

# **Makalah Metode DevOps Dengan Tools ANSIBLE**



Untuk memenuhi tugas mata kuliah Metodologi Desain Perangkat Lunak Praktik

disusun oleh:

- |                         |              |
|-------------------------|--------------|
| 1. Novian Yuda Permana  | (5200411317) |
| 2. Khafid Nur Maulana   | (5200411491) |
| 3. Rizky Diar Panuntun  | (5200411499) |
| 4. Alfaeni Syafa Safira | (5200411511) |

**TEKNIK INFORMATIKA**

**FAKULTAS SISTEM INFORMASI DAN ELEKTRO**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

## DAFTAR ISI

BAB I .....	3
PENDAHULUAN .....	3
A. Latar Belakang.....	3
B. Rumusan Masalah.....	3
C. Tujuan Penelitian .....	3
BAB II .....	4
PEMBAHASAN .....	4
D. Apa itu Ansible .....	4
E. Apa tujuan tools Ansible .....	4
F. Apa manfaat tools Ansible .....	4
G. Apa cara berjalan tools Ansible.....	4
H. Apa sajakah kelebihan dan kekurangan tools ansible .....	5
BAB III .....	6
METODE ANSIBLE.....	6
I. Review Jurnal .....	6
BAB IV.....	14
PENUTUP .....	14
J. Perbandingan Dengan Metode Waterfall.....	14
K. Perbandingan Metode Prototype .....	15
L. Perbandingan Dengan Metode RAD .....	17

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Pada era serba digital ini perangkat lunak atau software berkembang dengan pesatnya dan memudahkan manusia di dalam kehidupan. Dalam kehidupan manusia teknologi perangkat lunak atau software menjadikan teknologi atau perangkat lunak menjadi salah satu system yang dapat memberikan banyak kemudahan dalam kehidupan manusia . Perangkat lunak atau software adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaannya (user manual). Pada masa sekarang perangkat lunak sudah melekat dengan kehidupan manusia karena ketika manusia menggunakan perangkat lunak sangat dapat membantu segala aktivitas dalam kehidupan manusia.

Perlunya metode yang mendukung pengembangan perangkat lunak yaitu Development dan Operation. Metode DevOps tidak hanya mengelola bagian pengembangan tetapi juga mengelola bagian pengoperasian. Hasil perancangan menunjukkan bahwa DevOps menjadi pendekatan tepat agar perangkat lunak dikembangkan dari kecil menjadi besar, step by step tetapi tanpa kehilangan penelusuran antara rilis produk. Dan yang lebih mendasar, DevOps mampu memperkecil gap antara pengembang dengan pengguna aplikasi. Melalui metode DevOps, pengembangan dan pengoperasian memiliki keterhubungan sebagai timbal balik antara pengembangan dengan pengoperasian maupun sebaliknya.. Banyaknya tools yang digunakan DevOps seperti AWS, GitHub, docker, Codeship, JUnit, Ansible dan masih banyak lainnya.

Pada laporan yang telah kami buat, akan menjelaskan terkait apa itu tools ansible, tujuan, manfaat, cara menjalankan, kelebihan dan kekurangannya. Mari kita membaca untuk memahi konsep dan tujuan metode Ansible.

### **B. Rumusan Masalah**

1. Apa itu tools Ansible?
2. Apa tujuan tools Ansible?
3. Apa manfaat tools Ansible?
4. Apa cara menjalankan tools Ansible?
5. Apa sajakah kelebihan dan kekurangan tools ansible?

### **C. Tujuan Penelitian**

1. Mengetahui apa itu tools Ansible
2. Mengetahui tujuan tools Ansible
3. Mengetahui manfaat metode Ansible
4. Mengetahui cara menjalankan tools Ansible
5. Mengetahui apa saja kelebihan dan kekurangan tools ansible

## **BAB II**

### **PEMBAHASAN**

#### **D. Apa itu Ansible**

Ansible adalah penyediaan perangkat lunak sumber terbuka , manajemen konfigurasi , dan alat penerapan aplikasi yang memungkinkan infrastruktur sebagai kode . Ini berjalan pada banyak sistem mirip Unix, dan dapat mengonfigurasi sistem mirip Unix serta Microsoft Windows . Ini termasuk bahasa deklaratif sendiri untuk menggambarkan konfigurasi sistem . Ansible ditulis oleh Michael DeHaan dan diakuisisi oleh Red Hat pada tahun 2015. Ansible tidak memiliki agen , sementara terhubung dari jarak jauh melalui SSH atau Windows Remote Management (memungkinkan eksekusi PowerShell jarak jauh ) untuk melakukan tugasnya.

#### **E. Apa tujuan tools Ansible**

dengan Ansible, kita dapat melakukan pengelolaan server yang lebih mudah dan terstruktur. Selain itu dapat menghemat waktu kita untuk mengeksekusi proses konfigurasi atau deployment yang sama terhadap beberapa server yang kita miliki.

