BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitan sebelumnya yang telah dilakukan oleh beberapa orang tentang *security assessment* menjadi salah satu bahan yang digunakan sebagai referensi dan landasan pemikiran dalam melakukan penelitian ini. Pada bagian ini peneliti akan menjabarkan tentang bagaimana para peneliti tersebut melakukan penelitian dengan tempat dan kasus yang berbeda – beda.

Pada tahun 2015 Sugandh Shah dan B.M Mehtre melakukan kegiatan *security assessment* dengan menggunakan metode *Vulnerability Assessment and Penetration Testing* (VAPT) dan *Tool* Nirikshak dengan versi 1.0 di *Institute for Development and Research in Banking Technology*. Pada penelitian ini Sugandh Shah dan B.M Mehtre menjelaskan bahwa metode ini merupakan metode proaktif yang efektif untuk meningkatkan keamanan sistem informasi pada suatu organisasi. Namun, masih terdapat kekurangan salah satunya pada *tool* yang digunakan yaitu nirikshak. *Tool* versi ini masih dalam pengembangan sehingga *plugin* dan kedalaman dalam melakukan *vulnerability scanning* masih kurang jika dibandingkan *vulnerability scanner* lain seperti Nessus atau OWASP ZAP.

Pada tahun 2016 Mohammed Akour dan Izzat Alsmadi (Akour & Alsmadi, 2016) melakukan kegiatan *vulnerability assessment* menggunakan *tool* Nexpose yang dikembangkan oleh Rapid7 LLC (Rapid7, 2007). *Vulnerability assessment* ini dilakukan di 20 Universitas di Jordania. Hasil dari *vulnerability assessment* ini menujukan *tool* ini dapat melakukan *vulnerability scan* dengan cukup efektif dalam *moderate level passive testing*. Banyak dari universitas yang menjadi sampel belum mengimplementasikan desain keamanan hal ini dibuktikan dengan ditemukannya 436 kerentanan 76 diantaranya adalah kerentanan dengan tingkat kritikal, 339 dengan tingkat berbahaya, dan 21 tingkat menengah.

Pada tahun 2016 penelitian yang dilakukan oleh Tashia Indah Nastiti mengatakan bahwa Universitas Gadjah Mada memiliki *website* yang berisi data

tentang nomor jaminan sosial, kartu kredit dan data sensitif lainnya (Nastiti, 2016). Oleh sebab itu dibutuhkan sebuah kegiatan untuk melakukan pengujian keamanan untuk mengevaluasi sistem keamanan pada *website* tersebut. Kegiatan ini menggunakan *tool* OWASP ZAP untuk mencari celah keamanan. Terdapat kurang lebih sepuluh celah keamanan yang ditemukan pada *website* tersebut. Tujuan dari penelitian ini adalah mengevaluasi dan memastikan proses keamanan yang dijalankan oleh *website* tersebut sudah berjalan dengan baik

Pada tahun 2017 Fikri Zulfi melakukan kegiatan evaluasi keamanan pada web aplikasi SISTER (Sistem Informasi Terpadu) Universitas Jember (ZULFI, 2017). Penelitian ini mengacu pada metode VAPT (*Vulnerability Assessment and Penetration Testing*) untuk mengetahui kelemahan yang dapat menyebabkan kegagalan proses bisnis pada web aplikasi SISTER. Tujuan dari penelitiannya adalah untuk mengevaluasi dan memberikan usulan perbaikan pada sistem keamanan web aplikasi SISTER. Beberapa *tools* pendukung yang digunakan pada penelitian ini adalah W3af dan OWASP ZAP yang digunakan untuk pemindaian kelemahan pada web aplikasi SISTER, Metasploit yang digunakan untuk mengendalikan komputer jarak jauh dan Nmap yang digunakan untuk mendeteksi *port* yang digunakan pada *web* aplikasi tersebut. Hasil yang didapatkan dari penelitian ini adalah ditemukannya beberapa kelemahan, seperti SQL *Injection*, *Cross Site Scripting* dan lain lain.

