BAB III

DESAIN PENELITIAN/METODOLOGI

3.1. Metodologi Pengumpulan Data

Proses mengumpulkan data yang diperlukan untuk penelitian ini, Penulis akan menggunakan metode B*lack Box Testing* dalam melakukan uji coba terhadap semua target yang telah didefinisikan pada batasan masalah. Dengan menggunakan metode ini, maka Penulis juga hanyan melakukan penelitian terhadap target yang telah ditentukan dalam batasan masalah. Oleh karena itu, Penulis akan memfokuskan pada pemeriksaan kelemahan dalam sistem :

- *Web Server*
- *Web Apps*
- *Database Server*

Data mengenai target yang akan diuji coba dapat Penulis peroleh dengan menggunakan metode observasi. Penulis dapat melakukan inventarisasi semua target dengan melihat dan berinteraksi secara langsung dengan mesin target, atau melakukan aktifitas pengumpulan informasi dengan teknik *scanning* menggunakan *tools Penetration Testing*.

Tabel III.1 Output hasil dari Penetration Testing

Komponen Web	Output
Web Server	- HTTP Banner Disclosure - Trace Method - Cross Site Scripting - HTTP Directory Listing
Web Apps	- System Path Disclosure - Web Apps Version
Database Server	- Database Version - Username & Password - Sql Injection Attack

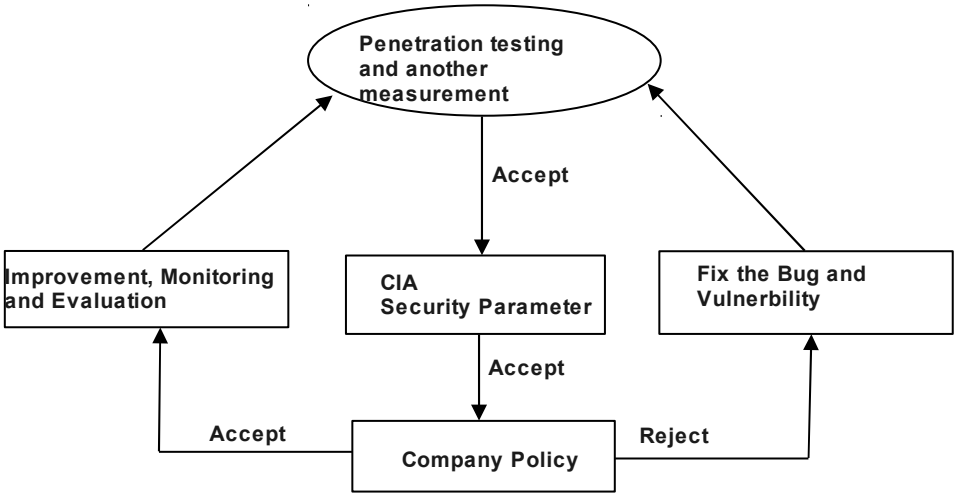
3.2. Metode Penelitian dan Alat Ukur yang digunakan

Penulis menggunakan metode Kuantitatif sebagai metode penelitian yang digunakan. Dasar penggunaan metode ini adalah karena hasil penetration testing yang berupa jenis insiden, akan di jumlahkan untuk kemudian dibandingkan dengan parameter Sistem Informasi. Jumlah insiden tersebutlah yang menjadi dasar penentuan metode ini. Dengan metode ini, Penulis akan membandingkan hasil yang diperoleh menggunakan alat ukur yang dipilih dengan parameter keamanan yang menjadi ukuran pada keamanan web portal Pemerintah.

Alat ukur yang digunakan adalah penetration testing yang menggunakan cara yang dilakukan *Cracker* maupun hacker. Oleh karena itu perangkat yang digunakan serta cara yang dilakukan adalah mengadopsi dari cara dan perangkat yang digunakan oleh para Cracker maupun hacker. Penulis menggunakan metode CIA (*Confidentiality, Integrity* dan *Availability*) sebagai parameter keamanan Sistem Infomasi (SI). Alasan digunakannya metode ini adalah :

- Konsep ini digunakan di hampir semua literatur terkait Sistem Informasi dan Komputer
- Konsep ini juga menjadi acuan untuk standarisasi International,yaitu ISO17799 yang merupakan " *Code of Practice for Information Security Management*
- Konsep ini terbilang sederhana, sehingga dengan segala keterbatasan yang ada pada Penulis maka konsep ini sangat tepat untuk digunakan.

