SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

ANALISA KEAMANAN APLIKASI BERBASIS WEB MENGGUNAKAN *TOOLS* ARACHNI (STUDI KASUS : DINAS KOMUNIKASI DAN INFORMATIKA DAERAH ISTIMEWA YOGYAKARTA)

Diajukan Untuk

TUGAS AKHIR

Dimas Ramadhani

0110220281

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

SEKOLAH TINGGI TEKNOLOGI TERPADU NURUL FIKRI

ANALISA KEAMANAN APLIKASI BERBASIS WEB MENGGUNAKAN *TOOLS* ARACHNI (STUDI KASUS : DINAS KOMUNIKASI DAN INFORMATIKA DAERAH ISTIMEWA YOGYAKARTA)

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar S1

Dimas Ramadhani

0110220281

PROGRAM STUDI TEKNIK INFORMATIKA

DEPOK

AGUSTUS 2024

HALAMAN PERNYATAAN ORISINALITAS

Skripsi/Tugas Akhir ini adalah hasil karya penulis, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : **Dimas Ramadhani**

NIM : **0110220281**

Depok, Selasa 13 Agustus 2024

Tanda Tangan

Dimas Ramadhani

HALAMAN PENGESAHAN

Skripsi/Tugas Akhir ini diajukan oleh :

Nama : Dimas Ramadhani

NIM : 0110220281

Program Studi : Teknik Informatika

Judul Skripsi : Analisa Keamanan Aplikasi Berbasis Web Menggunakan Tools Arachni (Studi Kasus : Dinas Komunikasi dan Informatika Daerah Istimewa Yogyakarta)

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Strata 1 pada Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri

DEWAN PENGUJI

Pembimbing	Penguji
Henry Saptono, S.Si., M.Kom.	April Rustianto, S.Komp., M.T.

Ditetapkan di : Depok

Tanggal : 23 Juli 2024

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan skripsi/Tugas Akhir ini. Penulisan skripsi/Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana komputer Program Studi Teknik Informatika pada Sekolah Tinggi Teknologi Terpadu Nurul Fikri Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi penulis untuk menyelesaikan skripsi/tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Allah SWT.
2. Orang tua dan semua anggota keluarga yang telah memberikan dorongan baik secara moril maupun materil dalam penyelesaian tugas ini.
3. Bapak Dr. Lukman Rosyidi selaku Ketua Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
4. Ibu Anik Budiati, S.Kom., M.Eng. Selaku Perwakilan dari Dinas Komunikasi dan Informatika Daerah Istimewa Yogyakarta yang telah memperbolehkan menjadi tempat dan subjek penelitian.
5. Ibu Tifani Nabarian, S.Kom, M.T.i. selaku Ketua Program Studi Teknik Informatika Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
6. Bapak Dr. Lukman Rosyidi, M.T., M.M. selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama berkuliah di Sekolah Tinggi Teknologi Terpadu Nurul Fikri.
7. Bapak Henry Saptono, S.Si, M.Kom. selaku Dosen Pembimbing Tugas Akhir penulis dalam menyelesaikan penulisan ilmiah ini.
8. Para Dosen di lingkungan Sekolah Tinggi Teknologi Terpadu Nurul Fikri yang telah membimbing penulis dalam menuntut ilmu yang telah diberikan.

Dalam penulisan ilmiah ini tentu saja masih banyak terdapat kekurangan-kekurangan yang mungkin disebabkan oleh keterbatasan kemampuan dan pengetahuan yang penulis miliki. Walaupun demikian, penulis telah berusaha menyelesaikan penulisan ilmiah ini sebaik mungkin. Oleh karena itu apabila terdapat kekurangan di dalam penulisan ilmiah ini, dengan rendah hati penulis menerima kritik dan saran dari pembaca.

Akhir kata, penulis berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, Selasa 13 Agustus 2024

Dimas Ramadhani

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Sekolah Tinggi Teknologi Terpadu Nurul Fikri, saya yang bertanda tangan di bawah ini:

Nama : Dimas Ramadhani

NIM : 0110220281

Program Studi : Teknik Informatika

Jenis karya : Skripsi / Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada STT-NF **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty - Free Right*)** atas karya ilmiah saya yang berjudul :

Analisa Keamanan Aplikasi Berbasis Web Menggunakan Tools Arachni
(Studi Kasus : Dinas Komunikasi dan Informatika Daerah Istimewa Yogyakarta)

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini STT-NF berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : Selasa, 13 Agustus 2024

Yang Menyatakan

Dimas Ramadhani

ABSTRAK

Nama : Dimas Ramadhani
NIM : 0110220281
Program Studi : Teknik Informatika
Judul : Analisa Keamanan Aplikasi Berbasis Web Menggunakan Tools Arachni (Studi Kasus : Dinas Komunikasi dan Informatika DIY)

Tugas Akhir/Skripsi ini membahas tentang analisa keamanan aplikasi berbasis web, pada era digital saat ini, berkembangnya teknologi memudahkan menerima informasi dari berbagai sumber melalui internet. Namun, semakin meningkatnya teknologi pada informasi dan komunikasi juga muncul ancaman yang menyerang kerentanan sistem. Berdasarkan data statistik laporan anomali traffic yang terjadi di Indonesia berjumlah 21.420.466 *traffic*. Oleh karena itu, peningkatan keamanan sistem menjadi penting sebagaiupaya untuk melindungi sistem dari anomali dan ancaman yang tidak diinginkan. Salah satu instansi yang memanfaatkan *website* sebagai layanan penting adalah Dinas Komunikasi dan Informatika Daerah Istimewa Yogyakarta. Diskominfo DIY memiliki banyak sekali situs untuk menunjang kinerja instansi terkait dan pemerintahan yang tidak luput dari data penting, banyaknya situs yang dikelola oleh Diskominfo DIY menjadikan sebuah pertanyaan, bagaimanakah tingkat keamanan situs yang dikelola, maka dari itu diperlukannya evaluasi kemanan dengan menggunakan metode *Vulnerability Assesment* (VA), Arachni adalah *tools* yang cocok digunakan dalam evaluasi keamanan, penelitian ini berhasil menemukan 291 kerentanan pada beberapa web Diskominfo, kerentanan akan dianalisa menggunakan pendekatan OWASP *top* 10 dan akan diuji coba untuk menghasilkan profil keamanan dan bukti kerentanan yang valid. Hasil peneltian ini VA adalah salah satu metode yang cocok untuk menganalisa keamanan pada aplikasi web yang dikelola Diskominfo DIY, serta profil keamanan yang harus menjadi perhatian adalah pengelolaan hak akses direktori dan file yang ada pada *website*.

Kata kunci : Kerentanan, Evaluasi, Diskominfo DIY, Arachni, OWASP, *Website*, Keamanan

ABSTRACT

Name : Dimas Ramadhani

NIM : 0110220281

Study Program : Informatic

Title : Security Analysis of Web-Based Applications Using Arachni Tools (Study Case: DIY Communication and Informatics Agency)

The focus of this final research are about the security analysis of web-based applications. In the current digital era, technological advancements facilitate the retrieval of information from various sources via the internet. However, with the increasing use of information and communication technology there are vulnerabilities in systems are also rising. Based on statistical data, the report has identified 21.420.466 *anomalous traffic incidents in Indonesia. Therefore, enhancing system security is crucial to protect against unwanted anomalies and threats. One of the institutions that utilize websites as essential services is the Communication and Information Agency of the Special Region of Yogyakarta (Diskominfo DIY). Diskominfo DIY manages numerous sites to support its operational and governmental functions which involve handling important data. The extensive number of sites managed by Diskominfo DIY raises questions about their security levels. Hence, security evaluation using Vulnerability Assessment (VA) methods is necessary. Arachni is identified as a suitable tool for conducting security evaluations. This research successfully identified 291 vulnerabilities across several Diskominfo websites. These vulnerabilities were analyzed using the OWASP top 10 approach and tested to generate security profiles and valid vulnerability evidence. The results of this research indicate that VA is an appropriate method for analyzing the security of web-based applications for Diskominfo DIY websites. Security profile concerns highlighted include the management of directory and file access rights on websites.*

Key words : *Vulnerability, Assesment, Diskominfo DIY, Arachni, OWASP, Websites, Security*

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar belakang

Perkembangan Teknologi Informasi dan Komunikasi secara pesat di Indonesia sekarang ini membuat kemudahan memperoleh serta melihat data yang ada di internet semakin mudah. Informasi secara terpusat ini ditulis dalam *website* yang dipublikasikan melalui internet yang dengan mudah ditemukan di mana saja. Aplikasi yang berkolaborasi dengan *website* diminati karena fleksbilitas akses dari aplikasi, dibanding dengan aplikasi lain, dan aplikasi berbasis web ini sangat ringan untuk digunakan tidak seperti aplikasi yang lain[1]. Hal tersebut membuat bertambahnya jumlah penggunaan layanan aplikasi berbasis web dari tahun ke tahun, karena hanya menggunakan internet menampilkan aplikasi berbasis *website* yang dapat memudahkan pengguna dalam menerima informasi bahkan sampai bertukar informasi. Di era digital seperti sekarang para instansi atau organisasi organisasi semakin yakin untuk membuat aplikasi berbasis *website* yang mempermudah instansi atau organisasi dalam melakukan input data dan menampilkan berbagai macam infomasi untuk para penggunanya.

Dinas Komunikasi dan Informatika Daerah Istimewa Yogyakarta (Diskominfo DIY) adalah lembaga pemerintahan yang berfungsi sebagai sarana dan wadah pemberian layanan informasi kepada masyarakat umum khususnya warga Yogyakarta[2]. Diskominfo DIY memiliki banyak sekali *website* yang bertujuan memberikan informasi bagi warga DIY untuk mendapat informasi seperti kesehatan, pendidikan, bahkan sampai aduan masyarakat. Selain itu Diskominfo DIY dapat menunjang pelayanan masyarakat dan instansi pemerintahan yang terkait, salah satu contohnya dengan layanan penambahan *Subdomain* dan *Hosting*, *Subdomain* bertujuan mengganti IP menjadi nama unik untuk mempermudah mengakses alamat

suatu aplikasi web maupun sistem elektronik, dan *Hosting* bertujuan supaya *file* dan data yang telah di tampilkan oleh suatu *domain* dapat disimpan ke *data center* milik Diskominfo DIY. dilansir dari Rencana Kominfo DIY 2023-2026 *Data Center* menggunakan 9 (sembilan) unit *server* fisik yang diperuntukan mengoperasikan 138 (seratus tiga puluh delapan) *server virtual* dan lebih dari 250 aplikasi dari seluruh Organsasi Perangkat Daerah di wilayah Pemda DIY[3]. Dari data diatas terdapat banyaknya *website* yang dipublikasi dan dikembangkan oleh Diskominfo DIY untuk menunjang kinerja pelayanan masyarakat dan instansi yang terkait. Banyaknya website yang dikelola Diskominfo DIY dan peran penting yang dimiliki oleh Diskominfo DIY mendasari latar belakang pada penelitian ini.

Kebergantungan sarana informasi menggunakan aplikasi berbasis *website* menjadikan isu kerentanan keamanan *website* saat ini sedang marak maraknya terjadi di Indonesia, banyak *website* yang setiap harinya tanpa sadar disusupi oleh aktivitas anomali yang diluar dari kinerja *website* itu sendiri. Dari data Id-SIRTII (*Indonesia Security Incident Response Team On Internet Infrastructure / Coordination Center*), *traffic* yang berpotensi memiliki anomali sejumlah 21.420.466 *traffic*[4]. Data tersebut menyadarkan para pengguna internet dan perusahaan yang mengembangkan aplikasi berbasis web bahwa begitu banyak ancaman bagi sistem yang mereka gunakan dan data pribadi pengguna maupun data perusahaan. Tentu data-data pengguna dan perusahaan yang ada di *website* tersebut menjadi sasaran empuk apabila tidak mementingkan pada sisi keamanannya. Bocornya data membuat kerugian yang sangat besar bagi perusahaan, apabila data yang bocor dimiliki pihak yang tidak bertanggung jawab, dapat berdampak buruk dan dapat merugikan banyak pihak[5]. Dampak kebocoran data ini akan menimbulkan stigma bahwa tingkat keamanan sistem yang ada di Indonesia masih lemah, hal ini membuat pengguna layanan enggan menggunakan cara digital karena data yang akan diolah memiliki potensi mengalami kebocoran. Maka dari itu kini pengembang dan publisher *website* harus menentukan cara

bagaimana mencegah kerentanan pada sistem mereka, metode apa yang akan mereka gunakan untuk mencari informasi tentang kerentanan sistem mereka. perusahaan yang mengembangkan aplikasi berbasis *website* memiliki peran untuk menguji coba setiap *website* yang dimiliki, bahkan melakukan uji coba keamanan bagi *website* baru atau pembaruan *website* menjadi sebuah kewajiban untuk memperkecil kerentanan yang ada.

Uji coba kerentanan memiliki beberapa tahapan *Vulnerability Assesment*. *Vulnerability assessment* dapat mendefinisikan, mengidentifikasi, mengelompokkan dan memprioritaskan kerentanan dalam sistem web[6]. Tentunya *Vulnerability Assesment* harus di uji coba untuk memeriksa kebenarannya, karena beberapa *tools Vulnerability Assesment* memiliki sifat *False Positive*. Dengan adanya data kerentanan pengelola dapat melakukan pengukuran resiko kerentanan menggunakan pendekatan *Open Web Aplication Security Project* (OWASP) untuk mengetahui penanganan dan kategori kerentanan yang ada.

Penelitian ini akan melakukan analisa kerentanan ke beberapa situs yang disediakan oleh Diskominfo DIY untuk dilakukannya analisa menggunakan *tools Vulnerability Assesment* berupa Arachni dengan berfokus pada keamanan dari aplikasi berbasis *website*, dan hasil dari *Vulnerability Assesment* akan dilakukan uji coba dan dianalisa menggunakan pendekatan OWASP oleh peneliti setelah itu akan dikembangkan oleh pihak Diskominfo DIY untuk meminimalisir kerentanan yang ada pada setiap *domain* dan *sub domain* yang dikembangkan oleh Diskominfo DIY.

1.2 Rumusan Masalah

Dari latar belakang diatas peneliti merumuskan beberapa permasalahan seperti berikut.

1. Menentukan metode untuk Analisa keamanan aplikasi berbasis web di Diskominfo DIY.
2. Bagaimanakah profil keamanan dari aplikasi berbasis web di Diskominfo DIY.

1.3 Tujuan dan Manfaat Penelitian

a. Peneliti memiliki tujuan dari penelitian ini sebagai berikut.
    1. Menentukan metode analisia keamanan aplikasi berbasis web.
    2. Mengetahui profil keamanan dari aplikasi berbasis web.

b. Penelitian ini memiliki manfaat seperti berikut.
    1. Menjadikan *Vulnerability Assesment* sebagai pedoman keamanan bagi Diskominfo DIY, untuk upaya meningkatkan tingkat keamanan aplikasi berbasis web.
    2. Mempermudah Diskominfo DIY dalam mempercepat proses kendali keamanan.
    3. Penulis dapat menghasilkan penelitian ini sebagai karya tulis untuk dapat dikembangkan dan disitasi.

1.4 Batasan Masalah

Merujuk rumusan masalah dan tujuan penelitian yang dikemukakan sebelumnya, maka peneliti menganggap perlu untuk memberikan batasan dari penelitian ini sebagai berikut.

1. Metode Analisa keamanan berbasis web ini tidak dilakukan secara manual dan hanya dilakukan secara *automated* (temuan Arachni).
2. Pengujian sistem keamanan aplikasi web menggunakan Teknik *Blackbox*.
3. Peneliti tidak menguji bagian *database*, *OS*, dan jaringan, melainkan hanya aplikasi web.
4. Peneliti hanya menggunakan pendekatan OWASP, dan tidak keseluruhannya, hanya yang memiliki kaitan dengan aplikasi web.
5. Peneliti mendapatkan 4 website yang sudah mendapat izin untuk diteliti dan diangkat menjadi tugas akhir yaitu, dis***dag.jogjaprov.go.id, bir***an.jogjaprov.go.id, rs***ra.jogjaprov.go.id, dan mo***bj.jogjaprov.go.id.

1.5 Sistematika Penulisan

Sistematika penulisan dalam penelitian ini disusun sebagai berikut.

BAB 1 Pendahuluan

Dalam bab ini membahas tentang Latar Belakang Masalah tentang kerentanan yang dibutuhkan setiap instansi penyedia aplikasi berbasis web di Indonesia khususnya Diskominfo DIY, Rumusan Masalah yang disusun untuk bagaimana cara meminimalisir kerentanan yang ada, Tujuan Penelitian yang ditujukan untuk menjawab rumusan masalah yang sudah dibuat, Manfaat Penelitian diharapkan dapat membuat sistem yang terpelihara bagi Diskominfo DIY dengan cara melakukan kendala keamanan, dan Sistematika Penulisan yang menjelaskan isi tiap bab.

BAB 2 Kajian Literatur

Dalam bab ini membahas tentang teori tentang kerenatan aplikasi berbasis web dan data – data kerentanan yang berkaitan dengan pembahasan dari penelitian yang dilakukan, selain membahas tentang kerentanan, pada bab ini membahas tentang WAVS (*Web Application Vulnerability Scanner*) yang akan menjadi alat untuk pencarian kerentanan dan profil dan latar belakang instansi yang menjadi subjek penelitian.

