Penerapan Automated Security Testing Menggunakan OWASP ZAP pada Continuous Integration/Continuous Deployment (CI/CD)

Ade Irmansyah, I Putu Hariyadi

Universitas Bumigora, Mataram, Indonesia

Correspondence : *e-mail*: adegigihpraka@gmail.com

Abstrak

Ketersediaan teknologi informasi sangat membantu pada proses pemberian layanan yang cepat dan tepat. Namun dibalik kemudahan dan kecepatan pelayanan yang ditawarkan terdapat tantangan dan kendala dari segi isu – isu terkait dengan tranksasi data dan informasi yang ada dalam sistem itu sendiri, keamanan menjadi salah satu isu yang kerap kali muncul dan menjadi tantangan tersendiri dalam pengembangan teknologi dan sistem. Berdasarkan kajian yang penulis lakukan dapat disimpulkan bahwa analisis dan uji keamanan pada aliran CICD perlu dilakukan secara konsisten untuk memastikan bahwa sistem yang akan digunakan benar – benar telah siap dari segi keamanannnya. Berbagai metode dan tools ditawarkan untuk memudahkan proses pengujian keamanan pada sistem. Pada penelitian ini, penulis akan mengangkat metode Automated Security Testing (AST) sebagai metode pengujian sistem keamanan sistem dengan menggunakan alat dukung Dynamic Automatic Security Test (DAST) dengan alat bantu OWASP ZAP. Penelitian ini bertujuan untuk mengetahui bagaimana integrasi dan penerapan OWASP ZAP dalam aliran CI/CD serta mengetahui sejauhmana cakupan pengujian keamanan yang dilakukan oleh OWASP ZAP terhadap sistem yang diuji. Proses pengujian penerapan OWASP ZAP ini akan dilakukan pada Sistem keuangan yang merupakan bagian dari Sistem Informasi Perguruan Tinggi (SIMPT) yang dikembangkan oleh cv.xyz. Penelitian ini diharapkan dapat memberikan solusi yang baik kepada perusahaan mengenai bagaimana proses penerapan AST pada aliran CICD sehingga masalah yang dihadapi dapat terselesaikan dengan baik. Selain itu melalui penelitian ini penulis berharap bisa mendapatkan informasi terkait kendala pada penerapan metode AST dengan menggunakan alat dukung DAST dengan tools OWASP ZAP. Berdasarkan analisis yang telah dilakukan dapat ditarik kesimpulan bahwa proses integrasi OWASP ZAP pada aliran CICD dapat dilakukan namun perlu adanya perbaikan pada proses eksekusi OWASP ZAP. Perlu dilakukan perbaikan dan koreksi kembali terkait proses running dan listening pada port prooxy yang digunakan

Kata kunci: *CICD, AST, DAST, OWASP ZAP, SIMPT.*

Abstract

The availability of information technology is very helpful in the process of providing fast and accurate services. However, behind the ease and speed of service offered, there are challenges and obstacles in terms of issues related to data and information transactions within the system itself, security is one issue that often arises and becomes a challenge in technology and system development. Based on the author's study, it can be concluded that security analysis and testing in the CICD flow need to be carried out consistently to ensure that the system to be used is truly ready in terms of security. Various methods and tools are offered to facilitate the process of security testing on the system. In this study, the author will raise the Automated Security Testing (AST) method as a system security testing method using the Dynamic Automatic Security Test (DAST) support tool with the OWASP ZAP tool. This study aims to determine how the integration and implementation of OWASP ZAP in the CI/CD flow and to determine the extent of security testing carried out by OWASP ZAP on the system being tested. The testing process for the implementation of OWASP ZAP will be carried out on the financial system which is part of the Higher Education Information System (SIMPT) developed by cv.xyz. This research is expected to provide a good solution to the company regarding how the AST implementation process in the CICD flow so that the problems faced can be resolved properly. In addition, through this research the author hopes to obtain

information related to the obstacles in implementing the AST method using DAST support tools with OWASP ZAP tools. Based on the analysis that has been done, it can be concluded that the OWASP ZAP integration process in the CICD flow can be done but needs improvements in the OWASP ZAP execution process. Improvements and corrections are needed regarding the running and listening processes on the proxy ports used..

***Keywords**: CICD, AST, DAST, OWASP ZAP, SIMPT*

1. Pendahuluan

Seiring dengan perkembangan ilmu pengetahuan dan teknologi saat ini, kebutuhan akan sistem informasi semakin tinggi. Implementasi dan pemanfaatan teknologi serta sistem informasi merambah kesegala lini kehidupan masyarakat. Ketersediaan teknologi informasi sangat membantu pada proses pemberian layanan yang cepat dan tepat. Namun dibalik kemudahan dan kecepatan pelayanan yang ditawarkan terdapat tantangan dan kendala dari segi isu – isu terkait dengan tranksasi data dan informasi yang ada dalam sistem itu sendiri, keamanan menjadi salah satu isu yang kerap kali muncul dan menjadi tantangan tersendiri dalam pengembangan teknologi dan sistem.

