

Pengujian IaC Berbasis *DevOps* dan *Ansible* Menggunakan Metode *Black Box Testing*

I Putu Agus Eka Pratama¹, Putu Bayu Suarnata Wahyu Putra²

^{1,2}Department of Information Technology, Faculty of Engineering, Udayana University, Indonesia

Article Info

Article history:

Received Feb 16, 2022

Revised Jun 06, 2022

Accepted Jul 12, 2022

Keywords:

Ansible

Black Box Testing

DevOps

functionality

Infrastructure as Code (IaC)

ABSTRACT

The development of information technology, one of which is the most dominant is the development of web-based applications and services, always followed by an increase in the need for devices and computing resources for services on computer networks. This requires cost and time for the configuration and development process, so there is a need for automation and a combination of development and operational processes. DevOps-based Infrastructure as Code (IaC) using Ansible, is proposed as a solution to this problem by combining development and operational processes, configuring the host node, and running the Ansible playbook from the host server. The Black Box Testing method was chosen as the testing method in this study, to determine the functionality of the DevOps-based IaC system using Ansible. There are three stages of testing, namely: testing the web server's status using HTTPD on the host node, testing web content using Ansible, and testing the appearance of web content according to the configuration. The test results show that all three stages of the test were successful, so this research can prove that DevOps-based IaC using Ansible can facilitate the development of web-based applications and services.

Copyright © 2022 Universitas Indraprasta PGRI.
All rights reserved.

Corresponding Author:

I Putu Agus Eka Pratama

Department of Information Technology, Faculty of Engineering

Udayana University

Jl. Raya Kampus Unud, Jimbaran, Badung, Bali.

Email: eka.pratama@unud.ac.id

1. PENDAHULUAN

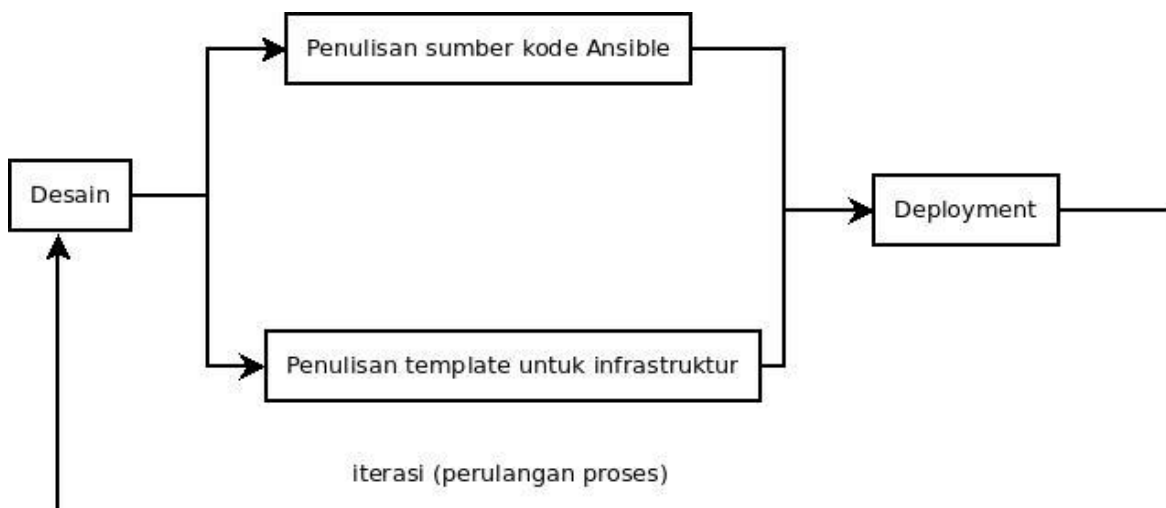
Perkembangan Teknologi Informasi (TI) yang makin pesat, diikuti dengan peningkatan jumlah dan kualitas layanan di jaringan komputer, terutama *internet*. Salah satu bentuk aplikasi dan layanan di *internet* tersebut adalah *website* beserta *web server*. Perkembangan aplikasi dan layanan berbasis *web*, turut mempengaruhi kebutuhan akan ketersediaan perangkat dan sumber daya komputasi yang makin besar, untuk mendukung ketersediaan layanan. Peningkatan jumlah perangkat dan sumber daya komputasi, akan membebani perusahaan penyedia layanan dari sisi biaya dan waktu untuk development dan konfigurasi.

