Jurnal Ilmiah Informatika (JIF)
| ISSN (Print) 2337-8379 | ISSN (Online) 2615-1049 |
Jurnal online di akses di http://ejournal.upbatam.ac.id/index.php/jif

Implementasi *Static Application Security Testing* Menggunakan Jenkins CI/CD Berbasis *Docker Container* Pada PT. Emporia Digital Raya

Abriza Mahandis Shama[1], *Dian W. Chandra*[2]

[1]*Universitas Kristen Satya Wacana, Jl. Diponegoro No.52-60, Salatiga 50711, Indonesia*
[2]*Universitas Kristen Satya Wacana, Jl. Diponegoro No.52-60, Salatiga 50711, Indonesia*

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 08 Juli 2021
Revisi Akhir: 16 Juli 2021
Diterbitkan *Online*: 10 September 2021

KATA KUNCI

SAST

Jenkins

CI/CD

Container

KORESPONDENSI

No HP: 0895416011508
E-mail: abriza.mhsm@gmail.com

A B S T R A C T

PT. Emporia Digital Raya is an fintech company. The product include web and android application. However, in the deployment system, PT Emporia Digital Raya still uses an ancient system with a single vm system and only uses git for deployment to server. Even though at this time the deployment process of an application has grown very far. Therefore, in this study will created a system which is currently popular being used. This system is called DevSecOps. DevSecOps will need a tools like Jenkins, Sonarqube, and Docker. The core of this system is the automation process where the deployment process is no longer done manually as before. With this system, it is proved to help speed up developer work and improve code quality.

1. PENDAHULUAN

Seiring dengan berkembangnya zaman proses dalam *deployment* aplikasi makin mengalami kemajuan. Banyak cara untuk deployment suatu aplikasi ke server. Dalam arsitektur kuno proses deployment masih menggunakan metode SCP yaitu copy file to server kemudian diikuti dengan Git. Dan yang paling terkini adalah *Continuous Integration dan Continuous Delivery* atau biasa disebut CI/CD. Di PT Emporia Digital Raya system deployment masih menggunakan metode Git dan tanpa proses *security testing*. Untuk itu butuh system yang mampu mendeteksi tingkat kerentanan dalam aplikasi secara CI/CD

SAST adalah proses *vulnerability assessment*. SAST dapat menemukan berbagai kerentanan dan lubang keamanan yang terdapat pada kode aplikasi yang sedang dikerjakan. SAST merupakan salah satu dalam suatu proses DevSecOps dimana dengan tujuan untuk mempercepat suatu development dan mengurangi kerentanan suatu kode program [1]. Ada banyak tools SAST yang digunakan tapi untuk perancangan kali ini penulis menggunakan Sonarqube untuk *vulnerability assessment.* Untuk menjalankan SAST install tools yang digunakan developer bisa cek hasilnya dengan menaruh repositorynya di tools tersebut. Namun hal itu akan membuat ribet developer karena developer harus manual menyalin dulu alamat reponya lalu meletakkannya ke tools SAST. Untuk itu dibutuhkan sistem yang mampu menjalankan semua itu secara otomatis dan cepat. Sistem itu adalah Jenkins, Jenkins adalah tools *open source* yang mampu build projek dan deploy projek dengan ringan dan cepat secara otomatis. Jenkins adalah tools yang paling populer untuk CI/CD. Jenkins juga dilengkapi ratusan plugin untuk mendukung optimalisasi dalam proses CI/CD.

Dengan adanya SAST menggunakan Jenkins CI/CD diharapkan aplikasi yang didevelop oleh PT Emporia Digital Raya tidak ada lagi kerentanan yang ada serta proses *deployment* menjadi lebih cepat karena dengan metode CI/CD.

2. TINJAUAN PUSTAKA

2.1. Static Application Security Testing (SAST)

SAST adalah proses awal untuk mengidentifikasi berbagai kerentanan dan kelemahan di dalam *source code* seperti SQL Injection. SAST tidak membutuhkan *system requirement* tertentu untuk menjalankannya. SAST dapat dijalankan secara otomatis, tentunya hal itu akan menghemat waktu dalam proses *Software Development Life Cycle* (SDLC).

2.2. Sonarqube

Sonarqube merupaan salah satu tools SAST. Sonarqube adalah otomasi review kode program untuk mendeteksi *bugs*, *vulnerabilities*, dan *code smells* [2]. Sonarqube dapat diintegrasikan dengan CI/CD tools seperti Jenkins, CircleCI, AWS CodeBuild, Azure DevOps, Atlassian Bamboo, atau Travis CI.

