

Pengujian Implementasi Disaster Recovery Menggunakan Ansible

Miftahul Ulum^{1, a)}, M. Agung Nugroho^{1, b)}, Edi Iskandar^{1, c)}, Sudarmanto^{1, d)}

¹⁾*Informatika, Universitas Teknologi Digital Indonesia, Jalan Raya Janti No.143, Bantul, Yogyakarta, Indonesia*
Author Emails

^{a)}ulm.mftah@gmail.com

^{b)}corresponding author: m.agung.n@utdi.ac.id

^{c)}darmanto@utdi.ac.id

^{d)}edi_iskandar@utdi.ac.id

Abstract. *Cloud-based technology is a technology that constructs the internet as a central server for managing data and also user applications. Because the smooth process of processing data in a company/organization depends on IT infrastructure, data protection and recovery become essential, especially in damage or natural disasters. Files or data stored and processed by a company/organization will increase in size so that centralized and organized data storage is needed, namely by managing data in a data centre. The data backed up will be stored in a replica of the main data centre; the replica can act as an alternative if the primary data centre is undergoing maintenance or maintenance (down). Consequently, we need a system that can overcome obstacles when the data centre is down and requires a new data centre that is fast and reliable to handle the backup with ansible.*

Keywords:

Ansible, Ansible playbooks, disaster recovery, the backup file

Abstraksi. Teknologi yang banyak digunakan saat ini yaitu teknologi komputer berbasis sistem cloud yang merupakan sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan juga aplikasi pengguna. Karena kelancaran proses mengolah data pada suatu perusahaan/organisasi tergantung kepada infrastruktur TI, maka proteksi data dan recovery menjadi penting, khususnya jika terjadi kerusakan atau bencana alam. File atau data- data yang disimpan dan diolah suatu perusahaan/organisasi akan bertambah besar sehingga diperlukan penyimpanan data yang terpusat dan terorganisir yaitu dengan manajemen data pada suatu data center. Data yang di backup akan tersimpan dalam replika dari data center utama, replika tersebut dapat berperan sebagai alternative jika data center utama sedang mengalami pemeliharaan atau maintenance (down). Oleh maka dibutuhkan suatu sistem yang bisa mengatasi kendala saat data center down dan membutuhkan data center yang baru dengan cepat dan sama seperti sebelumnya yaitu menggunakan Ansible.

Kata Kunci : Ansible, Ansible playbooks, disaster recovery, backup file

PENDAHULUAN

Teknologi Informasi dan Komunikasi saat ini sangat berkembang dengan baik dan pesat serta memasuki berbagai bidang dalam kehidupan manusia. Baik dalam bidang teknologi, kesehatan, pendidikan, dan media hiburan lainnya. Dari semua bidang informasi yang ada di atas, salah satunya di bidang teknologi. Teknologi yang banyak digunakan saat ini yaitu teknologi komputer berbasis sistem cloud yang merupakan sebuah teknologi yang menjadikan internet sebagai pusat server untuk mengelola data dan juga aplikasi pengguna. Karena kelancaran proses mengolah data pada suatu perusahaan/organisasi tergantung kepada infrastruktur TI, maka proteksi data dan recovery menjadi penting, khususnya jika terjadi kerusakan atau bencana alam [1]. File atau data-data yang disimpan dan diolah suatu perusahaan/organisasi akan bertambah besar sehingga diperlukan penyimpanan data yang terpusat dan terorganisir yaitu dengan manajemen data pada suatu data center. Data yang disimpan pada data center merupakan data yang memiliki nilai bagi organisasi, dengan manajemen serta proteksi pengolahan data yang baik akan membuat data