Umumnya, perusahaan dan organisasi seringkali melakukan pengembangan sebuah *software* dengan cara menggabungkan tim pengembangan (*development*) dengan tim operasional (*operation*). Kedua tim ini saling melengkapi, di mana tim *development* bertugas menciptakan fungsi baru dan memperbaiki kelemahan (*bug*) pada *software*, sedangkan tim *operational* bertugas membuat infrastruktur pendukung yang handal, sehingga menjadikan *software* yang telah dikembangkan oleh tim *development* dapat berjalan dengan baik dan stabil[1]. Berdasarkan kepada hal ini, muncullah *Development and Operation (DevOps)* sebagai sebuah metode yang menggabungkan sisi pengembangan dan operasional, sehingga pengembangan dan konfigurasi pada server dan aplikasi dapat dilakukan secara lebih cepat, efektif, dan efisien, melalui otomatisasi dan penyediaan blueprint konfigurasi yang dapat diterapkan pada *server*[2].

Metode *DevOps* menawarkan salah satu model solusi untuk menyelesaikan permasalahan terkait kebutuhan akan perangkat dan sumber daya komputasi yang besar, dalam bentuk *Infrastructure as Code (IaC)*. IaC mengutamakan adanya otomatisasi, dengan cara menyediakan media berupa *script* kode konfigurasi yang

dapat dijalankan pada *server*, untuk dapat langsung menerapkan konfigurasi secara langsung[3]. Terdapat sejumlah pilihan perangkat lunak untuk mengimplementasikan IaC, salah satunya adalah *Ansible*. *Ansible* merupakan *provisioning tool* dari *Red Hat Linux* untuk membantu pencatatan setiap proses *deployment* dan konfigurasi berulang terhadap sejumlah *server* dan di jaringan untuk mengotomatisasi *cloud provisioning*, manajemen konfigurasi, penerapan aplikasi, *orchestration* dan kebutuhan TI lainnya[4].

Ansible terdiri atas tiga bagian, yaitu: 1.)*Ansible Inventory* sebagai file penyimpanan *host* yang akan menjalankan *tasks* yang telah ditetapkan sebelumnya, 2.)*Ansible Playbook* sebagai *file* tempat menyimpan *tasks* yang akan dijalankan oleh *host* pada *inventory*, dan 3.)*Ansible Tasks* sebagai perintah atau aktifitas yang dijalankan oleh *hosts*. Di dalam penelitian ini, implementasi IaC berbasis *Ansible* dilakukan dari tahapan desain hingga *deployment*, dengan melibatkan sejumlah proses dan tahapan di dalamnya. Gambar 1. di bawah ini menunjukkan bagan proses dari tahapan-tahapan pada IaC berbasis *Ansible*:



Gambar 1. Bagan IaC berbasis *Ansible*

Berdasarkan kepada skema bagan proses yang ditampilkan pada Gambar 1. di atas, urutan proses dimulai dari tahapan desain untuk IaC. Desain menjadi tahapan dasar di dalam proses implementasi IaC berbasis *Ansible* menggunakan metode *DevOps*. Pada tahapan desain, dibuatkan bagan, arsitektur jaringan, atau desain dari sistem secara keseluruhan. Tahapan selanjutnya adalah penulisan sumber kode *Ansible*, sebagai bentuk implementasi dari desain yang telah dibuat. Penulisan sumber kode pada *Ansible* juga diikuti dengan penulisan *template* untuk infrastruktur yang telah didesain sebelumnya. Berdasarkan hasil implementasi berupa penulisan sumber kode *Ansible* dan penulisan *template* untuk infrastruktur, maka kemudian akan dilanjutkan ke tahapan terakhir berupa proses *deployment*. Pada tahapan *deployment* ini, dilakukan proses *deployment* terhadap sistem yang telah diimplementasikan. Apabila terdapat kesalahan ataupun kekurangan, maka proses dapat diulang kembali ke tahapan desain.

Terdapat sejumlah penelitian sebelumnya yang telah dilakukan oleh para peneliti mengenai *DevOps* dan IaC berbasis *Ansible* serta metode pengujian yang digunakan di dalamnya, yang menjadi *state of the art* dari penelitian ini. Penelitian pertama membahas mengenai konsep dari IaC sebagai sebuah tool *DevOps* berupa *script* kode konfigurasi untuk dijalankan pada *server* baru secara otomatis, dengan menggunakan metode *Topology and Orchestration Specification for Cloud Applications (TOSCA)*[5]. Penelitian kedua mengemukakan tentang implementasi IaC pada industri di mana semua otomatisasi *DevOps* menggunakan kode infrastruktur yang berkaitan dengan IaC dengan model pengujian studi literatur[6]. Penelitian ketiga memaparkan tentang format bahasa baru pada IaC dari format data JSON dan YAML, dengan mengusulkan sebuah *query language* bernama *Biznet Query Language (BQL)*, untuk mengelola IaC agar dapat melakukan CRUD secara lebih interaktif dengan menggunakan metode *Read Eval Print Loop (REPL)*[7]. Penelitian keempat membahas mengenai penilaian terhadap kode program dari IaC melalui percobaan terhadap kode pada IaC beserta penilaian terhadap kode berbasis metode *literature review*, dengan hasil bahwa IaC agnostik terhadap teknologi terapan[8].

