BAB I PANDAHULUAN

I.1. Latar Belakang

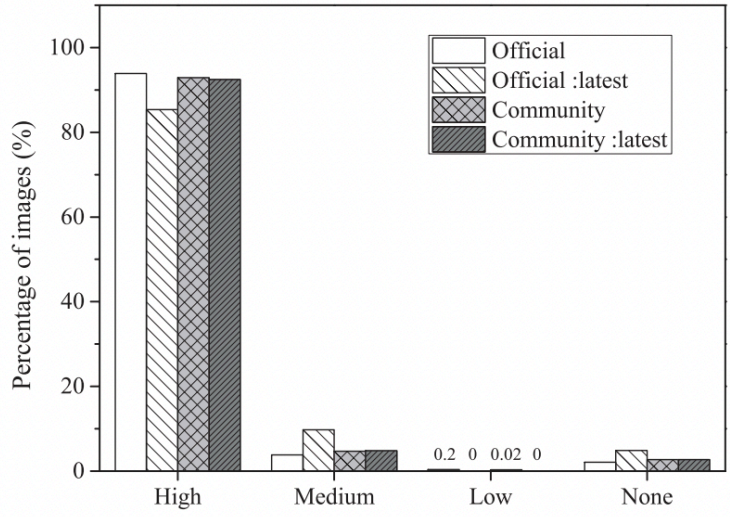
Virtualisasi dan *container* merupakan bagian yang sangat penting dalam penerapan *cloud computing*, karena sangat berpengaruh pada efisiensi pengelolaan sumber daya infrastruktur *cloud computing*. Hasil survey dari sdxcentral yang dilakukan pada tahun 2015 menunjukan bahwa 94% responden yang berasal dari kalangan perusahaan telah mengadopsi teknologi *container* sejak 12 bulan terakhir. Teknologi *Virtual Machine* dimanfaatkan untuk menyelesaikan masalah heterogenitas (perbedaan versi *library* atau *tools* dari beberapa aplikasi *web*). Peningkatan jumlah aplikasi *web* yang harus di *hosting* harus diikuti dengan peningkatan kualitas ataupun kuantitas sumber daya, terlebih saat hadirnya kebutuhan *High availability* dari layanan *web* tersebut. Teknik kontainerisasi (virtualisasi berbasis *container*) hadir sebagai solusi dan menjadi tren saat ini. *Container* telah menjadi teknik yang populer untuk menjalankan beberapa layanan aplikasi pada satu *host*.

Docker adalah salah satu teknologi berbasis *container* yang paling banyak digunakan. Docker mendistribusikan aplikasi (misalnya *Apache*, MySQL, Joomla) dalam bentuk *images*. Setiap *images* berisi perangkat lunak aplikasi target serta pustaka pendukung dan *file* konfigurasinya. Akibatnya, *Images* Docker menyediakan cara yang nyaman untuk menyimpan dan mengirimkan aplikasi. Docker memiliki komunitas pengembangan dan tempat berbagi *Images* Docker yang bernama Docker Hub. Docker Hub telah diperkenalkan secara public pada tahun 2014 sebagai layanan *cloud registry service* untuk berbagi *application images*. *Images* pada Docker didistribusikan menggunakan *repositories* sehingga memungkinkan pengembangan dan pemeliharaan *images* pada Docker dapat berversi atau memiliki banyak versi.

Docker Hub memiliki dua jenis repositori *public* yaitu repositori resmi dan repositori komunitas. Repositori resmi bersifat *public* dan *images* ini memiliki sertifikat dari vendor (misalnya, Canonical, Oracle, Red Hat, dan Docker). Sebaliknya, repositori

komunitas dapat dibuat oleh pengguna atau organisasi mana pun. Menurut Rui Shu, pada penelitian yang berjudul : "*A Study of Security Vulnerabilities on Docker Hub*" terdapat hampir 100 repositori resmi dan mengidentifikasi sekitar 100.000 repositori komunitas publik. Pada bulan Januari 2015, survei dari perusahaan Forrester menunjukkan bahwa keamanan menjadi perhatian utama ketika memutuskan apakah akan *deploy* suatu *container*.

