BAB 2
TINJAUAN PUSTAKA

2.1 Profil Instansi

Dinas Informasi dan Pengolahan Data Angkatan Darat (Disinfolahtad) merupakan Badan Pelaksana Pusat di tingkat MABESAD yang berkedudukan langsung dibawah pimpinan Kasad dalam menyelenggarakan fungsi Pembina sistem informasi dan pengolahan data secara elektronis di tingkat pusat meliputi pembangunan, pengembangan, pemeliharaan dan penyiapan sistem informasi TNI AD. Disinfolahtad terletak di Jalan Veteran Raya No. 5 Jakarta Pusat, DKI Jakarta. Disinfolahtad memiliki fungsi utama sebagai Pembinaan Fungsi (Binfung), Pembinaan Sistem Informasi Administrasi (Sisfomin), Pembinaan Sistem Informasi Kekuatan dan Lingkungan Operasi (Sisfokuat), Pembinaan Materiil Sistem Informasi (Binmatsisfo), dan Dukungan Pengolahan Data (Duklahta) [1].

2.2 Logo Instansi

Gambar 2.1. Logo Disinfolahtad

Adapun arti dan makna dari logo Disinfolahtad adalah, *Widyuta Awasara Anindyaguna***,** *Widyuta* memiliki arti kilat, petir, halilintar (bermakna cepat). *Awasara* memiliki arti saat yang tepat dan baik (bermakna tepat). *Anindyaguna* memiliki arti keunggulan yang sempurna (bermakna akurat). Dari keseluruhan makna tersebut, maka *Widyuta Awasara Anindya Guna*

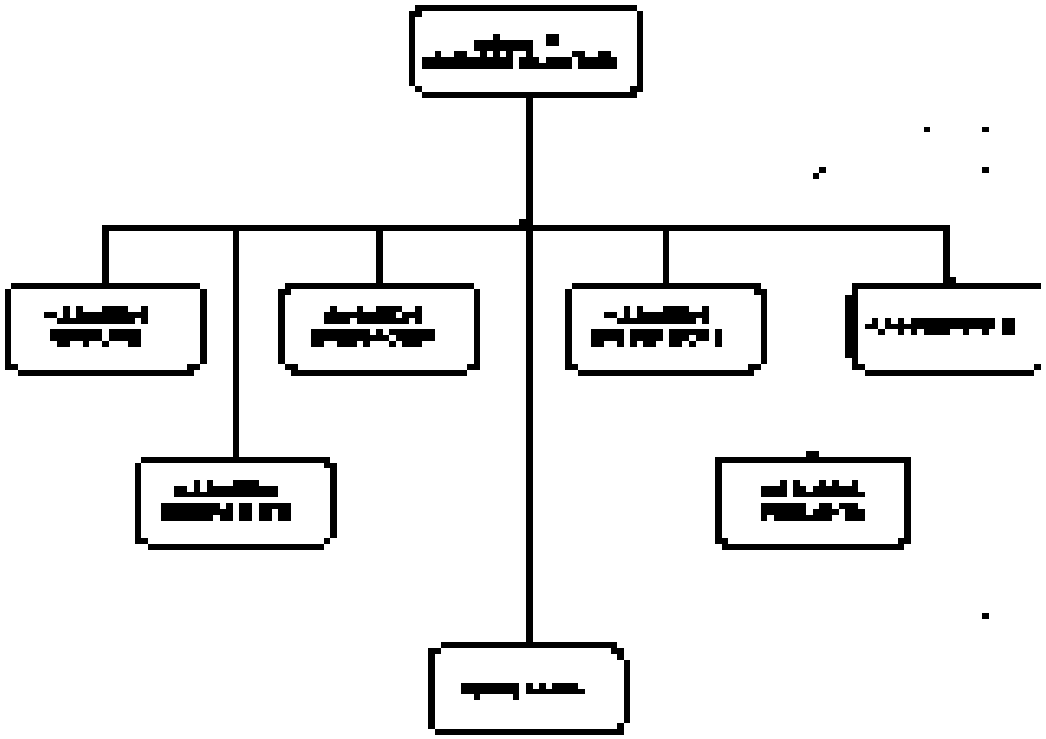
mengandung makna bahwa inti jiwa keprajuritan senantiasa bekerja, bertindak dan berbuat secara cepat dengan tepat sasaran dan penuh ketelitian sehingga benar-benar dapat dipertanggung jawabkan.

2.2.1 Arti dan Makna Simbol INFOLAHTA AD

1. Bintang Bersudut Lima merupakan lambang TNI AD yang merupakan Induk Organisasi Disinfolahtad. Memiliki ciri atau cita-cita yang tinggi dan luhur berdasarkan Pancasila.
2. Segitiga merupakan lambang yang menunjukkan bahwa Disinfolahtad harus mampu menyajikan informasi yang dibutuhkan oleh semua Pembina dalam setiap Strata Organisasi dalam jajaran TNI AD dengan tepat guna mendukung pembuatan keputusan yang arif dan bijaksana.
3. Garis Menyilang merupakan lambang tekad untuk mengikuti kemajuan teknologi informasi dan pengolahan data, juga melambangkan tekad untuk maju terus beriringan dengan perkembangan zaman.
4. Layar Pemantau Komputer menunjukkan bahwa dalam pelaksanaan tugas, Disinfolahtad menggunakan peralatan pengolahan data elektronis (komputer) sebagai sarana utama.
5. Daun Lontar memiliki makna cikal bakal peralatan media yang digunakan untuk menyimpan atau mencatat data. Delapan ujung pada bagian dalam (tengah) untuk menyatakan hari jadi Disinfolahtad yaitu tanggal 8 Maret.
6. Tangkai Daun Lontar melambangkan sifat personel Disinfolahtad yang tanggap, terampil dan teliti dalam melaksanakan tugas melalui pembangunan sistem, pengoperasian sistem dan dukungan teknisnya menghasilkan informasi yang cepat, tepat dan akurat.
7. Warna Jingga mempunyai arti keberanian yang agung dan dinamis. Artinya, dalam mengerjakan tugas penuh dengan keberanian, tidak putus asa, dan dinamis dalam berfikir, bertindak dan berinisiatif.
8. Warna Cream mempunyai arti netral, obyektif, dan adil. Artinya, dalam memberikan pelayanan dan penyajian informasi tidak memihak dan obyektif.