BAB 3 Metodologi Penelitian

Dalam bab ini berisi tahapan dan rancangan penelitian serta pemaparan metode *Vulnerability Assesment* yang digunakan untuk melakukan penelitian, pencarian, dan pengumpulan data yang berkaitan dengan penelitian.

BAB 4 Implementasi dan Analisa

Dalam bab ini penulis menjelaskan analisa dari hasil *Vulnerability Assesment* menggunakan pendekatan kerentanan aplikasi berbasis web yaitu OWASP serta melakukan uji coba terhadap kerentanan agar memvalidasi kerentanan yang ada.

BAB 5 Penutup

Dalam bab ini penulis memberikan saran dan kesimpulan dari penelitian yang dilakukan.

BAB II
KAJIAN LITERATUR

2.1 Definisi

Peneliti memberikan penjelasan dasar terkait informasi dan teori yang digunakan dalam peneltian ini.

2.1.1 Aplikasi Berbasis *Website*

Website adalah contoh dari salah satu media yang memiliki banyak halaman dan saling berkaitan antara halaman utama dan halaman menu pada *website*, *website* juga dapat menampilkan informasi dengan berbagai macam bentuk, mulai dari tulisan berupa teks, gambar, dokumen, video, suara, dan animasi. Dalam satu *website* dapat di atur posisi dan media apa saja yang digunakan untuk memberikan kenyamanan penyampaian informasi kepada pengguna *website*. *Website* memiliki komponen berupa *domain* yang berfungsi merubah alamat *website* dari angka menjadi nama serta menampilkan alamat dari *website* (URL), *Hosting* berfungsi sebagai media penyimpanan bagi *website*. Untuk mengembangkan situs web dalam mode penerbitan Internet, diperlukan beberapa aplikasi yaitu *Web Server*, *Database*, dan *Browser*[7].

Aplikasi berbasis web adalah aplikasi yang dibuat dengan mengimplemetasikan bahasa pemrograman HTML, PHP, CSS, JS untuk membuat *website* dan memerlukan *web server* sebagai wadah dan *browser* untuk menjalankannya, seperti *Chrome*, *Firefox* atau *Opera*, *Internet Explorer*, *Microsoft Edge* dan *browser* lainnya[8]. Aplikasi web merupakan salah satu *website* dengan ekosistem yang terstruktur dalam bentuk program dan infrastruktur komputer yang memungkinkan pengguna *website* bisa melakukan interaksi serta menampilkan data dari suatu *database server*, dengan cara menggunakan fitur yang ada didalamnya melalui koneksi internet dan mengaksesnya menggunakan alamat *website* yang dimasukan ke *browser*. Kemudian data yang sudah dimasukan akan ditampilkan kembali ke

pengguna *website* sebagai informasi yang dihasilkan secara dinamis oleh aplikasi web melalui web browser. Berbeda dengan aplikasi APK yang mana program APK harus melakukan instalasi pada perangkat, aplikasi web ini dapat di akses dan digunakan hanya menulis alamat *website* menggunakan browser dengan jaringan internet maupun jaringan lokal yang disediakan oleh perusahaan. Penggunaan aplikasi berbasis web memudahkan pemusatan informasi yang ada di perusahaan untuk disebar luaskan ke pengguna layanan aplikasi web, informasi akan ditampilkan dalam sebuah tampilan *website* untuk mempercepat kinerja perusahaan dan pertukaran informasi. Selain itu kegunaan dari aplikasi web ini adalah pemusatan data, data yang terpusat dan kemudahan akses data adalah fitur utama yang membuat aplikasi web lebih populer dan lebih mudah diterapkan di berbagai bidang kehidupan[9]. Banyak sekali bidang kehidupan yang terbantu dengan adanya aplikasi web ini mulai dari bidang kesehatan, bidang ekonomi, bidang sosial, bidang kebudayaan, dan bidang Pendidikan, salah satu contohnya Diskominfo DIY yang menjadi penyedia *website* untuk menunjang informasi dan data secara terpusat bagi masyarakat DIY.

2.1.2 Dinas Komunikasi dan Informatika DIY

Dinas Komunikasi dan Informatika DIY merupakan unsur pelaksana urusan Pemerintahan Daerah di bidang komunikasi, informatika dan persandian di Daerah Istimewa Yogyakarta. Pembentukan Dinas Komunikasi dan Informatika DIY merupakan implementasi dari Undang-Undang Nomor 23 Tahun 2014 tentang Pemerintahan Daerah dan Peraturan Pemerintah Nomor 18 Tahun 2016 tentang Perangkat Daerah yang mengamanatkan kepada setiap pemerintah daerah untuk menyelenggarakan urusan pemerintahan wajib yang tidak berkaitan dengan pelayanan dasar, antara lain mencakup komunikasi dan informatika, statistik dan persandian.

2.1.2.1 Tugas

Dinas Komunikasi dan Informatika DIY mempunyai tugas membantu Gubernur DIY untuk melaksanakan urusan pemerintahan bidang komunikasi dan informatika dan urusan pemerintahan bidang persandian.

2.1.2.2 Fungsi

Diskominfo memiliki banyak fungsi seperti:

1. Penyusunan program kerja dinas.
2. Perumusan kebijakan teknis bidang komunikasi dan informatika serta urusan persandian.
3. Pelayanan pengelolaan informasi dan komunikasi publik.
4. Penyelenggaraan ekosistem provinsi cerdas.
5. Penyelenggaraan teknologi informasi dan komuniaksi yang terintegrasi dalam Sistem Pemerintahan Berbasis Elektronik.
6. Pelayanan keamanan informasi dan persandian
7. Fasilitasi integrasi data dan informasi elektronik
8. Pelaksanaan koordinasi, pembinaan, dan pengawasan urusan pemerintahan bidang komunikasi, informatika dan persandian yang menjadi kewenangan Pemerintah Kabupaten/Kota.
9. Pelaksanaan kegiatan kesekretariatan.
10. Pelaksanaan dekonsentrasi dan tugas pembantuan.
11. Fasilitasi pembinaan reformasi birokrasi Dinas
12. Fasilitasi penyusunan kebijakan proses bisnis Dinas.
13. Pemantauan, evaluasi, dan penyusunan laporan pelaksanaan kebijakan bidang komunikasi, informatika dan persandian.
14. Penyusunan laporan pelaksanaan tugas Dinas.
15. Pelaksanaan tugas lain yang diberikan oleh Gubernur sesuai dengan tugas dan fungsi Dinas.

2.1.2.3 Visi

Visi Gurbernur yang dibantu oleh Diskominfo DIY adalah Terwujudnya PANCAMULIA bagi masyarakat Jogja melalui Reformasi Kalurahan, Pemberdayaan Kawasan Selatan, serta Pengembangan Budaya Inovasi dan Pemanfaatan Teknologi Informasi.

2.1.2.4 Misi

Misi dari gubernur yang dibantu oleh diskominfo DIY meliputi.

1. Meningkatkan kualitas hidup-kehidupan-penghidupan, pembangunan yang inklusif dan pengembangan kebudayaan melalui reformasi kalurahan.
2. Memberdayakan Kawasan Selatan dengan mengoptimalkan dukungan infrastruktur, peningkatan kapasitas SDM, dan perlindungan /pengelolaan sumber daya setempat.
3. Meningkatkan budaya inovasi dan mengoptimalkan kemanfaatan kemajuan teknologi informasi.
4. Melestarikan lingkungan dan warisan budaya melalui penataan ruang dan pertanahan yang lebih baik.

2.1.2.5 Tujuan

Terwujudnya penyelenggaraan pemerintahan yang transparan dan akuntabel berbasis teknologi informasi dan komunikasi (TIK).

2.1.2.6 Sasaran

Target atau sasaran yang dimiliki oleh Diskominfo DIY meliputi.

a. Meningkatnya penyelenggaraan sistem pemerintahan berbasis elektronik (SPBE).
b. Meningkatnya keterbukaan informasi penyelenggaraan pemerintahan.
c. Meningkatnya tata kelola penyelenggaraan urusan pemerintahan di perangkat daerah.

Diskominfo DIY memiliki berbagai macam jenis aplikasi berbasis *website* yang paling banyak adalah berjenis profil perusahaan dan sistem informasi, penelitian ini menguji 4 *website* yang diantaranya 3 profil perusahaan dan 1 sistem informasi, semua *website* yang akan diteliti memiliki *domain* *.jogjaprov.go.id. Setelah peneltian menghasilkan data maka penulis melakukan analisa menggunakan pendekatan OWASP untuk mengukur tingkat kerentanan yang ada pada *website* tersebut.

2.1.3 Konsep Keamanan dan Kerentanan Aplikasi berbasis web

2.1.3.1 Konsep Keamanan

Menurut Kamus Besar Bahasa Indonesia, kata dasar keamanan atau aman memiliki arti bebas dari ancaman, bahaya, dan bebas dari gangguan. Keamanan memiliki arti dalam keadaan aman atau dalam keadaan ketentraman. Keamanan aplikasi web adalah sebuah gagasan untuk menciptakan *website* sesuai dengan fungsinya meskipun *website* sedang diganggu. Konsep ini mengandalkan kontrol keamanan yang direkayasa ke dalam sistem aplikasi web untuk melindungi asset dan datanya dari orang yang tidak dikenal dan berpotensi berbuat jahat. Keamanan aplikasi web adalah pengelolaan sumber daya, termasuk alat, infrastruktur, dan data komputer, untuk melindungi sifat kerahasiaan, integritas, dan ketersediaan informasi dengan mengimplementasi aplikasi penunjang, pendidikan terkait keamanan, dan teknologi yang digunakan sesuai dengan pedoman teknis yang berlaku[10]. Terdapat 4 klasifikasi keamanan pada sebuah sistem yaitu[11] :

1. Keamanan Fisik

   Keamanan yang berfokus pada pihak *external* termasuk pihak yang berkaitan dengan alat dan media yang digunakan untuk menjalankan sistem.

2. Keamanan Sosial

   Keamanan yang berfokus kepada pereorangan dan pihak *internal*, kerentanan ini berkaitan dengan indentitas dan profil dari pihak yang

memiliki akses, karena semua asset dan data bergantung pada pemegang akses.

3. Keamanan Data

   Keamanan yang berfous untuk mengolah data, menyimpan data dan akses data, kerentanan ini berkaitan dengan data yang dimodifikasi serta penambahan dan pengurangan data.

4. Keamanan Operasi

   Keamanan yang berfokus pada prosedur atau aturan yang berlaku untuk mengatur dan mengelola keamanan pada sistem. Namun kelemahan sistem aplikasi berbasis web masih ada dan bisa saja menjadi senjata yang dapat disalahgunakan oleh pihak yang tidak bertanggung jawab, menjadikan kelemahan pada sistem aplikasi berbasis web sebuah kerentanan yang sangat berbahaya.

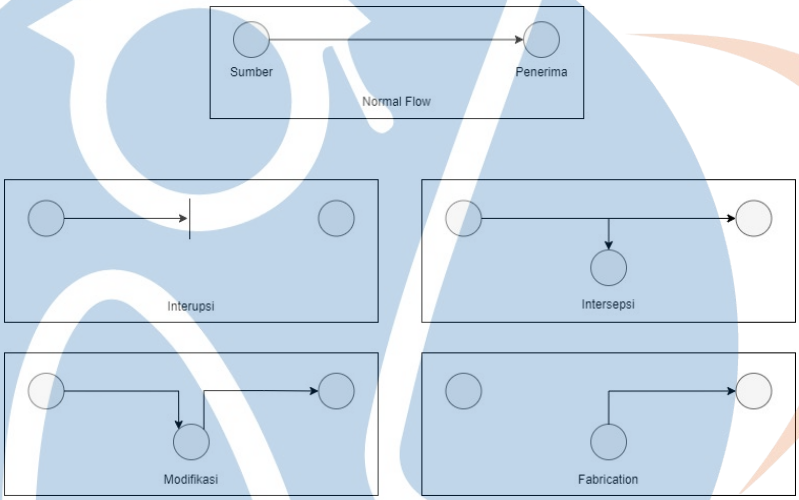
2.1.3.2 Konsep Kerentanan

Menurut Kamus Besar Bahasa Indonesia, kata dasar kerentanan atau rentan memiliki arti mudah terinfeksi, dan peka atau mudah merasa. Kerentanan sendiri memiliki arti mudah terinfeksinya sesuatu yang menghasilkan akibat yang tidak diduga secara tiba - tiba. Kerentanan aplikasi web adalah kelemahan yang dapat disalahgunakan dan memiliki dampak yang mempengaruhi kinerja, data dan informasi yang ada di aplikasi berbasis web secara tiba tiba dan tidak diketahui. Rendahnya kesadaran akan kemanan siber pada pengguna dan pembuat aplikasi berbasis web membuat kerentanan dapat dengan mudah ditemukan. Apabila kerentanan tidak dapat dikendalikan maka akan dapat merugikan semua pihak yang berkaitan dengan sistem tersebut. Keberagaman serangan dan ancaman dari luar ada banyak sekali perlu di ketahui beberapa tipe dan jenis serangan yang ada, untuk mengetahui langkah pencegahan serangan dan ancaman[12].

2.1.3.3 Tipe – Tipe Kerentanan

A. Ancaman Keamanan

Ancaman keamanan adalah tindakan jahat yang berpotensi merusak sistem atau aset pada aplikasi berbasis web, Jenis-jenis ancaman biasanya menyerang dapat dikategorikan dalam empat kategori berdasarkan kriteria target serangan yaitu[13]:



Gambar 2.1 Klasifikasi Ancaman[13]

1. *Interruption*

Interusi adalah upaya yang mengganggu dan mengancam ketersediaan informasi. Pihak yang tidak bertanggung jawab dapat mengganggu dengan melakukan pemberhentian proses pemberian informasi kepada penerima informasi.

2. *Interception*

Intersepsi adalah upaya yang mengancam kerahasiaan. Pihak yang tidak bertanggung jawab dapat mengakses informasi yang bukan miliknya, apabila informasi yang sudah diakses dapat dilihat dengan jelas atau diunduh maka berpotensi untuk melakukan hal-hal yang merugikan pihak lain.

3. *Modification*

Modifikasi adalah upaya yang mengancam isi dari sistem. Pihak yang tidak bertanggung jawab dapat mengakses informasi dan dapat juga merubah

isi dari informasi yang akan di kirim dan diterima dengan maksud tertentu dan dapat merugikan pihak lain.

4. *Fabrication*

Upaya yang mengancam keaslian informasi. Pihak yang tidak bertanggung jawab membuat informasi palsu dengan mengatasnamakan suatu pihak, agar penerima informasi percaya dan memberikan informasi lebih.

B. Serangan Keamanan

Serangan dibagi menjadi dua tipe yang dikemukakan oleh William Stalling dalam salah satu bukunya yang membahas kriptografi dan keamanan jaringan, diantaranya adalah serangan aktif dan pasif[12].

1. Aktif

Serangan yang berupaya untuk mengubah dan mempengaruhi proses pada suatu sistem yang sedang beroprasi.

2. Pasif

Serangan yang berupaya untuk mempelajari untuk menggunakan informasi yang ada pada sistem namun tidak mempengaruhi proses dari sistem.

Beberapa contoh serangan yang di jelaskan oleh paryati di tahun 2008, tidak luput dari jenis keaktifannya, contoh serangan aktif dan pasif diantaranya adalah [14].

1. *Interusion*

Serangan aktif yang membobol akses sistem untuk masuk ke dalam sistem.

2. *Denial of Service*

Serangan aktif yang dapat melumpuhkan sistem.

3. *Hijacking*

Serangan pasif yang dapat memonitor sistem untuk memata matai pergerakan atau aktifitas pengguna pada sistem.

4. *Sniffing*

Serangan pasif ini dapat menangkap paket yang berlalulalang di situs *website*.

5. *Spoofing*

Serangan pasif yang memberikan informasi palsu berupa laman web, hal ini bertujuan untuk mendapatkan informasi rahasia dari pengguna laman web.

6. *Defacing*

Serangan pasif yang membuat beberapa fitur pada laman *website* tidak berfungsi.

7. *Virus*

Serangan aktif yang berupa kode yang sudah diprogram dan di jalankan tanpa sengaja oleh pengguna sistem, menyebabkan sistem bekerja diluar kehendak, banyak fiur tidak bekerja, kehilangan data secara mendadak dan, sistem operasi yang digunakan terasa berat.

8. *Trojan*

Serangan aktif dengan teknik penyamaran agar tidak dideteksi sebagai bahaya oleh sistem yang tujuannya merusak sistem.

9. *Worm*

Serangan aktif yang berupa program duplikasi media atau penambahan kinerja sistem secara extrim yang dapat merusak dan memperlambat kinerja sistem.