#### **F. Apa manfaat tools Ansible**

1. Menghemat lebih banyak waktu. Hanya perlu satu kali membuat file konfigurasi dan dapat digunakan secara berkali-kali.
2. Dokumentasi konfigurasi infrastruktur lebih jelas. Semua konfigurasi yang dilakukan pada infrastruktur tertulis dalam baris kode (file yml) sehingga memudahkan untuk sysadmin lain mengerti apa saja konfigurasi yang telah dilakukan pada infrastruktur.
3. Dapat menerapkan Version Control. Karena konfigurasi tersimpan dalam bentuk baris kode, dan apapun yang berbentuk teks dapat diterapkan version control, maka konfigurasi dalam infrastruktur juga dapat diterapkan version kontrol jika menggunakan Ansible.

#### **G. Apa cara menjalankan tools Ansible**

1. Ansible Ad-Hoc. Cara menjalankan ansible berupa command-line instruksi via terminal
2. Ansible Playbooks. Cara menjalankan ansible yang terdiri dari playbook atau file YML dan file YML terdiri dari task, dimana task akan memanggil modul-modul ansible. Ansible playbook akan di jalanka secara berurutan
3. Ansible Tower. Ansible Tower merupakan automation untuk kalangan Enterprise dan juga berbayar. Ansible Tower memiliki banyak kemudahan di bandingkan ansible tipe lain. Ansible Tower lebih mudah untuk mengontrol server di karenakan memiliki halaman UI, selain itu juga module-module lain. Untuk keamanan ansible Tower juga sudah menyediakan secara langsung.

## **H. Apa sajakah kelebihan dan kekurangan tools ansible**

### **1. Kelebihan**

- a. Mudah digunakan. Ansible mudah digunakan karena dapat langsung digunakan dengan 1 command line.
- b. Mudah dipahami. Selain digunakan langsung dengan 1 command line, ansible juga mendukung pengumpulan tugas yang biasa disebut dengan ansible-playbook, playbook ini berisikan pekerjaan pekerjaan yang akan dieksekusi secara berurutan. Playbook ini dibuat dengan format yaml yang mudah dipahami. Dengan menggunakan dan juga semua konfigurasi saat ini tetap sama seperti konfigurasi di hari berikutnya.
- c. Terdokumentasi. Jika saat ini kalian melakukan setup 1 server dan beberapa bulan kemudian diminta lagi untuk setup beberapa server berdasarkan sebelumnya yang telah disetup, hasilnya mungkin beda. Nah dengan penggunaan ansible, cukup membuat 1 konfigurasi dan bisa dieksekusi berkali kali dengan hasil yang sama.
- d. Support Command Bash. Ansible memiliki banyak module yang bisa digunakan namun jika kalian ingin mengeksekusi perintah tertentu namun belum ada dimodule ansible, anda tetap bisa mengesekusinya dengan module command dan kemudian bisa kalian isi dengan command yang ingin dieksekusi.

### **2. Kekurangan**

- a. Penerapan harus merubah kebiasaan atau kebudayaan dalam pengembangan proyek.
- b. Membutuhkan Software Engineering yang ahli.
- c. Memerlukan kolaborasi yang kuat.

### BAB III

#### METODE ANSIBLE

##### I. Review Jurnal

<b>JUDUL JURNAL</b>	Otomatisasi Keamanan Pada Router Mikrotik Menggunakan Ansible
<b>PENULIS</b>	1. Ahmad Givari Adi Prasetyo 2. I Putu Hariyadi
<b>PENDAHULUAN</b>	<p>Pandemic covid 19 membuat aktifitas belajar dan bekerja dilakukan dari rumah. Untuk mendukung kegiatan tersebut maka dibutuhkan koneksi Internet. Salah satu perangkat yang sangat penting untuk koneksi Internet adalah router. Sehingga keamanan router menjadi hal yang perlu diperhatikan.. dari sistus avast.com selama dua tahun terakhir telah di prediksi peningkatan malware berbasis router dan juga perubahan dalam karakteristik serangan tersebut, Avast memprediksi peningkatan pembajakan router yang digunakan untuk mencuri kredensial perbankan, misalnya, di mana router yang terinfeksi menyuntikkan frame HTML berbahaya ke halaman web tertentu saat ditampilkan di ponsel. Elemen baru ini dapat meminta pengguna seluler untuk memasang aplikasi perbankan baru, misalnya, dan aplikasi jahat ini kemudian akan menyimpan pesan autentikasi. Router akan terus digunakan sebagai target serangan, tidak hanya untuk menjalankan scripts kriminal atau memata-matai pengguna, akan tetapi juga bertindak sebagai tautan perantara dalam serangan berantai.</p>
<b>METODOLOGI</b>	<p>Metodologi penelitian yang digunakan pada penelitian ini yaitu metode waterfall. Metode waterfall merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode waterfall adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear.</p> <p>Dari lima tahapan yang ada, penulis hanya menggunakan 4 (empat) tahapan yaitu :</p> <ol style="list-style-type: none"> <li>1. analisa kebutuhan</li> <li>2. desain sistem</li> <li>3. penulisan kode program</li> <li>4. pengujian program.</li> </ol>
Analisa Kebutuhan	<p>Analisa Kebutuhan</p> <p>Pada tahapan ini dilakukan analisa permasalahan dan kebutuhan untuk proses penelitian tentang Otomasi Manajemen dan keamanan menggunakan ansible. Tahap ini terdiri dari dua tahapan yaitu pengumpulan data dan analisa data.</p>
Desain system	<p>Desain system</p> <p>Pada tahap ini dilakukan perancangan jaringan uji coba, rancangan pengalamatan IP, rancangan system otomasi dan kebutuhan perangkat lunak</p>