2.3. Version Control System

Version Control Sistem adalah sebuah sistem yang mencatat perubahan file dari waktu ke waktu. Sebagai contoh, seorang web designer akan menyimpan setiap versi dari rancangan yang dibuatnya. Dengan menggunakan VCS seperti git memungkinkan untuk kembali ke kode program sebelumnya, mengetahui perubahan yang terjadi, serta melihat siapa yang telah membuat perubahan pada suatu kode program [3].

2.4. Jenkins

Jenkins adalah tools *open source* yang mampu build projek dan deploy projek dengan ringan dan cepat secara otomatis [4]. Jenkins adalah tools yang paling populer untuk CI/CD. Jenkins juga dilengkapi ratusan plugin untuk mendukung optimalisasi dalam proses CI/CD.

2.5. Continuous Integration

Fase ini merupakan siklus pengecekan dalam siklus DevOps dimana proses pengembangan perangkat lunak dan operasional dilakukan. Setiap perubahan yang dilakukan developer sedini mungkin harus terdeteksi, Terlebih jika terdapat bug integrasi kode melibatkan kompilasi, urasan kode, *unit test, functional test, integration test* single packaging. Fase pengujian ini biasanya dibantu dengan beberapa tools misalnya Gitlab CI, Jenkins, Circle CI, Travis CI [5]. Namun pada penelitian kali ini penulis menggunakan Jenkins.

2.6. Continuos Delivery

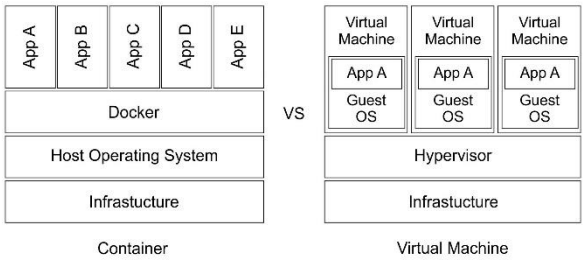
Dalam continuous delivery, proses otomasi dilakukan di hampir semua tahap termasuk deployment. Pada tahap ini terjadi proses sebuah pipeline Jenkins yang berjalan sukses dan berhasil deploy suatu aplikasi ke server. *Continuous Delivery* akan meneruskan proses dari *Continuous Integration* apabila berhasil dilakukan.

2.7. Docker

Docker adalah sebuah projek open source yang ditujukan untuk software engineer, DevOps engineer atau sysadmin untuk membangun, mengemas dan menjalankan aplikasi dimana pun di dalam sebuah container. Docker berfungsi sebagai virtualisasi di dalam sebuah sistem operasi atau sebuah server atau sebuah web server atau bahkan sebuah database server, dimana dengan menggunakan virtualisasi ini, diharapkan developer dapat mengembangkan aplikasi sesuai dengan spesifikasi server [6].

2.8. Container

Container adalah sebuah abstraksi layer aplikasi yang membungkus package diperlukan oleh aplikasi [7]. Container merupakan suatu sistem operasi virtualisasi yang ringan yang dan menjadi alternatif dari teknik virtualisasi berbasi hypervisor. Container berbeda dengan virtualisasi yang memerlukan sistem operasi tersendiri untuk masing-masing mesin virtual, sedangkan banyak container dapat berbagi sistem operasi yang sama.



Gambar 1. Container vs Virtual Machine

Oleh karena itu beban hardware untuk menjalan container akan lebih ringan bila dibandingkan dengan VM karena VM menjalankan beberapa OS pada setiap mesinnya dan menjalankan aplikasi pada setiap OS-nya sedangkan container hanya membutuhkan satu OS dan dapat menjalankan banyak aplikasi dengan dependensi yang berbeda-beda. Data dalam container juga bersifat terisolasi yaitu tidak tidak akan keluar ke vm induknya [8].

3. METODOLOGI

Dalam sebuah penelitian dibutuhkan sebuah tahapan agar dapat dibuat acuan dalam pembuatan penelitian. Adapun untuk tahapan penelitian penelitian ini dapat dilihat di gambar berikut.

