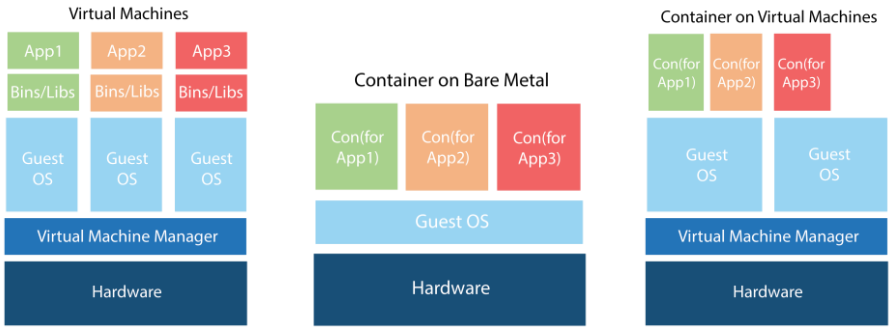
BAB I PENDAHULUAN

I.1 Latar Belakang

Kontainer (*container*) memiliki sejarah panjang dalam dunia komputasi. Tidak seperti *virtualisasi hypervisor,* dimana satu atau lebih mesin independen yang berjalan secara virtual pada perangkat keras fisik melalui lapisan perantara, kontainer justru berjalan di ruang pengguna di atas kernel sistem operasi. Akibatnya, kontainer sering disebut virtualisasi tingkat sistem operasi.



Gambar I. 1 Teknis kerja VM, kontainer pada *Bare Metal* dan kontainer pada VM

Menurut Saleh Dwiyatno et al., (2020), kontainer adalah virtualisasi pada level sistem operasi dimana tiap proses atau aplikasi yang dijalankan tiap kontainer memiliki kernel yang sama. Kontainer memungkinkan organisasi untuk mempercepat siklus pengiriman dan dengan cepat menskalakan operasi untuk memenuhi permintaan pasar yang bergerak cepat saat ini. Dengan semakin banyaknya organisasi yang memindahkan beban kerja mereka ke kontainer, ekosistem kontainer terus berkembang untuk mengakomodasi lingkungan yang semakin dinamis. Penggunaan kontainer terus meningkat setiap tahunnya seperti pada laporan Datadog. Datadog merupakan laman yang fokus membahas monitoring performa aplikasi terutama Docker. Penelitian ini didasarkan pada laporan penggunaan kontainer yang telah dibuat sebelumnya seperti laporan orkestrasi kontainer dan laporan penelitian Docker. Laporan tersebut menjelaskan bahwa terdapat lebih dari 1,5 miliar kontainer yang dijalankan oleh puluhan ribu

pelanggan Datadog untuk memahami bagaimana registri *image*, jaringan, dan teknologi lain digunakan di lingkungan kontainer dunia nyata (Datadog, 2020).

Dengan hadirnya kontainer Docker dan segala kemudahan didalamnya, tentu juga perlu diperhatikan mengenai keamanan dan risiko dari penggunaan Docker tersebut. Karena memang tak dapat dipungkiri bahwa semakin berkembangnya teknologi maka akan bertambah pula peluang risiko yang akan terjadi. Risiko tersebut bisa diakibatkan karena adanya kerentanan pada suatu sistem. Kerentanan tersebut bisa menjadi sasaran utama pada serangan siber seperti *hacking, malware*, dan sebagainya oleh pihak yang tidak bertanggung jawab demi kepentingan pribadi.

Untuk mengukur celah atau *vulnerability* pada aplikasi, dapat dilakukan dengan melakukan pemindaian (*scanning*). Pada pengukuran pemindaian akan menemukan risiko *vulnerability* sehingga dapat menemukan evaluasi dan strategi atau rekomendasi yang tepat untuk mengurangi atau menghilangkan risiko. *Vulnerability scanning* tentu dilakukan melalui *scanner* (pemindai). Pada penelitian ini akan dilakukan *scanning* menggunakan AlienVault dan Docker Bench for Security yang merupakan *opensource vulnerability scanner*. Dengan menerapkan *opensource vulnerability scanner,* maka akan lebih mudah untuk mendapatkan identifikasi *vulnerability*, kemampuan tindakan. dokumentasi, serta perizinan dan keamanan.

Selain harus memerhatikan keamanan suatu aplikasi, maka yang harus diperhatikan kembali adalah standardisasi secara global untuk menyamaratakan faktor-faktor prasyarat apakah yang harus terdapat pada suatu aplikasi agar aplikasi tersebut dapat dikatakan aman. Serta dapat membangun sistem yang mampu mendorong, meningkatkan, menjamin mutu dari keamanan aplikasi serta mampu memfasilitasi kehadiran aplikasi yang aman sesuai dengan standar skala global. Salah satu *framework* yang dapat dijadikan sebagai acuan atau