Pada tahun 2014 Indraneel Mukhopadhyay, Shilpam Goswami, Eshita Mandal melakukan penelitian untuk membandingkan kelebihan dan kekurangan dari *tool vulnerability scan* yang sering digunakan ketika melakukan web *penetration testing*. *Tool* yang dibandingkan yaitu Skipfish, Wapiti, Arachni, Nessus, w3af, Acunetix, Websecurify. *Tool – tool* tersebut dibandingkan dengan cara melihat fitur yang dimliki dan mencoba setiap *tool* untuk melakukan vulnerability scanning terhadap jenis – jenis kerentanan yang ada. Pada gambar 2.1 menujukkan hasil perbandingannya.

FEATURES	skipfish	Wapiti	Arachni	Nessus	w3af	Acunetix	Websecurify
Injection	√	√	√	√	√	√	√
Cross-site scripting(XSS)	√	√	√	√	√	√	√
Broken Authentication and Session Management	√		√	√	√	√	√
Insecure Direct Object Reference		√	√		√	√	√
Cross-site Request Forgery(CSRF)			√	√		√	
Security Misconfigurations	√		√	√			
Insecure Cryptographic Storage	√		√	√			
Failure to Restrict URL	√			√			
Insufficient Transport Layer Protection				√			
Unvalidated Redirect and Forwards	√						

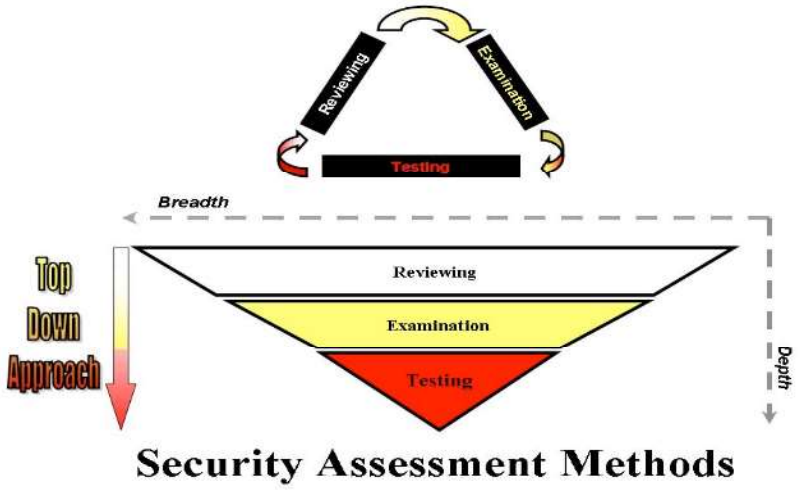
Gambar 2. 1 Perbandingan Dari Beberapa *Tool* (Mukhopadhyay, Goswami, & Mandal, 2014)

Dari gambar tersebut dapat disimpulkan *tool* Nessus memilik nilai tertinggi karena dari 10 jenis kerentanan yang diujikan Nessus bisa mendeteksi 8 kerentanan.

2.2 *Information Technology Security Assessment*

Information technology security assessment adalah pengukuran untuk suatu model keamanan pada sebuah sistem di organisasi atau perusahaan (Miles, Rogers, Fuller, Hoagberg, & Dykstra, 2004). Model keamanan adalah cara bagaimana keamanan sistem informasi diimplementasikan pada sebuah organisasi. Pengukuran ini bertujuan untuk memberikan informasi tentang celah keamanan sistem informasi yang terdapat pada organisasi atau perusahaan tersebut yang kemudian hasil pengukuran akan digunakan untuk meningkatkan keamanan dari sistem informasi.

Security assessment bergantung kepada tiga fase penilaian utama yang saling terkait, yaitu fase peninjauan, fase pemeriksaan, fase pengujian. Tiga fase tersebut dapat secara akurat menilai teknologi, orang, dan proses yang merupakan bagian dari keamanan(Aziz, 2011) sebagaimana dijelaskan dibawah.