terlindungi. Data yang di backup akan tersimpan dalam replika dari data center utama, replika tersebut dapat berperan sebagai alternative jika data center utama sedang mengalami pemeliharaan atau maintenance (down) [2]. Berdasarkan uraian diatas, maka dibutuhkan suatu sistem yang bisa mengatasi kendala saat data center down dan membutuhkan data center yang baru dengan cepat dan sama seperti sebelumnya yaitu menggunakan Ansible. Ansible [3] akan menolong terutama bagi server administrator konvensional maupun yang sudah bergerak ke arah DevOps. Saat menangani server dalam jumlah yang cukup besar, Ansible memberikan jalan untuk membuat penanganan server (meski dalam jumlah besar) menjadi lebih efisien. Dengan tool Ansible ini, kita akan membuat penanganan server menjadi lebih otomatis dan simpel [4]. Jadi, secara singkat, Ansible adalah salah satu jenis Configuration Management [5] Tools yang dapat digunakan untuk mengubah proses infrastruktur dari suatu program dari manual menjadi otomatis. Berdasarkan latar belakang di atas didapatkan rumusan masalah untuk kasus diatas yaitu bagaimana cara melakukan suatu recovery data aplikasi web yang berada pada suatu server menggunakan ansible untuk menghasilkan server yang baru yang siap digunakan. Penulis jurnal ini mempunyai maksud dan tujuan untuk mempermudah dalam melakukan install aplikasi pada beberapa server, melakukan disaster recovery untuk menghasilkan server yang baru yang siap digunakan, dan mempermudah pengembalian data secara otomatis.

TINJAUAN PUSTAKA

Disaster recovery merupakan sebuah perencanaan atau mekanis terhadap data agar secara integritas tetap terjaga walaupun terjadi hal-hal seperti bencana alam, kebakaran atau masalah teknis pada server. Merancang rencana *disaster recovery* [6] memerlukan penentuan lokasi alternatif untuk lokasi server cadangan. Hal ini dikarenakan jika terjadi crash pada server utama maka data pada server utama sebelumnya telah di backup ke server backup yang letaknya berbeda dengan server utama. Pertimbangan lokasi selanjutnya adalah lokasi cagar harus berada di luar radius mitigasi (gunung berapi, tsunami, sering banjir). Pertimbangan berikutnya adalah ketersediaan dan kualitas daya listrik/baterai, fiber line terdekat, IT/IS yang memiliki tujuan di lokasi tersebut, dan tidak rawan konflik. Mekanisme yang umum dilakukan dalam proses *disaster recovery* adalah *backup* dan *restore*. Backing Up atau sering disebut Backup yang berarti tidak hanya mencadangkan sekumpulan data saja, tetapi metode ini juga dapat mencadangkan keseluruhan sistem. Sehingga misalnya terjadi masalah pada level server, maka proses ini dapat melakukan restore data keseluruhan sehingga layanan dapat berjalan kembali. Metode Backup ini membutuhkan waktu downtime yang bervariasi tergantung dari besaran restore data [7].

Ansible adalah mesin otomatisasi *open source* yang mengotomatisasi penyediaan perangkat lunak, manajemen konfigurasi, dan pemasangan aplikasi. Ansible disertakan sebagai bagian dari distro Fedora Linux, yang dimiliki oleh RedHat Inc., dan juga tersedia untuk Red Hat Enterprise Linux, CentOS, dan Scientific Linux melalui Paket Ekstra untuk Enterprise Linux (EPEL) dan juga untuk sistem operasi lain. Seperti kebanyakan manajemen konfigurasi perangkat lunak [8]. Ansible memiliki dua tipe server, yaitu mesin pengontrol dan node. Pertama, terdapat satu mesin pengontrol tunggal dimana mesin ini akan melakukan orkestrasi atau mengatur setiap node yang ada. Node dikelola oleh mesin pengontrol melalui SSH. Mesin pengontrol memetakan lokasi dari setiap node melalui *inventory*. Untuk mengatur node, Ansible menyebarkan modul-modul node yang ada melalui SSH [3]. Modul-modul ini disimpan sementara di setiap node dan berkomunikasi dengan mesin pengontrol melalui protokol JSON. Berbeda dengan manajemen konfigurasi perangkat lunak populer seperti *Chef*, *Puppet*, dan *CFEngine* - Ansible menggunakan arsitektur tanpa agen. Dengan arsitektur berbasis agen, node harus memiliki daemon terpasang secara lokal yang berkomunikasi dengan mesin pengendali. Dengan arsitektur tanpa agen, node tidak diharuskan memasang dan menjalankan daemon untuk terhubung dengan mesin pengendali. Jenis arsitektur ini mengurangi overhead [9] [10] pada jaringan dengan mencegah node untuk terus mengecek apakah mesin pengendali telah siap atau tidak.