Metode ini juga sangat tepat untuk dijadikan parameter penelitian yang bersifat kuantitatif, karena hasil yang diperoleh merupakan hasil baku yang dapat dihitung.



Gambar III.1 Daur Hidup Keamanan Sistem Informasi dalam Sebuah Organisasi

Untuk membandingkan kondisi keamanan aktual dari sebuah SI, dengan parameter keamanan yang telah ditentukan, maka kita harus menentukan alat ukur yang digunakan untuk membandingkan kondisi tersebut. Penulis menggunakan metode *Penetration Testing* sebagai alat ukur ini, hal ini didasari oleh beberapa pertimbangan, antara lain :

- *Penetration Testing* merupakan alat ukur yang relatif murah, terlebij lagi tersedianya tools berbasis open source dan dapat juga secara gratis
- *Penetration Testing* memerlukan waktu yang relatif lebih cepat, sehingga waktu keseluruhan proses menjadi lebih sedikit
- *Penetration Testing* mencakup aspek khusus yaitu sisi teknologi dari sebuah Sistem Informasi, sehingga proses penelitian bisa fokus dengan keterbatasan yang ada pada Penulis.

3.3. Prosedur dari Penetration Testing pada Web Aplikasi

Selain menggunakan cara-cara yang biasa dilakukan oleh *Cracker,* Penulis juga menggunakan prosedur yang disarankan oleh ISACA (*Information System Audit and Control Association*), yaitu sebuah asosiasi bagi auditor Sistem Informasi. Ada beberapa bagian yang disarankan oleh ISACA yang dapat melengkapi tahapan *Penetration Testing,* oleh karena itu Penulis juga mengadopsi prosedur dari ISACA ini. Karena sifat penelitian yang hanya memfokuskan pada

keamanan internal SI dari Organisasi serta secara global pada web portal Pemerintah yang akan diteliti sebanyak kurang lebih 65 web site portal pemerintah pusat. Perlu di ketahui Penulis dalam melaksanakan Penetration Testing terhadap web site portal yang akan diteliti, Penulis menggunakan sebuah tools *Penetration Testing* untuk memeriksa kelemahan web portal tersebut melalui tahapan analisa kelemahan sistem, terdiri dari :

- Scanning pada web server untuk melakukan komunikasi, setelah itu melakukan *banner grabbing* atau mencari informasi dari banner yang diperlihatkan oleh service yang berjalan
- Pemeriksaan terhadap kelemahan individual pada software yang sudah didapat sebelumnya. Daftar dari kelemahan ini bisa menggunakan daftar yang di keluarkan oleh OWASP Top Ten

Setelah proses diatas selesai, proses selannjutnya adalah ekploitasi dengan beberapa tahapan yaitu :

- Memeriksa tingkat serangan yang bisa diterima oleh organisasi, pada beberapa tipe serangan yang sifatnya merusak (*SQL Injection, Local and Remote File Include, Cross-site Scripting, Cross-site Request Forgery, Information Disclosure Problems, Session Security Problems,* dan lebih banyak lagi termasuk OWASP TOP 10)
- Pada beberapa kasus *Penetration Testing* semakin tinggi level akses yang didapatkan, maka semakin tinggi pula nilai proyek yang dijalankan
- Melakukan serangan sesuai dengan jenis kelemahan yang telah didapat, dan menggunakan *exploit* yang sesuai

3.4. Proses Penetration Testing

Penetration Testing yang akan dilakukan oleh Penulis akam mengikuti cara yang biasa dilakukan oleh *Cracker* dalam menembus keamanan sebuah sistem web portal. Cara – caranya ada sebagai berikut :

- ***Information Gathering*** **:** Dalam proses ini penulis akan berusaha untuk melakukan pengumpulan informasi dari web portal target. Pengumpulan

(*information gathering*) tersebut dilakukan sebuah *crawl url* direktori dari *web apps* tersebut yang terdiri dari *web page* target itu sendiri