Salah satu kasus terkait dengan isu kebocoran data *privacy* yang terjadi pada Tokopedia, pada penelitian [1] yang melakukan analisis terkait Kebocoran Data *Privacy* pada E-Commerce yang mendapatkan hasil bahwa data *privacy* yang ada pada *E-Commerce* Tokopedia mengalami kebocoran dan telah terjadi penjualan data *privacy* pelanggan, sehingga diperlukan adanya perbaikan regulasi pada pengelolaan dan keamanan data *privacy* tersebut. Adanya kasus terkait dengan kebocoran data *privacy* pengguna ini tentunya memberikan dampak negative pada proses implementasi dan penggunaan sistem secara lebih jauh.

Berdasarkan hal tersebut maka diperlukan adanya regulasi dan uji keamanan yang lebih spesifik terhadap sistem informasi yang dibangun sebelum sistem tersebut digunakan oleh Masyarakat. Proses pengujian keamanan pada sistem dapat dilakukan secara terintegrasi pada aliran *continuous Integration Continuous Deployment (CI CD).* Seperti penelitian yang dilakukan oleh [2] dimana pada penelitian ini dilakukan integrasi uji keamanan pada aliran CI/CD untuk menguji proses integrasi alat keamanan *Dynamic Security Testing* pada *CI CD pipelines*. Penelitian tersebut memberikan rujukan kepada pengembang terkait bagaimana teknis dan hasil dari proses integrasi serta masalah – masalah yang ada yang selanjutnya akan dikaji lebih lanjut. Penelitian lain dilakukan oleh [3] yang mengkaji terkait pedoman, tantangan dan maanfaat implementasi Devops untuk pengembangan dan operasi perangkat lunak didapatkan bahwa pipeline CICD dapat diterapkan untuk meningkatkan kualitas, membangun, menguji dan rilis dengan cepat dan andal serta mengembangkan perangkat lunak modern. Pada buku *The DevOps Handbook* (Kim, et al, 2021) juga disebutkan bahwa keamanan harus menjadi bagian dari aktivitas siklus hidup pembuatan sistem, tanpa pengujian keamanan dalam siklus CICD dapat menimbulkan risiko yang cukup berbahaya. Hal ini menegaskan bahwa implementasi CICD tanpa mempertimbangkan aspek keamanan dapat menciptakan risiko yang lebih besar daripada manfaat yang diperoleh, sehingga impelemntasi keamanan pada aliran CICD sangat di perlukan untuk menjamin kualitas dan keandalan sistem.

Berdasarkan kajian yang penulis lakukan dapat disimpulkan bahwa analisis dan uji keamanan pada aliran CICD perlu dilakukan secara konsisten untuk memastikan bahwa sistem yang akan digunakan benar – benar telah siap dari segi keamanannnya. Berbagai metode dan tools ditawarkan untuk memudahkan proses pengujian keamanan pada sistem. Pada penelitian ini, penulis akan mengangkat metode *Automated Security Testing (AST)* sebagai metode pengujian sistem keamanan sistem dengan menggunakan alat dukung *Dynamic Automatic Security Test (DAST)* dengan alat bantu OWASP ZAP. Penelitian ini bertujuan untuk mengetahui bagaimana integrasi dan penerapan OWASP ZAP dalam aliran CI/CD serta mengetahui sejauhmana cakupan pengujian keamanan yang dilakukan oleh OWASP ZAP terhadap sistem yang diuji.

Proses pengujian penerapan OWASP ZAP ini akan dilakukan pada Sistem keuangan yang merupakan bagian dari Sistem Informasi Perguruan Tinggi (SIMPT) yang dikembangkan oleh cv.xyz. Penelitian ini diharapkan dapat memberikan solusi yang baik kepada perusahaan mengenai bagaimana proses penerapan AST pada aliran CICD sehingga masalah yang dihadapi dapat terselesaikan dengan baik. Selain itu melalui penelitian ini penulis berharap bisa mendapatkan informasi terkait kendala pada penerapan metode AST dengan menggunakan alat dukung DAST dengan tools OWASP ZAP.

2. Metode Penelitian

Penelitian ini menggunakan metode Network Development Life Cycle dimana pada penelitian ini hanya diambil 3 tahapan yaitu Tahapan analisis, tahapan desain dan tahapan simulasi prototyping. Secara rinci berikut detail untuk tiap tahapan yang dilakukan.

