BAB II
LANDASAN TEORI

2.1 *Web Vulnerability*

World Wide Web (WWW atau *Web1*) merupakan salah satu "*killer applications*" yang menyebabkan populernya *Internet*. WWW dikembangkan oleh Tim Berners-Lee ketika bekerja di CERN (Swiss). Sejarah dari penemuan ini dapat dibaca pada buku karangan Tim Berners-Lee. Kehebatan *Web* adalah kemudahannya untuk mengakses informasi, yang dihubungkan satu dengan lainnya melalui konsep *hypertext*. Informasi dapat tersebar di manamana di dunia dan terhubung melalui *hyperlink* dan <http://www.w3.org>. (Budi, 2002)

Berkembangnya WWW dan *Internet* menyebabkan pergerakan sistem informasi untuk menggunakannya sebagai basis. Banyak sistem yang tidak terhubung ke *Internet* tetapi tetap menggunakan basis *Web* sebagai basis untuk system informasinya yang dipasang di jaringan *Intranet*. Untuk itu, keamanan sistem informasi yang berbasis *Web* dan teknologi *Internet* bergantung kepada keamanan sistem *Web* tersebut. Arsitektur sistem *Web* terdiri dari dua sisi: *server* dan *client*. Keduanya dihubungkan dengan jaringan komputer (*computer network*). Selain menyajikan data-data dalam bentuk statis, sistem *Web* dapat menyajikan data dalam bentuk dinamis dengan menjalankan program. Program ini dapat dijalankan di server (misal dengan CGI, servlet) dan di *client* (applet, Javascript). Sistem *server* dan *client* memiliki permasalahan yang berbeda.

Keamanan *server WWW* biasanya merupakan masalah dari seorang administrator. Dengan memasang *server WWW* di sistem, maka membuka akses (meskipun secara terbatas) kepada orang luar. Apabila server terhubung ke *Internet* dan memang *server WWW* disiapkan untuk publik, maka harus lebih berhati-hati sebab membuka pintu akses ke seluruh dunia.

Adanya lubang keamanan di sistem *WWW* dapat dieksploitasi dalam bentuk yang beragam, antara lain : (Budi, 2002)

a. Informasi yang ditampilkan di *server* diubah sehingga dapat mempermalukan perusahaan atau organisasi (dikenal dengan istilah *deface*1).
b. Informasi yang semestinya dikonsumsi untuk kalangan terbatas (misalnya laporan keuangan, strategi perusahaan, atau *database client*) ternyata berhasil disadap oleh saingan (ini mungkin disebabkan salah *setup server*, salah *setup router / firewall*, atau salah *setup authentication*).
c. Informasi dapat disadap (seperti misalnya pengiriman nomor kartu kredit untuk membeli melalui WWW, atau orang yang memonitor kemana saja melakukan *web surfing*).
d. *Server* diserang (misalnya dengan memberikan *request* secara bertubi-tubi) sehingga tidak bisa memberikan layanan ketika dibutuhkan (*denial of service attack*).
e. Untuk server *web* yang berada di belakang *firewall*, lubang keamanan di *server web* yang dieksploitasi dapat melemahkan atau bahkan menghilangkan fungsi dari *firewall* (dengan mekanisme *tunneling*).

Berikut ini menurut Lauri Auronen (2002) beberapa celah keamanan yang biasa menyerang sistem keamanan di *WWW* yang dapat dikelompok sebagai berikut :

a. ***Detecting backend systems*****.** Penggunaan kode template, komentar dalam kode *HyperText Markup Language* (HTML) atau bahkan bentuk *Uniform Resource Locator* (URL) yang digunakan dalam aplikasi *web* menyediakan informasi tentang sistem *backend* atau lingkungan pengembangan aplikasi *web*. Terutama penggunaan kode *template* dapat membuktikan berbahaya (NALNEESH, 2000). Kode ini dapat tersedia secara luas dan dapat mengandung *bug* yang mudah dianalisa dari *source code*. Informasi ini dapat digunakan untuk mengeksploitasi kerentanan khusus atau mempersempit fokus pada pembacaan kerentanan yang dilakukan pada sistem.