Penelitian kelima membahas mengenai penerapan IaC menggunakan *Ansible* dalam bentuk modul imperatif dengan operasional *idempotent* yang dapat dijalankan berkali-kali tanpa mengubah hasil yang dikeluarkan dengan metode pengujian empiris[9]. Penelitian keenam menguraikan tentang implementasi *automation* pada manajemen *server* dengan menggunakan *Ansible* berbasis *DevOps*, untuk kemudian diujikan menggunakan metode *experimental*[10]. Penelitian ketujuh menguraikan tentang pemanfaatan *Ansible* untuk otomatisasi perangkat pada jaringan komputer serta kemudahan konfigurasi pada studi kasus laboratorium

komputer dengan menggunakan metode *Network Development Life Cycle (NDLC)*[11]. Penelitian kedelapan mengemukakan tentang analisis perbandingan manajemen konfigurasi di dalam jaringan *Cloud Computing* menggunakan konsep *DevOps* berbasis *Ansible* dan *Shell Script* pada studi kasus *AWS Cloud*, dengan metode pengujian *experimental*[12]. Penelitian kesembilan mengemukakan tentang *automation provisioning* berbasis *DevOps* pada studi kasus penanganan *server website* dengan menggunakan kombinasi *Ansible* dan *Vagrant* dan menggunakan metode pengujian *experimental*[13]. Penelitian kesepuluh mengemukakan tentang proses otomatisasi dan konfigurasi pada proses *routing* di dalam jaringan menggunakan perangkat *router* dengan memanfaatkan *Ansible* dengan metode pengujian *Research and Development (RnD)*[14]. Penelitian kesebelas memaparkan mengenai pemanfaatan metode *Action Research* di dalam melakukan pengujian terhadap IaC pada *IaaS Cloud Computing*, yang diimplementasikan menggunakan *Proxmox* dan *Xen Server*[15]. Penelitian kedua belas menguraikan tentang Implementasi *web server* berbasis IaC menggunakan *Terraform* dengan metode pengujian *DevStack*[16]. Penelitian ketiga belas menguraikan tentang implementasi IaC pada *Infrastructure as a Service (IaaS) Cloud Computing* dengan melakukan pengujian berbasis *Load Balancing*[17].

Berdasarkan kepada penelitian-penelitian tersebut, terdapat salah satu hal yang belum dilakukan oleh para peneliti tersebut, yaitu melakukan pengujian fungsionalitas dari sistem IaC berbasis *Ansible* yang diimplementasi. Untuk itu, di dalam penelitian ini diusulkan untuk penggunaan metode pengujian *Black Box Testing*. Hal ini bertujuan untuk dapat mengetahui fungsionalitas sistem di dalamnya pasca implementasi. Rumusan masalah dalam bentuk pertanyaan penelitian adalah: a.)Bagaimana cara melakukan pengujian IaC berbasis *DevOps Ansible* menggunakan metode *Black Box Testing*? b.)Apa saja langkah-langkah pengujian yang dilakukan untuk menguji sistem IaC berbasis *Ansible* tersebut?

2. METODE

2.1. Bahan dan Alat

Untuk menunjang implementasi penelitian ini, digunakan sejumlah hardware dan software pendukung. Hardware yang digunakan di dalam penelitian ini adalah sebuah notebook dengan spesifikasi processor Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz (8 CPUs) 2.0 GHz, RAM 8GB, dan harddisk 512GB. Software yang digunakan di dalam penelitian ini adalah sistem operasi Linux Ubuntu, Linux CentOS 7, Windows 10, VMWare, dan Ansible. Selain itu juga digunakan modem internet untuk koneksi ke dalam jaringan.