Survei menemukan bahwa dari berbagai masalah keamanan, masalah Kerentanan & Malware adalah yang terbesar. Oleh karena itu, pada penelitian ini berhipotesis bahwa kompleksitas konfigurasi perangkat lunak dalam Docker *Hub* yang dikombinasikan dengan sejumlah besar *Images* yang dibuat oleh berbagai pihak dan memiliki perbedaan versi menghasilkan sistem *vulnerability*. Intuisi ini membawa penelitian ke pertanyaan utama dalam penelitian ini adalah : Bagaimana status kerentanan yang ditemukan pada Docker *Hub Images*?



Gambar I. 1 Kerentanan pada Docker Images (Shu, 2022)

Dalam diagram diatas menunjukkan bahwa kerentanan paling tinggi dihasilkan oleh Docker Official *Images*. Penelitian ini akan memberikan proses analisis kerentanan pada Docker *Images* yang tersedia secara umum di Docker Hub. Dari sampel *Data* Docker *Images* yang digunakan adalah Docker dan Joomla. Docker dan Joomla merupakan

official *images* dari Docker. Pengunaan *Images* Docker disini untuk analisis *vulnerability* sebagai *platform* yang mengemas berbagai *file* perangkat lunak atau aplikasi ke dalam suatu *Container*. Sedangkan penggunaan *Images* Joomla digunakan sebagai evaluasi eksperimental *vulnerability* sebuah sistem aplikasi berbasis *website* dimana keamanan *website* pada saat ini menjadi prioritas di setiap aset IT. Sebuah aplikasi berbasis *website* memiliki *Content Management System* (CMS). Joomla yang merupakan salah satu CMS terpopuler sering kali di sebut sebagai *tools* yang termudah dalam manajemen website, terutama jika berada pada level pemula mengenai dunia website. Kepopuleran inilah yang menjadi alasan bahwa Joomla menjadi hal yang menarik oleh *hacker*. Joomla memiliki *Official Images* pada Docker hal ini dapat membantu penelitian mengenai *vulnerability* pada Images Joomla.

Untuk menjawab pertanyaan utama diatas, penelitian ini membangun kerangka kerja yang secara otomatis dapat menjelaskan proses mulai dari menemukan, mengunduh, dan menganalisis *vulnerability* pada Docker *Images*. Sehingga diberikan kontribusi membangun sistem *Vulnerability Management* pada *Container* Docker berdasarkan standar *Cyber Resilience Review* (CRR). Menurut ISAO *Standards Organization, Cyber Resilience Review* (CRR) adalah *framework Cybersecurity* untuk mengevaluasi ketahanan operasional dan praktik keamanan *Cyber* pada organisasi.

Selain memiliki fokus penelitian pada kerangka kerja *Vulnerability Management*, penelitian ini juga menganalisis penilaian proses mitigasi dari aset IT yang memiliki perbedaan versi. Dengan memiliki data kenaikan versi suatu *Software Container*, maka proses mitigasi kerentanan dapat dianalisis. Selanjutnya membandingkan *tools container scanner* yang akan digunakan yaitu Aquasec dan Anchore sebagai *tools* yang bersifat *open Source* untuk *Container Scanner* khususnya Docker. Sehingga dapat mengetahui seberapa jauh Aquasec dan Anchore dalam mendeteksi kerentanan pada Docker *Images*. Hal ini juga akan menjadi perbandingan evaluasi untuk mengakomodasikan *Vulnerability Management*.

Berdasarkan uraian tersebut, maka dilakukan penelitian dalam bentuk tugas akhir dengan judul: “Analisis *Vulnerability Management* pada *Container* Docker

Menggunakan Aquasec dan Anchore Berdasarkan Standar *Cyber Resilience Review* (CRR)". Penelitian ini menghasilkan usulan yaitu *self-assessment* suatu *Vulnerability Management Container* Docker menurut standar *Cyber Resilience Review* (CRR) dan kajian perbandingan *tools Open Source* sebagai rekomendasi *tools* yang bekerja secara optimal dalam melindungi serta menangani keamanan pada suatu sistem.

I.2. Perumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan permasalahan untuk penelitian ini adalah sebagai berikut:

1. Bagaimana menelusuri *vulnerability* pada *Container* Docker?
2. Bagaimana *vulnerability* dengan membuat dua *Container*?
3. Bagaimana variasi *vulnerability scanner software*?
4. Bagaimana data teknis hasil *vulnerability scanning* yang dikelola berdasarkan suatu standar?