b. ***Session hijacking*****.** HTTP adalah *stateless*. Aplikasi *Web* adalah sering perlu untuk tindakan pengguna mengikat ke *stateful* sesi tunggal. HTTP dibuat *stateful* dengan membuat objek sesi pada sisi server dan menyimpan *identifier* obyek ini, sesi disebut *identifier*, dalam sebuah cookie di *browser* klien atau sebagai parameter berlalu dalam URL setiap permintaan klien. Mengubah sesi pengenal di *cookie* atau URL pada sisi *client* agar sesuai dengan identifier sesi pengguna lain dapat digunakan untuk membajak sesi ini (NALNEESH, 2000). Pengidentifikasi sesi ini dapat diketahui dengan mendengarkan lalu lintas jaringan atau dengan menebak.

c. ***Cookie poisoning.*** *Cookie* dalam beberapa aplikasi berisi informasi bisnis aktual tertentu, seperti item disimpan dalam *shopping cart* dan daftar harga

mereka. Informasi ini dapat dengan mudah berubah dan jika tidak ada mekanisme otentikasi yang berada di tempat, untuk memeriksa validitas *cookie* di sisi *server*, ini mengarah kepada kompromi dari aplikasi. (NALNEESH, 2000).

d. ***Form manipulation.*** *Form* HTML dapat disimpan ke dalam disk pada sisi *client* dan dapat di *edit*. Hal ini menjadi masalah jika ada *field* tersembunyi dalam bentuk yang berisi data yang dianggap tidak berubah, seperti harga *item*. Sekali lagi, jika mekanisme otentikasi yang cukup yang tidak pada tempatnya, ini menyebabkan serangan mungkin terjadi (NALNEESH, 2000). Bidang Formulir juga dapat memiliki kendala seperti panjang maksimum. Perubahan ini dapat menyebabkan *buffer overflows* dalam aplikasi *web*.

e. ***URL parameter tampering***. Aplikasi *Web* sering mengambil parameter sebagai bagian dari URL yang dikirim oleh *browser*. Serangan terhadap parameter URL adalah serangan paling mudah perusakannya, karena setiap pengguna dapat mengklik pada *address bar browser* mereka dan di ketik ke dalam parameter baru.

f. ***HTTP header modification.*** *Header* HTTP digunakan untuk melewati beberapa variable antara agen pengguna dan *server web*. Variabel ini termasuk *cookie* yang telah diatur oleh situs, URL pengarah dan bahasa dari agen pengguna. Seperti formulir isian data, variabel-variabel ini juga dapat secara bebas dimodifikasi oleh pengguna dengan alat yang cocok. Ini dapat digunakan untuk, misalnya, *cross-site scripting* dan *SQL injection query.*

g. ***Bypassing intermediate forms in a multipleform set.*** Karena HTTP adalah *stateless*, hal itu bisa mungkin bisa juga tidak untuk menjamin bahwa halaman yang diakses dalam urutan yang telah ditentukan. Pengguna bisa menebak dan ketik alamat halaman lain. Hal ini menimbulkan masalah dalam bentuk *multiple-set* jika bentuk yang kemudian bergantung pada masukan yang diberikan dalam bentuk sebelumnya. Hal ini dapat menyebabkan masalah pada fungsionalitas dari aplikasi *web* itu sendiri atau kompromi dari *platform* yang mendasari, biasanya melalui *buffer overflow*. (NALNEESH, 2000)

h. ***SQL Injection Queries.*** Masukan yang diberikan oleh user dalam bentuk yang dikirim untuk diproses ke aplikasi *backend*. Jika bidang ini tidak cukup di validasi, mereka mungkin berisi karakter dengan arti khusus dalam SQL. Jika masukan tersebut kemudian digabungkan sebagai bagian dari query SQL karakter khusus dapat digunakan untuk membangun sebuah *query* SQL yang sudah di modifikasi, maka dapat mengarah ke modifikasi informasi yang bocor dan berbahaya dari *database*. (NALNEESH, 2000)

i. ***Cross-Site Scripting.*** Ketika pengguna dapat mengirim informasi ke aplikasi *web* melalui beberapa mekanisme dan informasi yang kemudian ditampilkan untuk pengguna lain, adalah kemungkinan untuk dimasukan *script* HTML yang berbahaya. Kode ini, ketika ditampilkan ke pengguna lain, akan tampak berasal dari aplikasi *web* itu sendiri. Kode kemudian dapat digunakan untuk berbagai serangan informasi kompromi yang rahasia. Daftar serangan menggunakan *cross-site scripting* (XSS) dapat ditemukan di CERT *advisory* . (Mike dkk, 2009).