Gambar 2. 2 Security Assessment Process (Aziz, 2011)

2.2.1 Fase Peninjauan

Fase peninjauan merupakan proses dilakukan pengumpulan informasi terkait sistem yang akan dilakukan proses *assessment*. Proses ini dapat berupa wawancara kepada pihak organisasi. Informasi yang dikumpulkan mencakup evaluasi kebijakan, prosedur, aplikasi, dan jaringan untuk menemukan kerentanan. Fase peninjauan ini dilakukan untuk memahami bagaimana sistem bekerja.

2.2.2 Fase Pemeriksaan

Fase pemeriksaan adalah proses dilakukannya pemeriksaan teknis dari sisi sistem atau jaringan komputer untuk mengidentifikasi celah keamanan yang berada pada sistem tersebut. Proses ini termasuk analisis teknis pada *firewall*, *intrution detection system*, dan perangkat jaringan komputer pada sistem.

2.2.3 Fase Pengujian

Fase pengujian biasa disebut juga dengan *penetration testing* adalah proses mencari celah keamanan, yang memungkinkannya masuk ke dalam sistem atau jaringan. Metode peninjauan dan pemeriksaan akan memberikan informasi yang berguna untuk pengujian kedepannya.

2.3 *Vulnerability Assessment & Penetration Testing*

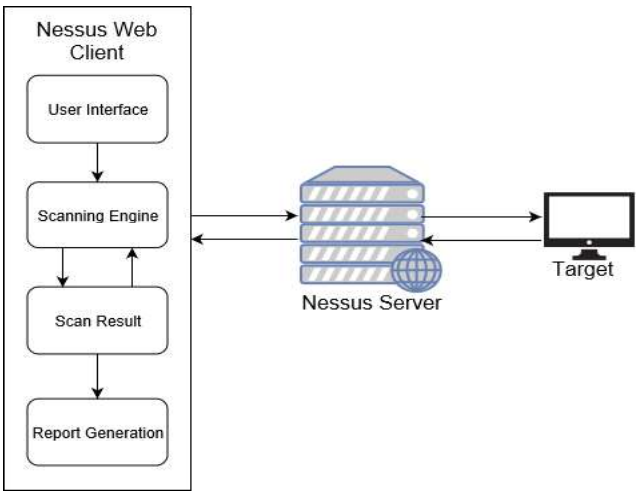
Vulnerability Assessment & Penetration Testing merupakan sebuah metode yang digunakan dalam melakukan penilaian keamanan terhadap suatu sistem informasi. VAPT adalah gabungan dari dua aktivitas yaitu, *Vulnerability*

Assessment dan *Penetration Testing*. *Vulnerability Assessment* adalah sebuah proses pemindaian sistem, *software*, atau jaringan komputer untuk menemukan kelemahan atau *loophole* yang terdapat pada sistem. Kelemahan ini bisa berupa *backdoor* yang bisa digunakan untuk menyerang sistem(Goel & Mehtre, 2015).

Penetration Testing adalah tahap selanjutnya dari *vulnerability assessment. Penetration Testing* merupakan tahap eksploitasi pada sistem dilakukan dengan cara legal untuk menemukan celah keamanan yang mungkin dieksploitasi pada sistem. Pada tahap ini *tester* memiliki hak untuk mencoba masuk ke dalam sistem dan mengeksploitasinya(Goel & Mehtre, 2015).