- ***Attack and Penetration Testing* :** Selain menjalankan eksploitasi dikemas untuk aplikasi web, dampaknya adalah dapat menghasilkan kustom serangan *Cross Site Scripting (XSS), SQL Injection dan Remote File* Inclusion. Serangan yang dilancarkan dengan menganalisis halaman pertama yang diidentifikasi selama pengumpulan informasi yang mungkin rentan terhadap keamanan tersebut. Sejalan dalam proses testing tersebut menciptakan eksploitasi untuk membuktikan apakah kerentanan menimbulkan ancaman aktual. Teknik ini aman untuk produksi server, karena mereka tidak berusaha untuk merusak aplikasi web.
- ***Clean up and Reporting* :** Pada tahap akhir ini proses *penetration testing* tersebut aman, karena tidak menginstal atau menjalankan kode apapun pada web server selama pengujian. Pada akhirnya mendapatkan data tentang kelemahan dieksploitas untuk ditindaklanjuti terkait pada sistem dan data untuk opsi perbaikan.

Tabel III.2. Tools serta Kegunaan dalam *Information Gathering, Attack dan Penetration Testing*

Nama Tools	Penjelasan
Web Securify	Websecurify adalah keamanan web terintegrasi pengujian lingkungan web, yang dapat digunakan untuk mengidentifikasi kerentanan web dengan menggunakan browser otomatisasi canggih, penemuan dan *fuzzing* teknologi. Platform ini dirancang untuk melakukan tes manual otomatis kerentanan serta dan terus ditingkatkan dan *fine-tuned* oleh tim kelas dunia web penguji keamanan aplikasi penetrasi dan mendapat *feed back* yang baik dari komunitas *open sourc*e.
	Web Vulnerability Scanner, sebuah alat uji

Web Cruiser	penetrasi web yang efektif dan ampuh yang akan membantu Anda dalam audit website. Memiliki Scanner Kerentanan dan serangkaian alat keamanan. Hal ini dapat mendukung situs pemindaian serta POC (Bukti konsep) untuk kerentanan web: SQL Injection, Cross Site Scripting, XPath Injection dll Jadi, WebCruiser juga merupakan alat injeksi SQL otomatis, alat injeksi XPath, dan Cross Site Scripting.

3.4.1. Alur Kerja Pada Aplikasi Penetration Testing

Secara global alur kerja dari aplikasi tersebut ada sebagai berikut :

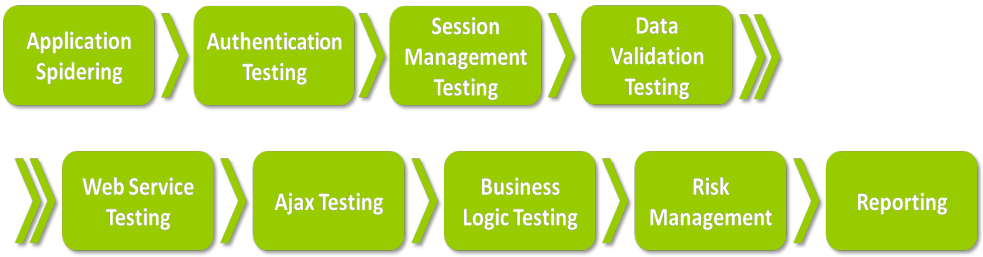
Gambar III.2. Alur Kerja Aplikasi Penetration Testing [IVI11]

Penjelasan :

- ***Log on to online portal***, pada tahap awal kita masuk dalam aplikasi dan mengetik alamat situs target yang akan kita jadikan *penetration testing* tersebut
- ***Schedule test from online portal***, tahap kedua ini kita menunggu hingga ada komunikasi antara *user penetration testing* dengan situs target tersebut
- ***Test conducted automatically*****,** serangan dilancarkan ke dalam situs target seperti *SQL Injection, Local and Remote File Include, Cross-site Scripting, Cross-site Request Forgery, Information Disclosure Problems, Session Security Problems,* dan lebih banyak lagi termasuk OWASP TOP 10 .