j. ***3rd Party Software Misconfiguration*****.** Perangkat lunak terkonfigurasi menyebabkan berbagai masalah. Biasanya *misconfigurations* ini memungkinkan beberapa serangan aplikasi *web* lainnya. Gaur, dalam artikelnya, memberi contoh yang *misconfiguration* dapat menyebabkan daftar direktori yang tidak sah.

k. ***Forceful Browsing*****.** Dapat berarti membuat beberapa permintaan ke *server web* dengan pola URL komponen aplikasi *web* khas seperti program CGI.

2.2 *Web Scurity*

Di masa lalu, tujuan utama dari komputer adalah untuk menyimpan informasi yang diperlukan oleh organisasi untuk kegiatan harian dari organisasi itu sendiri. Komputer hanya digunakan sebagai alat pusat pengolahan data saja, dan terbatas hanya untuk penggunaan di internal organisasi saja. Oleh karena itu, ancaman keamanan komputer biasa dan pada dasarnya berhubungan dengan staf di organisasi (misalnya: penyalahgunaan *account*, pencurian, atau data manipulasi oleh para pengguna). Menangani ancaman ini mudah sekali bagi organisasi, dikarenakan hampir tidak ada kemungkinan ancaman dari eksternal. Ancaman ini umumnya ditangani dengan dengan menjaga komputer dengan informasi penting di sebuah ruangan khusus dan terisolasi, serta secara manual di verifikasi bahwa data pada komputer belum dirusak.

Namun, penggunaan komputer sejak dari awal hingga saat ini telah berubah secara radikal. Sekarang organisasi dalam menggunakan komputeruntuk menyimpan data yang dapat diakses dari lokasi mana saja di dunia ini. Selain itu, komputer tidak lagi digunakan hanya dalam organisasi. Komputer banyak

digunakan oleh individu dan pemakai rumah tangga untuk berkomunikasi lebih cepat di seluruh dunia. Karena penggunaan yang luas seperti itu, ancaman keamanan komputer secara alami telah meningkat. Banyak ancaman keamanan terjadi dalam bentuk pencurian *virtual* di *Internet*. Dalam hitungan detik, seorang pencuri *virtual* dapat mengakses sistem dan mencuri informasi penting, seperti *password* dan nomor kartu kredit. Kerusakan juga bisa dilakukan dengan infiltrasi sistem dan informasi tentang itu dengan melewatkan *virus* dan *worm*.

Saat ini, *Internet* telah menjadi media di mana orang dapat terhubung. Ini adalah *platform* di mana jutaan komputer, seluruh saham, dunia dan akses informasi. Transaksi bisnis *e-commerce* seperti pasar *online*, kini menjadi kenyataan. Namun dengan evolusi dari *Web* dan peningkatan penggunaan dalam setiap aspek kehidupan, kebutuhan akan keamanan *web* sudah menjadi keharusan. Ada kekhawatiran beberapa kunci yang terkait dengan keamanan *Web*: Seberapa aman sistem yang mengontrol pertukaran informasi di *Web*? Seberapa aman informasi yang disimpan pada banyak komputer di seluruh *Web*? Ini adalah fakta diketahui bahwa apa yang dapat digunakan juga dapat disalahgunakan. Sebagai contoh, *e-commerce* telah membuat hidup kita lebih mudah, namun ada beberapa risiko yang melekat. Mengikuti alur pemikiran ini, kita perlu rencana keamanan *Web* dalam suatu organisasi baik di tingkat sistem dan data. Keamanan di tingkat sistem memastikan bahwa sistem tidak *hacked* sejauh itu jatuh. Keamanan di tingkat data menjamin bahwa informasi pada sistem tidak dirusak. Harus selalu ingat bahwa jika informasi organisasi adalah *hacked* baik melalui jaringan atau melalui cara lain, bisa dikenakan biaya berat untuk perusahaan. Sebuah kegagalan