9. Warna Biru Langit mempunyai arti kesungguhan dan kesanggupan. Artinya, dalam melaksanakan tugas penuh dengan kesanggupan dan kesungguhan yang didasari oleh rasa kesadaran dan tanggung jawab.
10. Warna Kuning Emas mempunyai arti keagungan atau keluhuran, dan kebijaksanaan. Artinya, apapun tugas yang diberikan akan dilaksanakan dengan penuh tanggung jawab dan bijaksana [11].

2.3 Struktur Organisasi



Gambar 2.2. Struktur Organisasi

Dalam struktur organisasi yang berjalan di Disinfolahtad terdiri dari Eselon Pimpinan, Eselon Pembantu Pimpinan, dan Eselon Pelayanan. Adapun detail tugas dan kewajiban dari masing-masing eselon sebagai berikut [1] :

1. Sebagai eselon pimpinan, Kepala Dinas Informasi dan Pengolahan Data Angkatan Darat (Kadisinfolahtad) memiliki tugas dan kewajiban sebagai Pimpinan Satuan, sebagai Staf Khusus Kasad, dan sebagai Pembina

Fungsi. Dalam melaksanakan tugas dan kewajibannya Kadisinfolahtad bertanggung jawab langsung kepada Kasad dan dalam pelaksanaan tugas sehari-hari akan dikoordinansikan oleh Wakasad.

2. Sebagai eselon pembantu pimpinan, terdiri dari
    a. Kepala Sub Dinas Pembinaan Fungsi (Kasubdisbingfung) merupakan pembantu Kadisinfolahtad yang bertanggung jawab untuk menyelenggarakan kegiatan di bidang Pembinaan Fungsi Satuan Infolahta. Dalam melaksanakan tugasnya, Kasubdisbingfung dibantu oleh tiga kepala bagian yang terdiri dari Kepala Bagian Pembinaan Satuan (Kabagbinsat), Kepala Bagian Penataran dan Pelatihan (Kabagtarlat), dan Kepala Bagian Penelitian dan Pengembangan (Kabaglitbang). Dalam melaksanakan tugas dan kewajibannya, Kasubdisbingfung bertanggung jawab langsung kepada Kadisinfolahtad.
    b. Kepala Sub Dinas Pembinaan Materiil Sistem Informasi (Kasubdisbinmatsisfo) merupakan pembantu Kadisinfolahtad yang bertanggung jawab untuk menyelenggarakan kegiatan di dalam bidang Perbendaharaan Materiil Sistem Informasi yang dalam pelaksanaannya bertanggung jawab langsung kepada Kadisinfolahtad. Dalam melaksanakan tugasnya, Kasubdisbinmatsisfo dibantu oleh tiga Kepala Bagian yang terdiri dari Kepala Bagian Pengadaan (Kabagada), Kepala Bagian Pengendalian dan Pendistribusian (Kabagdaldisi), dan Kepala Bagian Pemeliharaan dan Perbaikan (Kabagharkan).
    c. Kepala Sub Dinas Pembinaan Sistem Informasi Administrasi (Kasubdisbinsisfomin) merupakan pembantu Kadisinfolahtad yang bertanggung jawab untuk menyelenggarakan kegiatan di dalam bidang Pembinaan Sistem Informasi Administrasi yang dalam pelaksaannya bertanggungjawab langsung kepada Kadisinfolahtad. Dalam melaksanakan tugasnya, Kasubdisbinsisfomin dibantu oleh empat Kepala Bagian yang

terdiri dari Kepala Bagian Sistem Informasi Administrasi Personel (Kabagsisfominpers), Kepala Bagian Sistem Informasi Administrasi Logistik (Kabagsisfominlog), Kepala Bagian Sistem Informasi Administrasi Perencanaan dan Anggaran serta Pengawasan dan Pemeriksaan (Kabagsisfominrengarwasrik), dan Kepala Bagian Program Aplikasi Sistem Informasi Administrasi (Kabagprogapsisfomin).

d. Kepala Sub Dinas Pembinaan Sistem Informasi Dukungan Operasional (Kasubdisbinsisfoops) merupakan pembantu Kadisinfolahtad yang bertanggung jawab untuk menyelenggarakan kegiatan di bidang Pembinaan Sistem Informasi Dukungan Operasional yang dalam pelaksanaannya bertanggung jawab langsung kepada Kadisinfolahtad. Dalam melaksakanan tugasnya, Kasubdisbinsisfoops dibantu oleh empat kepala bagian yang terdiri dari Kepala Bagian Sistem Informasi Intelijen (Kabagsisfointel), Kepala Bagian Sistem Informasi Teritorial (Kabagsisfoter), Kepala Bagian Sistem Informasi Operasi (Kabagsisfoops), dan Kepala Bagian Program Aplikasi Sistem Informasi Operasional (Kabagprogapsisfoops).
e. Kepala Sub Dinas Dukungan Pengolahan Data (Kasubdisduklahta) merupakan pembantu Kadisinfolahtad yang bertanggung jawab untuk menyelenggarakan kegiatan di bidang kesiapan dan pengoperasian sarana prasarana pengolahan data dan layanan informasi serta dukungan teknis yang dalam pelaksanaanya bertanggung jawab langsung kepada Kadisinfolahtad. Dalam melaksanakan tugasnya, Kasubdisduklahta dibantu oleh tiga kepala bagian yang terdiri dari Kepala Bagian Instalasi Pengolahan Data (Kabagstallahta), Kepala Bagian Peranti Keras dan Lunak (Kabagperankerlun), dan Kepala Bagian Sistem Jaringan Komunikasi Data (Kabagsisringkomta).