	<p>1. Desain Jaringan Rancangan tersebut mensimulasikan jaringan dari sebuah perusahaan yang memiliki lima gedung dimana setiap gedung terdiri dari satu router untuk interkoneksi dengan jaringan yang berbeda-beda.</p> <p>2. Rancangan jaringan uji coba Rancangan jaringan tersebut di simulasikan menggunakan virtualisasi VMWare Workstation.</p> <p>3. Rancangan Sistem Otomasi Pengamanan Router Mikrotik Pada rancangan system otomasi pengamanan router Mikrotik menjelaskan tentang alur dari system otomasi yang di jalankan sehingga dapat mengotomasi router</p> <p>4. Rancangan pengalamatan IP Pada rancangan pengalamatan IP untuk rancangan jaringan uji coba otomasi Manajemen dan security pada Mikrotik ini terdapat 5 (lima) alamat network class C yang digunakan yaitu 192.168.169.0/24, 192.168.12.0/30, 192.168.13.0/30, 192.168.14.0/30, 192.168.15.0/30.</p> <p>5. Kebutuhan perangkat keras dan lunak Adapun kebutuhan perangkat keras dan perangkat lunak dalam penulisan skripsi ini adalah sebagai berikut:</p> <p>A. Kebutuhan Perangkat Keras (Hardware) Dengan menggunakan 1 (satu) Unit computer yang di dalamnya diinstallkan VMware Workstation dan dibuatkan 2 (dua) Virtual machin sebagai Mikrotik chr dan server ansible. Dan juga 1 (satu) unit computer itu juga akan difungsikan sebagai PC client.</p> <p>a. Mikrotik chr dengan spesifikiasi sebagai berikut:</p> <ul style="list-style-type: none"> <li>- Procesor : 1 core</li> <li>- Hardisk : 1 GB</li> <li>- Memory : 256 MB</li> </ul> <p>b. 1 server ansible dengan spesifikasi sebagai berikut:</p> <ul style="list-style-type: none"> <li>-Procesor : 1 core</li> <li>-Hardisk : 15 GB</li> <li>-Memory : 2 GB</li> <li>-Sistem Operasi: CentOS 7</li> </ul> <p>c. Komputer client dengan spesifikasi sebagai berikut:</p> <ul style="list-style-type: none"> <li>-Procesor : Intel Core i5</li> <li>-Hardisk : 1 TB</li> <li>-Memory : 8 GB</li> <li>-Sistem Operasi: Windows 10</li> </ul>
--	--

	<p>B. Kebutuhan Perangkat Lunak (Software)</p> <ol style="list-style-type: none"> <li>1. Ansible yang digunakan dalam penelitian kali ini menggunakan ansible versi 2.9.1. Ansible diinstall pada PC yang digunakan sebagai server ansible untuk melakukan otomasi.</li> <li>2. VMWare Workstation pro 15</li> <li>3. ISO CentOS 7 sebagai OS pada computer server dan digunakan untuk instalasi sistem operasi.</li> <li>4. Mikrotik chr yang digunakan dalam penelitian ini menggunakan versi yang berbeda-beda di antaranya Mikrotik chr version 6.44.6, 6.45.8, 6.45.9, 6.45.7, dan 6.46.2 merupakan versi routerOS yang di tunjukan untuk dijalankan sebagai mesin router virtual.</li> <li>5. Winbox sebagai aplikasi kecil yang dapat digunakan untuk melakukan administrasi terhadap Mikrotik routerOS dengan cepat dan dengan tampilan GUI.</li> </ol>
Penulisan kode program	<p>Penulisan kode program</p> <p>Pada tahap ini berisikan tentang konfigurasi pada perangkat jaringan dan pengkodean pada system otomasi yang akan dibuat.</p> <p>1 Instalasi Dan Konfigurasi Perangkat Jaringan</p> <p>Pada tahap ini memuat tentang instalasi dan konfigurasi yang dilakukan pada tiga perangkat yang terlibat yakni satu server, satu client, dan lima router berdasarkan rancangan uji coba.</p> <ol style="list-style-type: none"> <li>a. Instalasi dan konfigurasi server. <ol style="list-style-type: none"> <li>1. Install epel-release berfungsi untuk menginstalasi repository Extra Packages for Enterprise Linux(EPEL) .</li> <li>2. Instalasi ansible dimana tools yang di gunakan untuk mengotomasi router yang di inginkan.</li> <li>3. Membuat public key untuk dapat mengatur public key authentication untuk SSH dengan membuat key pair menggunakan utilitas ssh-keygen pada CentOS 7 Control Machine sehingga dapat mengakses Managed Machine yaitu router Mikrotik melalui SSH.</li> <li>4. Ansible Inventory, penerapan pengelompokan (group) untuk mengklasifikasi sistem dan menentukan sistem yang dikontrol berdasarkan waktu dan tujuan tertentu.</li> </ol> </li> <li>b. Instalasi dan konfigurasi server Ansible</li> </ol> <ol style="list-style-type: none"> <li>b. Instalasi dan konfigurasi router Mikrotik. <p>Sebelum dapat di konfigurasi Mikrotik tersebut harus di install menggunakan Mikrotik CHR pada VMWare Worstation 15 pro.</p> <ol style="list-style-type: none"> <li>1. Konfigurasi pengalamatan IP pada Mikrotik</li> <li>2. Routing static untuk dapat terkoneksi antara beda jaringan.</li> <li>3. Menyalin public key dari server ke Mikrotik.</li> </ol> </li> </ol>