- ***View reports online*****,** tahap akhir adalah mendapatkan laporan hasil eksploit yang sudah kita lakukan tadi

Berikut metodologi secara detail pada aplikasi *Penetration Testing web* :



Gambar III.3. Metodologi Aplikasi Penetration Tester [IVI11]

3.5. Analisa Hasil Penetration Testing

Dalam menetukan apakah sebuah sistem informasi yang diuji dapat dikatakan aman, maka penulis menggunakan metode kuantitatif untuk membandingkan hasil dari penetration testing, terhadap parameter keamanan yang telah di tentukan. *Penetration Testing* yang dijalankan diatas memiliki efek terhadap satu atau lebih dari parameter yang ada. Oleh karena itu, maka metode kuantitatif sangat tepat untuk digunakan dalam penelitian ini. Proses membandingkan hasil *Penetration Testing* dengan parameter klasifikasi serangan pada web portal, dengan menetapkan aspek dari parameter yang terkait dengan hasil yang didapatkan pada proses *Penetration Testing* itu.

Tabel III.3. Tabel Klasifikasi Serangan pada Web Portal sebagai Acuan dalam Analisa Terhadap Keamanan Web

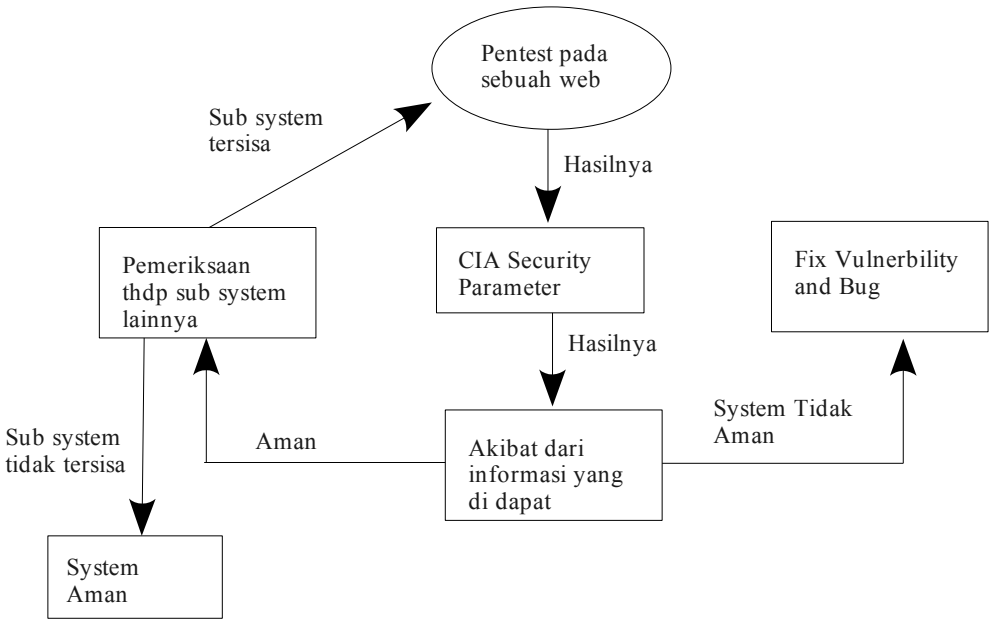
Tipe Serangan	Aspek Keamanan SI	Keterangan
Footprinting (cetak kaki/jejak)	Confidentiality	Menentukan topologi jaringan dan info terkait
Scanning (periksa keseluruhan)	Confidentiality	Memeriksa port yang terbuka, mengetahui service yang berjalan
Enumeration	Confidentiality	Mencari informasi

(daftar informasi lebih spesidik)		spesifik tentang target kelemahannya
Eavesdroping (mendengarkan paket)	Confidentiality	Sniffing merekam dan menaganalisa lalu lintas jaringan
Password Attack (serangan password)	Confidentiality Integrity	Proses untuk mendapatkan password
Software and Protocol Exploitation (perangkat dan protokol eksploitasi)	Confidentiality Integrity	Eksploitasi kelemahan yang ada pada software
Man-in the -middle attack (serangan di tengah user)	Confidentiality Integrity	Penyerang mendapat akses ke dalam media, melakukan modifikasi dan mengirimkan paket kembali ke network
DOS Attack (serangan pengiriman paket)	Availability	Serangan terakhhir yang meyebabkan sistem tidak berfungsi

Menggunakan tabel diatas, Penulis dapat menentukan tingkat keamanan sebuah Sistem Informasi. Keamanan harus selalu di evaluasi dan monitoring secara berkala agar selalu berada dalam kondisi 'aman' menurut kebijakan keamanan sistem informasi organisasi.