dalam keamanan jaringan juga bisa biaya organisasi dalam hal *goodwill* dan reputasinya. Tidak ada organisasi lain akan tertarik dalam melakukan bisnis dengan organisasi yang tidak bisa melindungi informasi dan sistem keamanan. Sebuah pelanggaran keamanan dapat didefinisikan sebagai akses ilegal terhadap informasi yang dapat mengakibatkan pengungkapan, penghapusan, atau perubahan informasi. Dengan kata lain, suatu pelanggaran keamanan terjadi ketika informasi atau sistem yang digunakan atau diakses untuk tujuan ilegal. Sebuah pelanggaran keamanan *web* dapat berlangsung dalam beberapa bentuk, seperti pelanggaran di jaringan organisasi, *hacking* ke dalam sistem atau jaringan, perubahan informasi organisasi atau individu, serangan virus, gangguan atau penolakan layanan, perusakan, dan pencurian.

Berikut ini adalah beberapa jenis umum pelanggaran keamanan : (Shweta Bhasin, 2003)

a. ***Accessing subscriber details to send spam email***

Untuk mempromosikan penawaran produk baru, perusahaan kartu kredit biasanya mengakses informasi pelanggan dari *database* penyedia layanan *e-mail*. Hal ini tentu saja tanpa sepengetahuan penyedia layanan *e-mail*.

b. ***Unauthorized access of confidential data to create fraudulent identities***

Seseorang mengakses rincian tambahan, seperti alamat tempat tinggal, nomor kontak, nomor jaminan sosial, dan detail nomor rekening dari *database* bank untuk menciptakan identitas palsu.

c. ***Eavesdropping***

Sebuah badan intelijen suatu negara terhubung ke jaringan negara lain untuk mengakses informasi pertahanan yang sensitif dan rahasia.

d. ***Promoting your organization on somebody else's Web site***

Sebuah organisasi membuat sebuah *server* untuk *host* situs di *Web*. Organisasi lain yang serupa mengakses *web server* ini dengan cara ilegal dan *host* beberapa halaman *web* di situs untuk mempromosikan organisasinya.

e. ***Using an automated script to try to log in to a computer system***

Seorang *hacker* menggunakan *script* otomatis untuk membuat berbagai upaya untuk *login* ke sistem komputer. Akibatnya, pengguna jasa yang berwenang *logon* ditolak oleh komputer karena komputer sedang sibuk karena menolak permintaan dari *hacker*.

f. ***Gaining unauthorized access to a mail server***

Akses yang tidak sah oleh seseorang untuk keuntungan *pribadi* ke *mail server* organisasi untuk mengirim dan menerima pesan *e-mail*.

g. ***Gaining unauthorized access to the network to gain information***

Seseorang menyusup ke jaringan bank atau perusahaan keuangan untuk mentransfer sejumlah besar uang ke rekening fiktif.

h. ***Virus attacks***

Virus biasanya menyebar melalui pesan *e-mail*. Sebuah serangan virus juga dapat terjadi dalam jaringan organisasi, dan melalui jaringan yang menyebar melalui *Internet*.

i. *DNS hijacking*

Domain Name System (DNS) adalah *database* yang peta nama domain ke alamat IP. Komputer yang terhubung ke *internet* menggunakan DNS untuk menyelesaikan URL ke alamat IP dari situs yang perlu diakses. Dalam pembajakan DNS, para *hacker* mendapatkan akses ke layanan DNS dan membuat perubahan dalam informasi yang memetakan nama domain ke alamat IP. Karena ini, pengguna akan diarahkan ke situs yang berbeda dari yang mereka inginkan untuk mengakses.

j. *DoS attacks*

Denial-of-service (DoS) serangan adalah serangan berbasis jaringan di mana pengguna resmi ditolak penggunaan layanan jaringan. Serangan DoS terjadi karena berbagai alasan, seperti penggunaan sumber daya yang tidak sah. Sebuah contoh umum serangan DoS adalah pengguna yang tidak sah menggunakan lokasi FTP untuk meng-*upload* volume data yang besar. Hal ini menyebabkan penyumbatan yang tidak perlu di ruangan penyimpanan dan menghasilkan lalu lintas jaringan yang ramai.