	<p>c. Konfigurasi pada client</p> <p>Konfigurasi pada client meliputi konfigurasi pengalamatan TCP/IP agar client dapat berkomunikasi dengan server ansible. Konfigurasi yang dilakukan pada client windows 10 adalah pengaturan alamat IP.</p>
Pengujian Program	<p>Pengujian Program</p> <p>Pada tahap ini memuat tentang langkah-langkah untuk uji coba hasil konfigurasi menggunakan beberapa skenario uji coba. Dalam melakukan skenario uji coba manajemen Mikrotik seperti pembuatan user, menonaktifkan user, backup, firewall filter, mac server, bandwidth test server, service, dan reboot. baik secara manual maupun otomatis, dilakukan percobaan skenario menggunakan 1 Mikrotik sebanyak 5 kali dan menggunakan 5 Mikrotik sebanyak 5 kali, serta melakukan perhitungan waktu yang dibutuhkan untuk manajemen Mikrotik.</p>
<b>HASIL DAN PEMBAHASAN</b>	<p>Pada tahap ini memuat tentang pembahasan hasil dari instalasi dan konfigurasi pada perangkat Mikrotik, hasil pembuatan ansible playbook untuk mengotomatisasi fitur keamanan pada Mikrotik, uji coba serta analisa terhadap hasil ujicoba.</p>
Hasil Uji Coba	<p>Hasil Uji Coba</p> <p>Pada hasil uji coba terdiri dari 2 (dua) bagian yaitu hasil verifikasi konfigurasi dan Ujicoba skenario ujicoba.</p> <p>1. Hasil Verifikasi Konfigurasi</p> <p>Terdapat 3 (tiga) hasil verifikasi konfigurasi yang dilakukan yaitu hasil verifikasi konfigurasi server, router, dan hasil verifikasi konfigurasi client.</p> <p>1. Hasil Verifikasi Konfigurasi Server</p> <p>Pada tahap ini terlihat beberapa hasil verifikasi hasil konfigurasi dari server.</p> <pre> login as: root root@192.168.169.3's password: Last login: Mon Mar  9 03:43:20 2020 [root@givar ~]# ip address 1: lo: &lt;LOOPBACK,UP,LOWER_UP&gt; mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00     inet 127.0.0.1/8 scope host lo         valid_lft forever preferred_lft forever     inet6 ::1/128 scope host         valid_lft forever preferred_lft forever 2: ens33: &lt;BROADCAST,MULTICAST,UP,LOWER_UP&gt; mtu 1500 qdisc pfifo_fast state UP group default qlen 1000     link/ether 00:0c:29:52:97:b0 brd ff:ff:ff:ff:ff:ff     inet 192.168.169.3/24 brd 192.168.169.255 scope global noprefixroute ens33         valid_lft forever preferred_lft forever     inet6 fe80::36d7:f7cf:4163:637d/64 scope link noprefixroute         valid_lft forever preferred_lft forever [root@givar ~]# </pre> <p>2. Hasil verifikasi konfigurasi router</p> <p>Hasil verifikasi pengalamatan ip address pada router R1 terlihat pada interface ether2 telah dikonfigurasi menggunakan alamat IP 192.168.169.254/24</p>

```

lgivar@MikroTik1 > ip address print
Flags: X - disabled, I - invalid, D - dynamic
# ADDRESS NETWORK INTERFACE
0 192.168.169.254/24 192.168.169.0 ether2
1 192.168.12.1/30 192.168.12.0 ether3
2 192.168.13.1/30 192.168.13.0 ether4
3 192.168.14.1/30 192.168.14.0 ether5
4 192.168.15.1/30 192.168.15.0 ether6
5 D 192.168.163.128/24 192.168.163.0 ether1