2.1.3.4 Penerapan Keamanan

Keamanan perlu di terapkan ke semua sistem, baik skala kecil maupun besar, keamanan sistem membuat kinerja sistem optimal, dan meminimalisir hal hal yang tidak diinginkan. Keamanan pada sistem juga melindungi semua asset yang ada pada sistem, data, informasi, kode, akun dan hal yang bersifat penting pada sistem. Keamanan sistem sesuai dengan apa yang disampaikan oleh G. J. Simons adalah bagaimana kita dapat mencegah ancaman dan serangan atau setidaknya melindungi sistem dari penyimpangan dan anomali

terhadap sumber daya dalam sebuah sistem, dalam usaha pencegahan maupun penanganan terhadap keamanan sistem, pihak yang membuat sistem perlu mengimplementasi tiga aspek penting dalam keamanan sistem, dengan sebutan CIA (*Confidentiality*, *Integrity*, *Availability*), baik saat proses pembuatan sistem sampai pada fase pemeliharaan sistem[15].

Gambar 2.2 Konsep segitiga CIA[15]

1. *Avaibility*

Ketersediaan (*Availability*). Merupakan aspek yang menjamin untuk ketersediaan data dan informasi saat dibutuhkan, aspek ini bertujuan agar sistem selalu bekerja secara optimal setiap saat.

2. *Integrity*

Integritas (*Integrity*). Merupakan aspek yang menjamin orisinalitas data, dengan cara untuk tidak adanya pengubahan data dengan tidak adanya izin dan sepengetahuan dari pihak yang berwenang, aspek ini bertujuan untuk menjaga keakuratan dan keutuhan informasi.

3. *Confidentiality*

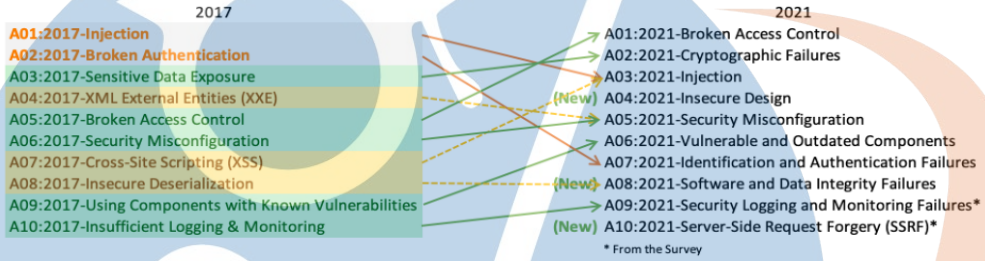
Kerahasiaan (*Confidentiality*). Merupakan aspek yang memastikan bahwa informasi hanya dapat diakses oleh orang yang berwenang.

2.1.4 OWASP

Open Worldwide Application Security Project (OWASP) adalah komunitas terbuka yang membahas secara detail bagaimana untuk mempermudah instansi untuk menyusun, mengembangkan, memperoleh,

mengoperasikan, dan memelihara aplikasi yang aman dan optimal. Semua yang ada dalam OWASP seperti, alat, dokumen, forum, bersifat gratis dan terbuka bagi siapa saja yang tertarik untuk meningkatkan keamanan aplikasi.

OWASP sering menampilkan *top* 10 kerentanan aplikasi berbasis web yang sering terjadi di dunia berdasarkan survey dari komunitas OWASP yang ada. Data *top* 10 kerentanan yang terjadi di dunia tahun 2021 seperti gambar berikut[16].



Gambar 2.3 Top 10 kerentanan OWASP[16]

Kategori yang ada pada map top 10 dari OWASP adalah :

A01:2021-*Broken Access Control*.

A02:2021-*Cryptographic Failures*.

A03:2021-*Injection*.

A04:2021-*Insecure Design*.

A05:2021-*Security Misconfiguration*.

A06:2021-*Vulnerable and Outdated Components*.

A07:2021-*Identification and Authentication Failures*.

A08:2021-*Software and Data Integrity Failures*.

A09:2021-*Security Logging and Monitoring Failures*.

A10:2021-*Server-Side Request Forgery*.

OWASP menjadi pedoman dan perbandingan kerentanan yang ditemukan oleh Arachni dengan data top 10 OWASP untuk menentukan kerentanan mana saja yang paling sering disalahgunakan dan sedang marak terjadi, untuk meningkatkan resistensi serangan yang sedang marak.

2.1.5 Metode Pengujian Kerentanan Aplikasi berbasis web

Pada pengujian kerentanan aplikasi berbasis web terpadat tiga teknik pengujiannya yakni *white box*, *black box*, dan *grey box*[17]:

1. *White Box*

White box testing merupakan pengujian yang berfokus pada sistem inti seperti *source code* program dan program pendukung sistem. Tujuan dari pengujian *white box* sebagai alat ukur keamanan dari kode program serta mengevaluasi sistem berdasarkan fungsionalitas penggunaan *source code* dan *software* pendukung dengan tepat, *whitebox* dilakukan secara bersama sama dengan kolaborasi dan dukungan informasi scara lengkap oleh pihak *internal*.

2. *Black Box*

Black Box adalah pengujian fungsional sistem. *Black box* ini menguji keseluruhan fungsi sistem seperti *database*, aplikasi web, *software*, secara lengkap.

3. *Grey Box*

Grey Box adalah pengujian yang dikombinasi dari pengujian *Black Box* dan pengujian *White Box*, menguji sistem secara keseluruhan berdasarkan spesifikasi namun dengan ijin yang valid, perbedaan dari *white box* adalah pengujian ini hanya mendapatkan beberapa informasi terkait sistem yang akan dilakukan pengujian.

Berdasarkan teori diatas, penelitian ini lebih condong dan mengarah ke pendekatan *black box*, karena yang dilakukan uji coba fungsionalitas keamanan sebuah aplikasi berbasis web, dan pihak Diskominfo DIY sudah memberikan ijin dan informasi terkait apa saja yang harus di uji kerentanannya.

2.1.6 Vulnerability Assesment

Vulnerability Assesment (VA) adalah analisis keamanan yang komprehensif dan mendalam seperti keamanan informasi, hasil analisis jaringan, metode manajemen, konfigurasi sistem, kesadaran keamanan aktor terkait dan keamanan fisik, untuk mengembangkan identifikasi semua potensi

kelemahan serius yang ada[18]. VA menjadi alat untuk pengembang *website* dan sistem untuk mencari kerentanan dari sistem dan web yang sedang masa testing, dengan cara kerja yang berfokus untuk mencari semua kerentanan pada subjek sistem yang menggunakan jaringan baik internet maupun lokal [19]. VA juga merupakan bagian dari manajemen *preventif* dalam pengendalian keamanan TI secara keseluruhan, VA menjadi metode untuk manajemen keamanan yang reliable dan selayaknya deteksi dengan IDS (*Intrusion Detection System*) dan pencegahan dengan *firewall* dan *antivirus*[20]. Waktu yang tepat untuk melakukan VA adalah pada saat masa implementasi sistem atau program baru pada sistem utama, idealnya VA sebaiknya dilakukan secara kontiniu, dan dilakukan secara berkala agar mengetahui keberadaan kerentanan yang ada pada sistem. Sangat berbahaya apabila ada pihak lain yang dapat melakukan eksploitasi untuk membongkar sistem melalui kerentanan yang tidak di sadari lebih dulu oleh pihak pengembang dan pengelola.

2.1.6.1 Kategori Kerentanan VA

VA dapat mengidentifikasi beberapa kategori kerentanan yang memiliki *severity*/dampak yang besar untuk sistem, berdasarkan data dari GitLab[21] dampak dikategorikan menjadi 5, yaitu :

1. *Critical*

   Kerentanan pada tingkat ini diasumsikan sebagai eksploitasi kelemahan yang dapat menyebabkan kebocoran dan penyalahgunaan pada sistem atau data secara penuh.

2. *High*

   Kerentanan pada tingkat ini dikategorikan sebagai kelemahan yang dapat menyebabkan penyerang mengakses sumber daya aplikasi atau akses ke data dan informasi.

3. *Medium*

   Kerentanan pada tingkat ini biasanya timbul karena kesalahan konfigurasi sistem atau kurangnya kontrol keamanan, kerentanan ini dapat

menyebabkan kerusakan akses pada sistem, yang mana dapat memberikan kemudahan untuk melakukan exploitasi lebih lanjut.

4. *Low*

Kerentanan pada tingkat ini berisi kelemahan yang mungkin tidak dapat dieksploitasi secara langsung namun menimbulkan kelemahan yang tidak perlu pada aplikasi atau sistem, kerentanan ini biasanya disebabkan oleh hilangnya kontrol keamanan, atau pengungkapan informasi yang tidak perlu dalam lingkungan sistem.

5. *Informational*

Kerentanan pada tingkat ini berisi informasi yang mungkin memiliki nilai, namun belum tentu terkait dengan kerentanan atau kelemahan tertentu.

2.1.6.2 Jenis Jenis Kerentanan

VA dapat mendeteksi berbagai jenis kerentanan yang ada pada aplikasi berbasis web, terutama pada *tools* yang berfokus mencari kerentanan pada aplikasi berbasis web atau WAVS (*Web Application Vulnerability Scanner)*, pada penelitian pendeteksian kerentanan oleh disebut bahwa WAVS berhasil menangkap beberapa kerentanan yang ada pada *website testing*, jenis kerentanan yang bisa di deteksi oleh WAVS meliputi[22].

1. *XSS*
2. *SQL Inject / NoSQL Inject*
3. *Dictionary Traversal*
4. *Traversal Code*
5. *Command Injection*
6. *Authenthication Inject*
7. *Authorization*
8. *Captcha*
9. *File Include*
10. *Database Server*

VA sangat mempermudah mendeteksi kerentanan yang ada pada aplikasi berbasis web, terutama dengan menggunakan WAVS, yang berfokus

pada kerentanan aplikasi web, ada banyak sekali *tools* dari WAVS ini, salah satunya adalah Arachni, Arachni salah satu dari WAVS yang andal dan *powerfull* dengan hasil yang teknis serta memiliki tingkat akurasi yang terbilang tinggi. Arachni bisa di bilang juga sebagai spesialis untuk menemukan kerentanan terutama pada aplikasi berbasis web.

2.1.7 Arachni

Arachni adalah salah satu *Web Aplication Vulnerability Scanner* (WAVS) yang bersifat *open source* dan gratis. Arachni dapat menelusuri kerentanan setiap jalur yang ada pada sistem web sampai ke sistem yang menjalankan web itu sendiri.

Arachni unggul dalam mengidentifikasi kerentanan dari daftar OWASP 2021, menjadikannya aset berharga dalam meningkatkan keamanan aplikasi web[23]. Arachni di bangun menggunakan bahasa pemrograman Ruby dan dapat berjalan di beberapa *platform* seperti Windows, Linux dan Mac OS, akan tetapi untuk mendapatkan potensi maksimalnya dianjurkan untuk menggunakan Linux. Arachni merubah namanya menjadi *Codename* SCNR dan kini berbayar, namun *source code* untuk menggunakan Arachni masih bisa digunakan pada laman *github* https://github.com/Arachni.

Pemilihan Arachni sebagai alat dalam menganalisa keamanan aplikasi berbasis web karena Arachni adalah salah satu tools yang *powerfull*¸serta mudah penggunaan dan pemasangannya. Selain itu tampilan Arachni termasuk tampilan yang paling mudah dimengerti, temuan yang ditemukan juga memiliki bukti yang jelas dan dapat menjadi pertimbangan untuk melakukan tindakan perbaikan sistem.

2.2 Penelitian Terkait

Peneliti melakukan riset terhadap jurnal terdahulu untuk dijadikan topik dan acuan dalam penelitian ini, berikut tabel penelitian terkait.

Tabel 2.1 Penelitian Terkait

No	Nama dan Tahun	Judul	Topik	*Tools*	Hasil
1	Khaled Abdulghaffar, Nebrase Elmrabit, Mehdi Yousefi, 2023	*Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners*	*Penetration Testing*	Union, Intersection, Arachni OWASP Zed Attack Proxy	Perbandingan hasil dari WAVS
2	Muhammad Yaqi, 2023	*Vulnerability Assesment* Dan *Penetration Testing* (VAPT) Menggunakan Metode *Zero Eentry Hacking* (ZEH) Terhadap *Website*	*Penetration Testing*	OWASP Zed Attack, Uniscan, Nmap, Acunetix	Analisa kerentanan dari WAVS OWASP ZAP, Uniscan, dan Acunetix
3	Febri Al Fajar, 2020	Analisis Keamanan Aplikasi Web Prodi Teknik Informatika UIKA Menggunakan Acunetix Web *Vulnerability*	*Vulnerability Assesment*	Acunetix	Analisa kerentanan dari WAVS Acunetix

Pada penelitian pertama, Khaled Abdulghaffar, Nebrase Elmrabit, Mehdi Yousefi, meneliti berbagai macam WAVS yang mana *tools* yang akan di gunakan pada penelitian ini, peneliti mengetest beberapa WAVS diantaranya Union, Intersection, Arachni, dan OWASP ZAP untuk menghasilkan perbandingan *tools* yang paling efektif dan memiliki tingkat akurasi penemuan kerentanan yang tinggi, membuat penulis ingin

mengembangkan topik Arachni sebagai *tools* penelitian kerentanan aplikasi berbasis web.

Pada Penelitian ke dua Muhammad Yaqi mencoba untuk melakukan uji coba keamanan *website* menggunakan metode VAPT dengan teknik *Zero Entry Hacking*, Muhammad Yaqi mendapatkan banyak sekali kerentanan yang ditemukan menggunakan beberapa *tools* WAVS, penulis melihat betapa riskannya web dari instani terkait, membuat penulis menjadikan *website* dari Diskominfo DIY sebagai subjek penelitian ini, namun kerentanan itu belum tervalidasi ada atau tidaknya, namun penelitian ini meberikan metode bagaimana melakukan *assessment* dengan berkolaborasi dengan OWASP.

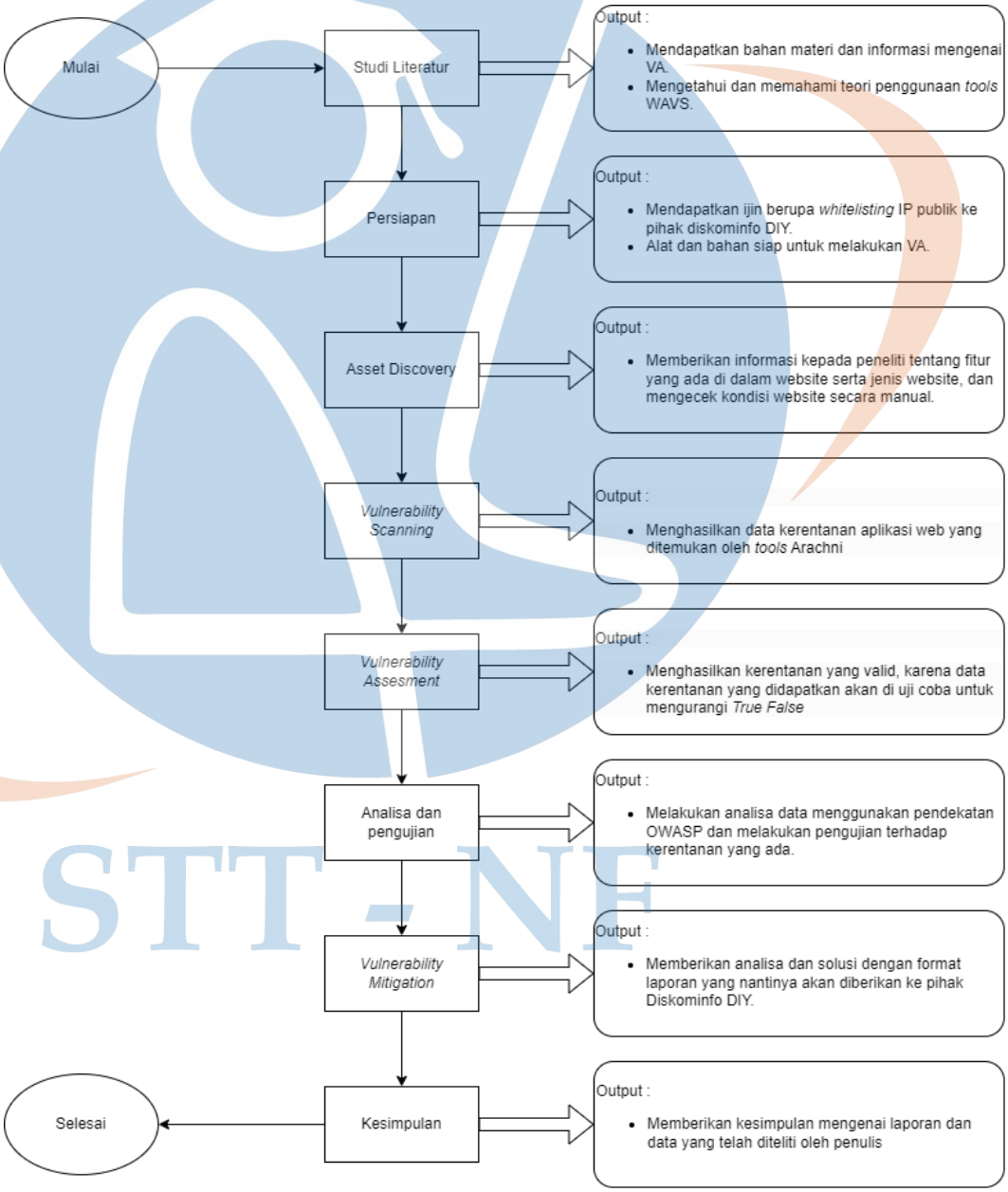
Pada Penelitian ke tiga Febri al Fajar mencoba uji kerentanan web instansi kampus dengan menggunakan Acunetix, Febri al Fajar mengatakan diperlukannya penyajian hasil audit keamanan ini harus menggunakan pendekatan yang memiliki standart keamanan, dari hal tersebut membuat penulis dapat menggembangkan Analisa pada penelitian ini menggunakan pendekatan OWASP.