k. *DDoS attacks*

Distributed denial-of-service (DDoS) serangan adalah bentuk canggih dari serangan DoS. Dalam serangan DDoS, sistem target diserang dari beberapa komputer di *Internet*. Tanpa pengetahuan pemilik, *hacker* menciptakan suatu aplikasi dan tempat aplikasi di beberapa lokasi di *Internet*. Aplikasi ini tidak terdeteksi, karena mereka tidak membahayakan sistem di mana mereka

berada. Ketika serangan diluncurkan, sistem target terganggu dari semua komputer yang berbeda yang memiliki aplikasi yang diinstal oleh *hacker*.

2.3 *Web Vulnerability Scanner*

2.3.1 *Web*

Web adalah alamat atau lokasi di dalam *internet* suatu halaman *web*, umumnya membuat dokumen *HTML* dan dapat berisi sejumlah foto atau gambar grafis, musik, teks bahkan gambar yang bergerak. Dengan menggunakan teknologi tersebut, informasi dapat diakses selama 24 jam dalam satu hari dan dikelola oleh mesin. (Pardosi, 2002:2).

2.3.2 *Database*

Database adalah representasi kumpulan fakta yang saling berhubungan disimpulkan secara bersama sedemikan rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikan rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Dalam maksud yang sama, bisa juga diartikan sebagai sekumpulan informasi yang disusun sedemikan rupa untuk dapat diakses oleh sebuah *software* tertentu. Basis data tersusun atas bagian yang disebut *field* dan *record* yang tersimpan dalam sebuah *file*. (Febrian, 2007:133).

2.3.3 *Vulnerability Scanner*

Vulnerability scanner adalah sebuah program komputer yang di desain untuk mencari dan memetakan sistem untuk kelemahan pada aplikasi, komputer atau jaringan. Meningkatnya penggunaan internet membuat semakin banyaknya *website* yang bermunculan khususnya di Indonesia, dengan adanya *website* membuat pengembang *website* dapat menyampaikan informasinya kepada pengguna *internet* dengan mudah, di Indonesia *website* – *website* pemerintahan pun telah banyak bermunculan, namun sangat disayangkan kejahatan internet di indonesia terus meningkat seiring bermunculannya ragam artikel yang membahas masalah hacking. (Perdana, 2010:4).

Jenis dari *tool vulnerability scanner* yaitu :

a. *Acunetix* (*www.acunetix.com*)

Acunetix website application scanner merupakan perangkat lunak yang dikembangkan untuk melakukan *scanning*. Kelebihan dari *tools* ini adalah kemampuannya untuk memberikan solusi dari kelemahan yang ditemukan dan mengelola *traceability* dari setiap *vulnerabilities* tersebut. Selain itu, *acunetix* menyediakan fungsi-fungsi tambahan yang dapat digunakan untuk melakukan pengujian lebih lanjut terhadap *website* yang diuji.

b. *w3af* (*www. w3af.org*)

w3af merupakan *tools open source* berbasis *python* yang tersedia secara gratis untuk semua *platform*. Fungsi yang ditawarkan tidak jauh berbeda dengan acunetix berkaitan dengan scanning vulnerabilities pada website. Perbedaannya *w3af* menampikan hasil *scan* yang lebih teknis dibandingkan *acunetix* dan tidak

memberikan solusi secara langsung untuk setiap *vulnerabilities* yang ditemukannya.

c. *Wireshark* (*www.wireshark.org*)

Wireshark adalah *packet analyzer gratis* dan *open-source*. *Tools* ini seringkali digunakan untuk menemukan masalah pada jaringan, pengembangan perangkat lunak dan protokol komunikasi, dan pendidikan. *Wireshark* bersifat *cross-platform* dan menggunakan *pcap* untuk meng-*capture* paket jaringan. Wireshark dapat berjalan pada hampir semua sistem operasi yang tersedia.