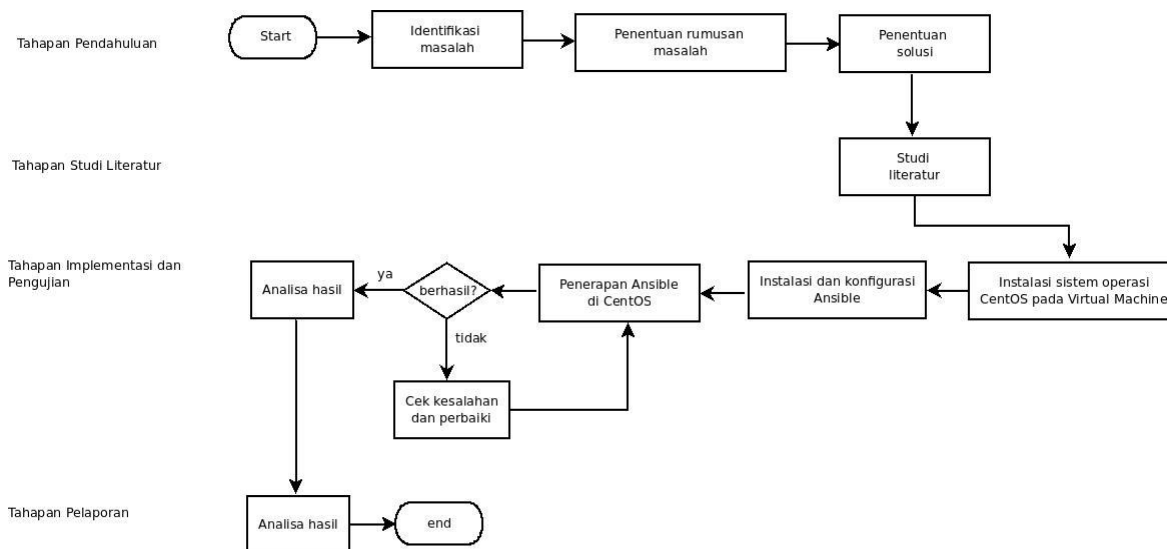
2.2. Metode Pengujian *BlackBox Testing*

Metode pengujian yang digunakan penelitian ini adalah *Black Box Testing*. *Black Box Testing* merupakan sebuah model dan metode pengujian yang dilakukan terhadap sebuah sistem dan software untuk mengetahui fungsionalitas yang terjadi di dalamnya, tanpa perlu pemahaman oleh penguji dan pengembang mengenai sumber kode dan proses yang terjadi di dalamnya[18][19]. Di dalam metode pengujian *Black Box Testing*, peneliti hanya perlu memasukkan satu atau beberapa buah perintah, kemudian mengamati luaran dari perintah yang dijalankan tersebut.

Di dalam penelitian ini, *Black Box Testing* dilakukan pasca implementasi IaC berbasis *Ansible*. Disediakan sejumlah tahapan pengujian beserta dengan urutan langkah yang dilakukan, untuk kemudian diamati hasil dari setiap langkah pengujian tersebut, kemudian dicatat dan didokumentasikan. Dari hasil pengujian *Black Box Testing*, akan dapat diperoleh kesimpulan mengenai jalannya fungsionalitas sistem sesuai dengan yang diharapkan.

2.3. Flowchart Penelitian

Flowchart penelitian merupakan bagan yang menggambarkan alur dan urutan langkah penelitian yang dilakukan di dalam penelitian, yang disesuaikan dengan metode dan metodologi penelitian, metode pengujian, dan skenario penelitian. Sebuah *flowchart* dapat terdiri atas satu buah tahapan atau lebih dari satu buah tahapan di dalam penelitian, sesuai dengan kebutuhan dan studi kasus. Di dalam penelitian ini, terdapat empat tahapan penelitian yang dilakukan, yang terdiri atas: 1) Tahap pendahuluan, 2)Tahapan studi literatur, 3.Tahapan implementasi dan pengujian, serta 4.)Tahapan pelaporan. *Flowchart* penelitian ditampilkan pada Gambar 2. di bawah ini:



Gambar 2. Flowchart penelitian

Berdasarkan kepada Gambar 2. di atas, setiap tahapan memiliki proses masing-masing, yang berkelanjutan dengan proses pada setiap tahapan lainnya. Pada tahapan pendahuluan, dimulai dari proses identifikasi masalah, penentuan rumusan masalah, dan penentuan solusi. Identifikasi masalah dilakukan dengan cara mengumpulkan data di lapangan terkait dengan studi kasus perlunya sebuah sistem untuk mengatasi permasalahan kebutuhan akan sumber daya komputasi yang besar, dengan memanfaatkan konsep DevOps dan otomatisasi. Pada penentuan rumusan masalah, dilakukan dalam bentuk pertanyaan-pertanyaan penelitian terkait dengan permasalahan. Pada penentuan solusi diajukan solusi berupa implementasi IaC dan pengujian sistem menggunakan metode Black Box Testing. Pada tahapan studi literatur, dilakukan kajian literatur dari sejumlah referensi terkait dengan penelitian, yang berasal dari paper pada jurnal dan proceeding. Pada tahapan implementasi dan pengujian, dilakukan implementasi IaC menggunakan Ansible pada Linux CentOS serta pengujian menggunakan metode Black Box Testing terhadap fungsionalitas sistem IaC yang dibangun, dengan studi kasus manajemen web server dan pengembangan aplikasi berbasis web. Pada tahapan pelaporan, dilakukan dokumentasi hasil implementasi dan pengujian ke dalam bentuk paper publikasi.