Pada Penelitian ini penulis menggabungkan semua penelitian terdahulu mulai dari hasil perbandingan WAVS yang diuji, memvalidasi kerentanan, pengembangan pendekatan standart keamanan, serta pengembangan penemuan kerentanan menggunakan Arachni, dengan *tools* WAVS yang berbeda berupa Arahni dan metode yang digunakan adalah *Vulnerability Asssesment* maka ditariklah judul Analisa Kerentanan Aplikasi Berbasis Web Menggunakan *Tools* Arachni dengan Studi Kasus *Website* Dinas Komunikasi dan Informatika DIY.

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian yang dilakukan memiliki beberapa tahapan yang akan dijelaskan seperti pada gambar berikut.



Gambar 3.1 Tahapan Penelitian

1. Studi Literatur

Pada tahap ini penulis mencoba memahami bacaan yang bersumber dari teori dan jurnal untuk menambah informasi dan data yang berkaitan dengan penelitian ini, serta membaca modul pembelajaran *Cyber Security*.

2. Persiapan

Pada tahapan ini penulis memberikan alamat IP publik pada perangkat yang akan menjalankan Arachni kepada Dinas Komunikasi dan Informatika DIY, tahap ini wajib dilakukan pada penelitian ini untuk mengawasi penelitian dan memudahkan pencarian kerentanan pada aplikasi berbasis web, tahap ini bertujuan agar tidak terblokir oleh sistem pertahanan paling luar dari *website* seperti *Firewall* (WAF), IDS maupun IPS. Penulis juga mempersiapkan *device* yang akan digunakan sebagai media untuk melakukan VA dengan mulai menginstall *virtual machine Kali Linux* dan Arachni sebagai *tools* VA.

3. Asset Discovery

Tahap selanjutnya melihat tampilan dan fungsi *website* untuk melihat fitur apa saja yang dapat diakses, ditampilkan dan digunakan scara umum. Serta mengecek apakah *website* yang akan menjadi subjek VA masih berjalan dan tidak ditutup atau tidak diaktifkan menggunakan *tools* WhatsWeb pada Kali Linux.

4. Vulnerability Scanning

Pada tahap ini penulis melakuakan *scan* dari *list website* yang disediakan oleh instansi dengan menggunakan Arachni dengan *profile* pengujian *automated* dan *full scan* yang akan menghasilkan hasil deteksi kerentanan dan bukti celah kerentanan yang ada pada website Diskominfo yang dijadikan target pengujian arachni.

5. *Vulnerability Assesment*

Pada tahap ini penulis mendapatkan data hasil kerentanan aplikasi web dengan bukti yang ditemukan oleh *tools* Arachni, akan tetapi penulis memastikan kerentanan yang ada, hal ini sangat penting dilakukan karena beberapa hasil kerentanan dari Arachni bersifat *False Positive* yang hanya dinyatakan kerentanan, namun tetap semua kerentanan akan dilaporkan terutama kerentanan yang terbukti bisa dieksploitasi dan bisa dilakukan.

6. Analisa dan Pengujian

Pada tahapan ini penulis melakukan pendataan dari keseluruhan kerentanan yang sudah berhasil dideteksi maka akan dilakukan analisa kerentanan yang paling banyak terjadi, analisa menggunakan pendekatan OWASP *top* 10 untuk melihat apakah *website* ini termasuk dalam kategori *website* yang rentan pada serangan yang sering terjadi di dunia menurut OWASP, selain itu pada tahap ini peneliti melakukan pengujian berdasarkan temuan Arachni.

7. Vulnerability Mitigation

Pada tahapan ini penulis membuat laporan yang berisi data dan solusi kerentanan yang ada untuk diserahkan kepada pihak Diskominfo DIY.

8. Kesimpulan,

Penarikan kesimpulan dilakukan dari hasil analisa dan data kerentanan yang ada pada *website*.

3.2 Rancangan Penelitian

Mengidentifikasi jenis penelitian yang akan dilakukan, metode analisis dan pengumpulan data yang akan digunakan serta penjelasan metode pengujian yang akan digunakan dan ruang lingkup penelitian.

3.2.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini adalah eksplanatif, yang mana penulis mendeskripsikan data hasil penelitian kepada pihak Diskominfo, dengan bertujuan memberikan data terkait profil keamanan dari

aplikasi berbasis web milik diskominfo, serta memberikan metode untuk menganalisa keamanan menggunakan WAVS.

3.2.2 Metode Analisis Data

Metode analisis menggunakan metode Kualitatif dengan menghasilkan data kerentanan yang ada dari hasil *vulnerability assesment*, setelah data berhasil diteliti maka dilakukan pendekatan OWASP untuk mengetahui mana saja kerentanan yang paling sering disalahgunakan di dunia, selain itu akan ada beberapa kerentanan yang ada di *website* untuk dijadikan kesimpulan pada profil keamanan *website*, selain itu penulis mendapatkan kerentanan apa saja yang ada pada *website*, dan validitas kerentanan yang ada menggunakan pengujian dengan membuka *evidence* atau temuan dari WAVS.

3.2.3 Metode Pengumpulan Data

Dalam penelitian ini, pengumpulan data dilakukan oleh penulis menggunakan metode sebagai berikut.

1. Studi Literasi

Penulis mendapatkan data dari mempelajari banyak bacaan - bacaan yang berkaitan dengan penelitian yang bersumber dari buku, jurnal, modul, dan *website* yang terpercaya, pada metode ini penulis banyak mendapatkan informasi dan data tentang *Vulnerability Assesment*, WAVS, Arachni, Diskominfo DIY, dan OWASP.

2. Eksperimen

Penulis mendapatkan data dari mempelajari hasil eksperimen atau uji coba dari *tools* pencarian kerentanan aplikasi berbasis web atau WAVS, hasil kerentatanan yang ada berupa kerentanan apa saja yang dideteksi oleh WAVS pada *website* yang dilakukan *scan*, lalu dari hasil *scan*, data kerentanan yang ada akan dianalisa menggunakan pendekatan OWASP dan divalidasi kerentanannya untuk mendapatkan hasil profil keamanan.

3.2.4 Metode Pengujian

Metode yang dilakukan adalah *Black Box Testing* karena peneliti melakukan pengecekan pada sistem aplikasi berbasis web saja, penelitian ini menggunakan tenknik fungsionalitas, yang mana memastikan semua fitur dan fungsi aplikasi berbasis web sesuai dengan kebutuhan dan tidak memiliki kerentanan yang fatal, instrumen yang digunakan pada pengujian ini adalah melakukan tes atau eksperimen dengan tahapan *:*

1. *Asset Discovery*

   Melakukan indentifikasi dan ujicoba fitur yang ada pada website seperti login, kolom pencarian dan kolom pesan.

2. *Vulnerability Scanning*

   Melakukan kegiatan *scaning* secara otomatis menggunakan *tools* Arachni yang hasilnya akan diolah menjadi *assessment* atau evaluasi.

3. *Vulnerability Assesment*

   Melakukan analisa terhadap kerentanan yang ada, menguji coba kerentanan yang suda ditemukan, serta mengevaluasi kerananan berdasarkan dampak dan tingkat ancaman bagi sistem berdasarkan OWASP *top* 10.

4. *Vulnerability Mitigation*

   Melakukan simpulan, saran perbaikan dan laporan kepada pihak Diskominfo agar dapat melakukan tindakan lebih lanjut untuk menutup celah kerentanan pada sistem.

3.2.5 Lingkup Pengembangan

1. Lokasi

   Penelitian ini dilakukan scara koaboratif dengan Dinas Komunikasi dan Informatika DIY yang beralamat Jl. Brigjen Katamso, Keparakan, Kec. Mergangsan, Kota Yogyakarta, Daerah Istimewa Yogyakarta. Penelitian ini dilakukan secara online.

2. Objek Penelitian

Penelitian ini menjadikan aplikasi berbasis *website* milik Diskominfo DIY sebagai objek penelitian, informasi dan data yang sudah di teliti akan diberikan ke narasumber yang berasal dari pihak Diskominfo DIY.

3. Alat dan Bahan

Alat (*hardware*) yang digunakan dalam penelitian ini diantaranya.

- Laptop *Asus ROG Zephyrus* GA503QR.
- Prosesor AMD *Ryzen* 9 5900HS.
- Kapasitas RAM 24 GB.
- Sistem Operasi Windows 10 64bit dan *Virtual Machine* Kali Linux.
- Koneksi Internet.

Adapun bahan (*software*) dalam penelitian ini diantaranya.

- *Document Reader* (*Microsoft Office*).
- *Code Reader* (*Visual Studio Code*).
- *Web browser* (*Google Chrome*).
- VMWare *Workstation Pro*.
- *Terminal* dan Arachni *Web Application Vulnerability Scanner*.

BAB IV
IMPLEMENTASI DAN ANALISA

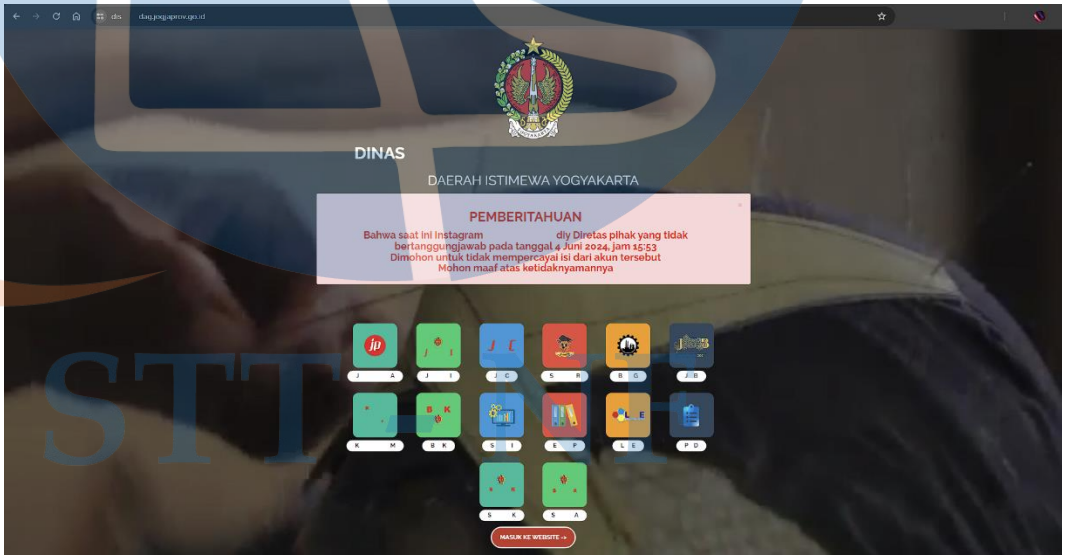
4.1 Analisa dan Perancangan

Sebelum melakukan penelitian analisa sistem diperlukan untuk mengetahui apa saja yang ada pada sistem mulai dari tampilan, fitur, sampai infrastruktur. Analisa diperlukan untuk mengukur apa saja kebutuhan yang harus tercukupi untuk melakukan analisa terhadap kerentanan yang ada.

4.1.1 Analisa Sistem

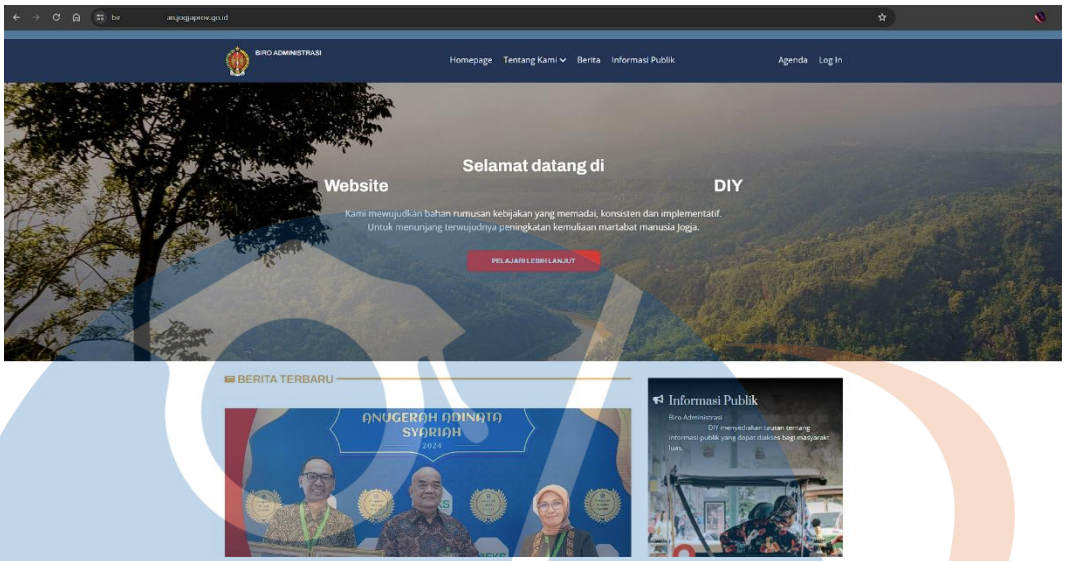
Sistem yang ada pada diskominfo sangat banyak jumlahnya, dari banyaknya jumlah situs, diskominfo mengidentifikasi setiap *website*nya menjadi sistem informasi dan profil persuahaan atau *company profile*, pada penelitian ini yang akan dijadikan objek analisa kerentanan berjumlah 4 situs, diantaranya 1 situs sistem informasi dan 3 situs *company profile* yaitu.

4.1.1.1 https://dis***dag.jogjaprov.go.id (*company profile*)



Gambar 4.1 *Website* dis***dag.jogjaprov.go.id

4.1.1.2 https://bir***an.jogjaprov.go.id (*company profile*)

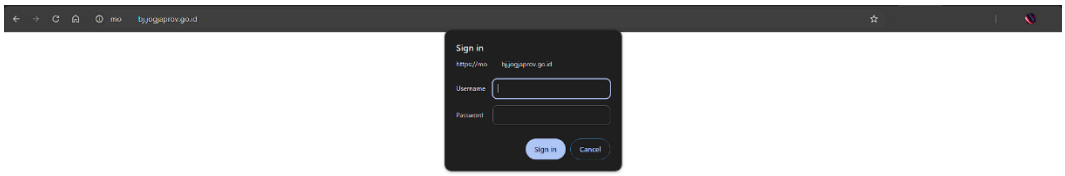


Gambar 4.2 *Website* bir***an.jogjaprov.go.id

4.1.1.3 https://rs***ra.jogjaprov.go.id (*company profile*)

Gambar 4.3 *Website* rs***ra.jogjaprov.go.id

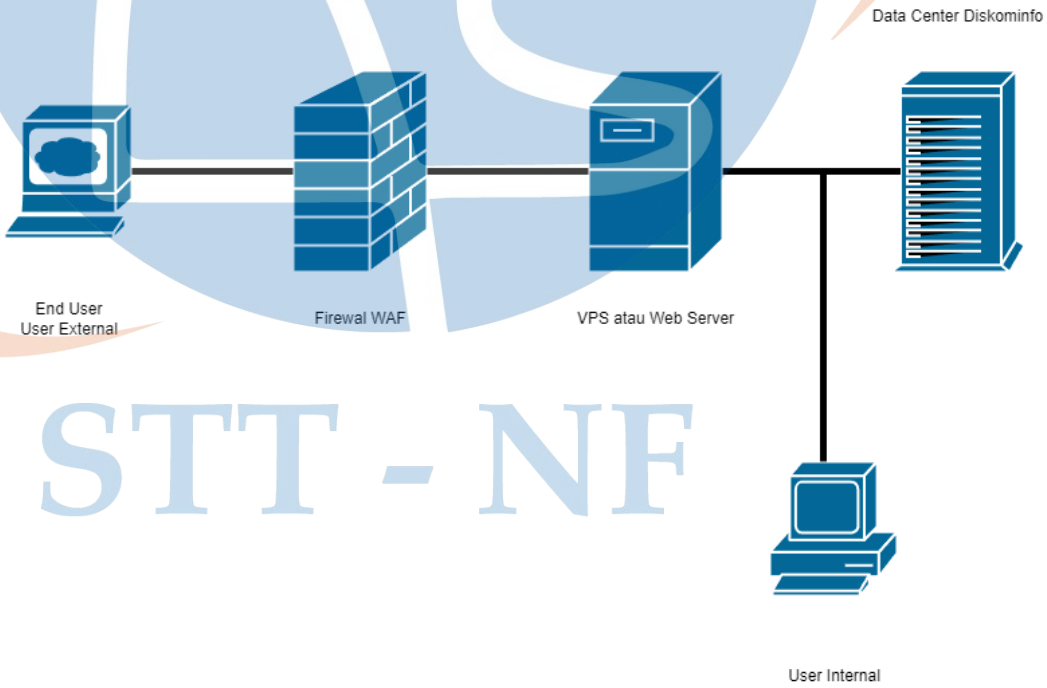
4.1.1.4 https://mo***bj.jogjaprov.go.id (sistem informasi)



Gambar 4.4 *Website* mo***bj.jogjaprov.go.id

4.1.1.5 Infrastruktur Sistem

Sistem yang akan dilakukan analisa adalah sistem yang berbentuk aplikasi berbasis web, seperti rata rata aplikasi berbasis web memiliki komponen seperti web *server*, dan *database server*. Pada situs situs yang dikelola oleh diskominfo terdapat WAF/*Firewall*, sebagai penghalang anomali dari luar agar tidak masuk ke dalam sistem utama. Berikut gambaran strukturnya.