Gambar 2. Metodologi Penelitian

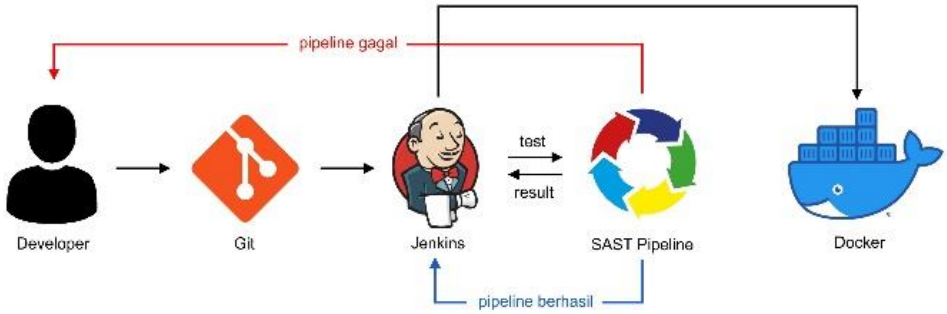
Penjelasan dari tahapan yang terdapat pada gambar 3 adalah pada tahap pertama yaitu identifikasi masalah. Mengapa perlu menerapkan SAST dalam proses deployment aplikasi. Tahap selanjutnya adalah studi literatur dimana mempelajari literatur yang sesuai dengan penelitian. Setelah dirasa cukup mempelajari literatur selanjutnya yaitu membuat perancangan arsitektur sistem.

Setelah semua persiapan telah dilalui, selanjutnya adalah implementasi dari perancangan sistem yang telah dibuat untuk membangun SAST menggunakan Jenkins CI/CD berbasis Docker Container.

4. HASIL DAN PEMBAHASAN

4.1. Alur Sistem

Pada penelitian kali ini penulis merancang keseluruhan sistem sebagai berikut.



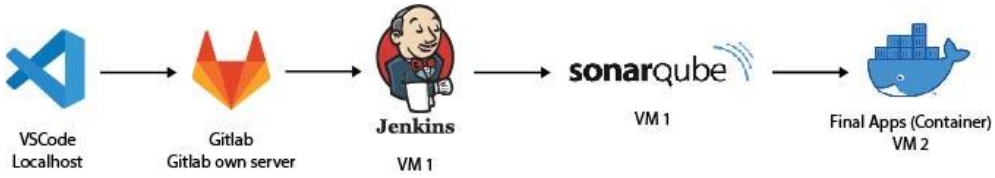
Gambar 3. Alur SAST dengan Jenkins

Proses dimulai dengan developer melakukan *commit* ke *reporsitory* lalu Jenkins akan mendeteksi bahwa developer telah melakukan *commit* dan akhirnya *pipeline* di Jenkins akan menjalankan SAST. Bila proses gagal maka akan dikembalikan ke developer dan developer wajib untuk memperbaiki kodenya, sedangkan bila lolos dari SAST maka akan masuk proses deploy oleh Docker. Bila telah berhasil proses SAST maka akan diteruskan untuk deploy ke server menggunakan Docker.

4.2. Arsitektur Sistem

Dalam suatu sistem DevOps pastinya ada arsitektur sistem yang dibuat. Dimana letak masing-masing tools yang digunakan.

Berikut adalah keseluruhan arsitektur sistem pada penelitian kali ini.



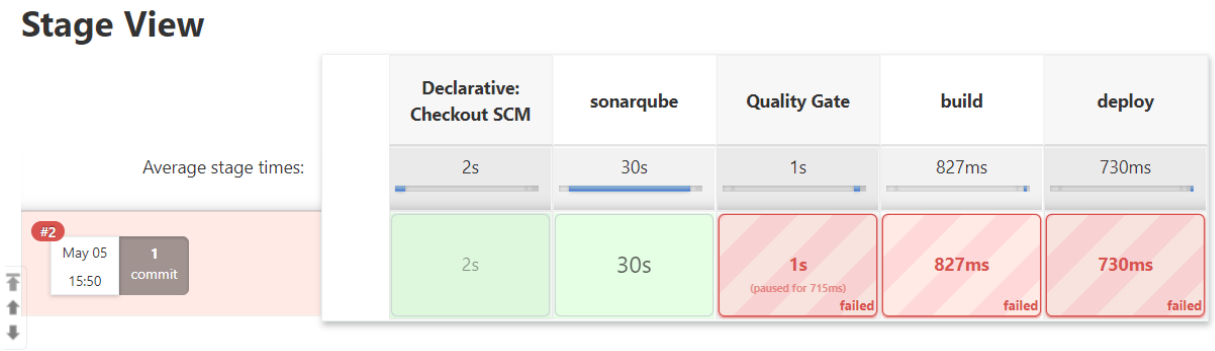
Gambar 4. Arsitektur Sistem

Dimulai dengan Visual Stuido Code yang berada di masing-masing local pc developer lalu dilanjutkan dengan sistem Git menggunakan Gitlab dan akan diteruskan ke Jenkins untuk proses CI/CD. Jenkins dipasang di private server sendiri. Lalu Jenkins akan meneruskan proses SAST yang menggunakan tools Sonarqube. Begitu juga dengan Sonarqube yang dipasang di private server sendiri. Apabila telah melewati proses SAST maka akan diteruskan untuk deploy menggunakan docker dan aplikasi final akan menjadi container. Container akan berada pada private