f. Kepala Sub Dinas Informatika (Kasubdisinfo) merupakan pembantu Kadisinfolahtad yang bertanggung jawab dalam menyelenggarakan kegiatan dalam bidang pengkajian, pengembangan ilmu pengetahuan dan teknologi informasi serta keamanan informasi yang berkaitan dengan pembinaan fungsi sistem informasi angkatan darat yang dalam pelaksaaannya bertanggung jawab langsung kepada Kadisinfolahtad. Dalam melaksanakan tugasnya, Kasubdisinfo dibantu oleh tiga kepala bagian yang terdiri dari Kepala Bagian Pusat Pengetahuan (Kabagpuspeng), Kepala Bagian Teknologi Informatika (Kabagtekinfo), dan Kepala Bagian Keamanan Informasi (Kabagkaminfo).

3. Sebagai Eselon Pelayanan, Sekretaris (Ses) Dinas Informasi dan Pengolahan Data Angkatan Darat (Disinfolahtad) merupakan unsur pelayanan Disinfolahtad yang bertanggung Jawab dalam menyelenggarakan kegiatan di bidang fungsi organik TNI AD yang dalam pelaksanaannya bertanggung jawab langsung kepada Kadisinfolahtad. Dalam melaksanakan tugasnya, Ses dibantu oleh lima kepala bagian yang terdiri dari Kepala Bagian Pengamanan dan Operasi (Kabagpamops), Kepala Bagian Teritorial (Kabagter), Kepala Bagian Umum (Kabagum), Kepala Bagian Tata Usaha dan Urusan Dalam (Kabagtuud), dan Kepala Bagian Perencanaan Program dan Anggaran (Kabagrenproggar).

2.4 Aplikasi Laporan Pelaksanaan Anggaran (LAPLAKGAR)

Sesuai dengan berlakunya Peraturan Menteri Keuangan Nomor 143/PMK.05/2018 mengenai Mekanisme Pelaksanaan Anggaran Belanja Negara di lingkungan Kemhan dan TNI, Aplikasi Laporan Pelaksanaan Anggaran (LAPLAKGAR) merupakan aplikasi yang bertujuan untuk membantu dalam melakukan pelaksanaan anggaran di lingkungan angkatan darat. Pelaksanaan Anggaran merupakan suatu kegiatan yang menyediakan informasi mengenai pelaksanaan dari satu periode seperti pendapatan, belanja atau pengeluaran dan sisa anggaran dalam satu periode. Informasi tentang pelaksanaan anggaran tersebut memerlukan suatu bentuk laporan sebagai sarana pengendalian, pengawasan dan pertanggung jawaban. Dalam penyusunan laporan pelaksanaan anggaran, diperlukan suatu pedoman dalam bentuk Petunjuk Teknis (JUKNIS).

JUKNIS penyusunan laporan pelaksanaan anggaran di Lingkungan TNI AD merupakan penjabaran dari petunjuk penyelenggaraan (JUKGAR) pelaksanaan anggaran di Lingkungan TNI AD. JUKNIS ini membahas mengenai tata cara Penyusunan Laporan Pelaksanaan Anggaran di Lingkungan TNI AD yang bersumber dari DIPA Petikan Satker yang menjadi pedoman pokok bagi pejabat perencanaan dan anggaran dalam menyusun laporan pelaksanaan anggaran di lingkungan TNI AD dari tingkat U.O. TNI AD sampai dengan Satker.

Ketentuan umum dalam JUKNIS diperlukan agar penyusunan laporan pelaksanaan anggaran di lingkungan TNI AD dapat dilaksanakan secara optimal, transparan, dan akuntabel. Ketentuan umum ini berisikan tentang tujuan, sasaran, sifat, peranan, organisasi, tugas dan tanggung jawab, syarat personel, teknis, sarana dan prasarana, serta faktor-faktor yang memengaruhi.

Tujuan dan Sasaran pada ketentuan umum dalam JUKNIS meliputi, memiliki tujuan untuk mewujudkan peningkatan kualitas pelaksanaan anggaran yang efektif, efisien, transparan, dan akuntabel berdasarkan peraturan dan perundang-undangan yang berlaku dengan sasaran terwujudnya laporan pelaksanaan anggaran DIPA Petikan Satker di Lingkungan TNI AD

yang efektif, efisien, transparan, akuntabel dan dapat dipertanggungjawabkan [2].

2.5 Aplikasi Berbasis Website

Aplikasi berbasis website merupakan sebuah aplikasi yang dikembangkan dengan menggunakan bahasa pemrograman, baik HTML, PHP, CSS, dan lainnya. Dimana aplikasi atau *software* berbasis website ini membutuhkan *web server* dan *browser* untuk menjalankan programnya. Aplikasi berbasis website dapat dijalankan melalui jaringan internet maupun intranet [12].