2.1. Tahapan Analisis

Pada tahapan ini dilakukan analisa mengenai kebutuhan sistem yang diperlukan pada proses merancang dan menerapkan *automated security testing* Pada aliran CICD. Berdasarkan analisis dari sistem yang akan diuji dan diimplementasikan berikut analisa kebutuhan perangkat keras dan perangkat lunak pada penelitian ini.

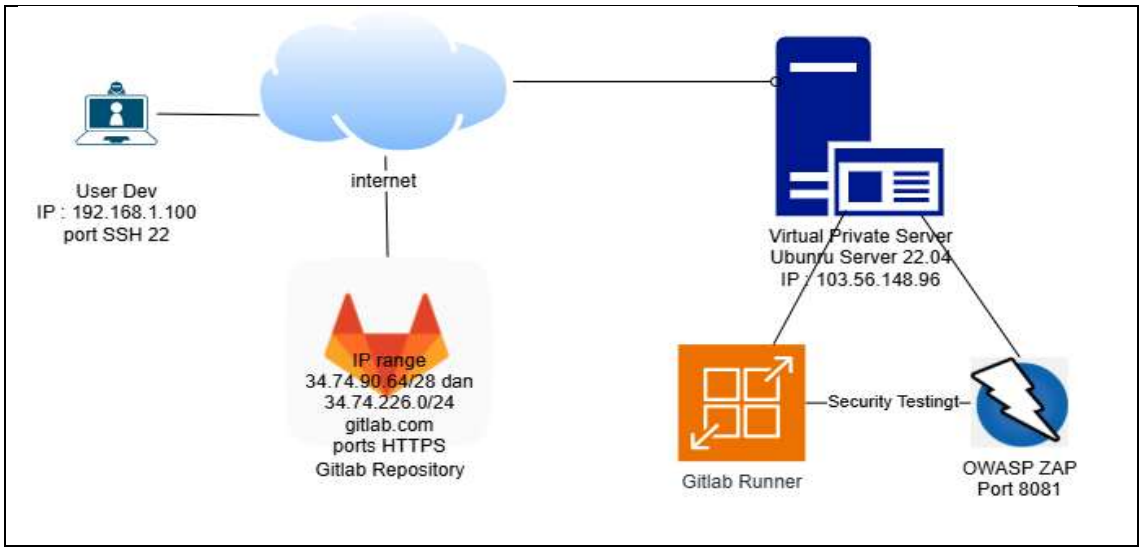
Berdasarkan kajian yang telah dilakukan metode pengujian yang akan penulis implementasikan ialah metode Autometic Security Testing dengan tools Dynamic Security Testing menggunakan OWASP ZAP. Tools ini merupakan alat pengujian keamanan yang berjalan diatas java runtime dan dinilai mampu melakukan pengujian yang lebih baik terhadap sistem.

2.2. Tahapan Desain

Pada tahapan ini dilakukan rancangan topologi, rancangan alur sistem yang akan digunakan pada penerapan Automated Security Testing pada aliran CICD. Pada tahapan ini penulis membahas secara detail rancangan dan model implementasi yang dilakukan pada proses integrasi Automated Security Testing dimana pada uji ini digunakan tools OWASP ZAP sebagai alat untuk melakukan uji keamanan yang dimaksud. Langkah yaang dilakukan pada tahapan desain ini ialah sebagai berikut :

2.2.1. Rancangan Jaringan

Berdasarkan analisa yang dilakukan penulis membuat rancangan topologi yang melibatkan penggunaan Virtual Private Server (VPS), reposiroty, gitlab, developper serta OWASP ZAP sebagai alat otomasi uji keamanan. Rancangan topologi seperti dilihat pada gambar 1.



Gambar 1. Diagram Topologi jaringan uji

Diagram topologi arsitektur pipeline CI/CD pada gambar 1 memiliki beberapa komponen diantaranya :

- Developer dengan IP local 192.168.1.100 terhubung dengan Source Code Repository github melalui internet dengan menggunakan HTTPS/SSH (port 443, 22).
- Source Code Repository (gitlab.com) terhubung dengan developper dan VPS Server melalui internet dengan menggunakan SSH
- Virtual Private Server (VPS) Sebagai Server Stagging yang merupakan Server fisik/VM yang menjalankan GitLab Runner, web server (Apache/Nginx), PHP, dan OWASP ZAP native.
- Gitlab Runner berjalan pada VPS Server sebagai mesin eksekusi pipeline yang berjalan di Ubuntu Server 22.04, dengan runner tipe shell tanpa docker.
- OWASP ZAP berjalan didalam Server VPS Stagging. Tools ini merupakan aplikasi keamanan berbasis Java yang berjalan secara native di runner, melakukan scanning keamanan