3. HASIL DAN PEMBAHASAN

3.1. Implementasi Sistem

Sistem operasi Linux CentOS diinstallkan pada komputer server, kemudian dilanjutkan dengan melakukan konfigurasi pada server dan node client. Kemudian dilakukan penentuan IP Address server dan node, konfigurasi SSH, dan konfigurasi Ansible. File konfigurasi dilakukan melalui perintah `#nano /etc/sysconfig/network-scripts/ifcfg-ens33` dan perintah `#nano /etc/sysconfig/network-scripts/ifcfg-ens34`. Pengalamanan jaringan di dalam penelitian ini menggunakan DHCP sehingga IP Address yang diperoleh untuk server dan node ditentukan secara otomatis oleh sistem yaitu 192.168.10.1/24 dan 192.168.10.2. Kemudian dilakukan konfigurasi SSH untuk persiapan di dalam mengimplementasikan IaC, dengan menggunakan perintah `#nano /etc/ssh/sshd_config`

Untuk konfigurasi Ansible di dalam penelitian ini meliputi konfigurasi Ansible pada CentOS, konfigurasi inventory Ansible, dan konfigurasi playbook Ansible. konfigurasi Ansible yang dilakukan pada CentOS untuk memasang paket tambahan Extra Packages for Enterprise Linux Repository (EPEL), menggunakan perintah `#yum -y install epel-repo` dan `#yum -y install ansible`. Konfigurasi inventory Ansible berfungsi untuk menentukan node-node yang akan menerapkan tasks yang terdapat pada playbook. Konfigurasi ini dilakukan pada file `/etc/ansible/hosts` yang memuat cluster host bernama `ansible_clients`, yang terdiri dari host node dengan IP Address 192.168.10.2 beserta akun SSH.

Konfigurasi Playbook dilakukan pada file `playbook.yml` menggunakan perintah `#nano playbook.yml`. Pengecekan konfigurasi Playbook dilakukan dengan perintah `#ansible-playbook playbook.yml --syntax-check`. Pada penganturan dilakukan penambahan perintah `export ANSIBLE_HOST_KEY_CHECKING=FALSE`, di mana perintah ini berfungsi untuk mencegah penggunaan key access ketika menggunakan SSH serta langsung menjalankan SSH dengan user dan password yang telah ditentukan pada inventory. Selanjutnya dilakukan perintah `ansible-playbook playbook.yml` untuk dapat menjalankan playbook. Gambar 3. di bawah ini menunjukkan perintah menjalankan playbook beserta status yang ditampilkan:

```
[root@192 ~]# export ANSIBLE_HOST_KEY_CHECKING=False
[root@192 ~]# ansible-playbook playbook.yml

PLAY [play book] *****

TASK [Gathering Facts] *****
ok: [192.168.10.2]

TASK [install httpd] *****
changed: [192.168.10.2]

TASK [run httpd] *****
changed: [192.168.10.2]

TASK [create webpage content] *****
changed: [192.168.10.2]

PLAY RECAP *****
192.168.10.2 : ok=4    changed=3    unreachable=0    failed=0    s
kipped=0    rescued=0    ignored=0

[root@192 ~]#
```

Gambar 3. Menjalankan Playbook beserta kelima status yang ditampilkan

Berdasarkan kepada Gambar 3. di atas, perintah `export ANSIBLE_HOST_KEY_CHECKING=False` dan perintah `#ansible-playbook playbook.yml` dapat dijalankan dengan baik, di mana terdapat lima status yang ditampilkan. Kelima buah status tersebut meliputi: a.) *Task gathering facts* ke *host node* di *IP Address* 192.168.10.2, b.) *Task install httpd* ke *host node* di *IP Address* 192.168.10.2 disertai dengan perubahan (*changed*), c.) *Task run httpd* ke *host node* di *IP Address* 192.168.10.2 disertai dengan perubahan (*changed*), d.) *Task create webpage content* untuk melakukan perubahan (*changed*) ke *host node* di *IP Address* 192.168.10.2, dan e.) *Play recap* ke *host node* di *IP Address* 192.168.10.2 dengan informasi bahwa semua proses berjalan dengan baik (*Ok=4, changed=3, unreachable=0, failed=0, skipped=0, rescued=0, ignored=0*).

3.2. Black Box Testing dan Hasil Pengujian

Pasca implementasi sistem, selanjutnya dilakukan proses pengujian dengan menggunakan metode Black Box Testing. Pada tahap pengujian ini, dilakukan pengujian terhadap hasil konfigurasi sistem dan perintah yang digunakan pada Ansible terhadap *host node* di *IP Address* 192.168.10.2 yang bertindak sebagai client dari *task playbook*. Terdapat tiga tahapan pengujian yang dilakukan, yaitu: menguji status HTTPD pada *host node* dan melihat status yang ditampilkan, membuat web content menggunakan *Ansible* dan menguji hasil task yang dilakukan, serta menguji tampilan *web content* yang telah dibuat sesuai dengan konfigurasi di file *index.html*.