2.6 Monitoring Keamanan

Monitoring keamanan merupakan sebuah upaya yang dapat dilakukan untuk menjaga keamanan sistem pada sebuah aplikasi. Monitoring keamanan juga dapat diartikan sebagai memantau, mengawasi atau melakukan kontrol dari sebuah sistem keamanan pada sebuah aplikasi. Dengan melakukan monitoring keamanan, maka dapat membantu untuk menjaga kerahasiaan, integritas, dan ketersediaan data. Proses monitoring ini perlu dilakukan secara berkala untuk memastikan bahwa sistem keamanan pada sebuah aplikasi tergolong aman dan berjalan dengan seharusnya [13].

2.7 Pengujian Celah Keamanan

Pengujian celah keamanan merupakan sebuah langkah untuk mencari titik kerentanan pada sebuah sistem atau aplikasi yang berfungsi untuk memastikan guna menjaga sistem atau aplikasi tersebut dari hal-hal yang dapat menimbulkan keresahan dari sisi *security*, pengujian celah keamanan ini bertujuan untuk menjaga data dan informasi penting agar mempertahankan integritas data dan informasi tersebut. Pada umumnya pengujian celah keamanan memanfaatkan sebuah metode untuk mendukung pencarian titik kerentanannya. Salah satu tipe atau jenis dari pengujian keamanan yang dapat digunakan yaitu dengan melakukan *Vulnerability assessment* [14].

2.8 Keamanan Informasi

Keamanan informasi pada umumnya berkaitan dengan menjaga keamanan informasi tersebut dari berbagai gangguan seperti pencurian, kerusakan, kehilangan, dan lainnya. Setiap perusahaan, instansi maupun organisasi penting untuk menjaga keamanan informasi sebagai suatu kebutuhan penting, terutama dalam melindungi aset yang menjadi tanggung jawab setiap organisasi tersebut [15]. Aspek dari keamanan informasi terdiri dari CIA TRIAD, yaitu kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*). Tujuan keamanan informasi adalah untuk mencegah penipuan atau mendeteksi kecurangan dalam sebuah sistem berbasis informasi, di mana informasi tersebut tidak memiliki bentuk fisik. Sistem keamanan informasi memiliki berbagai ancaman, seperti *virus, worm, Trojan horse,* serta ancaman baik dari dalam maupun luar sistem, yang disengaja atau tidak disengaja [16] . Dengan demikian, untuk memastikan keamanan data, maka diperlukan implementasi mekanisme perlindungan yang efektif [17].

2.9 *Threat*

Threat merupakan suatu potensi yang dapat menimbulkan kerugian pada keamanan informasi, *Threat* dapat disebabkan oleh *cracker*, *virus*, *hacker*, dan lainnya. Kecerobohan manusia juga dapat dikategorikan sebagai sebuah *Threat*, adapun jenis *Threat* seperti *Spyware*, *Adware*, *Hacker Attack, Denial of Service Attack, Virus,* dan lain-lain [16].

2.10 *Attack*

Attack dapat diartikan sebagai sebuah serangan, dimana penyerangnya disebut sebagai *Attacker*. *Attack* merupakan suatu kegiatan yang melibatkan penyerang dalam melakukan serangan atau tindakan yang merugikan. Jenis *Attack* dibagi menjadi *Active Attack* dan *Pasif Attack. Active attack* merupakan serangan yang dilakukan dengan cara meng-*intercept* koneksi serta melakukan modifikasi informasi maupun data hingga mengambil alih

sistem. Sedangkan passif *attack* merupakan serangan yang meng- *intercept* sebuah informasi tanpa melakukan sesuatu terhadap informasi tersebut [16].

2.11 *Vulnerability*

Vulnerability merupakan sebuah kelemahan pada pengoprasian, implementasi, maupun *design* sistem yang dapat dilakukan eksploitasi, sehingga mengganggu bahkan mengancam sistem tersebut. *Vulnerability* juga bisa dimanfaatkan oleh seseorang untuk mengambil keuntungan dari sistem yang berhasil di eksploitasi, *Vulnerability* umumnya berkaitan dengan tiga unsur keamanan sistem, seperti sistem yang rentan atau cacat, penyerang mengetahui dan mengakses kelemahan sistem, serta penyerang melakukan eksploitasi terhadap kelemahan sistem tersebut. Adapun penyebabnya dapat berasal dari user, perancang sistem, dan device yang digunakan oleh sistem tersebut. Terdapat beberapa faktor munculnya *Vulnerability*, seperti kompleksitas, familiaritas, koneksifitas, *password* management *flaws*, *operating system design flaws*, internet *website browsing*, *software bugs*, dan *unchecked user input*. Selain *Vulnerability* pada sistem, terdapat *Vulnerability* yang memungkinkan untuk menyerang sebuah software seperti aplikasi, sistem operasi, dan lainnya. Berikut merupakan macam-macam *Vulnerability* pada software yang banyak dijumpai [18] :

1. *Memory Safety Violations*
2. *Input Validation Errors*
3. *Race Conditions*
4. *Privilege-confusion Bugs*
5. *Privilege Escalation*
6. *User Interface Failures*

2.12 Jenis dan Teknik *Scanning*

Scanning dapat diartikan sebagai sebuah proses pemerolehan *intelligence Gathering*. Hasil yang diharapkan dari *Scanning* seperti menemukan mesin target, *IP address*, dan port, menemukan *Vulnerability* dan jenis *Vulnerability* pada mesin target, menemukan keberadaan *firewall*, dan lainnya. Adapun jenis dan teknik *Scanning* meliputi [19]:

1. Network *Scanning*

   Sebuah tahapan yang dapat dilakukan untuk identifikasi mesin target pada sebuah *network*.