2.2.2. Rancangan Pengalamatan IP

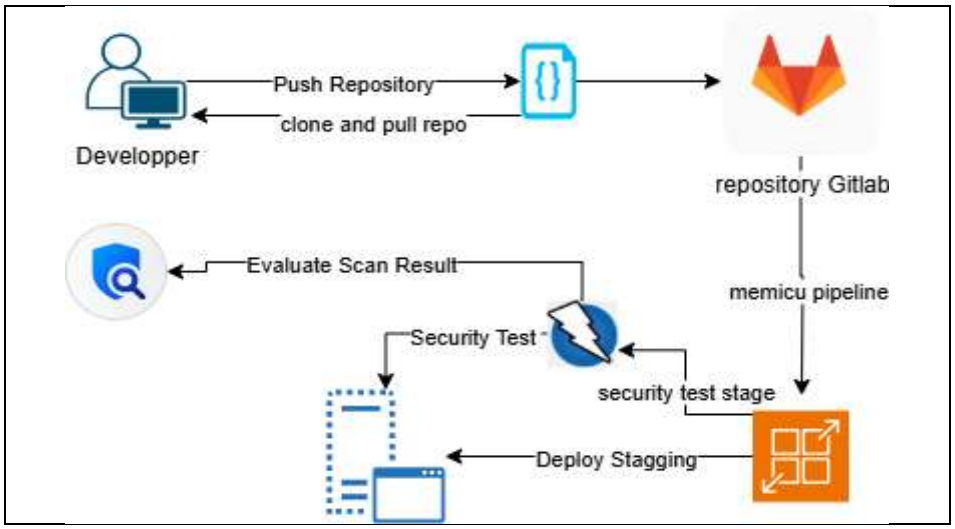
Pada tahapan ini penulis akan memetakan pengalamatan IP Address pda lingkungan sistem topologi ujicoba yang telah dibuat. Desain IP Address melibatkan penentuan pengelolaan alamat Ip yang digunakan di dalam sistem yang akan di ujicobakan. Adapun pengalamatan IP yang ada pada lingkungan sistem ujicoba yang akan digunakan dapat dilihat pada tabel 1.

Tabel 1. Pengalamatan IP

Nama Perangkat	Ip Address	Subnet	Keterangan
PC/Laptop	192.168.1.100	255.255.255.0	Perangkat Developper
Gitlab Repository	34.74.90.64/28 dan 34.74.226.0/24	-	Repository code
VPS	103.56.148.96	-	Server Stagging
Gitlab Runner	-	-	Eksekutor pipeline
OWASP ZAP	103.56.148.96 port 8081	-	Tools pengujian keamanan

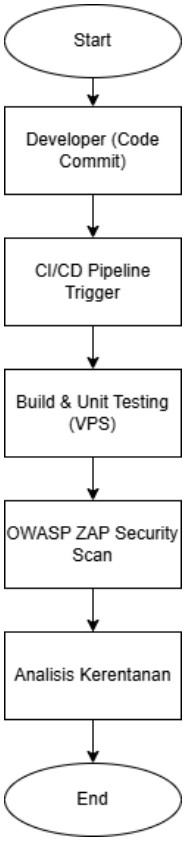
2.3. Rancangan Sistem

Pada tahapan ini penulis membuat rancangan sistem dan alur penerapan sistem Automated Security Testing (AST) dengan dynamic security test menggunakan alat OWASp ZAP. Rancangan sistem pengujian ini dapat dilihat pada gambar berikut ini :



Gambar 2. Rancangan Sistem Pengujian

Berdasarkan rancangan sistem diatas proses penerapan AST dan pengujian terdiri atas lima komponen utama yaitu Developper, gitlab repository, gitlab runner, OWASP ZAP dan VPS. Masing – masing komponen memiliki peranannya masing – masing dalam menjalankan proses siklus hidup yang telah dibuat dalam perintah gitlab-ci.yml. Secara sederhana rancangan sistem ini dapat digambarkan dalam bentuk diagram alur seperti yang tampak pada gambar 3, pada alur ini terdapat lima tahapan dengan melibatkan komponen – komponen yang telah yang telah digambarkan pada rancangan sistem uji coba CI/CD diatas tahapan ini meliputi code commit, CI/ CD pipeline trigger, Build dan Unit Testing (VPS), OWASP ZAP Security Scan dan analisis kerentanan.



Gambar 3. Alur proses Penerapan AST

3. Hasil dan Pembahasan

Berdasarkan hasil uji coba yang dilakukan pada proses penerapan *automated security test* pada CICD dengan menggunakan alat uji OWASP ZAP. Tahapan eksekusi CICD dilakukan dalam 3 stage yaitu test, deploy dan security test. .