Gambar 4.5 Infrastruktur Aplikasi berbasis Web Diskominfo DIY

Pada struktur diatas dapat dilihat *user external* harus melewati firewall terlebih dahulu agar bisa mengakses web *server* yang berisi komponen utama

atau code yang menjalankan aplikasi berbasis web milik diskominfo, ini merupakan salah satu langkah *preventif* dalam pencegahan anomali dan kerentanan yang akan terjadi pada web *server* dan *database server*. Namun langkah preventif ini tergolong *standart* dan masih kurang untuk memaksimalkan keamanan pada infrastruktur yang ada, maka dari itu dilakukannya *Vulnerability Assesment* yang dapat meminimalisir kerentanan yang ada.

Pada masing masing situs memiliki fitur yang hampir mirip, kemiripan fitur dapat dilihat karena situs situs yang akan dilakukan analisa keamanan memiliki fitur *login*, fitur *upload* dan *download file* yang dapat melakukan interaksi antara pengguna situs dan pengelola situs, dan fitur pencarian yang menggunakan data yang sudah di *upload* atau di tambahkan oleh pengelola maupun pengguna pada situs yang disimpan ke database, maka dari itu ada beberapa situs yang memiliki data penting yang berharga, seperti akun yang ada pada *website*, biodata diri, data dan surat perusahaan, sampai ke *file* code dari situs itu sendiri.

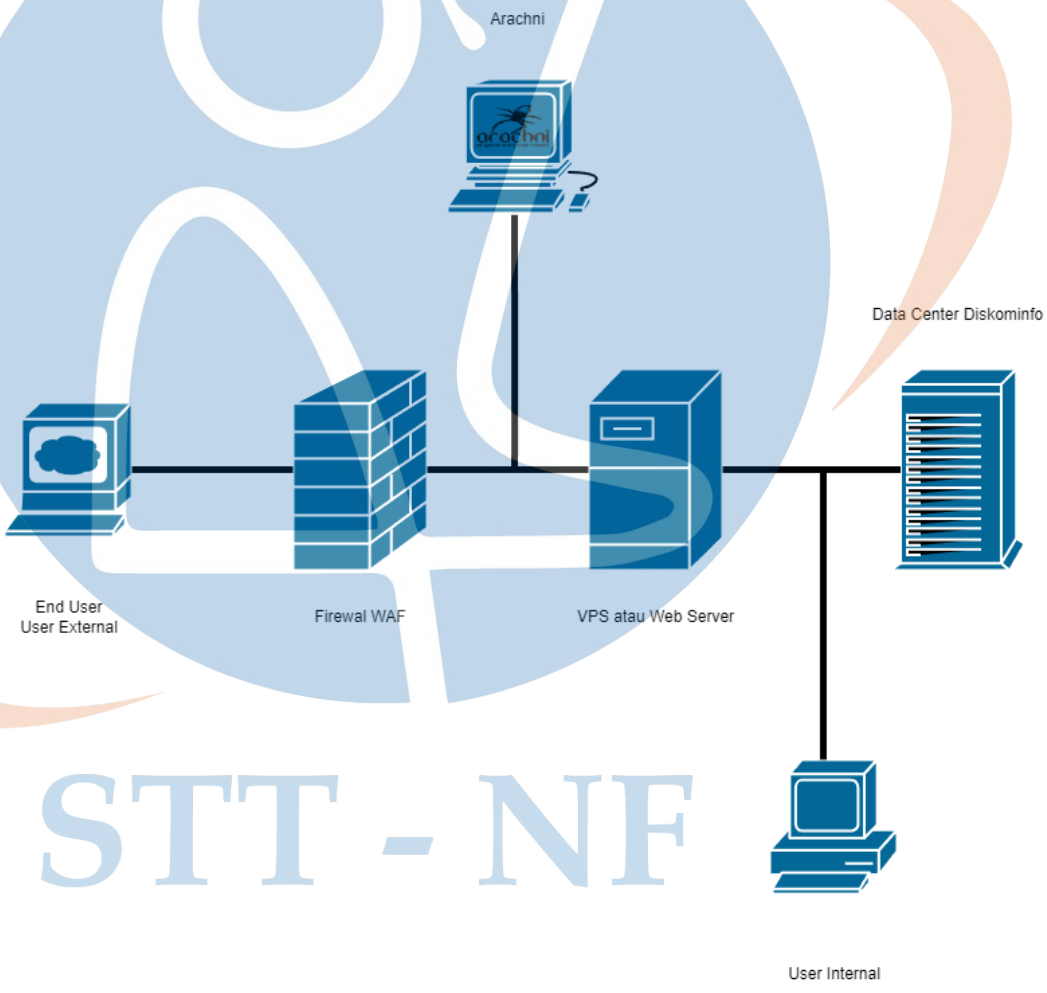
Untuk meminimalisir kerentanan atau kebocoran data, diperlukannya analisa dan pengendalian secara berkala khususnya pada keamanan *website*, salah satu pengendalian keamanan menggunakan metode penilaian kerentanan atau *Vulnerability Assesment* (VA), karena kebanyakan *tools* VA menguji coba semua fitur yang ada pada *website*.

4.1.2 Analisa *Tools*

Arachni adalah salah satu *tools* yang dapat melakukan *assessment* atau evaluasi keamanan dan kerentanan yang ada pada *website* khususnya aplikasi berbasis *website*, Arachni merupakan *tools* VA yang termasuk dalam kategori WAVS (*Web Application Vulnerability Scan*). Arachni dirasa cocok digunakan untuk melakukan analisa kemanan dan kerentanan yang ada pada *website* diskominfo, efektifitas dari VA untuk meminimalisir kerentanan dengan sering untuk melakukan VA, semakin sering melakukan VA

meningkatkan kewaspadaan terhadap kerentanan dan keamanan pada *website*.

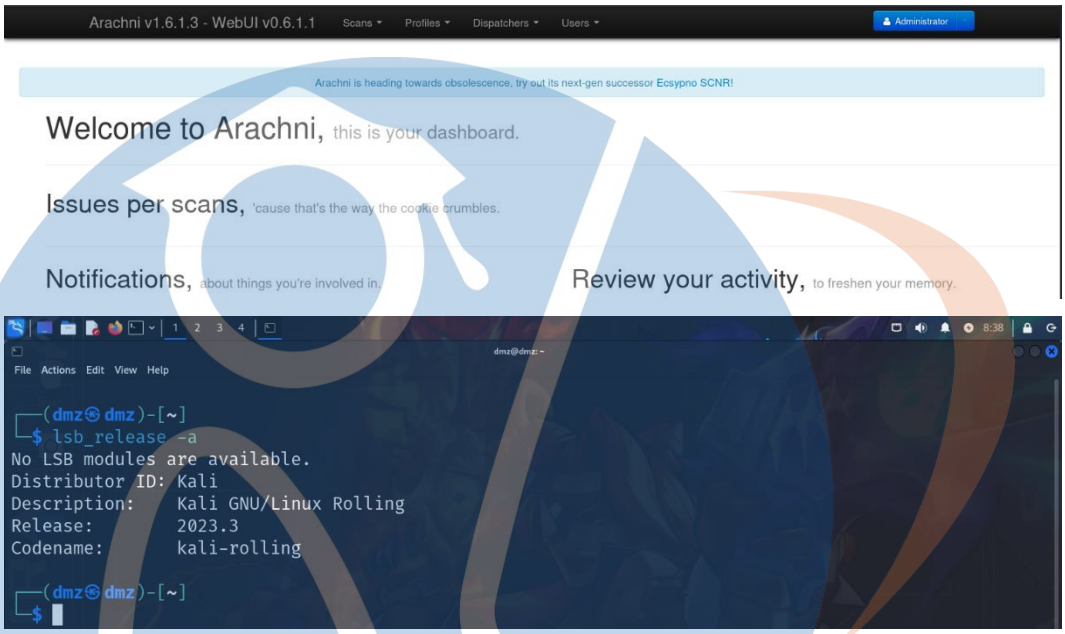
Vulnerability Assesment yang akan dilakukan lebih efektif apabila sudah melewati WAF, ada beberapa *action* yang dijalankan oleh *tools* akan diblokir/di*drop*, apabila belum bisa melewati WAF hasil yang ada akan lebih sedikit dari yang tidak menggunakan WAF, namun semua itu tergantung dari keamanan *website* itu sendiri. Struktur logik dari kegiatan VA ini adalah sebagai berikut.



Gambar 4.6 Gambaran Perencanaan VA

4.1.3 Analisa Kebutuhan Sistem

WAVS atau *Web Application Vulnerability Scaner* pada penelitian ini menggunakan Arachni, yang dijalankan menggunakan sistem operasi Kali Linux.



Gambar 4.7 Arachni dan Kali Linux

Untuk dapat menjalankan Arachni dan Kali Linux memilki spesifikasi dan kebutuhan minimum seperti pada tabel berikut[24].

Tabel 4.1 Spesifikasi yang dibutuhkan Kali Linux dan Arachni[24]

Komponen	**Minimum Spesifikasi**
Processor	Intel Core i3 atau AMD E1
Memory	2 GB of RAM
Storage	30 GB
Network	WLAN Atau LAN
Operating System	Kali Linux 2023.2
Software	Arachni v1.6.1.3

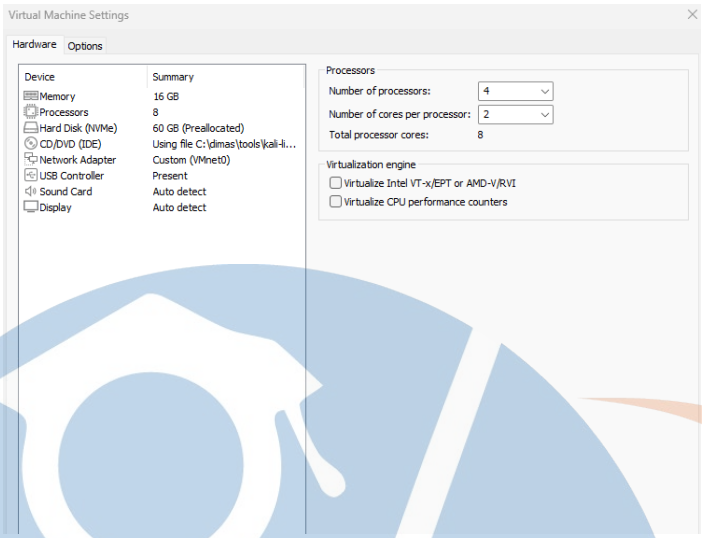
Pada penelitian ini menggunakan sistem operasi *Windows* sebagai sistem utama dibantu dengan aplikasi VMWare Workstation Pro (aplikasi menjalankan sistem *virtual*) untuk menjalankan Kali Linux secara *virtual* dengan spesifikasi:

Gambar 4.8 Spesifikasi Windows dan VMWare

Device yang digunakan pada penelitian ini spesifikasinya akan dipecah karena menggunakan sistem virtual, dengan spesifikasi seperti pada tabel dan gambar berikut.

Tabel 4.2 Spesifikasi Device yang digunakan

Nama Perangkat	Sistem Operasi	Prosesor	RAM	Penyimpanan	Konektifitas
Asus ROG Zephyrus GA503QR	Windows 11	16 CPU	24 GB	1000 GB	Wifi Card Intel AX210
	Kali Linux 2023.3	8 CPU	16 GB	60 GB	



Gambar 4.9 Spesifikasi Virtual Kali Linux

Sistem yang sudah berjalan membutuhkan *software* - *software* pendukung untuk melakukan VA seperti pada tabel berikut.

Tabel 4.3 *Software* yang Digunakan

Nama	Versi	Fungsi
VMWare Workstation Pro	17	Menjalankan Kali Linux secara *Virtual*
Terminal	Kali Linux 2023.3	Eksekusi perintah menjalankan Arachni
Arachni	1.6.1.3	Melakukan *Vulnerability Assesment*
Firefox *Browser*	115	Membuka laporan Arachni, dan membuka *evidence* atau bukti yang berupa *link* dari Arachni
VSCode	1.85.1	Membuka *file* yang memiliki *directory*
MS *Office*	2021	Membuka *document* yang berisi data yang ditemukan Arachni dari *website*

4.1.4 Rancangan Pengujian

Peneliti merancang bagaimana menguji data yang dihasilkan dari Arachni agar dapat diolah menjadi data kerentanan yang relevan, dan menjadi profil keamanan yang harus diperhatikan.

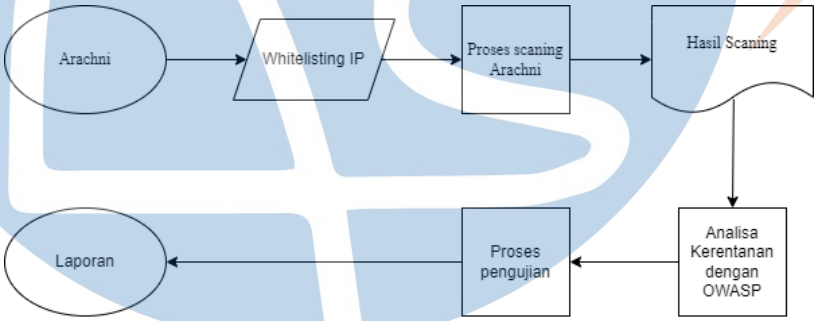
4.1.4.1 Dasar Pengujian

Pengujian pada penelitian ini menggunakan metode *Vulnerability Assesment* untuk menguji setiap fitur pada *website*nya yang dibantu

menggunakan *tools* Arachni, fitur yang diuji diantaranya *Path* atau direktori *website*, fitur *upload* dan *download file*. Penelitian ini menggunakan pendekatan OWASP *top* 10 untuk mengidentifikasi kerentanan yang paling sering disalahgunakan, data yang dihasilkan dari *tools* VA masih belum diidentifikasi dengan OWASP *top* 10 dan belum terbukti kerentananya, pengujian ini bertujuan untuk mendapatkan bukti hasil kerentanan dari Arachni agar memaksimalkan evaluasi untuk meningkatkan resistensi dari kerentanan yang paling sering terjadi.

4.1.4.2 Prosedur Pengujian

Pengujian kerentanan yang ditemukan akan di uji secara manual, dan juga menggunakan *link* temuan dari Arachni, dengan catatan pengujian harus menemukan Proof of Content, apabila pengujian berhasil menemukan PoC maka kerentanan akan terbukti dan dianggap valid (tidak *false positive*). Prosedur pengujian digambarkan seperti berikut.



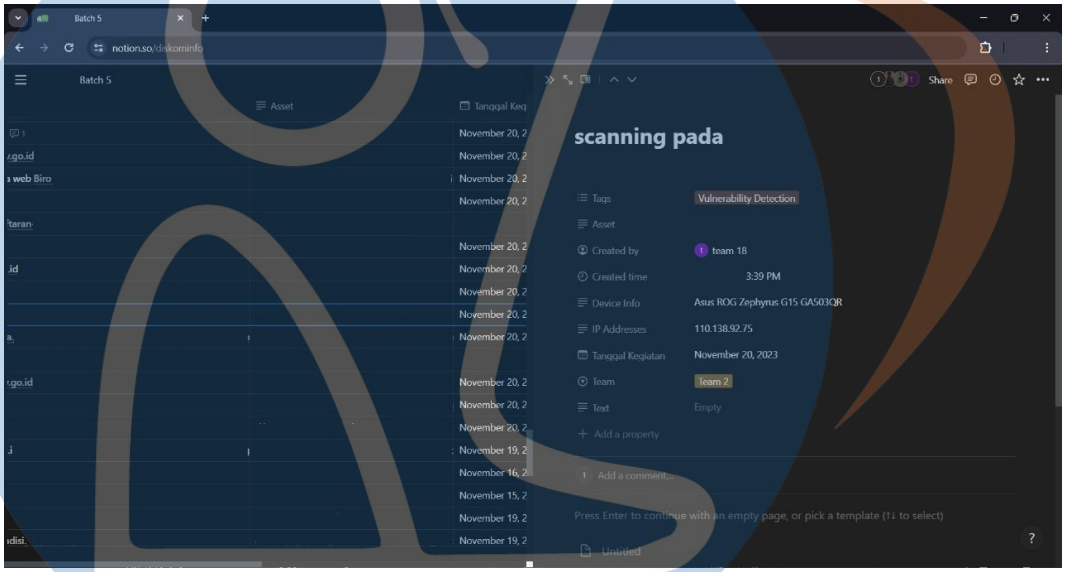
Gambar 4.10 Prosedur penelitian

4.2 Implementasi

Peneliti melakukan beberapa prosedur yang telah disetujui bersama oleh pihak diskominfo dalam melakukan penelitian ini, prosedur yang sudah disetujui ini menjadi tahapan wajib untuk dilakukan agar pihak diskominfo DIY mendapatkan laporan aktivitas dan monitoring *website* yang sedang dilakukan *Vulnerability Assesment*, serta merencanakan tahapan pengujian kerentanan.