2. Port *Scanning*

   Sebuah tahapan yang dapat dilakukan untuk identifikasi *port* pada mesin target yang berhasil di deteksi, *port scanning* dapat dibagi menjadi beberapa jenis, yaitu *Vanilla, Strobe, Fragmented Packets*, UDP, *Sweep*, FTP *Bounce*, dan *Stealth Scan*.

3. *Vulnerability Scanning*

   Sebuah cara yang dapat dilakukan untuk memindai keberadaan celah kerentanan dari sebuah sistem, sehingga dapat menentukan bagaimana cara melakukan eksploitasi. *Vulnerability Scanning* digunakan untuk mencari kelemahan sistem tanpa dilakukan eksploitasi.

2.13 *Vulnerability* Assessment

Vulnerability assesment merupakan sebuah proses yang dapat dilakukan untuk mengumpulkan atau memperoleh informasi terkait keberadaan *Vulnerability* pada sebuah sistem atau aplikasi yang kemudian hasilnya akan dilakukan dokumentasi. *Vulnerability assessment* atau *Vulnerability analysis* juga didefinisikan sebagai sebuah proses pendefinisian, pengkasifikasian serta identifikasi dari berbagai *Vulnerability* atau celah kerentanan [19]. *Vulnerability* Assessment dibagi menjadi 3 kategori, yaitu :

1. *Level High*

   Kerentanan yang ditemukan pada *level* ini sangat berpotensi tinggi mengancam keamanan sebuah sistem atau aplikasi.

2. *Level Medium*

   Kerentanan yang ditemukan pada *level* ini bersifat lokal dan upaya penanganan yang dapat dilakukan juga bersifat lokal.

3. *Level Low*

   Kerentanan pada *level* ini tergolong rendah, dimana upaya pencegahan termasuk memadai.

2.14 Eksploitasi

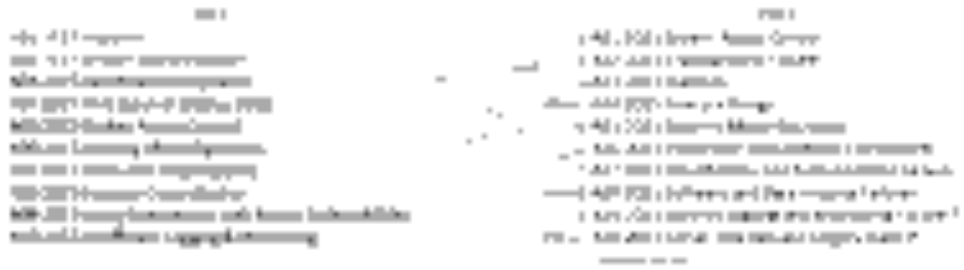
Eksploitasi dapat diartikan sebagai sebuah serangan yang berhasil dilakukan oleh *attacker* dengan memanfaatkan *Vulnerability* yang ada. Terdapat dua jenis eksploitasi yaitu *remote exploit* dan *local exploit*. *Remote exploit* merupakan sebuah serangan yang dilakukan dengan menggunakan jaringan komputer atau *network* tanpa harus melalui akses lokal pada sistem tersebut. Sedangkan *Local Exploit* merupakan sebuah serangan yang dilakukan melalui akses lokal kedalam sistem yang akan dilakukan eksploitasi [19].

2.15 *Open Web Application Security Project (OWASP)*

Open Web Application Security Project (OWASP) merupakan organisasi atau komunitas terbuka nonprofit terpercaya yang menyediakan alat, dokumentasi, dan lainnya untuk membantu para pengembang di seluruh dunia dalam melakukan peningkatan keamanan *software* aplikasi. Selain itu, OWASP bertujuan untuk membantu setiap orang maupun organisasi dalam melakukan analisis resiko keamanan aplikasi dengan tepat [20].

2.16 OWASP TOP 10

OWASP Top 10 merupakan dokumentasi yang disediakan oleh OWASP yang dapat membantu para pengembang untuk menyusun rangkaian pengujian yang lebih sistematis dan terarah dengan melakukan simulasi skenario serangan. Pengetahuan kerentanan ini dapat digunakan untuk menghasilkan kasus uji keamanan yang lebih terfokus sesuai dengan skenario serangan yang paling mungkin terjadi. Meskipun tujuan awal dari proyek OWASP Top 10 hanyalah untuk meningkatkan kesadaran di antara pengembang, namun OWASP Top 10 telah menjadi standarisasi keamanan aplikasi yang diakui secara umum [5]. Adapun OWASP TOP 10 tahun 2021 ditunjukkan pada gambar 2.3.



Gambar 2.3. OWASP TOP 10 2021 [5]

1. ***Broken Access Control***

   Broken Access Control merupakan sebuah kesalahan konfigurasi yang kurang tepat terhadap hak akses user yang menyebabkan data sensitif dan fungsi-fungsi tertentu dapat diakses diluar akses yang diberikan.

2. ***Cryptographic Failures***

   Cryptographic Failures merupakan sebuah kesalahan pada kriptografi dimana terdapat kegagalan atau kekurangan dalam penerapan yang berkaitan dengan kriptografi, hal ini sering mengakibatkan terbukanya informasi yang bersifat sensitif.

3. ***Injection***

   Injection merupakan sebuah kejadian dimana *attacker* mengirimkan data ilegal ke sebuah intepreter yang kemudian dieksekusi sebagai sebuah perintah tanpa mendapatkan hak akses atau otoritas yang valid.