Pada tahap pengujian status HTTPD di *host node* (192.168.10.2), dilakukan dengan cara menjalankan perintah `systemctl status httpd`, kemudian mengamati proses yang terjadi pada Ansible pasca HTTPD diaktifkan. Gambar 4. menunjukkan status HTTPD pada *host node*:

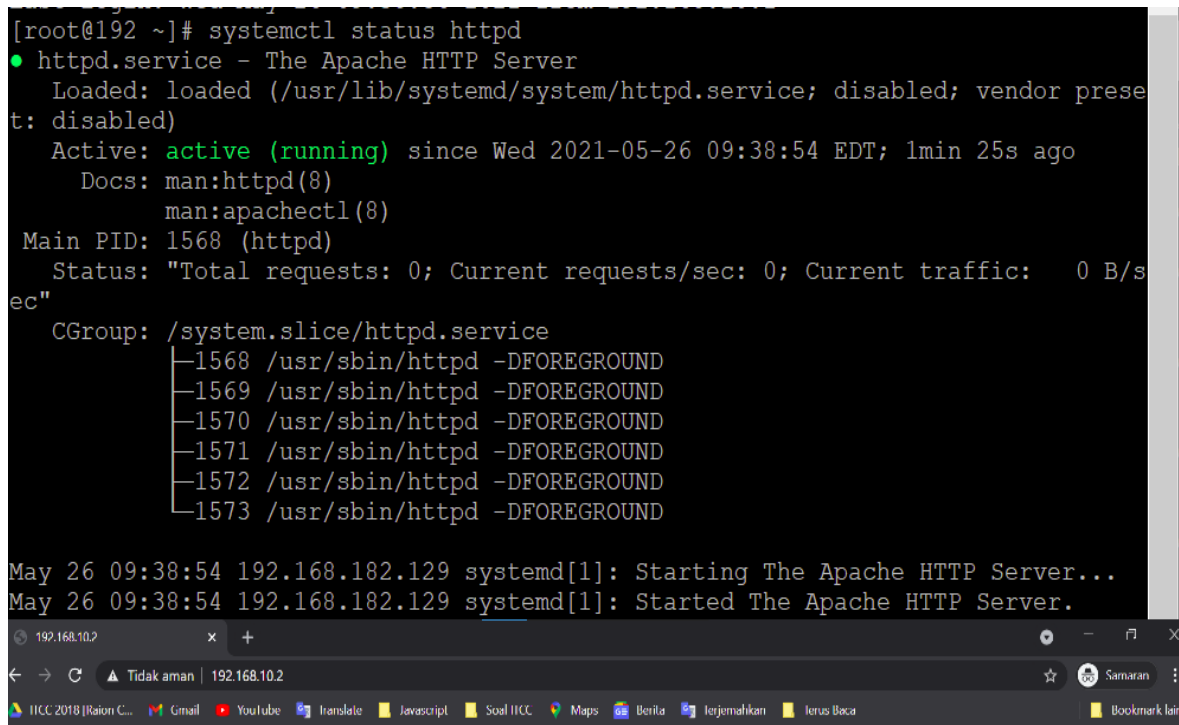
```
[root@192 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; vendor prese
t: disabled)
   Active: active (running) since Wed 2021-05-26 09:38:54 EDT; 1min 25s ago
     Docs: man:httpd(8)
           man:apachectl(8)
  Main PID: 1568 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:  0 B/s
ec"
   CGroup: /system.slice/httpd.service
           └─1568 /usr/sbin/httpd -DFOREGROUND
             └─1569 /usr/sbin/httpd -DFOREGROUND
               └─1570 /usr/sbin/httpd -DFOREGROUND
                 └─1571 /usr/sbin/httpd -DFOREGROUND
                   └─1572 /usr/sbin/httpd -DFOREGROUND
                     └─1573 /usr/sbin/httpd -DFOREGROUND

May 26 09:38:54 192.168.182.129 systemd[1]: Starting The Apache HTTP Server...
May 26 09:38:54 192.168.182.129 systemd[1]: Started The Apache HTTP Server.
```

Gambar 4. Pengujian status HTTPD

Berdasarkan kepada Gambar 4. di atas, terlihat bahwa paket HTTPD (*Apache HTTP Server*) berhasil dipasang dan berhasil dijalankan dengan Process ID (PID) 1568. Hal ini ditunjukkan dengan status *active (running)* yang muncul pada layar sistem.