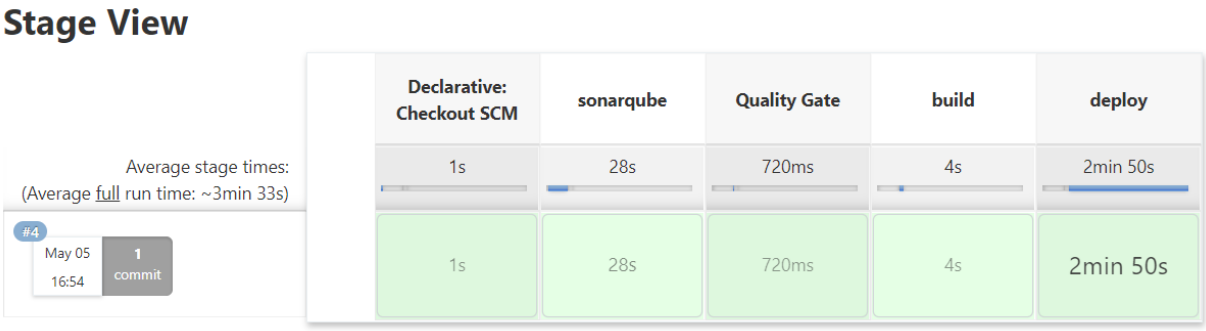
server yang berbeda dengan Jenkins dan Sonarqube. Hal itu dikarenakan untuk mempercepat proses CI/CD.

4.2.1. Jenkins Pipeline

Didalam Jenkins terdapat pipeline untuk menunjukkan proses CI/CD. Dalam sebuah pipeline terdapat banyak proses. Pada penelitian kali ini terdapat *Static Application Security Testing (SAST)* di dalam pipeline Jenkins. Berikut adalah pipeline yang terdapat SAST.



Gambar 4. Pipeline yang gagal

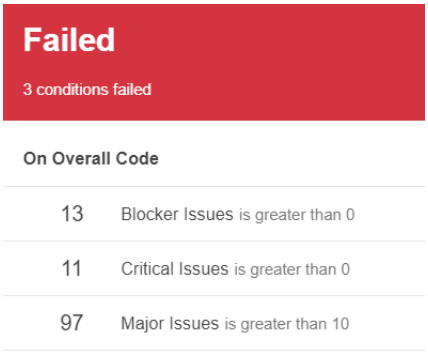


Gambar 5. Pipeline yang berhasil

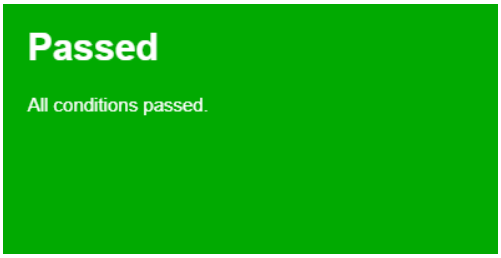
Penjelasan dari gambar 4 adalah pipeline yang gagal melewati proses SAST. Oleh karena itu perlu diperbaiki terlebih dahulu dan apabila setelah diperbaiki maka pipeline akan menunjukkan seperti gambar 5.

4.2.2. SAST Result

Hasil dari SAST akan ditampikan di halaman Sonarqube. Sonarqube akan menampilkan mengapa pipeline Jenkins menjadi gagal dan akan memberikan detail kerentanan yang terdapat pada kode program.



Gambar 6. Hasil kerentanan SAST



Gambar 7. Hasil dengan nihil kerentanan

Kedua gambar di atas adalah tampilan utama yang terdapat dalam Sonarqube dan menunjukkan jenins kerentanan yang ada dalam kode program. Hasil *failed* dan *passed* akan mempengaruhi pipeline Jenkins yang ada pada gambar 4 dan 5.