4. ***Insecure Design***

   Insecure Design merupakan masalah keamanan yang timbul dari keputusan yang buruk dalam merancang dan mengembangkan arsitektur atau desain sistem. Hal ini dapat meliputi penggunaan teknologi yang rentan, tata letak aplikasi yang buruk, atau kurangnya perlindungan terhadap serangan umum seperti *injection attacks*. Dalam konteks keamanan *web*, *insecure design* and *architecture* dapat memungkinkan penyerang untuk dengan mudah mengambil alih atau merusak sistem, mencuri data rahasia, atau melakukan serangan lainnya.

5. ***Security Missconfiguration***

   Security Misconfiguration adalah masalah keamanan yang terjadi akibat konfigurasi atau pengaturan yang salah atau tidak memadai pada *server*, aplikasi, atau perangkat lunak. Hal ini dapat mengakibatkan eksposur informasi sensitif, akses tidak sah ke sistem, atau kerentanan keamanan lainnya.

6. ***Vulnerable and Outdated Components***

   Vulnerable and Outdated Components merupakan masalah umum yang terjadi ketika perangkat lunak yang digunakan pada suatu sistem terdapat komponen yang rentan atau usang, sehingga dapat dieksploitasi oleh penyerang untuk memperoleh akses tidak sah atau merusak sistem. Beberapa contoh komponen yang rentan dan tidak terbaru antara lain sistem operasi, *framework*, dan *library*.

7. ***Identification and Authentication Failures***

Identifikasi dan kesalahan autentikasi merupakan masalah yang terjadi ketika sistem gagal mengidentifikasi dan mengotentikasi pengguna dengan benar, sehingga memungkinkan pengguna yang tidak sah atau orang yang tidak berhak untuk mengakses informasi atau sumber daya yang seharusnya tidak mereka akses. Beberapa contoh termasuk *password* yang lemah, penggunaan autentikasi yang tidak aman, dan tidak memeriksa sesi yang tidak sah atau kedaluwarsa.

8. ***Software and Data Integrity Failures***

Software and Data Integrity Failures merupakan serangan terhadap integritas data dan perangkat lunak. Contoh dari serangan ini adalah memodifikasi atau menghapus data secara tidak sah, melakukan perubahan pada perangkat lunak atau aplikasi untuk mempengaruhi kinerjanya, atau mengirimkan data yang dimodifikasi dengan tujuan untuk merusak integritasnya. Serangan ini dapat terjadi karena kurangnya validasi data atau penggunaan teknologi enkripsi yang lemah.

9. ***Security Logging and Monitoring Failures***

Security Logging and Monitoring Failures adalah ketidakmampuan sistem untuk menghasilkan, merekam, memantau, atau menganalisis kejadian keamanan yang terjadi di dalamnya. Kegagalan dalam *logging* dan monitoring keamanan dapat mempengaruhi kemampuan organisasi untuk mendeteksi serangan, menganalisis kejadian yang telah terjadi, dan memberikan respons yang tepat waktu dan efektif. Hal ini dapat menyebabkan hilangnya data sensitif, kerentanan yang tidak terdeteksi, dan bahkan mengancam kelangsungan hidup suatu organisasi.

10. *ServerSide Request Forgery*

ServerSide Request Forgery (SSRF) adalah kerentanan keamanan yang memungkinkan penyerang untuk memanipulasi permintaan server melalui aplikasi yang terdapat pada server itu sendiri. SSRF dapat terjadi ketika aplikasi web tidak memverifikasi input yang diberikan oleh pengguna dan kemudian mengirimkan permintaan dari server ke sumber yang tidak dipercayai atau bahkan ke server *internal*. Serangan SSRF dapat digunakan untuk mencuri data sensitif, memperoleh akses ke server *internal*, atau memanipulasi fungsi aplikasi yang terhubung ke jaringan lain.

2.17 API

Application Programming Interface (API) merupakan suatu antarmuka yang digunakan oleh program perangkat lunak untuk berkomunikasi dengan program lainnya, manusia, dan dunia melalui internet. Desain API mencerminkan banyak hal tentang program di baliknya, termasuk model bisnis, fitur produk, dan terkadang *bug*. API digunakan untuk memenuhi kebutuhan pertukaran informasi dengan penyedia data dalam menyelesaikan masalah tertentu. Fitur dan produk tambahan dapat dibuat dengan menggunakan data atau interaksi dari platform khusus melalui API, hal ini memungkinkan bisnis untuk mengembangkan produk yang unik dengan cepat dan memanfaatkan teknologi yang sudah ada [21].

2.18 OWASP ZAP API

OWASP ZAP API merupakan sebuah API yang disediakan oleh OWASP bagi para pengembang aplikasi yang ingin membuat alat atau pemindai untuk mendeteksi celah kerentanan pada aplikasi web. OWASP ZAP API menyediakan antarmuka pemrograman aplikasi yang memungkinkan pengguna untuk mengakses dan mengontrol fungsi dari OWASP ZAP. Dengan menggunakan API ini, pengguna dapat mengotomatisasi tugas-tugas pengujian keamanan dan mengintegrasikan ZAP dengan alat-alat keamanan lainnya untuk melakukan pengujian keamanan aplikasi web secara menyeluruh. API ini menyediakan sejumlah metode yang dapat digunakan untuk melakukan berbagai tindakan, seperti memulai dan menghentikan pemindaian, menetapkan target untuk pemindaian, dan mengambil hasil dari pemindaian. Untuk menggunakan OWASP ZAP API, diperlukan pembuatan permintaan HTTP ke titik akhir API, yang biasanya di-*hosting* pada mesin yang sama dengan aplikasi ZAP. API menggunakan desain *Representational State Transfer* (RESTful), yang berarti bahwa pengembang dapat menggunakan metode HTTP standar (seperti *GET*, *POST*, *PUT*, dan *DELETE*) untuk melakukan berbagai tindakan [22].

