasi *vulnerability scanner open source* bernama FASTT yang dapat dijalankan di localhost menggunakan Docker yang dapat mengidentifikasi celah keamanan pada website berdasarkan daftar dari OWASP Top 10 menggunakan metode Agile. Hasil pengujian fitur melalui *black-box* testing menunjukkan bahwa aplikasi FASTT berjalan sesuai dengan fungsionalitasnya. Hasil pengujian *scanning* aplikasi pada website yang berbeda menunjukkan bahwa aplikasi FASTT dapat mendeteksi celah keamanan dalam kategori *Low, Medium, High, dan Critical*. Dalam penelitian ini, penulis menyadari bahwa terdapat sejumlah kekurangan yang perlu diperbaiki dan banyak aspek yang masih dapat dieksplorasi lebih lanjut. Hal ini menunjukkan bahwa penelitian ini

memiliki potensi untuk dikembangkan dengan lebih mendalam di masa depan. Dengan mempertimbangkan hal tersebut, penulis ingin menyampaikan beberapa saran yang dapat dijadikan acuan untuk penelitian selanjutnya. Saran-saran ini diharapkan dapat membantu peneliti lain dalam memperluas cakupan penelitian dan meningkatkan kualitas hasil yang diperoleh. Beberapa rekomendasi yang dapat dipertimbangkan antara lain mengembangkan metode scanning agar akurasi deteksi dapat meningkat, membuat tampilan menjadi lebih *responsive* dan interaktif, dan menambah jumlah template agar dapat menguji celah keamanan yang beragam.

DAFTAR PUSTAKA

[1] Herman, I. Riadi, Y. Kurniawan, and A. I. Rafiq, "Analisis Keamanan Website Menggunakan Information System Security Asessment Framework (ISSAF)," *Jurnal Teknlogi Informatika dan Komputer MH. Thamrin*, vol. 9, no. 1, pp. 126–136, 2023.

[2] M. Aziz, "Vulnerability Assesment Untuk Mencari Celah Keamanan Web Aplikasi E-Learning Pada Universitas Xyz," *Jecsit*, vol. 1, no. 1, pp. 101–109, 2021.

[3] Mira Orisa and M. Ardita, "Vulnerability Assesment Untuk Meningkatkan Kualitas Kemanan Web," *Jurnal Mnemonic*, vol. 4, no. 1, pp. 16–19, 2021, doi: 10.36040/mnemonic.v4i1.3213.

[4] K. A. Suputri, M. D. Maharani, G. A. Pratama, N. D. I. Sudiasta Putri, I. M. E. Listartha, and G. A. J. Saskara, "Perbandingan Tools Vulnerability Scanning Pada Pengujian Sebuah Website," *Informatik : Jurnal Ilmu Komputer*, vol. 18, no. 3, p. 269, 2022, doi: 10.52958/iftk.v18i3.5133.

[5] J. Zhao and C. Liu, "Design and Implementation of SQL Injection Vulnerability Scanning Tool," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1742-6596/1575/1/012094.

[6] A. G. Buja, N. S. Khairuddin, N. A. Deraman, K. A. F. A. Samah, M. N. H. H. Jono, and N. A. M. Rashid, "Development of a browser extension for web application vulnerability detection, avoidance, and secure browsing (Vdas)," *International Journal of Advanced Technology and Engineering Exploration*, vol. 8, no. 77, pp. 537–544, 2021, doi: 10.19101/IJATEE.2020.762187.

[7] H. Patah and M. Zaid, "Perancangan Aplikasi Deteksi Kerentanan Sql Injection Dan Cross-Site Script Pada Aplikasi Berbasis Website Menggunakan Metode Waterfall," 2021.

[8] J. Chen, X. Chen, and B. Yu, "Design of web vulnerability scanner based on go language," *MATEC Web of Conferences*, vol. 336, p. 08010, 2021, doi: 10.1051/matecconf/202133608010.

[9] R. A. Ramdhani and N. Nurdiana, "RANCANG BANGUN APLIKASI ANALISIS STANDAR KEAMANAN WEBSITE DENGAN METODE SCANNING VULNERABILITY MENGGUNAKAN MODULE REQUESTS PYTHON," 2022.

[10] T. G. Laksana and S. Mulyani, "PENGETAHUAN DASAR IDENTIFIKASI DINI DETEKSI SERANGAN KEJAHATAN SIBER UNTUK MENCEGAH PEMBOBOLAN DATA PERUSAHAAN," *Jurnal Ilmiah Multidisiplin*, vol. 3, no. 1, pp. 109–122, 2024, doi: https://doi.org/10.56127/jukim.v3i01 .1143.

[11] R. Ashar, "Analisis Keamanan Open Website Menggunakan Metode OWASP dan ISSAF," *Jurnal Informasi dan Teknologi*, vol. 4, no. 4, pp. 187–194, 2022, doi: 10.37034/jsisfotek.v4i4.233.

[12] Deeptha R., K. Sujatha, D. Sasireka, R. Neelaveni, and R.Pavithra Guru, "Website Vulnerability Scanner," *Journal of Population Therapeutics and Clinical Pharmacology*, vol. 30, no. 15, Jan. 2023, doi: 10.47750/jptcp.2023.30.15.005.

[13] P. Jarupunphol, S. Seatun, and W. Buathong, "Measuring Vulnerability Assessment Tools' Performance on the University Web Application," *Pertanika J Sci Technol*, vol. 31, no. 6, pp. 2973–2993, 2023, doi: 10.47836/pjst.31.6.19.

[14] M. D. Purnomo and A. Chusyairi, "Pengujian Keamanan Sistem Mengunakan Metode Penetration Testing di Website Diskominfostandi Kota Bekasi," *Sistematis : Jurnal Ilmiah Sistem Informasi*, vol. 1, no. 1, pp. 92–101, 2024, doi: 10.69533.

[15] A. Triono, A. Setia Budi, and R. Abdillah, "Jurnal Multidisiplin Riset (JUMRI) IMPLEMENTASI PERETASAN SANDI VIGENERE CHIPHER MENGGUNAKAN BAHASA PEMROGRAMAN PYTHON," vol. 1, no. 1, pp. 1–9, 2023, [Online]. Available: https://jurnal.ittc.web.id/index.php/jumri