4.2.1 *Whitelisting* IP

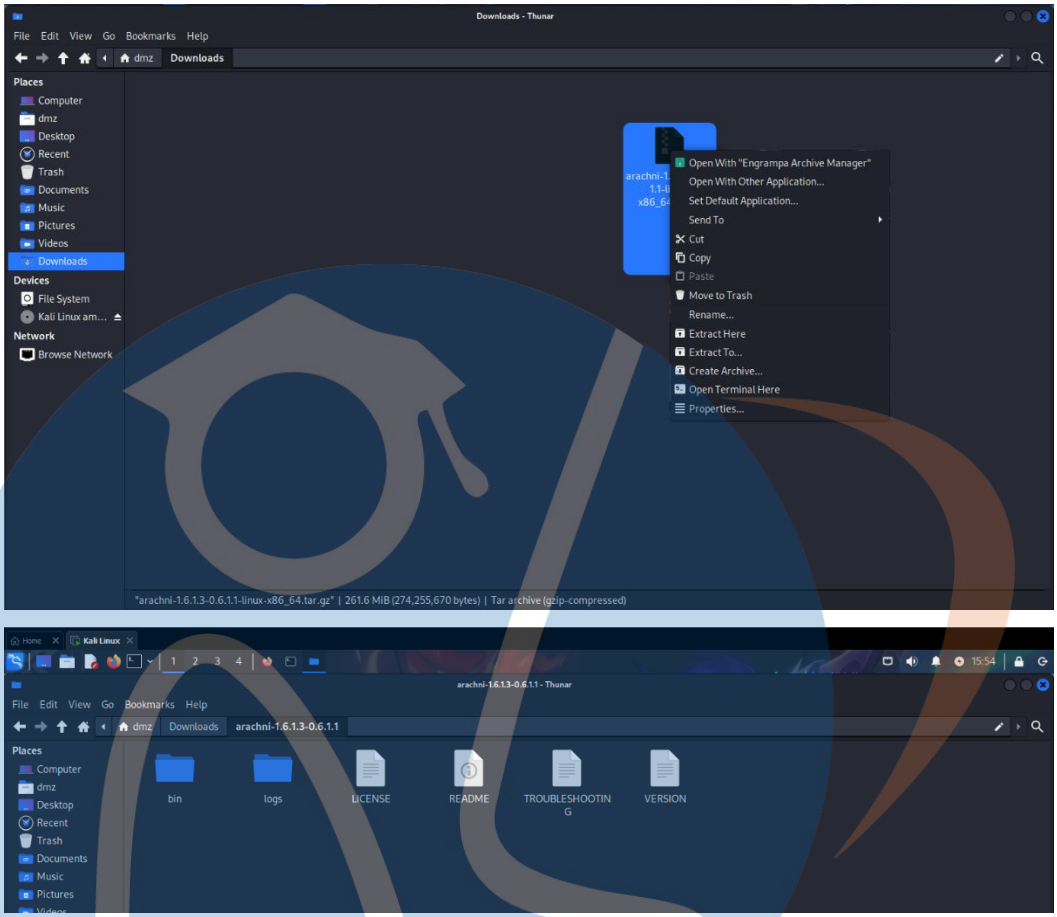
Whitelisting IP sangat penting karena untuk melewati WAF, agar pencarian kerentanan dapat berjalan secara maksimal, data seperti IP publik dan informasi *device* juga harus diberikan kepada pihak diskominfo DIY, hal ini penting untuk pihak diskominfo mengetahui mana IP dan *device* yang memang sedang melakukan pencarian kerentanan dan yang bukan, *Whitelisting* IP ini dilakukan dalam bentuk pengisian *log book* melalui Notion yang dikelola oleh diskominfo DIY seperti gambar berikut.



Gambar 4.11 *Whitelisting* IP dan *Device*

4.2.2 Instalasi dan penggunaan Arachni

Arachni adalah salah satu *tools* WAVS (*Web Application Vulnerability Scan*) yang sangat mudah digunakan dan diaplikasikan pada perangkat laptop ataupun komputer, Arachni juga salah satu *tools* yang bersifat *open source* sebelum berganti menjadi *enterprise* dengan nama SCNR yang di kembangkan oleh Ecsypno, untuk mengunduhnya dapat melalui *link* Git Hub *https://github.com/Arachni/Arachni*, dan dapat dijalankan melalui berbagai *platform* seperti Windows, Linux, dan Darwin. Pada penelitian kali ini menggunakan Linux dengan distro Kali, instalasi Arachni hanya perlu mengekstrak *file* tar.gz dari *file* unduhan github, seperti gambar berikut.

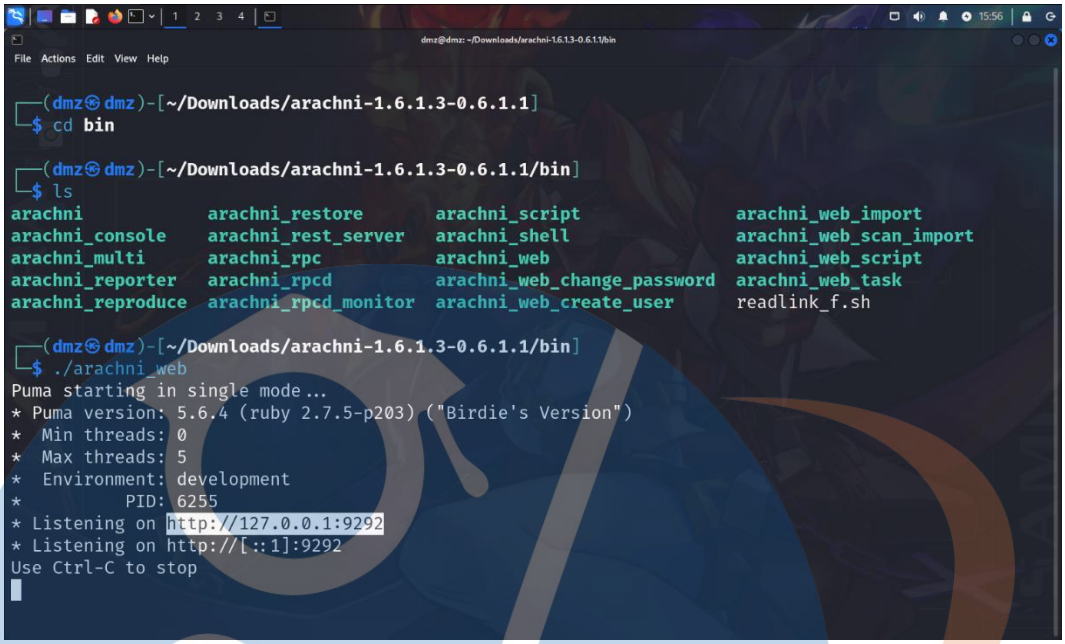


Gambar 4.12 Instalasi Arachni

Untuk menjalankan Arachni tergolong mudah cukup menggunakan terminal dan membuka direktori hasil ekstrak tar.gz yang tadi sudah di unduh seperti gambar 4.12, lalu ke direktori bin dan menjalankan Arachni dengan perintah.

```
./Arachni_web
```

Perintah tersebut digunakan untuk menjalankan Arachni Web UI apabila berhasil untuk menjalankan Arachni berbasis Web UI ini maka Arachni secara otomatis akan memeberikan IP *Access* berupa IP *local* 127.0.0.1 dengan *port* 9292 seperti pada gamabar berikut ini.

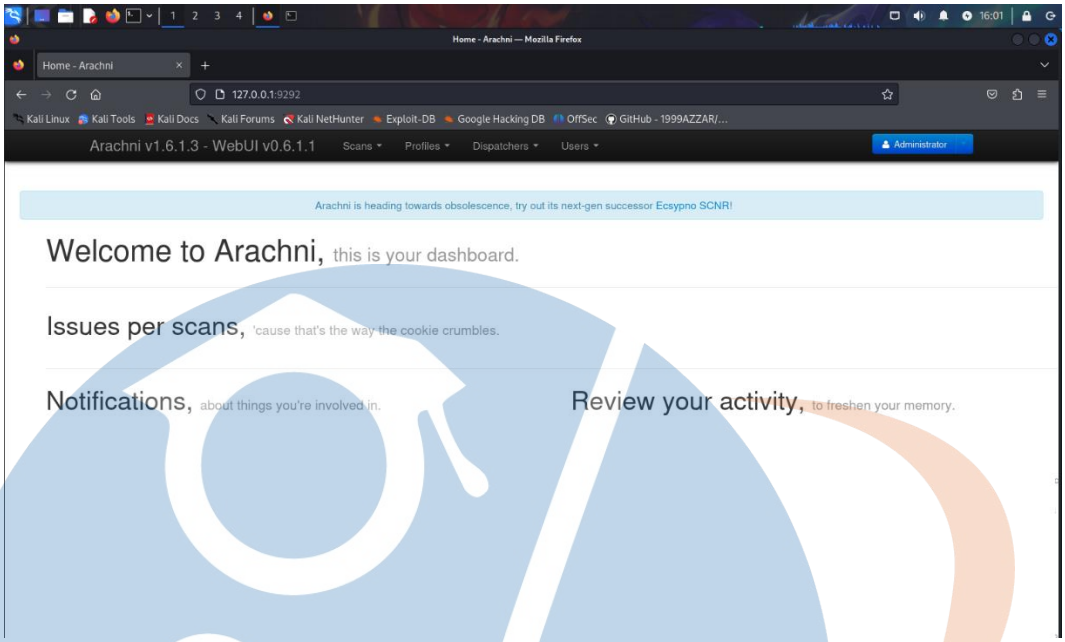


Gambar 4.13 Command menjalankan Arachni

Web UI Arachni sudah bisa di akses dan digunakan, menggunakan *login default* berupa email dan password, yang nantinya bisa di ubah.

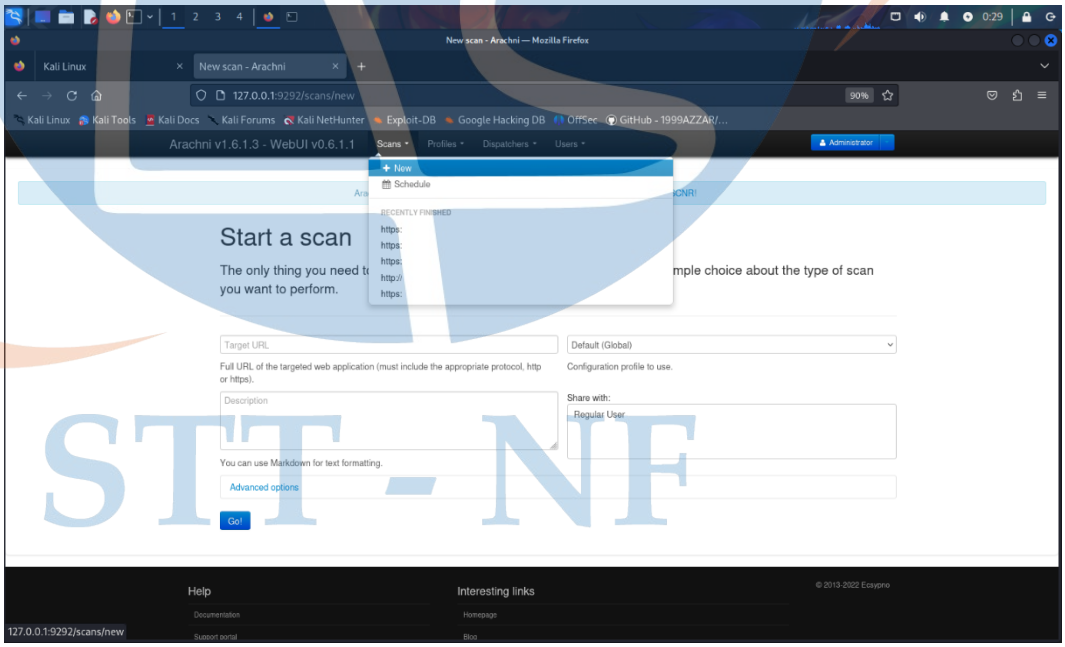
```
Email    = admin@admin.admin
Password = administrator
```

Apabila *login* telah berhasil akan dialihkan ke tampilan *dashboard* pada gambar berikut ini.



Gambar 4.14 Tampilan *login* dan *dashboard* Arachni

Penggunaan Arachni cukup mudah, hanya memilih menu *scans* dan memilih *new* seperti gambar di bawah ini

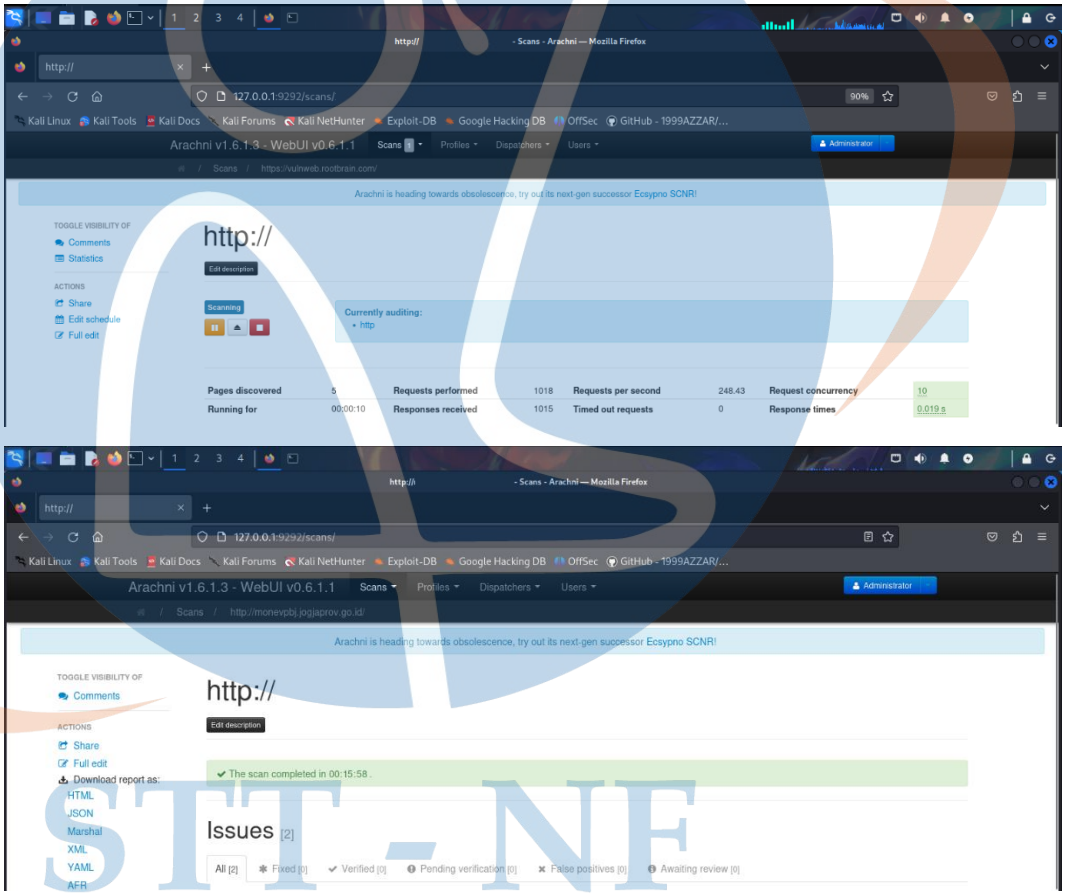


Gambar 4.15 Memulai pencarian kerentanan Arachni

Dan akan dialihkan ke menu *start a scan* dan memasukan *domain* dari *website* ke kolom *form target url*, *configuration profile* dibutuhkan apabila

terdapat case tertentu namun disini peneliti hanya menggunakan *global profile*, lalu tekan Go! Untuk memulai proses VA.

Dalam proses VA rentan waktu *tools* menyelesaikan VA minimal 10 menit dan maksimal 5 jam proses *scan* pada *tools* Arachni, waktu yang dibutuhkan biasanya menyesuaikan web yang di *scan*, apabila aplikasi berbasis web memiliki fitur dan *subsite* yang banyak, maka akan semakin lama juga untuk mendapatkan hasil VA dari *website* tersebut, seperti gambar berikut ini.



Gambar 4.16 Tampilan memulai dan selelai *scan*

Report Arachni ini yang akan diuji secara manual berdasarkan *evidence*/temuan *tools* Arachni yang nantinya akan dilakukan analisa dengan pendekatan OWASP *top* 10 yang nantinya akan ditarik menjadi profil keamanan *website*.