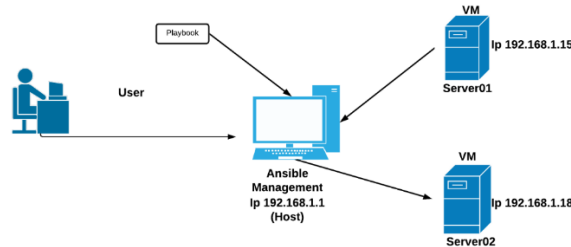
Penelitian sebelumnya terkait *disaster recovery* bertujuan untuk membangun rancangan disaster recovery plan dengan menggunakan ISO 24762:2008, apabila terjadi bencana terhadap data yang ada di Fakultas Teknik Universitas Pasundan yang berisi panduan dalam pemulihan terhadap bencana di Fakultas Teknik Universitas Pasundan berdasarkan ISO 24762:2008 [11]. Disaster recovery dengan menggunakan *Government Data Recovery Center* (GDRC) dapat diimplementasikan dengan menggunakan *rsync* untuk mekanisme backup dan restore dengan penerapan pada lembaga pemerintahan [2]. Penelitian ini berupa uji coba dan evaluasi pada sistem dan telah sesuai dengan standar yang ada dan dapat diaplikasikan guna mengoptimalkan manajemen jaringan pada proses *backup* dan *restore* operasional jaringan IPTEKnet BPPT. Mekanisme *backup* juga dapat menggunakan *bacula* yang merupakan salah satu pilihan backup server yang mengutamakan kecepatan transfer data, menampilkan informasi detail setelah

melakukan proses backup server, dan memiliki nilai *throughput* dengan data 25 GB sebesar 2.1 MBps untuk simulasi bandwidth 2 MBps dan 4.8 MBps dengan untuk simulasi dengan bandwidth 5 MBps. Namun jika pilihan backup server lebih mengutamakan keamanan data selama proses *backup* server berlangsung, dengan spesifikasi CPU dan *memory usage* yang rendah maka *rsync* merupakan pilihan yang tepat untuk melakukan backup server. Dengan hasil perbandingan 3% lebih rendah untuk *CPU Usage* dan 20 MB lebih rendah untuk *Memory Usage* [12]. Penelitian Markus [13] bertujuan untuk melakukan deploy aplikasi web secara otomatis menggunakan teknologi *ChatOps* dan Ansible. Penelitian ini menghasilkan peningkatan performa platform e-learning terutama waktu memuat halaman menjadi lebih cepat dan dapat berjalan secara offline.

METODE PENELITIAN

Dalam membuat sistem penelitian ini akan menggunakan 3 buah server yaitu server host yang akan digunakan untuk melakukan konfigurasi ansible management yang nantinya akan digunakan untuk recovery data dari server 1 ke server 2. Server host akan di install menggunakan Operating System Linux Ubuntu 20.04 sedangkan untuk server yang lain akan di install Operating System Linux Ubuntu Server 16.04. Pada server host akan di install Ansible yang akan digunakan untuk memmanagement server lainnya. Selanjutnya pada server lainnya akan dihubungkan pada ansible yang ada pada server host supaya saat melakukan recovery data menggunakan ansible dapat berjalan. Dalam penelitian ini, akan dilakukan konfigurasi supaya ketiga server tersebut dapat terhubung satu sama lain. Setelah ketiga server terhubung selanjutnya server host akan membuat ansible-playbook yang digunakan untuk menginstall beberapa aplikasi yang dibutuhkan untuk menjalankan web ke server lainnya secara bersamaan dan otomatis. Selanjutnya setelah aplikasi yang dibutuhkan terinstall maka server host akan melakukan management pada ansible untuk melakukan recovery data ataupun backup data lokal dari server 1 ke server 2 menggunakan ansible-playbook.

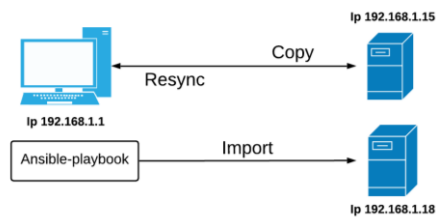
Untuk merancang sistem ini menggunakan ansible-playbook yang akan digunakan untuk melakukan manajemen ke beberapa server. Berikut topologinya:



GAMBAR 1. Arsitektur Sistem Pemodelan

Dapat dilihat pada gambar diatas seorang user melakukan konfigurasi menggunakan ansible untuk membuat suatu playbook yang akan digunakan untuk memmanagement server-server yang terhubung pada host diatas.