3.1. Hasil Uji Coba

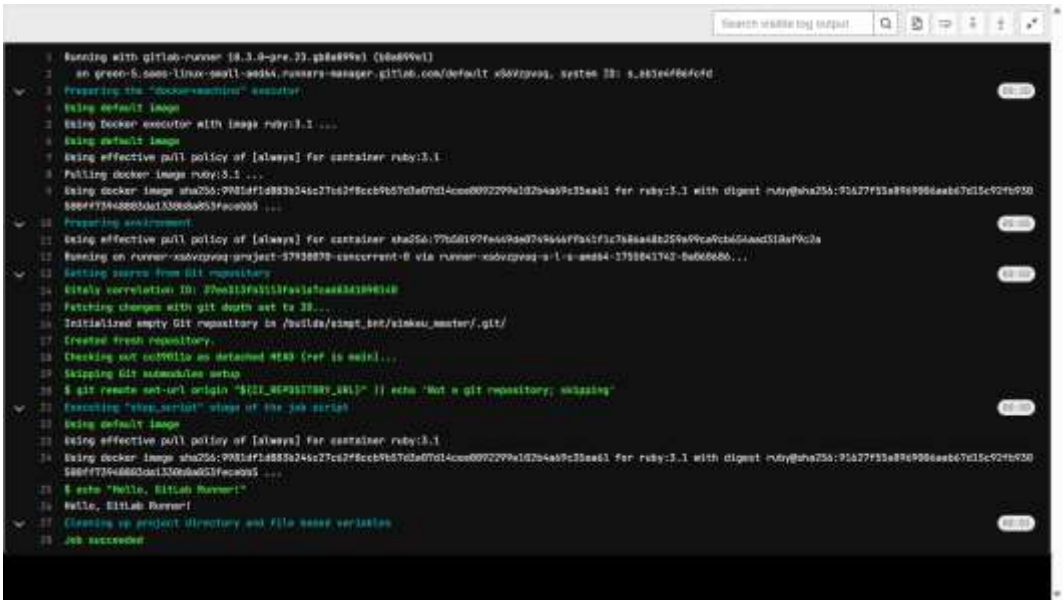
Tahapan awal yang dilakukan pada proses uji coba ini adalah pengujian koneksi gitlab runner. Tahapan test dilakukan untuk mengetahui apakah gitlab runner telah berjalan ataukah tidak, pengujian dilakukan dengan menggunakan seperti pada script berikut :

```
Stage : test
test:
       stage: test
       script:
       echo "Hello, Gitlab Runner!"
```

Setelah diinstruksi di push maka sistem gitlab runner akan menerima instrukri untuk menjalankan perintah yang telah diberikan, dan menjalankan *job test* seperti hasil pada gambar 4. Berdasarkan hasil yang terlihat proses pengujian gitlab runner berhasil (*passed)* yang menandakan bahwa gitlab runner telah berjalan pada repository project

Gambar 4. hasil eksekusi stage test

Lebih lanjut detail hasil pengujian dapat dilihat pada gambar 5.