4.3 Evaluasi Sistem dan Pengujian

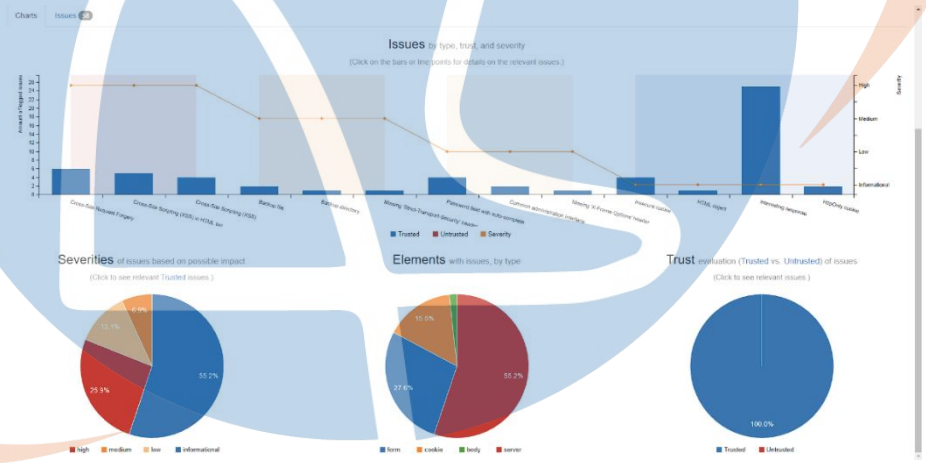
Peneliti mendapatkan evaluasi sistem berupa laporan VA, laporan VA berisi kerentanan dan *health map* dari setiap *website* yang di lakukan scan dari *tools* Arachni, dari data ini akan dilakukan pengujian dengan prosedur yang berkaitan dengan kerentanan yang ditemukan untuk meminimalisir *True False* dari *tools* Arachni.

4.3.1 Hasil Implementasi

Arachni menghasilkan laporan berbentuk diagram batang dan diagram *pie*, dengan persentasi kerentanan yang ada pada *website* yang di scan.

4.3.1.1 https://dis***dag.jogjaprov.go.id (*company profile*)

Didapati hasil *scan* dari *tools* arahni dalam bentuk diagram seperti gambar berikut ini.



Gambar 4.17 Hasil kerentanan dis***dag.jogjaprov.go.id

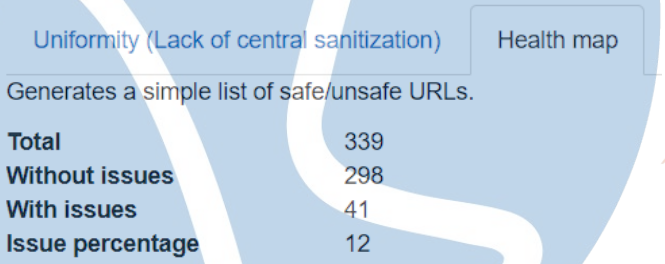
Arachni berhasil mendapat 58 *issue* atau bukti kerentanan, berikut rincian kerentananya dalam bentuk tabel.

Tabel 4.4 Kerentanan dis***dag.jogjaprov.go.id

Vulnerabilities	***Severities***	***Trust Level***	***Encounter***
Cross-Site Request Forgery	*High*	*Trusted*	6
Cross-Site Scripting (XSS) in HTML tag	*High*	*Trusted*	5
Cross-Site Scripting (XSS)	*High*	*Trusted*	4

Vulnerabilities	*Severities*	*Trust Level*	*Encounter*
Backup File	*Medium*	*Trusted*	2
Backup Directory	*Medium*	*Trusted*	1
Missing 'Strict-Transport-Security' header	*Medium*	*Trusted*	1
Password field with auto-complete	*Low*	*Trusted*	4
Common administration interface	*Low*	*Trusted*	2
Missing 'X-Frame-Options' header	*Low*	*Trusted*	1
Insecure Cookie	*Informational*	*Trusted*	4
HTML Object	*Informational*	*Trusted*	1
Interesting Response	*Informational*	*Trusted*	25
HttpOnly Cookie	*Informational*	*Trusted*	2
Total			58

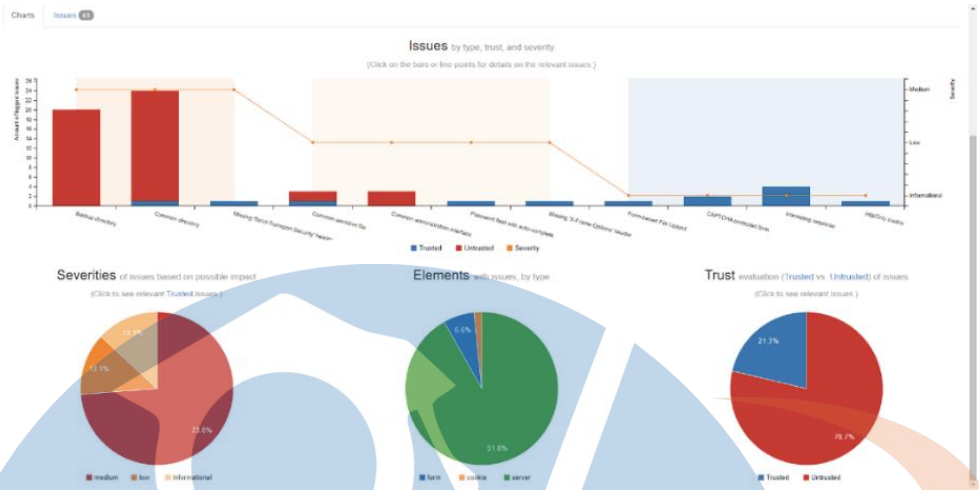
Arachni juga menghasilkan sampel *health map* seperti gambar berikut.



Gambar 4.18 *Health Map* dis***dag.jogjaprov.go.id

4.3.1.2 https://bir***an.jogjaprov.go.id (*company profile*)

Didapati hasil *scan* dari *tools* arahni dalam bentuk diagram seperti gambar berikut ini.



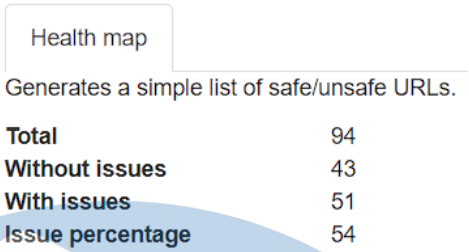
Gambar 4.19 Hasil kerentanan bir***an.jogjaprov.go.id

Arachni berhasil mendapat 61 *issue* atau bukti kerentanan, berikut rincian kerentananya dalam bentuk tabel.

Tabel 4.5 Kerentanan bir***an.jogjaprov.go.id

Vulnerabilities	***Severities***	***Trust Level***	***Encounter***
Common Directory	*Medium*	*Trusted*	1
Backup Directory	*Medium*	*Untrusted*	20
Common Directory	*Medium*	*Untrusted*	23
Missing 'Strict-Transport-Security' header	*Medium*	*Trusted*	1
Common Sensitive File	*Low*	*Trusted*	1
Common Sensitive File	*Low*	*Untrusted*	
Password Field with Auto-Complete	*Low*	*Trusted*	1
Common Administration Interface	*Low*	*Untrusted*	3
Missing 'X-Frame-Options' Header	*Low*	*Trusted*	1
Form-based File Upload	*Informational*	*Trusted*	1
CAPTCHA Protected form	*Informational*	*Trusted*	2
Interesting Response	*Informational*	*Trusted*	4
HttpOnly Cookie	*Informational*	*Trusted*	1
Total			61

Arachni juga menghasilkan sampel *health map* seperti gambar berikut.


Health map

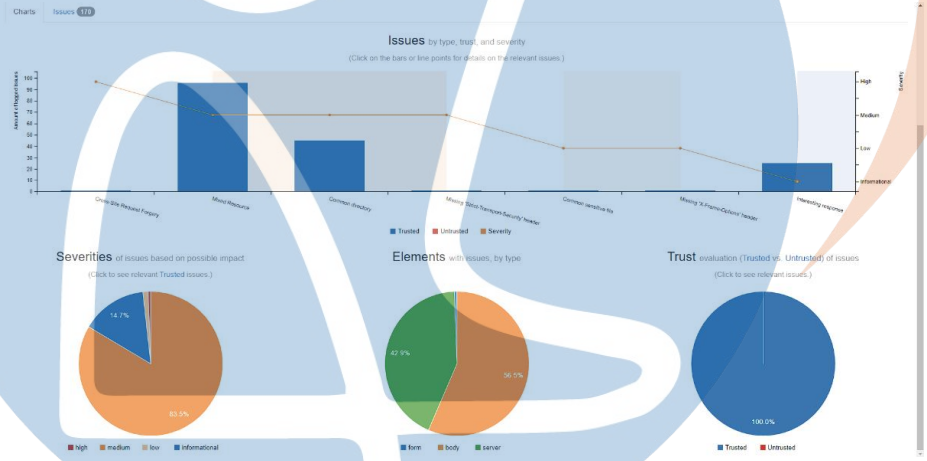
Generates a simple list of safe/unsafe URLs.

Total	94
Without issues	43
With issues	51
Issue percentage	54

Gambar 4.20 *Health Map* bir***an.jogjaprov.go.id

4.3.1.3 https://rs***ra.jogjaprov.go.id (*company profile*)

Didapati hasil *scan* dari *tools* arahni dalam bentuk diagram seperti gambar berikut ini.



Gambar 4.21 Hasil Kerentanan rs***ra.jogjaprov.go.id

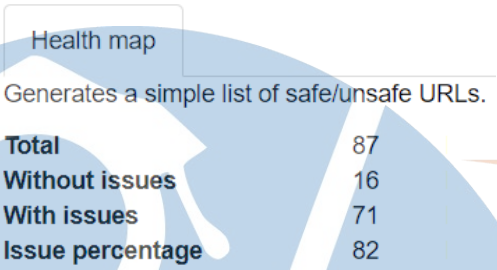
Arachni berhasil mendapat 170 *issue* atau bukti kerentanan, berikut rincian kerentananya dalam bentuk tabel.

Tabel 4.6 Kerentanan rs***ra.jogjaprov.go.id

Vulnerabilities	***Severities***	***Trust Level***	***Encounter***
Cross-Site Request Forgery	*High*	*Trusted*	1
Mixed Resource	*Medium*	*Trusted*	96
Common Directory	*Medium*	*Trusted*	45
Missing 'Strict-Transport-Security' header	*Medium*	*Trusted*	1
Common sensitive file	*Low*	*Trusted*	1
Missing 'X-Frame-Options' header	*Low*	*Trusted*	1

Vulnerabilities	*Severities*	*Trust Level*	*Encounter*
Interesting Response	*Informational*	*Trusted*	25
Total			170

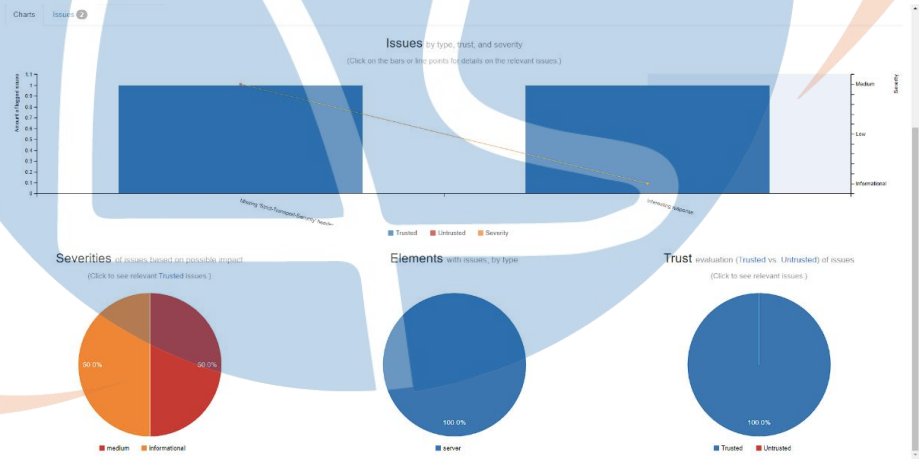
Arachni juga menghasilkan sampel *health map* seperti gambar berikut.



Gambar 4.22 *Health Map* rs***ra.jogjaprov.go.id

4.3.1.4 https://mo***bj.jogjaprov.go.id (sistem informasi)

Didapati hasil *scan* dari *tools* arahni dalam bentuk diagram seperti gambar berikut ini.



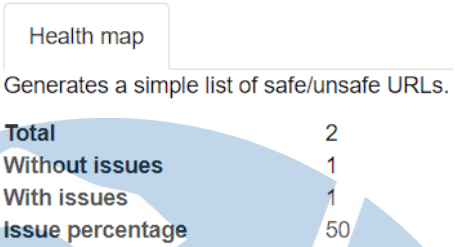
Gambar 4.23 Hasil Kerentanan mo***bj.jogjaprov.go.id

Arachni berhasil mendapat 2 *issue* atau bukti kerentanan, berikut rincian kerentananya dalam bentuk tabel.

Tabel 4.7 Kerentanan mo***bj.jogjaprov.go.id

Vulnerabilities	*Severities*	*Trust Level*	*Encounter*
Missing 'Strict-Transport-Security' header	*Medium*	*Trusted*	1
Interesting Response	*Informational*	*Trusted*	1
Total			2

Arachni juga menghasilkan sampel *health map* seperti gambar berikut.



Gambar 4.24 Health Map mo***bj.jogjaprov.go.i

4.3.2 Hasil Pengujian

Peneletian ini akan menguji kerentanan yang mana saja kerentanan yang ditemukan dan termasuk ke dalam kerentanan OWASP *top* 10 September 2021, serta peneliti mencoba validasi dengan mencari bukti kerentanan menggunakan hasil *assessment* dari Arachni (Automated) pengujian kerentanan secara manual dan *tools* yang berkaitan dengan kerentanan agar meminimalisir *true false* dari Arachni, apabila berhasil ditemukan maka terbukti kerentananya.

4.3.2.1 https://dis***dag.jogjaprov.go.id (*company profile*)

Hasil pengujian ditampilkan dalam bentuk tabel seperti berikut.

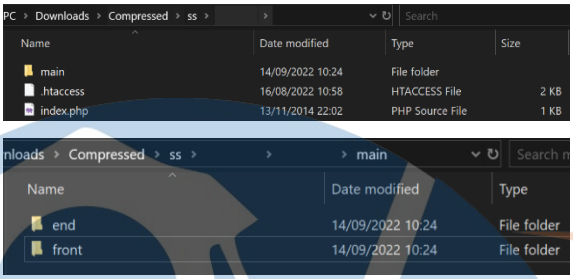
Tabel 4.8 Pengujian dis***dag.jogjaprov.go.id

Kerentanan	Jumlah	OWASP Top 10	Bukti
Cross-Site Scripting (XSS)	4	*A3-Injection*	Tidak terbukti
Cross-Site Scripting (XSS) in HTML tag	5		
Missing 'Strict-Transport-Security' header	1	*A5-Security Misconfiguration*	Tidak Terbukti
Backup Directory	1	*A1-Broken Access Control*	Terbukti
Backup file	2		Tidak Terbukti
Cross-Site Request Forgery	6		Terbukti

4.3.2.2 *Backup Directory*

Peneliti menemukan PoC (*Proof of Concept*) sebuah direktori *file* berupa .zip yang akan diunduh otomatis apabila menggunakan *link* tautan

dis***dag.jogjaprov.go.id/***.zip, *file* - *file* di dalam direktori ini terdapat *folder* front end dan backend dari salah satu aplikasi yang berjalan pada *website*, yang mana *file* dan *folder* ini berisi coding seperti gambar berikut.



Gambar 4.25 PoC dis***dag.jogjaprov.go.id

4.3.2.3 https://bir***an.jogjaprov.go.id (*company profile*)

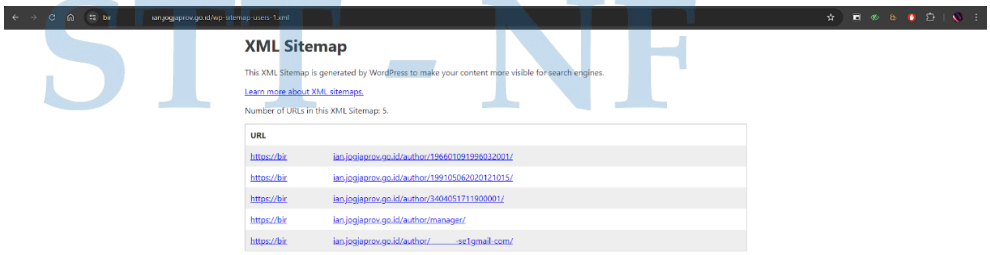
Hasil pengujian ditampilkan dalam bentuk tabel seperti berikut.

Tabel 4.9 Pengujian bir***an.jogjaprov.go.id

Kerentanan	Jumlah	OWASP Top 10	Bukti
Missing 'Strict-Transport-Security' header	1	*A5-Security Misconfiguration*	Tidak Terbukti
Backup Directory	1	*A1-Broken Access Control*	Tidak Terbukti
Common Sensitive Directory	1		Terbukti

4.3.2.4 *Common Sensitive File*

Peneliti menemukan PoC (*Proof of Concept*) menemukan sebuah direktori yang berisi tautan *link* XML, dan berisi nama user yang ada di dalam *website* itu yang takutnya banyak disalahgunakan dalam mengatas namakan user tersebut, dengan bukti seperti gambar berikut.