```

### 3. Hasil Verifikasi Konfigurasi Client

Hasil verifikasi pengalamatan IP Address pada client Windows 10 terlihat pada interface VMnet1, telah dikonfigurasi menggunakan alamat IP 192.168.169.2/24

```

Command Prompt
Ethernet adapter VMware Network Adapter VMnet1:

Connection-specific DNS Suffix . : 
Link-local IPv6 Address . . . . . : fe80::34f7:cae8:eb8b:b62e%12
IPv4 Address. . . . . : 192.168.169.2
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 

```

### 2. Ujicoba Berbasis Skenario

Terdapat 8 (delapan) skenario yang dilakukan untuk mengujicoba sistem otomasi keamanan router Mikrotik yang telah dirancang secara berurut.

pembuatan dari playbook bernama main.yml yang berguna untuk mengotomasi router.

```

---
- name: main
  hosts: routers
  connection: network_cli
  gather_facts: no
  tasks:
    - name: terkait mac
      include_tasks: mac_server_main.yml
    - name: add user
      include_tasks: add_user_main.yml
    - name: backup
      include_tasks: backup_main.yml
    - name: btest server
      include_tasks: btest_server_main.yml
    - name: firewall
      include_tasks: firewall_main.yml
    - name: m_service
      include_tasks: m_service_main.yml
    - name: m_user
      include_tasks: m_user_main.yml

```

Proses pembuatan struktur file main.yml menggunakan module ansible telah dilakukan maka untuk mengakses dan mengeksekusi struktur file main.yml menggunakan perintah 'ansible-playbook main.yml'.

```

[root@givar ~]# ansible-playbook main.yml
PLAY [main] *****

TASK [terkait mac] *****
included: /root/mac_server_main.yml for 192.168.169.254, 192.168.12.2, 192.168.13.2, 192.168.14.2, 192.168.15.2

TASK [mac_server_main] *****
ok: [192.168.169.254]
ok: [192.168.15.2]
ok: [192.168.13.2]
ok: [192.168.14.2]
ok: [192.168.12.2]

TASK [debug] *****
ok: [192.168.169.254] => {
  "result": {
    "ansible_facts": {
      "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "failed": false,
    "stdout": [
      ""
    ],
    "stdout_lines": [
      {
        ""
      }
    ]
  }
}

```

Telah terlihat bahwa router R1, R2, R3, R4, R5 telah terdampak oleh scenario yang di otomasi dapat dilihat pada tabel dibawah ini

No	Skenario	R1	R2	R3	R4	R5
1	Add user	✓	✓	✓	✓	✓
2	Disable user admin	✓	✓	✓	✓	✓
3	Btest server	✓	✓	✓	✓	✓
4	Mac ping server	✓	✓	✓	✓	✓
5	IP service	✓	✓	✓	✓	✓
6	ICMP filter	✓	✓	✓	✓	✓
7	backup	✓	✓	✓	✓	✓
8	reboot	✓	✓	✓	✓	✓

Analisa Hasil Ujicoba

Analisa Hasil Ujicoba  
Rangkuman Hasil Ujicoba Perbandingan Waktu Pembuatan file backup, disable bandwidth test server, firewall filter, manajemen service, manajemen user, disable mac server, user add. Pada 1 (satu) router mikotik dan pada 5 (lima) router Mikrotik dilakukan secara Manual dan Otomasi. Hasil analisa terlihat pada tabel dibawah ini:

**Tabel Perbandingan Waktu Pembuatan 8 fitur keamanan secara manual dan otomasi pada satu router**

Percobaan Pembuatan 8 skenario	Waktu	
	Manual	Otomasi
Pertama	3 menit 4 detik	27 detik
Kedua	2 menit 43 detik	25 detik
Ketiga	2 menit 47 detik	24 detik
Keempat	2 menit 46 detik	25 detik
Kelima	2 menit 40 detik	26 detik
<b>Rata-Rata</b>	<b>2 menit 56 detik</b>	<b>25 detik</b>
<b>Minimal (Waktu Tercepat)</b>	<b>2 menit 40 detik</b>	<b>24 detik</b>
<b>Maksimal (Waktu Terlama)</b>	<b>3 menit 4 detik</b>	<b>27 detik</b>