Pada tahap pengujian pembuatan web content dengan menggunakan Ansible, dilakukan pembuatan sebuah file index web bernama *index.html*, yang di dalamnya ditambahkan konten dari proses untuk menjalankan task copy pada *playbook Ansible*. File HTML diletakkan pada path */var/www/html/* dengan nama file *index.html*. Pengujian selanjutnya yaitu menampilkan web page sesuai dengan konfigurasi yang telah dilakukan, di mana apabila IP Address dari host node diakses melalui *web browser*, maka akan menampilkan konten sesuai dengan konfigurasi pada *Ansible*. Gambar 5. di bawah ini menunjukkan pembuatan web content menggunakan Ansible dan tampilan pada web browser saat diakses:



Gambar 5. Membuat web content menggunakan Ansible dan tampilan di web browser

Berdasarkan kepada Gambar 5. di atas, terlihat bahwa file *index.html* telah berisikan sebuah *content* dari proses untuk menjalankan task copy dari *playbook Ansible*. Gambar 5. di atas tersebut memperlihatkan bahwa task tersebut berhasil dijalankan, ditunjukkan dengan adanya content dari file *index.html* yang telah sesuai dengan konfigurasi dari *playbook* yang dijalankan. Selain itu, tampilan *web page* yang sesuai dengan konfigurasi yang dilakukan, di mana ketika alamat IP dari *host node* diakses melalui web browser, ditampilkan informasi bahwa *webpage* dibuat dengan menggunakan *Ansible*, yang mana merupakan konten yang telah dikonfigurasi pada *Ansible*. Dalam hal ini *Ansible Playbook* yang telah diatur dan dikonfigurasi, dapat diakses dan berjalan dengan baik serta telah menerapkan otomatisasi di dalamnya secara sederhana. Berdasarkan kepada ketiga tahapan Black Box Testing yang dilakukan, kemudian dirangkum ke dalam sebuah tabel, sebagaimana ditampilkan pada Tabel 1. di bawah ini:

Tabel 1. Hasil Pengujian Black Box Testing

No	Pengujian	Langkah Pengujian	Hasil
1	Menguji status HTTPD pada host node dan melihat status yang ditampilkan	Menjalankan perintah <code>systemctl status httpd</code> → mengamati proses yang terjadi	Berhasil
2	Membuat web konten menggunakan Ansible dan menguji hasil task yang dilakukan,	Membuat web konten menggunakan Ansible → menguji hasil task yang dilakukan → mengamati proses yang terjadi	Berhasil
3	Menguji tampilan web content yang telah dibuat sesuai dengan konfigurasi pada file <i>index.html</i> .	Mengakses IP Address host node melalui web browser → mengamati tampilan yang muncul pada web browser apakah telah sesuai dengan konfigurasi pada file <i>index.html</i> .	Berhasil

Berdasarkan hasil *Black Box Testing* pada uraian dan Tabel 1. di atas, ditampilkan kolom pengujian, langkah pengujian, dan hasil yang diperoleh (dengan status berhasil atau gagal). Tabel menunjukkan bahwa ketiga tahapan memperoleh hasil berhasil, sehingga dapat dikatakan bahwa semua proses pada *Ansible* telah dapat berjalan dengan baik sesuai dengan hasil konfigurasi dan perintah-perintah yang diberikan terhadap sistem.

4. PENUTUP

Berdasarkan kepada implementasi dan pengujian *Black Box Testing* yang telah dilakukan, dapat diperoleh kesimpulan bahwa pengujian IaC berbasis *DevOps Ansible* dengan menggunakan metode *Black Box Testing* telah berjalan dengan baik, di mana *Black Box Testing* mampu menunjukkan hasil pengujian fungsionalitas pada sistem IaC yang dibangun, di mana setiap tahapan pengujian berhasil dilakukan sesuai dengan urutan langkah-langkah yang ditentukan. Adapun ketiga tahapan pengujian *Black Box Testing* pada pengujian IaC berbasis *DevOps Ansible* meliputi pengujian status web server menggunakan HTTPD di *host node*, pengujian web konten menggunakan *Ansible* beserta pengujian task di dalamnya, serta pengujian tampilan web konten sesuai dengan konfigurasi pada file *index.html*. Ketiga tahapan pengujian ini menunjukkan bahwa konfigurasi dapat dilakukan dengan lebih cepat dan mudah berbasiskan *Playbook* pada *Ansible*. Dengan suksesnya ketiga tahapan pengujian *Black Box Testing* tersebut, maka dapat dikatakan bahwa implementasi IaC berbasis *DevOps Ansible* telah berjalan dengan baik.

UCAPAN TERIMA KASIH

Terima kasih penulis ucapkan kepada rekan-rekan komunitas *Linux* dan *Open Source* Indonesia, rekan-rekan *DevOps* Indonesia, keluarga, serta Universitas Udayana atas dukungan selama penelitian ini, sehingga penelitian ini dapat terselesaikan dengan baik.