3.5.1. Penetapan Ukuran Keamanan Sistem Informasi

Tetap pada acuan untuk klasifikasi yang dibuat oleh Krutz [KRU01], dimana sebuah Sistem Informasi dikatakan aman jika informasi dapat di percaya integritasnya, dapat diakses oleh pihak yang memiliki hak dan kapanpun dibutuhkan.



Gambar III.4 . Alur Penetapan Ukuran Keamanan Sistem Informasi

Untuk menentukan apakah sebuah sistem dikatakan aman atau tidak, maka kita harus melakukan pemeriksaan terhadap komponen-komponen dari sistem tersebut, atau bisa disebut *Subsystem*. Dalam menentukan tingkat keamanan dari sistem informasi target yan menjadi obyek penelitian, maka penulis akan memberikan penilaian terhadap semua *SubSystem*. *SubSystem* yang akan di periksa keamanannya telah di tetapkan dalam batasan masalah. Dalam penelitian ini, target dikatakan aman jika :

- Informasi yang bersifat *Internal Use Only* tidak dapat terungkap oleh pihak yang tidak berhak
- Informasi yang bersifat *Company Confidential* tidak dapat terungkap oleh pihak yang tidak berhak

3.5.2. Analisa Terhadap Kelayakan Penetration Testing

Menentukan apakah *Penetration Testing* layak untuk dijadikan sebagai alat ukur keamanan sistem informasi, maka diperlukan sebuah metode untuk melakukan pengujian. Penulis menggunakan metode kuantitatif dalam pengujian ini, dimana hasil dari *Penetration Testing* akan dibandingkan dengan parameter keamanam informasi. Penulis merangkum semua jenis insiden terkait dengan

parameter keamanan yang ada, lalu melakukan pengelompokan terhadap insiden tersebut. Pengelompokan insiden itu terbagi menjadi 3 yaitu :

- Insiden yang dilakukan oleh *Cracker*
- Insiden yang ditimbulkan oleh *Cracker*, namun tidak secara langsung
- Insiden yang di timbulkan *Cracker* secara langsung

Dengan mempelajari jenis insiden terhadap kemanan SI dari berbagai literatur, seperti [COL01] [SCR00] dan [STO02], serta melihat efeknya dengan parameter keamanan SI (CIA) maka Penulis mendapatkan sebuah tabel yang bisa digunakan sebagai acuan penenetuan kelayakan dan efektifitas *Penetration Testing* sebagai alat ukur maupun metode untuk mengevaluasi dan menganalisis sebuah keamanan suatu web portal..

Tabel III.4. Tabel Acuan dalam Analisa Terhadap Keamanan Web

Parameter CIA	**Jenis Insiden**	**Keterangan**
Authentication (otentikasi)	Brute Force Insufficient Authentication Weak Password Recovery Validation	Insiden yang disebabkan serangan Cracker secara langsung
Authorization (otorisasi)	Credential/Session Prediction Insufficient Authorization Insufficient Session Expiration Session Fixation	Insiden yang disebabkan serangan Cracker secara tidak langsung
Command Execution Confidentiality (eksekusi perintah)	Buffer Overflow Format String Attack LDAP Injection OS Commanding SQL Injection SSI Injection XPath Injection	Insiden yang disebabkan serangan Cracker secara langsung
Client-Side Attacks Integrity (serangan sisi client)	Content Spoofing Cross-site Scripting	Insiden yang disebabkan serangan Cracker secara langsung
Information Disclosure (keterbukaan informasi)	Directory Indexing Information Leakage Path Traversal Predictable Resource Location	Insiden yang disebabkan serangan Cracker secara langsung
Logical Attacks	Abuse of Functionality	Insiden yang

Availability (serangan logis ketersediaan)	Denial of Service Insufficient Anti-Automation Insufficient Process Validation	disebabkan serangan Cracker secara langsung

Tabel diatas adalah sebagai acuan analisa keamanan web aplikasi, sebafau acuan dalam merumuskan sebuah model ancamana keamanan web aplikasi maupun web portal.