2.4 Nessus

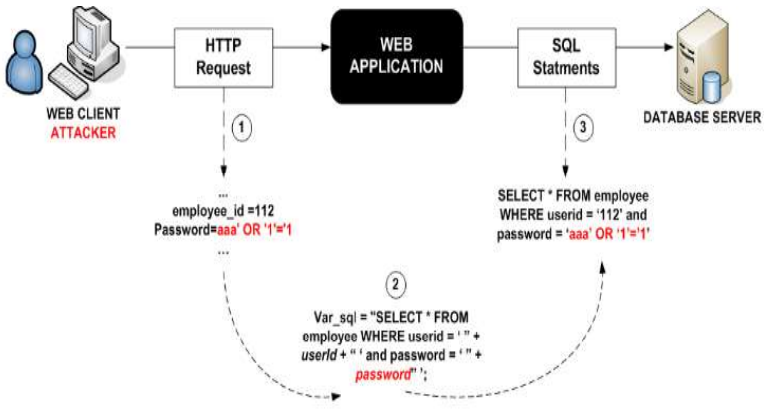
Nessus adalah salah satu produk penilaian kerentanan keamanan yang paling banyak digunakaan pertama kali dirilis oleh Renaud Deraison pada tahun 1998. alat ini telah menjadi salah satu alat pemindaian kerentanan yang paling populer digunakan di seluruh industri selama 15 tahun terakhir (Kumar, 2014). Nessus menyediakan pemindaian kerentanan untuk perangkat jaringan, *virtual host*, sistem operasi, basis data, aplikasi web, dan jaringan hibrid IPv4 / IPv6 (Li, Liang, Yang, & Chen, 2010). Nessus menggunakan *Common Vulnerability and Exposure* (CVE) sebagai standarnya. CVE adalah standar untuk penamaan kerentanan keamanan informasi . Nessus menggunakan *Common Vulnerability and Exposure* (CVE) sebagai standarnya. CVE adalah standar untuk penamaan kerentanan keamanan informasi (Li et al., 2010). Salah satu fitur yang menarik dari Nessus adalah ini merupakan aplikasi *open source* dan banyak orang yang berkontribusi setiap hari. Akan ada *plug-in* untuk kerentanan baru dalam beberapa hari setelah kerentanan keamanan dirilis ke publik (Mukhopadhyay et al., 2014).



Gambar 2. 3 Nessus Scanner Work Flow

2.5 SQL *Injection*

SQL Injection adalah jenis serangan dimana penyerang menambahkan Structured Query Language ke *input box* pada sebuah *website* untuk mendapatkan hak akses atau mengganti data. SQL *Injection* memungkinkan penyerang untuk mengirimkan perintah secara langsung ke *database* aplikasi (Tajpour, Ibrahim, & Sharifi, 2012).



Gambar 2. 4 SQL *Injection Process*

Gambar 2.3 diatas merupakan gambar bagaimana serangan SQL *Injection* terjadi. Penyerang mengirimkan http *request* berisi SQL statemen yang telah dimodifikasi dari *web client* menuju *server* dengan tujuan menggati SQL statemen

yang asli menjadi SQL statemen yang telah dimodifikasi untuk memperoleh data yang diinginkan dari *database*. Menurut (Nagpal, Chauhan, & Singh, 2015; Nagpal, Singh, Chauhan, & Panesar, 2015; Tajpour et al., 2012) SQL *injection* terbagi menjadi beberapa jenis sebagai berikut:

2.5.1 *Tautologies*

Jenis serangan ini menyisipkan token SQL ke pernyataan permintaan bersyarat untuk dievaluasi selalu benar. Jenis serangan ini digunakan untuk melewati kontrol otentikasi dan akses ke data dengan mengeksploitasi bidang *input* rentan yang menggunakan klausa WHERE.

2.5.2 *Logical Incorrect Queries*

ketika permintaan ditolak, pesan kesalahan dikembalikan dari *database* termasuk informasi *debugging* yang berguna. Pesan kesalahan ini membantu penyerang untuk menemukan parameter rentan dalam aplikasi dan akibatnya basis data aplikasi. Bahkan penyerang menyuntikkan masukan sampah atau token SQL dalam kueri untuk menghasilkan kesalahan syntax, ketik ketidakcocokan, atau kesalahan logis dengan tujuan.

2.5.3 *Union Query*

Dengan teknik ini, penyerang bergabung dengan kueri yang disuntikkan ke kueri aman oleh kata UNION dan kemudian bisa mendapatkan data tentang tabel lain dari aplikasi.