standardisasi keamanan aplikasi yaitu CIS Control. Justin Gratto (2020) pada unggahahnnya di laman *Securicy.com* mengatakan bahwa CIS Control V7.1 merupakan serangkaian 20 praktik terbaik yang dapat memandu melalui proses pembuatan strategi keamanan siber. Penelitian menunjukkan bahwa menerapkan CIS Control dapat mengurangi risiko serangan dunia maya yang berhasil di perusahaan sebanyak 85%. Beth Smith (2020), pada unggahannya di laman *itsecurity.org* mengatakan bahwa CIS Control telah merilis tolok ukur konfigurasi keamanan pertama untuk Docker 1.6, yang membuat lebih dari 80 rekomendasi untuk mengonfigurasi dan mengoperasikan Docker di lingkungan produksi. Tolok ukur dibuat berdasarkan konsensus dengan perwakilan dari Docker, VMware, Skala Kognitif, Bursa Efek Internasional, Rakuten, dan CIS. Beberapa contoh tolok ukur akan dibahas di bab selanjutnya.

Maka dari itu berdasarkan permasalahan di atas, pada penelitian ini akan dijelaskan bagaimana hasil analisis mengenai *vulnerability* pada *Vulnerable Docker* menggunakan *vulnerability scanner* yakni AlienVault dan Docker Bench for Security dengan mengacu pada *framework* CIS Control. Hasil analisis tersebut nantinya dapat memberikan beberapa rekomendasi untuk membantu memperkuat sistem Docker untuk kedepannya.

I.2 Perumusan Masalah

Berdasarkan uraian latar belakang di atas, perumusan masalah dalam penelitian ini adalah sebagai berikut:

a. Bagaimana hasil *vulnerability scanning* yang dihasilkan oleh hasil *scanning* melalui AlienVault dan Docker Bench for Security?
b. Bagaimana hasil analisis risiko yang dihasilkan oleh hasil *scanning* melalui AlienVault dan Docker Bench for Security?
c. Bagaimana dasar-dasar kontrol pada pengujian *Vulnerable Docker* berdasarkan *framework* CIS Control?

I.3 Tujuan Penelitian

Penelitian ini bertujuan untuk:

a. Mengetahui hasil *vulnerability scanning* yang dihasilkan oleh hasil *scanning* melalui AlienVault dan Docker Bench for Security
b. Mengetahui hasil analisis risiko yang dihasilkan oleh hasil *scanning* melalui AlienVault dan Docker Bench for Security
c. Memahami dasar-dasar kontrol pada pengujian *Vulnerable Docker* berdasarkan *framework* CIS Control

I.4 Batasan Penelitian

Batasan pada penelitian ini adalah sebagai berikut:

1. Penelitian dibatasi pada lingkungan sistem
2. Pembahasan risiko berdasarkan *vulnerability* dan *threat* pada sistem aplikasi dan Docker
3. Penelitian hanya memberikan analisis *vulnerability* pada *Vulnerable Docker* sesuai dengan *framework* CIS Control

I.5 Manfaat Penelitian

Manfaat penelitian ini yakni sebagai berikut:

1. Secara Teoritis, penelitian ini memberikan analisa risiko berdasarkan *vulnerability* pada *Vulnerable Docker* dan eksploitasi yang akan dideteksi menggunakan AlienVault dan Docker Bench for Securitty serta menyediakan pemahaman kontrol berdasarkan *framework* CIS Control
2. Secara Praktis, penelitian ini dapat memjelaskan mekanisme penentuan serta penguatan terhadap *vulnerability* pada Docker

I.6 Sistematika Penulisan

Tugas akhir ini diuraikan dengan sistematika penulisan sebagai berikut:

Bab I Pendahuluan

Pada bab ini berisi uraian mengenai konteks permasalahan, latar belakang permasalahan, perumusan masalah, tujuan penelitian, batasan, manfaat penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Bab ini berisi literatur yang relevan dengan permasalahan yang diambil dan dibahas pula hasil-hasil referensi maupun penelitian terdahulu yang dapat digunakan untuk merancang dan menyelesaikan masalah seperti teori mengenai Docker, CIS Control, risiko, ancaman, AlienVault, Docker Bench for Security, dan standar *framework* CIS Control.

Bab III Sistematika Penyelesaian Masalah

Metodologi penyelesaian merupakan penjelasan metode / konsep / kerangka kerja yang telah dipilih pada bab Tinjauan Pustaka. Pada bab ini dijelaskan langkah-langkah dan sistematika penelitian secara rinci untuk penyelesaian permasalahan.

Bab IV Perancangan Sistem Terintegrasi

Bab ini berisi penjelasan berupa skenario perancangan sistem serta penggunaan AlienVault dan Docker Bench forSecurity terhadap *Vulnerable Docker*.

Bab V Analisa dan Evaluasi Hasil Percobaan

Pada bab ini, disajikan hasil rancangan, temuan, analisis dan pengolahan data yang akan dijadikan acuan atau rekomendasi terhadap kontrol terhadap *Vulnerable Docker* untuk perbaikan kedepannya.

Bab VI Kesimpulan dan Saran

Pada bab ini dijelaskan kesimpulan dari penyelesaian masalah yang dilakukan serta jawaban dari rumusan permasalahan yang ada pada

bagian pendahuluan. Saran dari solusi dikemukakan pada bab ini untuk penelitian selanjutnya.