Untuk merancang pengujian ini menggunakan ansible-playbook yang akan di jalankan. Berikut topologinya:



GAMBAR 2. Arsitektur Pengujian

Dapat dilihat pada gambar diatas server host dengan ip 192.168.1.1 akan melakukan copy web server dari server 1 menggunakan ansible yang bertujuan supaya jika server 1 down atau bermasalah maka data dari server 1 aman berada di server host. Selanjutnya server host akan melakukan recovery web server ke server 2 atau server baru yang nantinya akan menghasilkan server yang sama seperti server 1 sebelumnya. Proses recovery berupa proses copy, backup dan import dibuat menggunakan ansible-playbook. Jasi saat ansible di jalankan otomatis akan melakukan recovery web server.

HASIL DAN PEMBAHASAN

Tahap implementasi sistem ini akan membahas proses implementasi sistem pada rancangan yang telah dibuat sebelumnya, pada tahapan implementasi ini terdapat beberapa potongan kode program inti dari Ansible Management. Proses backup database merupakan proses ansible playbook dimana didalam konfigurasinya terdapat beberapa perintah untuk melakukan suatu proses mem-backup database dari server 1 ke server host. Proses backup database ditunjukkan pada gambar 3.

```
root@DESKTOP-ICCB079:/etc/ansible# ansible-playbook backupdb.yml

PLAY [server01] *****

TASK [Gathering Facts] *****
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host server01 should use /usr/bin/python3, but is using
/usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using
the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature
will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
ok: [server01]

TASK [create a backup] *****
changed: [server01]

PLAY RECAP *****
server01 : ok=2  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0

root@DESKTOP-ICCB079:/etc/ansible#
```

GAMBAR 3. Proses Backup Database

Dalam proses backup web server merupakan proses ansible playbook dimana didalam konfigurasinya terdapat beberapa perintah untuk melakukan suatu proses mem-backup sekumpulan data web server dari server 1 ke server host. Proses backup web server ditunjukkan pada gambar 4.

```
root@DESKTOP-ICCB079:/etc/ansible# ansible-playbook backupweb.yml

PLAY [server01] *****

TASK [Gathering Facts] *****
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host server01 should use /usr/bin/python3, but is using
/usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using
the discovered platform python for this host. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature
will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in
ansible.cfg.
ok: [server01]

TASK [Ansible zip directory example] *****
ok: [server01]

TASK [Copy dari Remote ke Host] *****
changed: [server01]

PLAY RECAP *****
server01 : ok=3  changed=1  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

GAMBAR 4. Proses Backup Web Server

Setelah itu dapat dilihat data dari web server sudah terbackup pada direktori yang diinginkan. Dapat dilihat pada gambar 5.

```
root@DESKTOP-ICCB079:/etc/ansible# cd /tmp/
root@DESKTOP-ICCB079:/tmp# ls
backup.zip server01 tmpn9elzblp
root@DESKTOP-ICCB079:/tmp#
```

GAMBAR 5. Hasil backup Data Web Server

Proses recovery web server merupakan proses ansible playbook dimana didalam konfigurasiya terdapat beberapa perintah untuk melakukan suatu proses recovery web server dari server host ke server 2. Setelah proses recovery web server selesai kemudian data dari wes server tersebut akan di compress secara otomatis. Proses recovery web server ditunjukkan pada gambar 6.

```
root@DESKTOP-ICCB079:/etc/ansible# ansible-playbook recoveryweb.yml
PLAY [server03] *****
TASK [Gathering Facts] *****
[DEPRECATION WARNING]: Distribution Ubuntu 16.04 on host server03 should use /usr/bin/python3, but is using /usr/bin/python for backward compatibility with prior Ansible releases. A future Ansible release will default to using the discovered platform python for this host. See https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information. This feature will be removed in version 2.12. Deprecation warnings can be disabled by setting deprecation_warnings=False in ansible.cfg.
ok: [server03]
TASK [Install unzip] *****
ok: [server03]
TASK [Copy File .zip] *****
ok: [server03]
TASK [Ansible zip directory example] *****
ok: [server03]
PLAY RECAP *****
server03 : ok=4  changed=0  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

GAMBAR 6. Recovery Web Server

Dan pada gambar 7 dapat dilihat hasil dari recovery tersebut akan di compress secara otomatis pada server 2.



```
root@lum:/# cd /var/www/
root@lum:/var/www# ls
backup.zip html lum.com
root@lum:/var/www# _
```