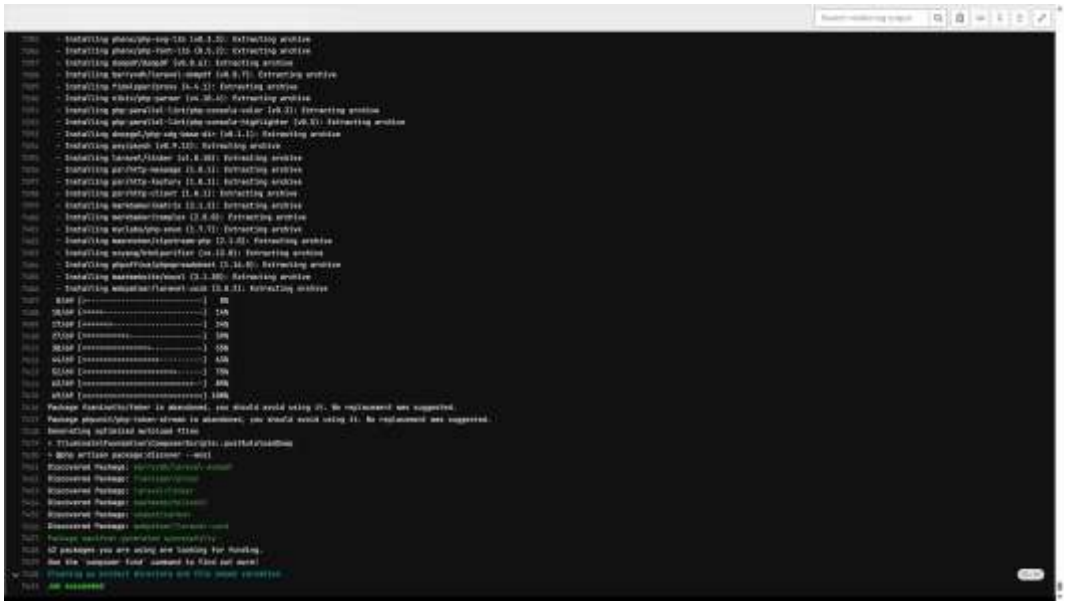


Gambar 5. hasil eksekusi stage test

Selanjutnya dilakukan pengujian deployment sistem untuk memastikan project dapat dikirim dengan baik kedalam server script pengujian pada tahapan ini dapat dilihat pada listing code berikut ini :

```
deploy :
  stage: deploy
  image: alpine:latest
  before_script:
    - apk add --no-cache openssh-client rsync bash sshpass
     # Menambahkan host key server ke known_hosts
    - mkdir -p ~/.ssh
    - ssh-keyscan -H $DEPLOY_IP >> ~/.ssh/known_hosts
      script:
    - sshpass -p "$DEPLOY_PASS" rsync -avz --delete --
exclude='.git/'   --exclude='storage/' --exclude='.env' ./
      $DEPLOY_USER@$DEPLOY_IP:$DEPLOY_PATH
   - sshpass -p "$DEPLOY_PASS" ssh -o StrictHostKeyChecking=no
      $DEPLOY_USER@$DEPLOY_IP "cd $DEPLOY_PATH && composer install
-     -no-dev"
  only:
    - main
  when: on_success
```

Code ini merupakan job yang berjalan pada stage *deploy* menggunakan image Docker *apline:Latest.* Pada tahapan ini dilakukan instalasi *tool SSH/rsync* menambahkan host *key server known_hosts*. Selanjutnya dilakukan proses penyalinan project ke server remote dengan *rsync* melalui ssh. Tahap akhir dilakukan proses instalasi dengan menggunakan peritah *composer install* pada folder yang telah disiapkan.



Gambar 6. hasil running stage deploy

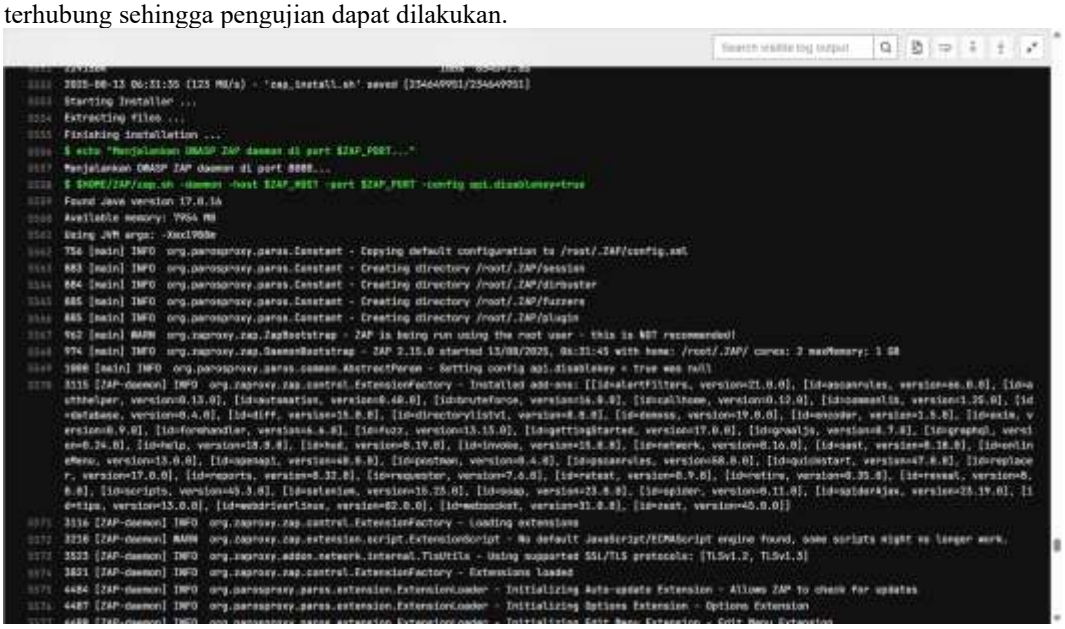
Berdasarkan hasil running stage yang ada pada gambar 3.3 dapat dilihat bahwa hasil deployment berjalan dengan lancar dan berhasil dilakukan proses penyalinan data ke server dan proses instalasi project pada folder yang telah ditetapkan.