4.2.3. Docker Container

Langkah terakhir dari sistem ini adalah apabila *pipeline* di Jenkins selesai maka Jenkins akan otomatis menjalankan Docker untuk *build* Container.

```
root@hndsme:/home/abriza_mhsm# docker images | grep "staff"
staff-system-handis        latest              dad7ec7054a8        13 minutes ago      919MB
root@hndsme:/home/abriza_mhsm# docker ps | grep "staff"
c9bb25eb535a        staff-system-handis:latest   "/usr/sbin/httpd -D …"   13 minutes ago      Up 13 minutes
.0.0.0:2205->80/tcp     staff-system-handis
root@hndsme:/home/abriza_mhsm#
```

Gambar 8. Container

4.3. Hasil dan Pengujian

Hasil penelitian kali ini adalah dapat mempercepat proses kerja developer yang ada pada PT. Emporia Digital Raya karena dibantu proses CI/CD serta meningkatkan kualitas kode program karena terdapat proses SAST. Adapun untuk pengujian dari penelitian ini dalam bentuk tabel sebagai berikut. Pengujian dibagi menjadi dua hal yaitu fungsional dan pengujian dengan sistem yang sebelumnya.

Tabel 1. Pengujian Fungsional

No	Pengujian	Hasil
1	Jenkins dapat menjalankan pipeline dengan otomatis	Valid
2	Pipeline dapat membaca hasil dari Sonarqube	Valid
3	Hasil SAST dari Sonarqube dapat menampilkan kerentanan	Valid
4	Docker dapat build container setelah pipeline selesai	Valid

Tabel 2. Perbandingan dengan sistem sebelumnya

No	Sistem sebelumnya	Sistem Sekarang
1	Semua aplikasi masih menggunakan satu vm	Semua aplikasi sudah menggunakan sistem container
2	Proses push ke server masih manual dengan git push dan git pull	Otomatis dengan sistem CI/CD
3	Tidak ada vulnerability assessment sebelum naik ke server	Terdapat proses SAST sebelum naik ke server

5. KESIMPULAN DAN SARAN

Dari penelitian ini dapat disimpulkan bahwa sistem DevSecOps menggunakan tools seperti Jenkins dan terdapat SAST yang berbasis Container. Di dalam sistem ini terdapat suatu proses CI/CD yang dapat mengotomasi suatu sistem deployment serta terdapat proses security testing yang dinamakan SAST. Dengan adanya sistem ini terbukti dapat membantu para developer yang ada pada PT. Emporia Digital Raya maupun developer lain dalam develop suatu aplikasi agar lebih cepat dan meningkatkan kualitas kode program yang sedang dikerjakan.

Adapun saran dari penelitian ini adalah dapat mengembangkan sistem ini menjadi lebih baik lagi seperti menambahkan proses DAST saat aplikasi sudah berhasil ke server atau meneruskan untuk deploy menggunakan Kubernetes.

DAFTAR PUSTAKA

[1] S. P. K. M. M. N. Rahul, "Implementation of DevSecOps using Open-Source tools," *Int. J. Adv. Res.*, vol. 5, no. 3, pp. 1050–1051, 2019, [Online]. Available: www.IJARIIT.com.

[2] "SonarQube Documentation | SonarQube Docs." https://docs.sonarqube.org/latest/ (accessed Feb. 24, 2021).

[3] J. M. Priera and R. T. Ganefi, "Automatic Deployment System Dengan Menggunakan Metode Continuous Integration Di Kakatu," *J. Ilm. Komput. dan Inform.*, 2017.

[4] "Jenkins." https://www.jenkins.io/ (accessed Feb. 25, 2021).

[5] T. Tohirin, S. F. Utami, S. R. Widianto, and W. Al Mauludyansah, "Implementasi DevOps Pada Pengembangan Aplikasi e-Skrining Covid-19," *Multinetics*, vol. 6, no. 1, pp. 15–20, 2020, doi: 10.32722/multinetics.v6i1.2764.

[6] R. A. Putra, "Analisa Implementasi Arsitektur Microservoces Berbasis Kontainer Pada Komunitas Pengembang Perangkat Lunak Sumber Terbuka (OpenDayLight DevOps Community)," *J. Sist. Infomasi Teknol. Inf. dan Komput. (Just It) Univ. Bina Nusant. Magister Manaj. Sist. Inf. Jakarta*, pp. 150–162, 2018.

[7] "What is a Container? | App Containerization | Docker." https://www.docker.com/resources/what-container (accessed Dec. 29, 2020).

[8] A. Frederius, M. Data, and W. Yahya, "Implementasi Penyimpanan Data Persisten pada Docker Swarm Menggunakan Network File System (NFS)," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 3, no. 2, pp. 9088–9096, 2019.

BIODATA PENULIS

Abriza Mahandis Shama
Mahasiswa Universitas Kristen Satya Wacana, Program Studi Teknik Informatika tahun 2017.
Email: abriza.mhsm@gmail.com

Dian W. Chandra
Dosen Universitas Kristen Satya Wacana, Program Studi Teknik Informatika.
Email: dian.chandra@uksw.edu