GAMBAR 7. Hasil Recovery pada Server 2

Proses recovery melibatkan beberapa proses seperti backup, copy dan import database yang akan dilakukan menggunakan ansible playbooks. Kode konfigurasi dari proses recover seperti dibawah

```
backupdb.yml
---
- hosts: server01
  become: true
  tasks:
  - name: create a backup
    mysql_db:
      name: HMJ_TI_Bebdahara
      state: dump
      target: /tmp/HMJ_TI_Bebdahara.sql
```

```
login_host: localhost
login_user: root
login_password: ulum1234
```

Script diatas merupakan konfigurasi ansible playbook untuk melakukan backup database nya. Database pada server01 yang akan di *recovery* nantinya dibackup terlebih dahulu dengan melakukan perintah tasks dengan nama *create a backup*, untuk database yang akan dibackup bernama *HMJ_TI_Bebdahara*, perintah *state: dump*, *target: /tmp/HMJ_TI_Bebdahara.sql* merupakan perintah untuk membackup databasenya ke direktori /tmp/ dengan nama *HMJ_TI_Bebdahara.sql* yang ada pada server host lalu untuk login_host: localhost, login_user: root, login_password: ulum1234 disesuaikan dengan isi databasenya agar dapat terkoneksi saat dijalankan nantinya jadi sesuai isi databasenya sendiri.

```
backupweb.yml
---
- hosts: server01
  tasks:
  - name: Ansible zip directory example
    archive:
      path:
        - /var/www/lum.com/bendahara
      dest: /tmp/backup.zip
      format: zip

  - name: Copy dari Remote ke Host
    fetch:
      src: /tmp/backup.zip
      dest: /tmp/
      mode: 0774
      flat: yes
```

Script di atas adalah isi konfigurasi ansible playbook untuk melakukan perintah pengambilan web server dari server01 ke server host. *Script hosts: server01* menandakan bahwa perintah akan dijalankan pada server01. Karena pada ansible belum mendukung perintah *copy folder*, maka folder webserver yang terletak pada *path:/var/www/lum.com/bendahara* di arsipkan dengan format zip dan kemudian di copy ke server host dengan perintah *yml fetch* yang mana *src: /tmp/backup.zip* merupakan asal file yang akan di copy kemudian *dest: /tmp/* direktori tujuan untuk meletakkan file hasil fetch ().

```
recoveryweb.yml
---
- hosts: server02
  tasks:
  - name: Install unzip
    become: true
    apt:
      name: unzip
      state: latest

  - name: Copy File .zip
    copy:
      src: /tmp/backup.zip
      dest: /var/www/
```



```
- name: Ansible zip directory example
unarchive:
  src: /var/www/backup.zip
  dest: /var/www/lum.com/
  remote_src: yes
```

Script di atas adalah isi dari ansible playbook untuk melakukan recovery web server dari server host ke server 2. Web server didapat dari perintah yang ada pada backupweb.yml sebelumnya. Lalu web server tersebut di recovery pada server 2 dengan menuliskan perintah `- name: Copy File .zip, copy:, src: /tmp/backup.zip`, web server yang akan di copy berapa pada direktori tmp dengan nama backup.zip setelah ketemu kemudian copy ke server 2 dengan perintah `dest: /var/www/` yang datanya akan ditempatkan pada direktori `var/www/` yang ada pada server 2.

Setelah itu unzip data yang telah di recovery sebelumnya pada server 2 dengan melakukan perintah `unarchive:src: /var/www/backup.zip`, selanjutnya data ditempatkan pada folder `var/www/lum.com` menggunakan perintah `dest: /var/www/lum.com/, remote_src: yes`

KESIMPULAN

Berdasarkan hasil ujicoba dalam studi kasus pada penelitian ini, ansible dapat melakukan suatu pekerjaan otomatisasi untuk menangani proses backup dan recovery. Dengan menggunakan ansible-playbook dapat melakukan proses recovery berupa copy, backup dan import secara otomatis dan dapat digunakan pada proses *deployment* server baru yang sama.

Pengembangan kedepannya, sistem backup dengan ansible dapat dirancang untuk memiliki interface yang mudah digunakan oleh semua jenis pengguna bukan hanya dari sisi administrator atau pengelola server saja.