Tahapan selanjutnya dilakukan pengujian stage *security-test* pada tahapan ini dilakukan pengujian code untuk proses pengujian keamanan. Pengujian ini dilakukan dengan pengecekan file instalasi OWASPZAP, proses ini dilakukan untuk memastikan bahwa perangkat sudah terinstal pada server ataukah tidak. Jika library OWASPZAP belum tersedia maka akan dilakukan instalasi OWASP ZAP pada server. Listing code lengkap dapat dilihat pada code dibawah ini :

```
security-test:
  stage: security-test
  before_script:
    # Install dependencies
    - apt update -y
    - apt install -y default-jdk python3-pip wget
    - export PATH=$PATH:$HOME/.local/bin

    # Install OWASP ZAP jika belum ada
    - if [ ! -d "$HOME/ZAP" ]; then
        wget
      https://github.com/zaproxy/zaproxy/releases/download/v2.15
      .0/ZAP_2_15_0_unix.sh -O zap_install.sh &&
        chmod +x zap_install.sh &&
        ./zap_install.sh -q -dir $HOME/ZAP;
      fi
  script:
    # Jalankan ZAP daemon di background di port 8080
    - echo "Menjalankan OWASP ZAP daemon di port $ZAP_PORT..."
    - $HOME/ZAP/zap.sh -daemon -host $ZAP_HOST -port $ZAP_PORT -
      config api.disablekey=true
    - sleep 10

    # Tunggu sampai port terbuka
    - echo "Menunggu port $ZAP_PORT terbuka..."
    - for i in {1..30}; do
        if nc -z $ZAP_HOST $ZAP_PORT; then
          echo "Port $ZAP_PORT sudah terbuka.";
```

```
          break;
        fi;
        echo "Menunggu ZAP... percobaan $i/30";
        sleep 2;
      done

    # Cek koneksi ke API ZAP
    - echo "Mengecek koneksi API ZAP..."
    - curl --fail http://$ZAP_HOST:$ZAP_PORT || (echo "Gagal
      menghubungi ZAP API" && exit 1)
    # Scan target
    - zap-cli --zap-url http://$ZAP_HOST:$ZAP_PORT open-url $
      TARGET_URL
    - zap-cli --zap-url http://$ZAP_HOST:$ZAP_PORT spider
      $TARGET_URL
    - zap-cli --zap-url http://$ZAP_HOST:$ZAP_PORT active-scan -
      r $TARGET_URL

    # Generate report HTML
    - zap-cli --zap-url http://$ZAP_HOST:$ZAP_PORT report -o
      zap_report.html -f html
      artifacts:
    paths:
      - zap_report.html
    when: always
    expire_in: 1 week
  only:
    - main
  when: on_success
```

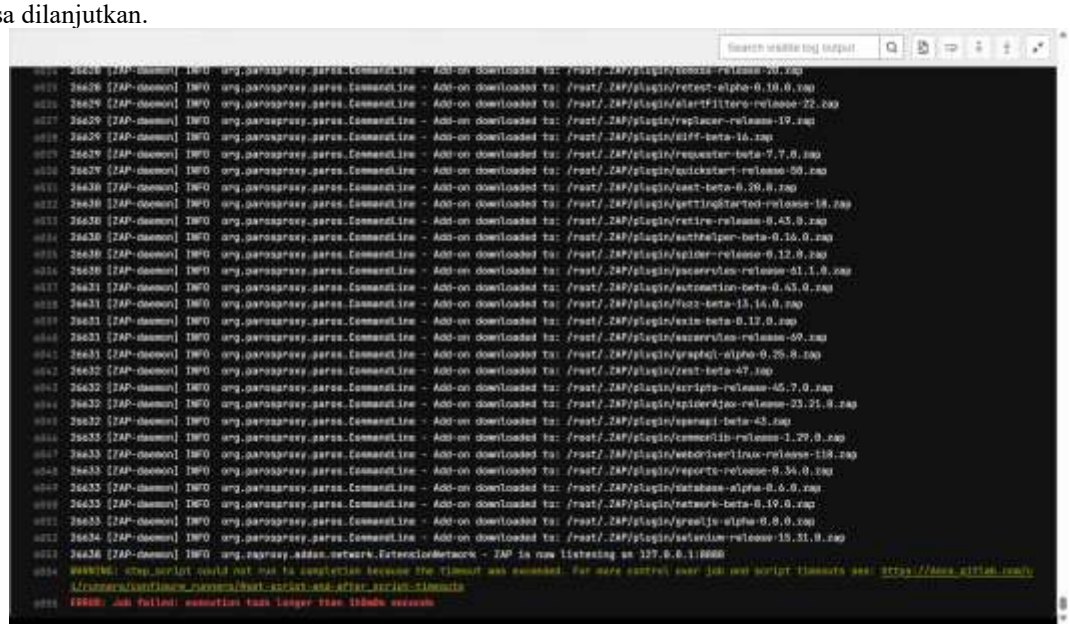
Pada gambar 7 dapat dilihat bahwa proses instalasi dapat berjalan dengan baik yang selanjutnya diikuti dengan proses untuk menjalankan daemon OWASP ZAP pada port 8080. Proses listening ini dilakukan untuk memastikan bahwa OWASP ZAP dapat berjalan dan komunikasi dengan port proxy dapat terhubung sehingga pengujian dapat dilakukan.