Gambar 4.26 PoC bir***an.jogjaprov.go.id

4.3.2.5 https://rs***ra.jogjaprov.go.id (*company profile*)

Hasil pengujian ditampilkan dalam bentuk tabel seperti berikut.

Tabel 4.10 Pengujian rs***ra.jogjaprov.go.id

Kerentanan	Jumlah	OWASP Top 10	Bukti
Missing 'Strict-Transport-Security' header	1	*A5-Security Misconfiguration*	Tidak Terbukti
Cross-Site Request Forgery	1	*A1-Broken Access Control*	Tidak Terbukti

Pada aplikasi *website* ini peneliti tidak menemukan kerentanan yang bisa menimbulkan resiko bahaya, *website* ini berisi tentang infomrasi tentang perusahaan yang bergerak dibidang kesehatan, tidak ada data sensitif dari dokter maupun pasien.

4.3.2.6 https://mo***bj.jogjaprov.go.id (sistem informasi)

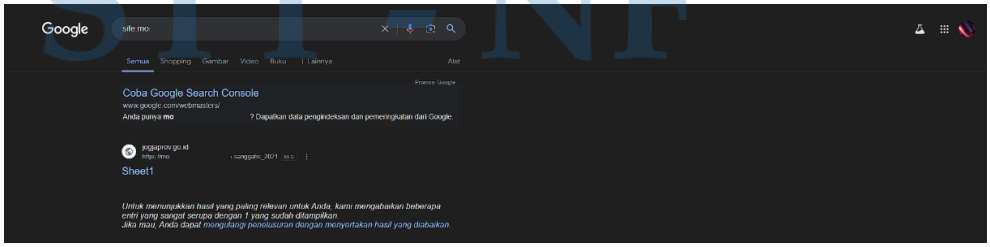
Hasil pengujian ditampilkan dalam bentuk tabel seperti berikut.

Tabel 4.11 Pengujian mo***bj.jogjaprov.go.id

Kerentanan	Jumlah	OWASP Top 10	Bukti
Missing 'Strict-Transport-Security' header	1	*A5-Security Misconfiguration*	Tidak Terbukti

4.3.2.7 *Interesting Response*

Pada *website* ini terbilang unik karena langsung diberikan tampilan *login*, peneliti sudah mencoba untuk *bruteforce* akun pada *website* ini namun tidak membuahkan hasil, namun ada *respond* yang unik apbila mencoba *dorking* menggunakan google *search*, terdapat *file* XLS excel yang berisi data – data sanggahan dan banding seperti gambar berikut.



Gambar 4.27 PoC mo***bj.jogjaprov.go.id

4.3.3 Solusi

Dari hasil pengujian peneliti dapat memberikan solusi kepada pihak pengelola agar dapat memperbaiki sistem keamanan pada *website* - *website* yang dikelola, karena banyak *file* - *file* pada *website* yang tidak seharusnya user melihat atau mengaksesnya, maka lebih baik dihapus atau disimpan pada *folder* yang memiliki hak akses yang lebih tinggi daripada *user* biasa, serta penutupan *path* - *path* yang masih *default*, agar semakin meminimalisir mudahnya mengetahui teknologi yang digunakan dalam *website*, seperti *file robots.txt*, dari path tersebut semua konten yang ada dalam *website* akan terbuka, mulai dari *user*, *comment*, sampai *content* dari *website*.

BAB V
KESIMPULAN DAN SARAN

1. Kesimpulan

1. *Vulnerability Assesment* (VA) merupakan salah satu metode untuk meningkatkan keamanan pada *website* dengan cara mencari kelemahan pada *website* yang ada, cara kerja VA dapat membantu untuk meningkatkan resistensi serangan dan ancaman pada sistem berdasarkan kerentanan yang ditemukan, informasi terkait kerentanan yang ada dapat mempermudah kendali keamanan sistem dan pemeliharaan sistem, apabila dilakukan secara kontinu dapat meminimalisir resiko sistem terkena serangan dan ancaman dari luar. Arachni adalah *tools* VA yang cocok untuk melakukan pencarian kerentanan terutama pada aplikasi berbasis web, tidak perlu melakuakn instalasi yang rumit dan tampilannya yang mudah dipahami, menjadikan VA menggunakan *tools* Arachni adalah salah satu metode analisa keamanan aplikasi berbasis web yang cocok untuk menjadi pedoman keamanan bagi Diskominfo DIY.
2. Pada data yang ada dan yang telah diuji dari penelitian ini, terdapat kurang lebih 291 kerentanan yang ditemukan oleh arachni, dari banyaknya kerentanan yang ditemukan arachni ada beberapa kerentanan yang sudah menjadi *common vulnerability* dan tidak memiliki dampak serius pada sistem aplikasi berbasis web namun tetap membutuhkan langkah penutupan celah, dan ada beberapa kerentanan yang serius dan perlu ditindak lanjuti seperti pada pembahasan Hasil Pengujian 4.3.2, dan hasil pengujian dapat dijadikan profil keamanan dari *website* yang sudah di lakukan pengujian.

   1 dari 4 aplikasi berbasis web milik diskominfo memiliki kerentanan yang paling parah, dan 3 lainnya adalah kerentanan yang common, kerentanannya adalah direktori *website* dan hak akses *website* yang kurang terorganisir menjadikan profil keamanan yang harus

diperhatikan oleh diskominfo adalah pengelolaan direktori dan hak akses pada *website* yang ada, karena masih banyak file - file dan informasi yang dapat diakses oleh user biasa dan dengan mudah dapat diambil dan disebar luaskan, terutama full code yang ada di *website*. Apabila disalahgunakan pihak luar dapat mengakses database utama dari *website* dengan menghubungkan controller db, dan membuat *website* phising yang sama persis tampilannya untuk mengelabui pengguna aplikasi. Kerentanan sudah ditemukan Arachni yang tidak memiliki PoC bisa menjadi pertimbangan untuk penguatan sistem milik Diskominfo DIY, karena kerentanan yang ditemukan menggunakan arachni memiliki bukti berupa *link* dari pengujian *automated* yang dilakukan Arachni untuk menjadi parameter celah kerentanan pada sistem.

2. Saran

Saran untuk pengelola agar dapat mengubah konfigurasi user untuk menentukan user yang dapat akses penuh dan tidak, serta mengelola penyimanan data yang sensitif dan penting tidak pada direktori terbuka ataupun direktori *default* atau bawaan sistem, sebisanya menghindari penggunaan konfigurasi *default*, serta menggunakan autentikasi pada file dan direktori tertentu.

Saran untuk penelitian selanjutnya agar menggunakan *tools* VA yang lain seperti Acunetix, OWASP ZAP, dan Neesus untuk pembanding hasil kerentanan yang ada agar menenemukan kerentanan secara maksimal, dan juga melakukan pengujian selanjutnya dengan metode VAPT, yang mana menguji dan meneliti lebih dalam terkait kerentanan yang ada.

DAFTAR PUSTAKA

[1] A. Budiman, S. Ahdan, and M. Aziz, "Analisis Celah Keamanan Aplikasi Web E-Learning Universitas Abc Dengan *Vulnerability Assesment*," *J. Komputasi*, vol. 9, no. 2, pp. 1–10, 2021, [Online]. Available: https://jurnal.fmipa.unila.ac.id/komputasi/article/view/2800

[2] A. Asfiyani and F. Junaedi, "Model Manajemen Produksi Konten Digital tentang COVID-19 Oleh Dinas Kominfo Daerah Istimewa Yogyakarta di Masa Pandemi," *J. Interak. J. Ilmu Komun.*, vol. 6, no. 1, Jan. 2022, doi: 10.30596/interaksi.v6i1.7424.

[3] D. K. dan I. P. D. D. I. Yogyakarta, "Rencana Strategis Pemerintah Daerah Daerah Istimewa Yogyakarta 2023-2026." pp. 1–72, 2022. [Online]. Available: https://birotapem.jogjaprov.go.id/tambahan/116.pdf

[4] Id-SIRTII/CC, "Laporan Bulanan Publik Hasil Monitoring Keamanan Siber Maret 2024," vol. 3, 2024, [Online]. Available: https://www.idsirtii.or.id/halaman/tentang/laporan-hasil-monitoring.html

[5] A. C. Kusuma and A. D. Rahmani, "Analisis Yuridis Kebocoran Data Pada Sistem Perbankan Di Indonesia (Studi Kasus Kebocoran Data Pada Bank Indonesia) Aditama Candra Kusuma , Ayu Diah Rahmani Fakultas Hukum , Universitas Pembangunan Veteran Jakarta Kemajuan teknologi sangat membantu manu," *J. Huk.*, vol. 5, no. 01, pp. 46–63, 2022, [Online]. Available: www.bi.go.id.

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

[7] Elgamar, *Konsep Dasar Pemrograman Website Dengan PHP*. Ahlimedia Book, 2020.

[8] R. Novria, B. Kurniawan, and Suryanto, "Aplikasi Pemesanan Makanan DI bebek dan Ayam Takaeng menggunakan Php dan Mysql," *J. Inform. dan Komput. (JIK)*, vol. 13, no. 1, pp. 15–26, 2022.

[9] S. Hendraputra *et al.*, "Pengantar Teknologi dan Informasi," *Yaya*, 2021.

[10] D. Fata, "Evaluasi risiko celah keamanan menggunakan metodologi open web application security project (OWASP) pada aplikasi web sistem informasi akademik (SIAKAD) UIN Ar-Raniry," pp. 1–58, 2023, [Online]. Available: https://repository.ar-raniry.ac.id/id/eprint/31189/%0Ahttps://repository.ar-raniry.ac.id/id/eprint/31189/1/Tugas Akhir - Darul Fata %28180705016%29.pdf

[11] J. Simarmata *et al.*, *Sistem Keamanan Data*. KitaMenulis.id, 2022. [Online]. Available: https://kitamenulis.id/2022/10/30/sistem-keamanan-data/

[12] M. Yaqi, *Vulnerability Assessment dan Penetration Testing (Vapt) Menggunakan Metode Zero Entry Hacking (Zeh) Terhadap Website Studi Kasus: Dinas Penanaman Modal …*. 2023. [Online]. Available: https://repository.uinjkt.ac.id/dspace/handle/123456789/73422%0Ahttps://repository.uinjkt.ac.id/dspace/bitstream/123456789/73422/1/MUHAMMAD

YAQI-FST.pdf
[13] J. A. Kusumaningtyas, “Analisa Pengelompokan Cyber Crime PadaPenerapan Electronic Commerce,” *J. Prodi Tek. Inform. UNW “Multimatrix,”* vol. 11, no. 1, pp. 9–19, 2019.
[14] Paryati, “Keamanan Sistem Informasi,” 2008.
[15] S. Nurul, Shynta Anggrainy, and Siska Aprelyani, “Faktor-Faktor Yang Mempengaruhi Keamanan Sistem Informasi: Keamanan Informasi, Teknologi Informasi Dan Network (Literature Review Sim),” *J. Ekon. Manaj. Sist. Inf.*, vol. 3, no. 5, pp. 564–573, 2022, doi: 10.31933/jemsi.v3i5.992.
[16] R. R. Yusuf and T. N. Suharsono, “Pengujian Keamanan Dengan Metode Owasp Top 10 Pada *Website* Eform Helpdesk,” *Pros. Semin. Sos. Polit. Bisnis, Akunt. dan Tek.*, vol. 5, p. 402, 2023, doi: 10.32897/sobat.2023.5.0.3132.
[17] I. R. Dhaifullah, M. Muttanifudin H, A. Ananda Salsabila, and M. Ainul Yaqin, “Survei Teknik Pengujian Software,” *J. Autom. Comput. Inf. Syst.*, vol. 2, no. 1, pp. 31–38, 2022, doi: 10.47134/jacis.v2i1.42.
[18] F. Wibowo, H. Harjono, and A. P. Wicaksono, “Uji Vulnerability pada *Website* Jurnal Ilmiah Universitas Muhammadiyah Purwokerto Menggunakan OpenVAS dan Acunetix WVS,” *J. Inform.*, vol. 6, no. 2, pp. 212–217, 2019, doi: 10.31311/ji.v6i2.5925.
[19] Y. Mulyanto, E. Haryanti, and J. Jumirah, “Analisis Keamanan *Website* Sman 1 Sumbawa Menggunakan Metode Vulnerability Asesement,” *J. Inform. Teknol. dan Sains*, vol. 3, no. 3, pp. 394–400, 2021, doi: 10.51401/jinteks.v3i3.1260.
[20] R. B. Baharsah, A. B. Purba, J. Mulyana, and C. I. Grahana, “Jurnal Inovasi Pengembangan Aplikasi dan Keamanan Informasi Nusantara,” *J. Inov. Pengemb. Apl. dan Keamanan Inf. Nusant.*, vol. 1, no. 1, pp. 1–10, 2023.
[21] Suzanne Selhorn and Marcel Amirault, “Vulnerability Severity Levels,” Gitlab. [Online]. Available: https://docs.gitlab.com/ee/user/application_security/vulnerabilities/severities.html
[22] I. Putu Mas Yuda Pratama, G. Agus Supriatmaja, K. Mahendra, I. Made Edy Listartha, and G. Arna Jude Saskara, “Perbandingan Vulnerability Analysis Pada *Website* Menggunakan Tools Wapiti, Skipfish, Dan ArachniArachni,” *J. Teknol. Inf.*, vol. 6, no. 2, pp. 187–193, 2022, [Online]. Available: http://testphp.vulnweb.com/.
[23] K. Abdulghaffar, N. Elmrabit, and M. Yousefi, “Enhancing Web Application Security through Automated Penetration Testing with Multiple Vulnerability Scanners,” *Computers*, vol. 12, no. 11, pp. 1–17, 2023, doi: 10.3390/computers12110235.
[24] K. Liniux, “Installation” gitlab, p. 1, 2023. [Online]. Available: https://gitlab.com/kalilinux/documentation/kali-docs/-/tree/master/installation/hard-disk-install

LAMPIRAN

Lampiran 1 Perizinan penelitian dari kampus

SEKOLAH TINGGI TEKNOLOGI
TERPADU NURUL FIKRI
STT - NF

Nomor : 009../S.Peng/BAAK/PRODI_TI/..VII../..2024
Perihal : **Permohonan Riset Tugas Akhir**

Yth. Kepala Dinas Kominfo DIY c.q. Anik Budiati, S.Kom., M.Eng.
Dinas Komunikasi dan Informatika
Jl. Brigjen Katamso, Keparakan, Kec. Mergangsan, Kota Yogyakarta
Daerah Istimewa Yogyakarta, 55152

Yang bertanda tangan di bawah ini, Kepala Program Studi ***Teknik Informatika*** Sekolah Tinggi Terpadu Nurul Fikri (STT-NF), ingin memberitahukan bahwa mahasiswa kami :

No	*Nama Mahasiswa*	*NIM*	*Jurusan*
1	Dimas Ramadhani	0110220281	Teknik Informatika

Dalam rangka mengerjakan penelitian ***Tugas Akhir*** mahasiswa kami bermaksud meminta izin untuk melakukan penelitian pada perusahaan Bapak/Ibu. Data hasil penelitian diperlukan semata-mata untuk kepentingan akademik, tidak untuk kepentingan komersial dan politik. Besar harapan kami Bapak/Ibu bersedia memberikan izin sehingga Penelitian Mahasiswa tersebut berjalan dengan baik.

Atas perhatian dan kerjasamanya, kami mengucapkan terima kasih.

Depok, Jum'at, 26 Juli 2024
Kepala Program Studi Teknik Informatika
Sekolah Tinggi Teknologi Terpadu Nurul Fikri

Tifani Nabarian, S.Kom., M.T.I.
NIP: 2200890201

Sekolah Tinggi Teknologi Terpadu Nurul Fikri
www.nurulfikri.ac.id | info@nurulfikri.ac.id
Kampus A, Jl. Situ Indah No. 116 Depok 16451 021 - 29842347
Kampus B1 & B2, Jl. Lenteng Agung Raya No. 20 - 21 Jakarta Selatan 12640 021 - 7863191

Lampiran 2 Perizinan dari Diskominfo DIY

https://bit.ly/Setfi-perizinan-dimas

Lampiran 3 Sertifikat yang diberikan Diskominfo DIY

Dinas Komunikasi dan Informatika
Daerah Istimewa Yogyakarta

Sertifikat
Apresiasi

Diskominfo DIY menyampaikan apresiasi dan terima kasih kepada

DIMAS RAMADHANI

yang telah melaksanakan kegiatan
Vulnerability Assessment pada sistem elektronik Pemda DIY
melalui Program Magang dan Studi Independen Bersertifikat (MSIB)

Partisipasi Anda berkontribusi dalam upaya meningkatkan
keamanan siber yang lebih baik

Yogyakarta, 5 Januari 2024

Kepala
Hari Edi Tri Wahyu Nugroho SIP., M.Si.

Dokumen ini ditandatangani secara elektronik menggunakan sertifikat elektronik yang diterbitkan oleh Balai Sertifikasi Elektronik (BSrE) Badan Siber dan Sandi Negara