d. *OWASP ZAP* (*www.owasp*)

Zed Attack Proxy (ZAP) buatan *The Open Web Security Project (OWASP)* merupakan *tools* penetration testing yang digunakan untuk menemukan vulnerabilities pada aplikasi *web. ZAP* menyediakan pemindaian otomatis dan seperangkat tools untuk menemukan vulnerabilities secara manual.

e. *Subgraph Vega* (*www.subgraph.com*)

Vega adalah aplikasi *open-source* yang dapat digunakan untuk menguji keamanan aplikasi web. Vega dapat membantu menemukan dan memvalidasi *SQL Injection, Cross-Site Scripting (XSS),* dan vulnerabilities lainnya.

2.4 *Acunetix Web Vulnerability Scanner*

Acunetix web vulnerability scanner adalah sebuah *software* yang berfungsi untuk melakukan scanning atas kelemahan yang bisa terjadi di suatu situs, *software* ini mampu memeriksa kelemahan *web* server maupun aplikasi *web*-nya dengan cepat, selain itu *software* ini juga memberikan saran yang harus dilakukan apabila ditemukan kelemahan pada *website* tersebut, untuk pengguna sistem

operasi windows perl interpreternya adalah software activeperl yang dapat didownload di http://www.activestate.com. (Perdana, 2010:5).

Menurut Sutanta (2008:11), Metode kualitatif dengan menggunakan beberapa *tools* berupa perangkat lunak dan cara-cara tertentu yang lazim digunakan untuk menguji keamanan aplikasi. Tahap-tahap yang dilakukan adalah sebagai berikut:

1. Tahap Inisiasi, pada tahap ini dilakukan penelusuran dan pengkajian *literatur-literatur* yang berhubungan dengan keamanan aplikasi.
2. Tahap Investigasi, pada tahap ini dilakukan penyelidikan terhadap *web server*, program aplikasi yang digunakan.
3. Tahap Pengujian, pada tahap ini dilakukan pengujian terhadap keamanan aplikasi dengan menggunakan *tools*, yaitu *Acunetix web vurnerability scanner* dengan metode yang lazim digunakan dalam pengujian keamanan aplikasi dan sistem.
4. Tahap Verifikasi, pada tahap ini dilakukan verifikasi terhadap keamanan aplikasi untuk pemberitahuan kepada admin untuk dilakukan perbaikan-perbaikan atas dasar hasil investigasi dan pengujian pada aspek pemrograman.

2.5 Penelitian Sebelumnya

Perdana, 2010. Judul penelitian "Analisis Perbandingan Aplikasi Web Vulnerability Scanner Antara Nikto dan Acunetix". Pada saat sekarang ini komputer – komputer di dunia dapat saling terhubung, semuanya terhubung ke dalam sebuah jaringan yang disebut dengan *internet*, kebutuhan akan informasi membuat jaringan *internet* di dunia berkembang dengan pesat, dapat dilihat

dengan banyaknya bermunculan *website* baik dalam negeri maupun luar negeri yang dapat memberikan kemudahan bagi masyarakat untuk mendapatkan informasi, pada saat ini kegiatan – kegiatan yang biasa dilakukan manusia seperti perdagangan atau jual beli dapat dilakukan di rumah dengan menggunakan komputer yang terhubung dengan *internet*.

Santoso, 2009. Judul penelitian "Analisis Vulnerability Aplikasi iFace IT Telkom Bandung". Dewasa ini kemajuan teknologi telah berkembang dengan baik. Pertukaran informasi dapat dilakukan dengan mudah tanpa harus mempermasalahkan lagi jarak antara pengirim dengan penerima informasi. Jejaring sosial merupakan salah satu media yang dapat menjawab hal tersebut. Dalam kasus kali ini jejaring sosial yang digunakan adalah jejaring sosial yang daerah cakupannya adalah internal kampus yaitu Aplikasi iFace di Institut Teknologi Telkom (IT Telkom) Bandung. Aplikasi ini berbasis website dan memiliki fungsionalitas hampir sama dengan *facebook*. Namun untuk iFace lebih dititik beratkan tentang informasi yang berhubungan dengan kampus IT Telkom.