2.5.4 *Piggy-backed Queries*

Dalam jenis serangan ini, penyusup mengeksploitasi *database* oleh pembatas kueri, seperti ";", untuk menambahkan kueri tambahan ke kueri asli. Dengan database serangan yang berhasil menerima dan menjalankan beberapa kueri berbeda. Biasanya kueri pertama adalah kueri yang sah, sedangkan kueri berikutnya bisa tidak sah. Jadi penyerang dapat menyuntikkan perintah SQL ke *database*.

2.5.5 *Stored Procedure*

Stored procedure adalah bagian dari *database* yang *programmer* bisa mengatur lapisan abstraksi tambahan pada *database*. Seperti prosedur yang

tersimpan dapat dikodekan oleh *programmer*, jadi, bagian ini adalah sebagai menyuntikkan dapat sebagai formulir aplikasi *web*. Tergantung pada prosedur tertentu yang tersimpan di *database* ada berbagai cara untuk menyerang.

2.5.6 *Inference*

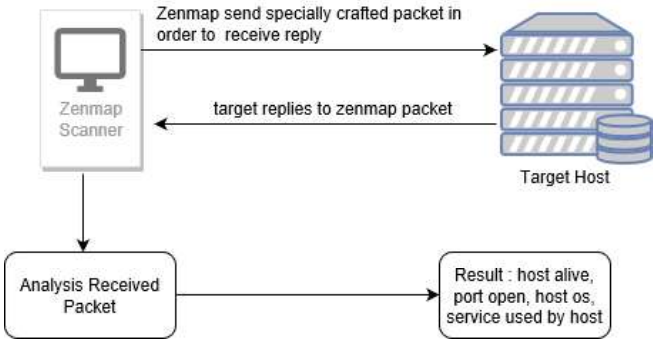
Dengan jenis serangan ini, penyusup mengubah perilaku *database* atau aplikasi. Ada dua teknik serangan terkenal yang didasarkan pada penyimpulan: *blind-injection* dan *timing attack*.

2.5.7 *Alternate Encodings*

Dalam teknik ini, penyerang memodifikasi kueri injeksi dengan menggunakan pengodean alternatif, seperti heksadesimal, ASCII, dan Unicode. Karena dengan cara ini, mereka dapat keluar dari filter pengembang yang memindai kueri masukan untuk "karakter buruk" yang dikenal khusus. Misalnya penyerang menggunakan char (44) bukan kutipan tunggal yang merupakan karakter buruk. Teknik ini dengan bergabung dengan teknik serangan lain bisa menjadi kuat, karena dapat menargetkan lapisan yang berbeda dalam aplikasi sehingga pengembang harus terbiasa dengan semuanya untuk menyediakan pengkodean defensif yang efektif untuk mencegah serangan pengkodean alternatif.

2.6 Nmap

Nmap adalah *tool* untuk eksplorasi jaringan atau mengaudit keamanan jaringan. *Tool* ini merupakan *tool* yang ideal untuk melakukan *vulnerability assessment* (Li et al., 2010). Nmap melakukan pemindaian terhadap satu *host* untuk mengetahui layanan yang digunakan (nama dan versi aplikasi), sistem operasi dan jenis filter firewall yang digunakan. Nmap berjalan disemua sistem operasi komputer, seperti Linux, Windows, dan Mac OS X (Wang & Yang, 2017). Nmap menjalankan setiap baris perintah menggunakan *command prompt* atau terminal. Nmap juga tersedia dalam versi GUI dengan nama Zenmap.



Gambar 2. 5 Zenmap Scanner Work Flow

2.7 SQLMap

SQLMap adalah alat pengujian penetrasi *open source* yang mengotomatiskan proses mendeteksi dan mengeksploitasi kelemahan SQL *injection*. Sqlmap memiliki mesin pendeteksi yang kuat, banyak fitur untuk tester penetrasi akhir dan berbagai *switch* yang berlangsung dari *fingerprinting database*, melalui pengambilan data dari *database*, untuk mengakses sistem file yang mendasari dan mengeksekusi perintah pada sistem operasi (SQLMap, 2006).