DAFTAR PUSTAKA

- [1] W.P. Lestari, A. Sujarwo, A., "DevOps: Disrupsi Pengelolaan ICT Pendidikan Tinggi," Seminar Nasional Aplikasi Teknologi Informasi, 2018, pp.26–31.
- [2] R.D. Feijter, "Towards the Adoption of DevOps in Software Product Organizations: A Maturity Model Approach, Journal of Department of Information and Computing Sciences Utrecht University, Vol.3 No.1, 2017, pp. 1–173.
- [3] A. Rahman, "Characteristics of Defective Infrastructure as Code Scripts in DevOps," Proceedings International Conference on Software Engineering, 2018, pp.476–479. <https://doi.org/10.1145/3183440.3183452>
- [4] T. Alfiandi, T.M Diansyah, R. Liza, "Analisis Perbandingan Manajemen Konfigurasi Menggunakan Ansible dan Shell Script Pada Cloud Server Deployment AWS," JITEKH, Vol.8, No.2, 2020, pp. 78-84.
- [5] M. Artac, T. Borovsak, E.D. Nitto, M. Guerriero, D.A. Tamburri, "DevOps: Introducing Infrastructure as Code," proceeding 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), pp.497-498. <https://doi.org/10.1109/ICSE-C.2017.162>
- [6] M. Guerriero, M. Garriga, D.A. Tamburri, F. Palomba, "Adoption, Support, and Challenges of Infrastructure-as-Code: Insights from Industry," 2019 IEEE International Conference on Software Maintenance and Evolution (ICSME). <https://doi.org/10.1109/ICSME.2019.00092>
- [7] E.T. Irawan, D. Bappedyanto, D. Hariyadi, A.S. Aziz, "Biznet Query Language Pada Infrastructre as Code," Jurnal Teknomatika, Vol.11, No.1, 2018.
- [8] J. Schwarz, A. Steffens and H. Lichter, "Code Smells in Infrastructure as Code," 2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC), 2018, pp. 220-228, doi: 10.1109/QUATIC.2018.00040.
- [9] S. Kokuryo, M. Kondo, O. Mizuno, "An Empirical Study of Utilization of Imperative Modules in Ansible," Proceedings 2020 IEEE 20th International Conference on Software Quality, Reliability, and Security, QRS 2020, pp. 442-449. <https://doi.org/10.1109/QRS51102.2020.00063>
- [10] A. Khumaidi, "Implementation of DevOps Method for Automation of Server Management Using Ansible," Jurnal Transformatika, Vol.18, No.2, 2021.
- [11] M.R. Affandi, P. Hatta, A. Efendi, "Otomatisasi Perangkat Jaringan Komputer Menggunakan Ansible Pada Laboratorium Komputer," Journal SMARTICS, Vol.6, No.2, 2020. DOI: <https://doi.org/10.21067/smartics.v6i2.4599>

- [12] T. Alfiandi, T. M. Diansyah, R. Liza, "Analisis Perbandingan Manajemen Konfigurasi Menggunakan Ansible dan Shell Script Pada Cloud Server Deployment AWS," *Jurnal Ilmiah Teknologi Harapan (JITEKH)*, Vol.8, No.2, 2020. DOI: <https://doi.org/10.35447/jitekh.v8i2.308>
- [13] N. Evianti, A.M. Wihandar, A. Kurniawan, "Automation Provisioning DevOps Website Server Menggunakan Ansible dan Vagrant," *Jurnal Nasional Informatika (JUNIF)*, Vol.2, No.2, 2021.
- [14] E.N. Fadhila, E.R. Gumelar, H.R. Pratama, G.M. Suranegara, "Otomasi Konfigurasi Routing pada Router menggunakan Ansible," *Journal Telecommunication, Network, Electronic, and Computer Technology (TELNECT)*, Vol.1, No.2, 2021.
- [15] Surahmat, A. Tenggono, "Analisis Perbandingan Kinerja Layanan Infrastructure as a Service Cloud Computing Pada Proxmox dan Xen Server," *Jurnal Manajemen Teknik MATRIK*, Vol.19, No.1, November 2019, pp.9-16, DOI : <https://doi.org/10.30812/matrik.v19i1.434>.
- [16] Y. Hidayat1, B. Arifwido, "Implementasi Web Server Menggunakan Infrastructure As Code Terraform Berbasis Cloud Computing," *Jurnal Format*, Vol.10, No. 2, 2021.
- [17] A. Saputra, H. Priyanto, N. Safriadi, "Implementasi Infrastructure as a Service pada Cloud Computing Menggunakan Metode Load Balancing," *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, Vol.8, No.4, 2020.
- [18] F.C. Ningrum, D. Suherman, S. Aryanti, H.A. Prasetya, A. Saifudin, "Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions," *Jurnal Informatika Universitas Pamulang*, Vol.4, No.4, 2019.
- [19] T.S. Jaya, "Pengujian Aplikasi dengan Metode Black Box Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)," *Jurnal Informatika Pengembangan IT*, Vol.3, No.1, 2018.