I.3. Tujuan Penelitian

Berdasarkan perumusan masalah, maka tujuan dari penelitian ini adalah sebagai berikut:

1. Mengidentifikasikan *vulnerability* pada *Container* Docker dengan menggunakan dua *vulnerability scanner* yang berbeda.
2. Mengidentifikasikan *vulnerability* antara *Container* Docker versi lama dan versi baru.
3. Mengetahui hasil perbandingan kinerja *vulnerability scanner* antara Aquasec dan Anchore
4. Melakukan analisis *Vulnerability Management* berdasarkan standar *Cyber Resilience Review* (CRR)

I.4. Manfaat Penelitian

Adapun manfaat pada penelitian ini sebagai berikut:

1. Teoritis
   - Memberikan kontribusi keilmuan terkait pengelolaan kerentanan pada Docker *Images*.

- Membantu untuk membuat *self-assessment* dalam proses *Vulnerability Management* menggunakan standar *Cyber Resilience Review* (CRR)

2. Praktis
   - Memberikan rekomendasi berupa kajian dan pembanding *tools open source container scanner* yang bermanfaat untuk menganalisa *vulnerability* pada Docker *Images*.

I.5. Batasan Masalah

Adapun batasan masalah pada penelitian ini sebagai berikut:

1. Penelitian ini terbatas pada *vulnerability management* pada Docker.
2. Penggunaan analisis penelitian ini hanya menggunakan 13 aspek *Vulnerability Management* menurut standar *Cyber Resilience Review* (CRR).
3. Implementasi penelitian ini menggunakan simulasi pada skala laboratorium

I.6. Sistematika Penelitian

Sistematika penulisan pada penelitian ini terdiri dari tujuh bab, yang tersusun sebagai berikut:

BAB I **PENDAHULUAN**

Bab ini berisi mengenai uraian latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah dan sistematika penelitian.

BAB II **KAJIAN TEORI**

Bab ini berisi mengenai literatur yang relevan dengan permasalahan yang dihadapi, penelitian terdahulu yang memiliki keterkaitan dengan penelitian yang sedang dilakukan, dan menjelaskan teori-teori yang digunakan seperti *Vulnerability Management, Cyber Resilience Review*

(CRR), *Docker Container, Joomla Container, Aquasec, Anchore*.

BAB III METODOLOGI PENELITIAN

Bab ini berisi penjelasan mengenai konseptual model untuk merumuskan solusi dari permasalahan yang ada. Selain itu, penelitian ini juga menjelaskan langkah-langkah dan sistematika penelitian yang digunakan. Metode penelitian menggunakan model konseptual dengan sistematika penelitian yang memiliki tahapan-tahapan berdasarkan standar *Cyber Resilience Review* (CRR).

BAB IV RANCANGAN PENGUJIAN

Bab ini berisi penjelasan rancangan sistem dan penggunaan *tools open Source* terhadap skenario yang dilakukan pada *vulnerability docker container* dalam bentuk simulasi menggunakan metode pengembangan analisis berbasis *Cyber Resilience Review* (CRR).

BAB V IMPLEMENTASI HASIL PENGUJIAN

Bab ini berisi pengujian yang dilakukan berdasarkan rancangan pengujian dengan menghasilkan *vulnerability* data *container* docker dan *container* joomla pada versi lama dan versi baru. Pengujian yang dilakukan menggunakan *tools* open *Source* khusus untuk *container scanner* yaitu Aquasec dan Anchore.

BAB VI ANALISIS HASIL PENGUJIAN

Bab ini berisi analisis dari hasil pengujian yang dilakukan mengenai *vulnerability* yang ditemukan pada docker *container* berdasarkan pengujian Aquasec dan Anchore dengan tahapan yang dilakukan berdasarkan standar *Cyber Resilience Review* (CRR).

BAB VII KESIMPULAN DAN SARAN

Bab ini berisi penjelasan kesimpulan dari penelitian yang telah dilakukan, rancangan sistem dan skenario pengujian, perancangan dan analisa usulan, serta memberikan saran untuk penelitian selanjutnya.