DAFTAR PUSTAKA

- [1] M. Ulum, "Implementasi Disaster Recovery Menggunakan Ansible," Skripsi, Stmik Akakom Yogyakarta, 2020. Accessed: Jan. 09, 2023. [Online]. Available: <https://Eprints.Utdi.Ac.Id/9036/>
- [2] A. Siahaan, M. Helmawaty, And F. Chairani Hisani, "Analisa Dan Perancangan Backup Dan Restore Data Center Secara Otomatis Dengan Rsync Pada Ipteknet Bppt," Undergraduate, Binus, 2010. Accessed: Jan. 09, 2023. [Online]. Available: http://Library.Binus.Ac.Id/Collections/Ethesis_Detail.Asp?Ethesisid=2010-1-00050-If
- [3] B. Meijer, L. Hochstein, And R. Moser, *Ansible: Up And Running*. O'reilly Media, Inc., 2022.
- [4] K. Marzuki, M. Mubiatma, And R. Azhar, "Otomasi Manajemen Vlan Intervlan Dan Dhcp Server Pada Cisco Menggunakan Ansible," *J. Inform. Dan Rekayasa Elektron.*, Vol. 4, No. 2, Pp. 171–180, Nov. 2021, Doi: 10.36595/Jire.V4i2.461.
- [5] J. Freeman And J. Keating, *Mastering Ansible: Effectively Automate Configuration Management And Deployment Challenges With Ansible 2.7*. Packt Publishing Ltd, 2019.
- [6] D. Y. Bernanda, Y. Charolina, O. Azhari, C. Pangrestu, And J. F. Andry, "Identification Of Potential And Planning For Disaster Recovery Using The Iso/Iec 24762 Standard At Xyz University," *J. Teknoinfo*, Vol. 17, No. 1, Art. No. 1, Jan. 2023, Doi: 10.33365/Jti.V17i1.2295.
- [7] W. Adhiwibowo, M. S. Suprayogi, And A. Nugroho, "Pengamanan Data Pada Aplikasi Sijalu Universitas Semarang Dengan Metode Remote Backup & Restore," *J. Pengemb. Rekayasa Dan Teknol.*, Vol. 14, No. 1, Art. No. 1, Jan. 2019, Doi: 10.26623/Jprt.V14i1.1217.
- [8] K. Okasha, *Network Automation Cookbook: Proven And Actionable Recipes To Automate And Manage Network Devices Using Ansible*. Packt Publishing Ltd, 2020.

- [9] I. Nida, “Implementasi Dan Analisis Efektivitas Ansible Sebagai Tools Automation Deployment Untuk Web Application Pada Cloud Server,” Other, Sekolah Tinggi Teknologi Terpadu Nurul Fikri, 2019. Accessed: Jan. 09, 2023. [Online]. Available: <https://Repository.Nurulfikri.Ac.Id/Id/Eprint/24/>
- [10] M. A. Nugroho, “Analisis Cluster Container Pada Kubernetes Dengan Infrastruktur Google Cloud Platform,” *Jipi J. Ilm. Penelit. Dan Pembelajaran Inform.*, Vol. 3, No. 2, 2018, Doi: <https://doi.org/10.29100/jipi.v3i2.651>.
- [11] Rizkiany Aulia Pratiwi, D. S. Doddy Ferdiansyah, And D. S. Iwan Kurniawan, “Perancangan Disaster Recovery Plan Pada Fasilitas Pusat Data Menggunakan Iso 24762:2008 (Studi Kasus : Fakultas Teknik Universitas Pasundan),” Other, Fakultas Teknik Unpas, 2019. Accessed: Jan. 09, 2023. [Online]. Available: [Http://Repository.Unpas.Ac.Id/40383/Http://Www.Teknik.Unpas.Ac.Id](http://Repository.Unpas.Ac.Id/40383/Http://Www.Teknik.Unpas.Ac.Id)
- [12] B. S. K. Adi, *Implementasi Dan Analisa Perbandingan Metode Rsync Dan Bacula Berbasis Linux Ubuntu 18.04 Lts Untuk Server Backup*. Universitas Telkom, S1 Teknik Telekomunikasi, 2019. Accessed: Jan. 09, 2023. [Online]. Available: <https://openlibrary.telkomuniversity.ac.id/pustaka/153795/implementasi-dan-analisa-perbandingan-metode-rsync-dan-bacula-berbasis-linux-ubuntu-18-04-lts-untuk-server-backup.html>
- [13] M. Juopperi, “Deployment Automation With Chatops And Ansible,” Thesis, Helsinki Metropolia University Of Applied Sciences, 2017.