**Tabel 3.2 Perbandingan Waktu Pembuatan 8 fitur keamanan secara manual dan otomasi pada lima router**

Percobaan Pembuatan 8 skenario	Waktu	
	Manual	Otomasi
Pertama	16 menit 2 detik	1 menit 44 detik
Kedua	16 menit 8 detik	1 menit 40 detik
Ketiga	16 menit	1 menit 47 detik
Keempat	15 menit 49 detik	1 menit 41 detik
Kelima	15 menit 53 detik	1 menit 44 detik
<b>Rata-Rata</b>	<b>16 menit</b>	<b>1 menit 43 detik</b>
<b>Minimal (Waktu Tercepat)</b>	<b>15 menit 53 detik</b>	<b>1 menit 40 detik</b>
<b>Maksimal (Waktu Terlama)</b>	<b>16 menit 2 detik</b>	<b>1 menit 47 detik</b>

Berdasarkan analisa hasil ujicoba keamanan router Mikrotik yang telah dilakukan maka diperoleh analisa hasil ujicoba sebagai berikut:

1. Perbandingan hasil ujicoba waktu Pembuatan file backup, disable bandwidth test server, firewall filter, manajemen service, manajemen user, disable mac server, user add Pada 1 (satu) router Mikrotik secara manual dengan rata-rata waktu 2 menit 56 detik, minimal waktu tercepat 2 menit 40 detik, waktu terlama 3 menit 4 detik. dan otomasi menggunakan ansible dengan rata-rata waktu 25 detik, minimal waktu tercepat 24 detik, waktu terlama 27 detik.
2. Perbandingan hasil ujicoba waktu Pembuatan file backup, disable bandwidth test server, firewall filter, manajemen service, manajemen user, disable mac server, user add Pada 5 (lima) router Mikrotik secara manual dengan rata-rata waktu 16 menit, minimal waktu tercepat 15 menit 53 detik, waktu

	<p>terlama 16 menit 2 detik. dan otomasi menggunakan ansible dengan rata-rata waktu 1 menit 43 detik, minimal waktu tercepat 1 menit 40 detik, waktu terlama 1 menit 47 detik.</p> <p>3. Ansible memanfaatkan modul <code>routeros_command</code> dimana <code>routeros_command</code> itu terdapat perintah Mikrotik cli yang terkait dengan pengamanan sehingga dapat di terapkan pada setiap router.</p>
<b>KESIMPULAN</b>	<p>Adapun kesimpulan yang dapat diambil berdasarkan hasil uji coba yang telah dilakukan adalah sebagai berikut:</p> <ol style="list-style-type: none"> <li>1. Ansible playbook yang dibuat berhasil digunakan untuk mengotomasi ke delapan fitur keamanan yaitu add user, disable user admin, disable service yang tidak perlu, disable btest server, disable mac ping server, firewall filter, backup, dan reboot.</li> <li>2. Sistem otomasi yang dibuat dapat mempercepat proses pembuatan kedelapan scenario pada satu router. Dilakukan sebanyak 5 (lima) kali percobaan dengan rata-rata Selisih waktu pembuatan kedelapan scenario pada satu router adalah 2 menit 31 detik lebih cepat dilakukan secara otomasi dari pada dilakukan secara manual.</li> <li>3. Sistem otomasi yang dibuat dapat mempercepat proses pembuatan kedelapan scenario pada lima router. Dilakukan sebanyak 5 (lima) kali percobaan dengan rata-rata Selisih waktu pembuatan kedelapan scenario pada lima router adalah 14 menit 17 detik lebih cepat dilakukan secara otomasi dari pada dilakukan secara manual.</li> </ol>

## **BAB IV**

### **PENUTUP**

#### **J. Perbandingan Dengan Metode Waterfall**

**Waterfall** merupakan suatu metode manajemen proyek yang sudah relatif lama. Waterfall menekankan pada sebuah desain proses yang berurutan yang dalam prosesnya terlihat seperti aliran air terjun dari proses perancangan konsep, analisis proyek, inisialisasi proyek, desain, pembuatan sistem, testing, implementasi dan perawatan.

##### **Beberapa ciri dari metode ini:**

- Proyek dibagi-bagi dalam beberapa fase yang saling berurutan.
- Penekanan pada perencanaan, jadwal, deadline dan implementasi keseluruhan sistem.
- Kontrol yang ketat dalam siklus hidup proyek dengan menggunakan bantuan dokumentasi tertulis.

##### **Kelebihan metode ini:**

- Relatif mudah dimengerti dan mudah digunakan.
- Requirement dari sistem bersifat stabil.
- Baik dalam manajemen kontrol.
- Bekerja dengan baik ketika kualitas lebih diutamakan dibandingkan dengan biaya dan jadwal/deadline.

##### **Kekurangan dari metode ini:**

- Semua kebutuhan sistem harus diketahui di awal
- Integrasi sekaligus di akhir sistem.
- Testing hanya dilakukan pada setiap akhir fase, bahkan tidak jarang testing hanya dilakukan di akhir-akhir proyek.
- Membutuhkan waktu yang cukup lama meski kadang proyeknya tidak terlalu besar.
- Perubahan requirement dapat merubah keseluruhan proses yang telah dilaksanakan.