2.19 Pemrograman Berorientasi Objek (OOP)

Object-oriented programming (OOP) merupakan sebuah model pemrograman yang diciptakan untuk memudahkan pengembang. OOP memungkinkan pemecahan masalah yang kompleks menjadi masalah-masalah yang lebih kecil dan lebih mudah dipahami. Prinsip utama dari OOP adalah melakukan segala sesuatu melalui objek. Objek adalah potongan kecil kode yang terdiri dari data dan perilaku. Dalam aplikasi, semua objek ini terhubung satu sama lain dan saling berbagi data untuk memecahkan masalah. Dengan OOP dapat membuat kode lebih mudah dipahami dan diorganisir. OOP dapat membuat pengembang yang ingin membuat aplikasi web *e-commerce* dapat memecah masalah menjadi objek-objek seperti produk, keranjang belanja, dan *checkout*, setiap objek ini memiliki data dan perilaku

tersendiri dan tentunya dapat berkomunikasi dengan objek lain dalam aplikasi untuk menyelesaikan tugas yang lebih besar. OOP juga memungkinkan pengembang untuk membuat kode yang lebih mudah dipelihara dan dimodifikasi, serta meningkatkan efisiensi dalam pengembangan aplikasi [23].

2.20 UML (Unified Modeling Languange)

Unified Modeling Languange atau UML merupakan alat diagram atau sebuah pemodelan visual yang digunakan untuk mendesain atau merancang dengan menggunakan bahasa grafis untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dari sistem perangkat lunak yang beriorentasikan pada objek. UML dapat membantu untuk memahami lebih dalam terkait perangkat lunak atau sistem yang akan dikembangkan di antara para pengembang dan pengguna [24].

2.20.1 Use Case Diagram

Use Case Diagram adalah sebuah diagram yang berfokus pada identifikasi kebutuhan fungsional atau fitur-fitur dari suatu sistem atau aplikasi yang akan dikembangkan. *Use Case Diagram* bertujuan untuk menggambarkan hubungan antara aktor (pengguna sistem) dan fungsi-fungsi (use case) yang tersedia dalam sistem tersebut [24].

2.20.2 *Activity Diagram*

Activity diagram merupakan sebuah diagram yang berfokus pada suatu kegiatan dalam sistem yang berurutan dan paralel dengan melibatkan setiap persyaratan fungsional sistem tersebut. *Activity diagram* biasanya digunakan untuk memodelkan alur kerja atau proses dalam sebuah sistem dan membantu dalam memvisualisasikan urutan aktivitas yang terlibat dalam menyelesaikan suatu tugas atau mencapai tujuan tertentu dalam sistem [24].

2.20.3 *Class* Diagram

Class diagram merupakan sebuah diagram yang digunakan untuk menggambarkan struktur sistem dalam hal kelas dan objek. *Class diagram* dapat memperlihatkan hubungan antara kelas, atribut kelas, dan metode kelas yang ada dalam sistem yang dirancang. Dengan menggunakan *class diagram*, pengembang perangkat lunak dapat memperjelas struktur kelas dan hubungan antara kelas yang ada dalam sistem [24].

2.20.4 *Sequence* Diagram

Sequence diagram adalah sebuah diagram yang digunakan digunakan untuk memodelkan interaksi antar objek dalam suatu skenario atau kasus penggunaan. Diagram ini menggambarkan objek-objek yang terlibat dalam skenario dan urutan pesan yang diperlukan untuk melaksanakan fungsionalitas yang dijelaskan dalam skenario tersebut. Dengan menggunakan diagram ini, kita dapat memvisualisasikan interaksi antar objek dan memahami bagaimana pesan dan data saling dipertukarkan antar objek dalam sistem [24].

2.21 Metode Pengujian Sistem

Metode pengujian sistem atau perangkat lunak merupakan sebuah aktifitas yang dilakukan atau direncanakan secara sistematis untuk melakukan pengujian atau melakukan evaluasi terhadap kemungkinan kesalahan dan juga kesesuaian dengan spesifikasi yang diinginkan pada pengembangan sistem [25].

2.21.1 Pengujian *Whitebox*

Whitebox testing merupakan teknik pengujian perangkat lunak yang berfokus untuk memeriksa desain dan kode sumber program yang digunakan untuk membuat perangkat lunak. Dalam pengujian ini, aplikasi diuji untuk melihat apakah fungsi-fungsi, masukan, dan keluaran telah sesuai dengan spesifikasi kebutuhan [26].

2.21.2 Pengujian Blackbox

Black Box testing merupakan metode pengujian perangkat lunak yang fokus pada pengujian fungsional tanpa memperhatikan desain dan kode program. Tujuan *blackbox testing* ini adalah untuk memastikan apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Hal ini dilakukan dengan cara membandingkan hasil yang dikeluarkan oleh perangkat lunak dengan spesifikasi fungsional yang telah ditentukan sebelumnya, tanpa melihat bagaimana cara kerja perangkat lunak tersebut di dalamnya. Pengujian ini dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan yang diharapkan oleh pengguna dan sesuai dengan spesifikasi yang diinginkan [25].

2.21.3 Pengujian UAT (User Acceptance Testing)

Pengujian UAT merupakan suatu pengujian yang dapat dilakukan untuk mencegah suatu kegagalan dalam proyek pengembangan perangkat lunak. UAT dilakukan oleh pengguna akhir untuk memastikan bahwa aplikasi yang dibangun telah memenuhi kebutuhan pengguna dengan baik. Dalam pengembangan perangkat lunak, terdapat tiga hal yang harus diperhatikan diantaranya meliputi UAT akan memperhatikan fungsi logika atau kode program, UAT akan mengukur kebutuhan pengguna dan tujuan pembangunan sistem sesuai dengan sistem yang dibangun, dan UAT akan membatasi bagaimana sistem tersebut selesai [27]. Pengujian ini dapat dilakukan dengan berbagai metode seperti wawancara dan kuesioner untuk mendapatkan hasil akhir penggunaan aplikasi sesuai dengan sisi pengguna [28].