Gambar 7. hasil running stage security-test

3.2. Pembahasan

Berdasarkan hasil pengujian yang telah dilakukan pada dasarnya proses awal aliran data pada jaringan CICD dapat berjalan dengan baik namun terdapat kendala pada saat proses listening OWASP ZAP daemon port 8080. Proses ini membutuhkan waktu yang lama dan setelah waktu runtime berjalan selama 60 menit proses listening dihentikan karena tdak dapat terjadi koneksi antara dengan port proxy yang telah dilakukan. Pada gambar 8 dapat dilihat bahwa proses listening terlalu lama sehingga proses tdak bisa dilanjutkan.



Gambar 8. hasil running stage pada proses listening OWASPZAP daemon port 8080

Pada tahapan ini dapat diketahui bahwa masih ada kekeliruan pada proses eksekusi OWASPZAP pada server dimana port 8080 tdak bisa terbuka. Dari segi integrasi sudah berjalan dengan lancar dan sesuai namun perlu dilakukan perbaikan pada open proxy agar komunikasi port proxy dapat berjalan dengan lancar dan uji proses scanning dapat dilakukan

4. Kesimpulan

Berdasarkan analisis yang telah dilakukan dapat ditarik kesimpulan bahwa proses integrasi OWASP ZAP pada aliran CICD dapat dilakukan namun perlu adanya perbaikan pada proses eksekusi OWASP ZAP. Perlu dilakukan perbaikan dan koreksi kembali terkait proses running dan listening pada port prooxy yang digunakan.

Daftar Pustaka

[1] S. Sukma Ayu, M. Irwan, and P. Nasution, “Analisis Kebocoran Data Privacy Pada E-Commerce Tokopedia,” vol. 2, no. 3, 2023, [Online]. Available: http://jurnal.jomparnd.com/index.php/jk

[2] T. Rangnau, R. V. Buijtenen, F. Fransen, and F. Turkmen, “Continuous Security Testing: A Case Study on Integrating Dynamic Security Testing Tools in CI/CD Pipelines,” in *Proceedings - 2020 IEEE 24th International Enterprise Distributed Object Computing Conference, EDOC 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 145–154. doi: 10.1109/EDOC49727.2020.00026.

[3] Meinahen B. Bansoma, “Pedoman, Tantangan, Dan Manfaat Implementasi Devops Untuk Pengembangan Dan Operasi Perangkat Lunak: Tinjauan Literatur Sistematis,” *TIMOR CERDAS – J. Teknol. Informasi, Manaj. Komput. dan Rekayasa Sist. Cerdas*, vol. 2, no. 1, 2024.

[4] Papagiannaki, K., Taft, N., Zhang, Z. L., & Diot, C. (2021). Long-term forecasting of internet backbone traffic. IEEE Transactions on Neural Networks and Learning Systems, 32(3), 1110-1123.

[5] Mockapetris, P. (2021). Domain names - concepts and facilities. RFC 1034. Internet Engineering Task Force.

[6] Metcalfe, R. M., & Boggs, D. R. (2022). Ethernet: Distributed packet switching for local computer networks. Communications of the ACM, 65(7), 63-69.

[7] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., & Turner, J. (2021). OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 51(2), 69-74.

[8] Galih Sundayana, M., & Lucia Kharisma, I. (2023). Rancang Bangun Layanan Private cloud Berbasis Infrastructure as a Service Menggunakan OpenStack dengan Metode Network Development Life Cycle(NDLC). Media Online), 4(1), 252–262. https://doi.org/10.30865/klik.v4i1.1001

[9] Hinden, R., & Deering, S. (2024). IP version 6 addressing architecture. RFC 4291bis. Internet Engineering Task Force.

[10] Hoffman, P., & McManus, P. (2023). DNS queries over HTTPS (DoH). RFC 8484. Internet Engineering Task Force.

[11] **I**BM Corporation. (2024). AI for network operations: Transforming IT infrastructure management. https://www.ibm.com/cloud/learn/ai-for-network-operations

[12] IEEE. (2023). IEEE 802.11 wireless LAN medium access control (MAC) and physical layer (PHY) specifications. IEEE Standard 802.11-2020

[13] Putu Hariyadi, I., Azhar, R., Santoso, H., Marzuki, K., & Made Yadi Dharma. (n.d.). IMPLEMENTASI CONTINUOUS INTEGRATION/CONTINUOUS DEPLOYMENT UNTUK MENGOTOMATISASI MANAJEMEN. In Desember (Vol. 4, Issue 2).

[14] Cisco Systems. (2023). Intent-based networking: The next evolution of networking. https://www.cisco.com/c/en/us/solutions/intent-based-networking.html

[15] Cisco Systems. (2024). Annual cybersecurity report 2024. https://www.cisco.com/c/en/us/products/security/security-reports.html