BAB I
PENDAHULUAN

1.1 Latar Belakang

Berbagai macam teknologi, alat, dan fitur telah berkembang secara pesat sebagai hasil dari kemajuan teknologi di bidang jaringan saat ini, salah satu contohnya internet. Internet biasanya terdiri dari jaringan komputer, server, *router*, dan perangkat lainnya yang terhubung satu sama lain melalui *protocol suite TCP/IP* standar, yang memungkinkan komputer di seluruh dunia untuk berkomunikasi dan mengakses data. Salah satu layanan yang dapat diakses oleh setiap orang yang terhubung ke internet adalah *website*, yang menjadikan internet sebagai wadah informasi. Dengan jumlah situs web yang semakin meningkat, internet menjadi wadah informasi yang mudah diakses untuk semua orang. *Website* juga sangat bagus untuk berkomunikasi dengan orang dan perusahaan; contohnya, banyak orang yang menggunakannya untuk melakukan jual beli atau promosi. Perkembangan *website* ini mendorong para *hacker* dan *cracker* untuk menjadikannya sebagai target utama, mulai dari mencoba memasuki sistem keamanan hingga mencuri data atau informasi yang dapat merugikan orang lain.

Ada sejumlah alat yang tersedia untuk mempermudah peninjauan dan koreksi kesalahan serta kerentanan aplikasi web. Memilih alat yang tepat sangat penting karena keberhasilannya bergantung pada seberapa rentan aplikasinya di lingkungannya. Alat-alat tersebut dapat berupa alat pengujian *white box* atau *black box* sesuai dengan kemampuan dan fiturnya. *White box* digunakan untuk menemukan dan memperbaiki kerentanan keamanan yang mungkin dieksploitasi oleh penyerang pada tahap produksi aplikasi web dengan menganalisis kode sumber dan struktur aplikasi web. Ada pilihan untuk melakukan analisis secara otomatis dengan alat khusus atau secara manual. Salah satu masalah utama dengan jenis uji keamanan aplikasi web ini adalah

jumlah waktu yang diperlukan untuk melakukan analisis kode lengkap, yang dipengaruhi oleh jumlah baris kode dan kompleksitasnya[11].

Dalam metode *black box*, keamanan aplikasi web diuji tanpa melihat kode sumber atau memahami strukturnya. Metode ini memungkinkan analisis aplikasi web dalam lingkungan simulasi yang mirip dengan lingkungan yang akan diamati oleh penyerang. Alat otomatis mengidentifikasi kerentanan dengan mengirimkan masukan khusus ke aplikasi web. Ini menghasilkan kemungkinan kerentanan yang dapat dimanfaatkan oleh penyerang[11].

Intrusion Detection System (*IDS*) adalah sebuah sistem yang melacak trafik secara *real-time* untuk mendeteksi upaya penyusupan terhadap sistem. Snort adalah perangkat lunak berbasis *IDS* yang dibuat dan dikembangkan oleh Martin Roesch dan kemudian menjadi proyek *open source*. Ini adalah salah satu perangkat *IDS* yang paling umum digunakan pada sistem server. Penerapan *Intrusion Detection System* (*IDS*) berbasis Snort dalam aplikasi dapat menghasilkan penghematan biaya perangkat lunak karena Snort adalah perangkat lunak *open source* yang handal dalam mendeteksi ancaman keamanan. Pengaturan utama Snort terfokus pada konfigurasi jaringan dan penggunaan rule Snort yang relevan. Kemampuan *IDS* Snort dalam mendeteksi serangan pada sistem keamanan bergantung pada ketersediaan rule yang sesuai. Dalam penelitian ini uji coba dilakukan dengan berbagai pola serangan untuk menilai kemampuan Snort dalam mendeteksi ancaman terhadap keamanan sistem.

Beberapa persoalan keamanan data dan informasi yang telah dijelaskan diatas dan adanya teknologi Snort, maka pada pengerjaan tugas akhir ini akan dibuat sebuah alanisis sistem keamanan yang diharapkan dapat mendeteksi kerentanan khusus-nya pada website dan diharapkan juga meningkatkan perlindungan terhadap data penting dan perangkat lain yang terhubung pada *website*. Sistem keamanan menggunakan Snort sendiri perlu diuji lebih lanjut terhadap beberapa metode, sehingga menjadi dasar bagi perancangan sistem keamanan yang lebih stabil.

Pada penelitan yang relevan sebelumnya ”*Analyzing the traffic of penetration testing tools with an IDS*” oleh F. R. Muñoz, dkk. pada tahun 2016 membahas tentang penggunaan beberapa *tools penetration* testing terhadap DVWA dan WackoPicko. Pada penelitian tersebut melalui beberapa serangan dan *tools*, SNORT tidak dapat mendeteksi semua serangan dari *tools-tools* yang digunakan. Meskipun SNORT memiliki ribuan aturan untuk berbagai macam serangan yang berbeda, SNORT tidak dapat mencakup serangan yang terperinci.

Peneliti dalam hal ini bermaksud melanjutkan penelitian sebelumnya dengan menambahkan *vulnerability scanning*. Merancang sistem menggunakan Snort dengan metode *Vulnerability scanning* dan *Penetration testing*, kemudian menganalisa hasil identifikasi keamanan terhadap serangan. Dengan variasi ujicoba ini diharapkan dapat memberikan kontribusi tambahan dalam kajian khususnya terkait SNORT dalam mendeteksi serangan dengan menggunakan metode *Vulnerability scanning* dan *Penetration testing*.

1.2 Rumusan Masalah

Berdasarkan pengertian dan penjelasan serta latar belakang masalah yang ada, maka pokok permasalahan yang akan dianalisis terbagi menjadi beberapa sub permasalahan, yaitu diantaranya adalah :

a. Bagaimana mengkonfigurasi sistem keamanan menggunakan Snort?
b. Bagaimana hasil dari *vulnerability scanning* pada DVWA dan WakcoPicko?
c. Bagaimana perbandingan hasil *vulnerability penetration testing* pada DVWA dan WakcoPicko ?

1.3 Tujuan Penelitian

Dalam penelitian ini terdapat beberapa tujuan diantaranya adalah sebagai berikut :

a. Mengkonfigurasi sistem keamanan menggunakan Snort.
b. Mendeskripsikan hasil dari *vulnerability scanning* pada DVWA dan WakcoPicko.

c. Mendeskripsikan perbandingan hasil *vulnerability penetration testing* pada DVWA dan WakcoPicko.

1.4 Batasan Masalah

Terdapat beberapa batasan masalah yang diangkat sebagai parameter pengerjaan tugas akhir ini diantaranya adalah :

a. Objek scanning yang diteliti yaitu DVWA dan WackoPicko.
b. Penelitian menggunakan 5 tools, yaitu Wapiti sebagai *Vulnerability Scanning* dan Burp Suit, Acunetix Manual Tools, OWASP ZAP, Postman sebagai *Penetration Testing*.
c. Hasil penelitian terdiri dari analisis dan laporan dari alat yang digunakan.