2.22 *Tools* Yang Digunakan

2.22.1 Visual *Studio Code*

Visual Studio Code adalah sebuah *text editor opensource* yang dapat digunakan untuk mengembangkan berbagai macam aplikasi dan proyek, termasuk aplikasi web, desktop, dan mobile. *Visual Studio Code* memiliki fitur-fitur seperti *syntax highlighting*, *debugging*, *auto completion*, *version*

control, dan lainnya. *Visual Studio Code* juga dapat diperluas melalui penggunaan ekstensi yang tersedia secara gratis di *marketplace*. *Visual Studio Code* dapat digunakan di berbagai sistem operasi, termasuk Windows, macOS, dan Linux [29].

2.22.2 Python

Python merupakan bahasa pemrograman *open source* beriorentasi objek yang mudah dipelajari karena pengkodeannya mudah dibaca, dipahami, dan ditulis. Python seringkali digunakan untuk mengembangkan aplikasi berbasis website. Dengan berbagai keunggulannya, membuat bahasa pemrograman python menjadi bahasa tingkat tinggi yang digemari oleh pemula hingga *expert* [30].

2.22.3 Flask

Flask merupakan kerangka kerja yang sering disebut dengan *micro framework*. Flask ditulis dengan menggunakan bahasa python. Flask menyediakan fitur dasar yang dapat digunakan oleh para pengembang seperti *routing*, integrasi dengan berbagai *library* dan *database*, serta *request* dan *response handling*. Flask banyak digunakan untuk mengembangkan web di berbagai tingkatan [31].

2.22.4 SQL Alchemy

SQLAlchemy merupakan sebuah alat atau *toolkit* dan ORM (*Object Relational Mapping*) yang dapat digunakan dalam *database* pyhton. SQLAlchemy dapat memungkinkan pengembang untuk berinteraksi dengan *database* yang beriorentasi objek. Keunggulan SQLAlchemy yaitu dapat mendukung berbagai jenis basis data [32].

2.22.5 Jinja2

Jinja2 merupakan sebuah *library* yang digunakan dalam bahasa pemrograman python yang dirancang agar aman, cepat dan fleksibel. Hal ini dikarenakan karena jinja2 menyediakan fitur seperti *variable substitution*,

inheritance, *control structures* dan sebagainya yang memungkinkan pengembang untuk membuat template dengan mudah dan lebih fleksibel. Dalam pengembangan aplikasi berbasis website, jinja2 marak kali digunakan bersamaan dengan kerangka kerja flask untuk menghasilkan tampilan yang dinamis [33].

2.22.6 Bootstrap

Bootstrap merupakan sebuah kerangka kerja *opensource* yang dapat digunakan oleh para pengembang untuk membangun antarmuka website. Kemampuannya dalam desain web yang responsif membuat bootstrap menjadi kerangka kerja populer. Dengan menggunakan bootstrap, maka pembuatan antarmuka seperti *layouting*, penyesuaian komponen lain seperti tombol, navigasi, dan lainnya menjadi lebih cepat dan mudah karena bootstrap menyediakan kode yang dapat dipakai untuk mengoptimalkan tampilan antarmuka untuk berbagai perangkat dan responsivitas berbagai ukuran layar [34].

2.23 *Tools* Pengujian Kerentanan Sistem

Tools pengujian kerentanan sistem merupakan sebuah alat yang digunakan untuk melakukan pengujian terhadap celah kerentanan dari sebuah sistem dengan melakukan *automated scanning* terhadap aplikasi yang akan diidentifikasi celah kerentanannya. Alat-alat yang digunakan bertujuan untuk membantu melakukan pengecekan terhadap keakuratan hasil dari identifikasi celah kerentanan yang ditemukan sesuai dengan hasil identifikasi aplikasi monitoring keamanan yang dibangun.

2.23.1 Zed Attack Proxy

Zed attack proxy atau ZAP merupakan sebuah alat atau aplikasi sumber terbuka yang dikembangkan oleh OWASP untuk melakukan identifikasi celah keamanan dengan cara melakukan pemindaian otomatis (Vulnerability Scanning). ZAP cocok digunakan bagi para pengembang untuk melakukan pengamanan awal yang berfokus pada keamanan aplikasi website. Selain itu,

ZAP sering digunakan oleh para pengembang atau penguji fungsional untuk melakukan uji penetrasi. ZAP juga memungkinkan untuk membantu pengembang dalam menemukan kerentanan keamanan secara manual [20].

2.23.2 Postman

Postman merupakan sebuah alat atau aplikasi yang digunakan untuk membangun, menguji, hingga melakukan modifikasi API. Postman memiliki alat untuk membangun API, yang terdiri dari perancangan, pengujian, dokumentasi, hingga publikasi [35]. Pada penelitian ini, postman digunakan untuk membantu menguji endpoint API OWASP ZAP dan juga menguji performa dari aplikasi monitoring keamanan. Pengujian API bertujuan untuk memastikan bahwa fungsi API pada aplikasi monitoring keamanan dapat berjalan dengan baik. Sedangkan pengujian performa bertujuan untuk memastikan bahwa aplikasi monitoring keamanan dapat berjalan sesuai dengan batas normal permintaan klien terhadap aplikasi.