#### **Tahap Yang diperlukan Pada Metode Waterfall**

##### **1. Requirement Analysis**

Sebelum melakukan pengembangan perangkat lunak, seorang pengembang harus mengetahui dan memahami bagaimana informasi kebutuhan pengguna terhadap sebuah perangkat lunak. Metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya. Informasi yang diperoleh kemudian diolah dan dianalisa sehingga didapatkan data atau informasi yang lengkap mengenai spesifikasi kebutuhan pengguna akan perangkat lunak yang akan dikembangkan.

##### **2. System and Software Design**

Informasi mengenai spesifikasi kebutuhan dari tahap Requirement Analysis selanjutnya di analisa pada tahap ini untuk kemudian diimplementasikan pada desain pengembangan. Perancangan desain dilakukan dengan tujuan membantu memberikan gambaran lengkap mengenai apa yang

harus dikerjakan. Tahap ini juga akan membantu pengembang untuk menyiapkan kebutuhan hardware dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat secara keseluruhan.

### **3. Implementation and Unit Testing**

Tahap implementation and unit testing merupakan tahap pemrograman. Pembuatan perangkat lunak dibagi menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Disamping itu, pada fase ini juga dilakukan pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah sudah memenuhi kriteria yang diinginkan atau belum.

### **4. Integration and System Testing**

Setelah seluruh unit atau modul yang dikembangkan dan diuji di tahap implementasi selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.

### **5. Operation and Maintenance**

Pada tahap terakhir dalam Metode Waterfall, perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya. Pemeliharaan meliputi perbaikan kesalahan, perbaikan implementasi unit sistem, dan peningkatan dan penyesuaian sistem sesuai dengan kebutuhan.

## **K. Perbandingan Metode Prototype**

**Metode Prototype** adalah teknik pengembangan sistem yang menggunakan prototype untuk menggambarkan sistem sehingga klien atau pemilik sistem mempunyai gambaran jelas pada sistem yang akan dibangun oleh tim pengembang.

### **Tahapan Dalam Metode Prototyp**

#### **Requirements Gathering and Analysis (Analisis Kebutuhan)**

Tahapan model prototype dimulai dari analisis kebutuhan. Dalam tahap ini kebutuhan sistem didefinisikan dengan rinci. Dalam prosesnya, klien dan tim developer akan bertemu untuk mendiskusikan detail sistem seperti apa yang diinginkan oleh user.

#### **Quick Design (Desain cepat)**

Tahap kedua adalah pembuatan desain sederhana yang akan memberi gambaran singkat tentang sistem yang ingin dibuat. Tentunya berdasarkan diskusi dari langkah 1 diawal.

#### **Build Prototype (Bangun Prototipe)**

Setelah desain cepat disetujui selanjutnya adalah pembangunan prototipe sebenarnya yang akan dijadikan rujukan tim programmer untuk pembuatan program atau aplikasi.

### **User Evaluation (Evaluasi Pengguna Awal)**

Di tahap ini, sistem yang telah dibuat dalam bentuk prototipe di presentasikan pada klien untuk di evaluasi. Selanjutnya klien akan memberikan komentar dan saran terhadap apa yang telah dibuat.

### **Refining Prototype (Memperbaiki Prototipe)**

Jika klien tidak mempunyai catatan revisi dari prototipe yang dibuat, maka tim bisa lanjut pada tahapan 6, namun jika klien mempunyai catatan untuk perbaikan sistem, maka fase 4-5 akan terus berulang sampai klien setuju dengan sistem yang akan dikembangkan.

### **Implement Product and Maintain (Implentasi dan Pemeliharaan)**

Pada fase akhir ini, produk akan segera dibuat oleh para programmer berdasarkan prototipe akhir, selanjutnya sistem akan diuji dan diserahkan pada klien. Selanjutnya adalah fase pemeliharaan agar sistem berjalan lancar tanpa kendala.

### **Kelebihan Metode Prototype**

Pertama-tama penting untuk memahami metode prototipe yang paling baik digunakan ketika sistem yang diinginkan perlu memiliki banyak interaksi dengan pengguna akhir.

Saat menggunakan model jenis ini, kesalahan biasanya dapat dideteksi lebih cepat dan umpan balik pengguna yang lebih cepat tersedia untuk menghasilkan solusi yang lebih baik. Dalam metodologi ini model kerja dari sistem disediakan, pengguna mendapatkan pemahaman yang lebih baik tentang sistem yang sedang dikembangkan.

Developer bisa bekerja menentukan kebutuhan klien dengan baik, Efisiensi waktu tinggi dalam pengembangan sistem serta Lebih mudah dalam penerapannya karena klien mengetahui apa yang dibutuhkan.

### **Kekurangan Metode Prototype**

Metode ini dapat meningkatkan kompleksitas. Rencana Anda mungkin mulai melampaui rencana awal Anda. Selain itu, Fokus pada prototipe terbatas dapat mengalihkan pengembang dari analisis lengkap proyek dengan benar. Namun itulah mengapa ada tahap penyempurnaan.

Klien terus menerus menambah requirement dari sistem, pegen dibuatkan yang seperti inilah seperti itulah, sehingga menambah kompleksitas pembuatan sistem. Sistem akan terhambat jika komunikasi kedua belah pihak tidak berjalan secara efektif.



## **L. Perbandingan Dengan Metode RAD**

**RAD** adalah salah satu metode pengembangan aplikasi yang kerap dipakai saat ini. Metode ini menekankan pada proses pembuatan aplikasi berdasarkan pembuatan prototype, iterasi, dan feedback yang berulang-ulang.

### **Tahapan RAD yang Perlu Dilakukan**

#### **1. Menentukan kebutuhan proyek**

RAD dimulai dengan menentukan kebutuhan sebuah proyek (project requirements). Pada tahap ini, tim perlu menentukan kebutuhan yang ingin dipenuhi dari sebuah proyek.

Kebutuhan ini tidak perlu spesifik. Tapi, sifatnya benar-benar umum dan jumlahnya bisa banyak. Baru dari situ, tim akan menentukan mana kebutuhan yang perlu diprioritaskan.

Setelah mendapatkan kebutuhan yang jelas, barulah tim menentukan hal-hal yang lebih detail. Misalkan seperti tujuan, timeline, dan budget yang diperlukan.

Semua anggota tim juga perlu memikirkan apa saja masalah yang mungkin muncul dalam proses pengembangan aplikasi. Tak lupa, mereka juga memikirkan strategi yang tepat untuk menyelesaikannya.

Intinya, tahap awal ini berguna untuk memberikan gambaran luas pada proyek yang ingin dikerjakan. Dalam prosesnya, tim bisa saja beralih ke hal-hal yang lebih spesifik.

#### **2. Membuat prototype**

Hal yang selanjutnya dilakukan adalah membuat prototype.

Developer secepat mungkin akan membuat prototype dari aplikasi yang diinginkan. Lengkap dengan fitur dan fungsi yang berbeda-beda.

Tujuannya, sekadar untuk mengecek apakah prototype yang dibuat sudah sesuai dengan kebutuhan klien.

Meski begitu, tahap ini bisa saja dilakukan berulang-ulang. Kadang juga melibatkan user untuk testing dan memberikan feedback.

Proses ini memungkinkan tim mempelajari error yang mungkin muncul ke depannya. Ini berguna untuk mengurangi error dan debugging.

Lewat tahapan ini, tim developer memiliki modal untuk membuat aplikasi yang mudah dipakai, stabil, tidak sering error, dan desainnya pun oke.

#### **3. Proses pengembangan dan pengumpulan feedback**

Setelah tahu aplikasi seperti apa yang ingin dibuat, developer mengubah prototype ke bentuk aplikasi versi beta sampai dengan final.

Jadi, bisa dibilang tahap RAD inilah yang cukup intens. Developer terus-menerus melakukan coding aplikasi, melakukan testing sistem, dan integrasi dengan bagian-bagian lainnya.

Karena itulah, developer menggunakan tools dan framework yang mendukung RAD agar cepat.

Apalagi proses ini terus diulang sambil terus mempertimbangkan feedback dari klien. Baik itu soal fitur, fungsi, interface, sampai keseluruhan aspek dari produk yang dibuat.

Nah, kalau prosesnya berjalan lancar, developer akan melanjutkan ke langkah berikutnya. Yaitu, finalisasi produk atau implementasi.

Kalau pun tidak, proses ini kemungkinan akan terus diulang. Pun, kalau apes-apesnya aplikasi tidak menjawab kebutuhan, developer akan kembali ke proses prototyping.

#### **4. Implementasi dan finalisasi produk**

Di sini, tugas utama developer adalah menambal kekurangan yang mungkin terjadi ketika proses pengembangan aplikasi.

Tugas ini termasuk melakukan optimasi untuk stabilitas aplikasinya, memperbaiki interface, hingga melakukan maintenance dan menyusun dokumentasi.

<b>Kelebihan RAD</b>	<b>Kekurangan RAD</b>
Kebutuhan aplikasi bisa berubah sewaktu-waktu.	Membutuhkan tim dengan skill teknis yang mumpuni.
Aplikasi dikembangkan berdasarkan kebutuhan dan keinginan user.	Memerlukan kolaborasi tim yang kuat.
Memperkecil kemungkinan error dan hal buruk lainnya.	Hanya cocok untuk proyek yang waktunya singkat.
Waktu pengembangan aplikasi bisa lebih cepat dan efektif.	Hanya cocok digunakan untuk mengembangkan aplikasi secara modular (fokus ke suatu fitur untuk dijadikan module terpisah).
Mempermudah proses integrasi.	Sulit diterapkan untuk mengembangkan aplikasi besar.