

MIKROTIK ROUTEROS SECURITY AUTOMATION WITH ANSIBLE



I PUTU HARIYADI

putu.hariyadi@universitasbumigora.ac.id

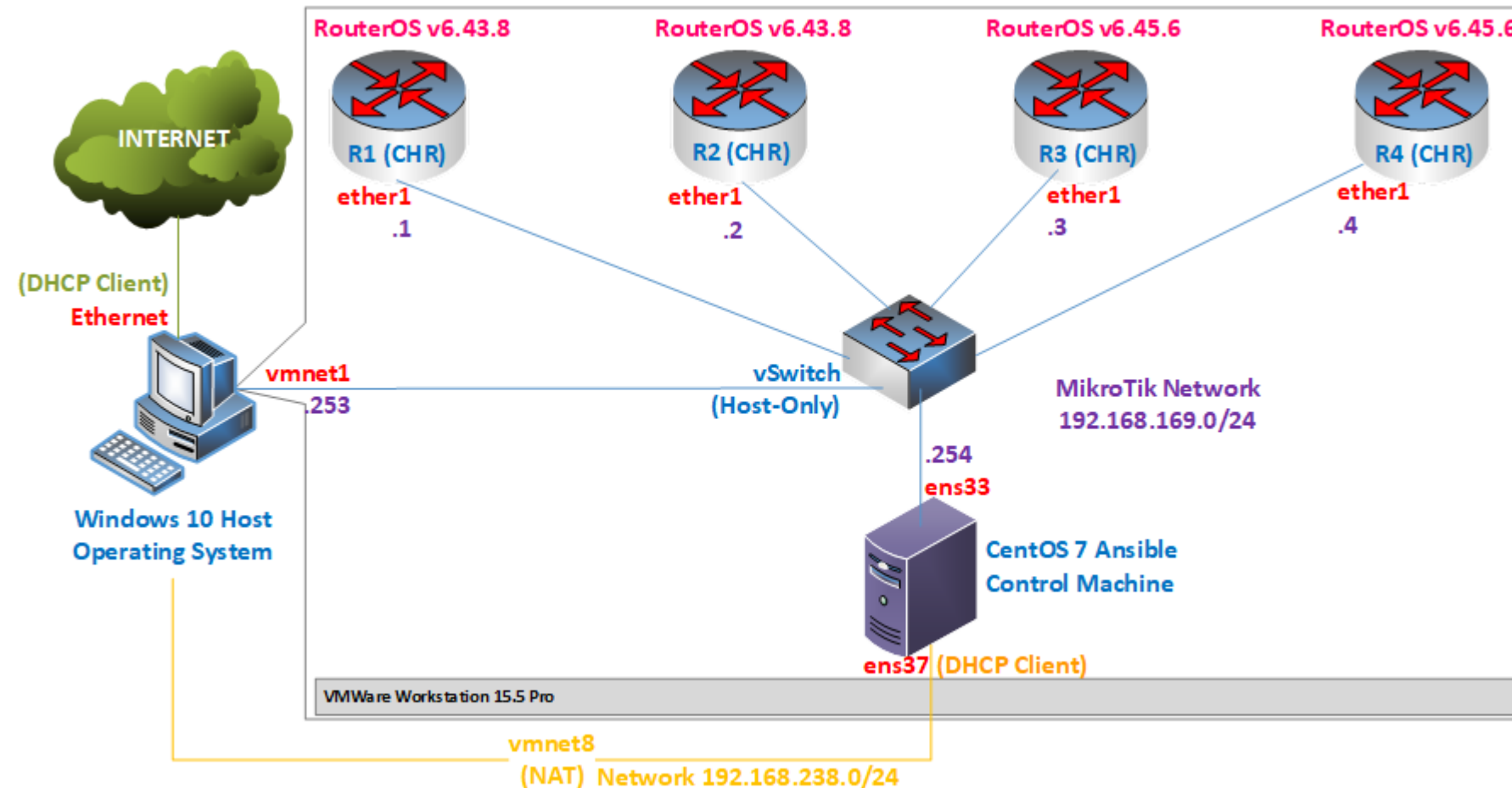
www.iputuhariyadi.net



MATERI

1. Pengamanan Router MikroTik.
2. Manajemen Konfigurasi dan Otomatisasi dengan Ansible.
3. Studi Kasus Ansible Playbook untuk mengotomatisasi keamanan MikroTik RouterOS.

RANCANGAN JARINGAN UJICоба



Rancangan jaringan ujicoba disimulasikan menggunakan VMWare Workstation 15.5 Pro. Terdapat 5 (lima) Virtual Machine (VM) yang dibuat yaitu:

- 1 (satu) VM Ansible Control Machine menggunakan CentOS 7. Interface **ens33** menggunakan *network connection mode host-only (vmnet1)*. Sedangkan untuk **ens37** menggunakan mode NAT (vmnet8).
- 4 (empat) VM MikroTik RouterOS dengan 5 (lima) interface yaitu **ether1**, **ether2**, **ether3**, **ether4** dan **ether5** menggunakan mode *host-only (vmnet1)*.

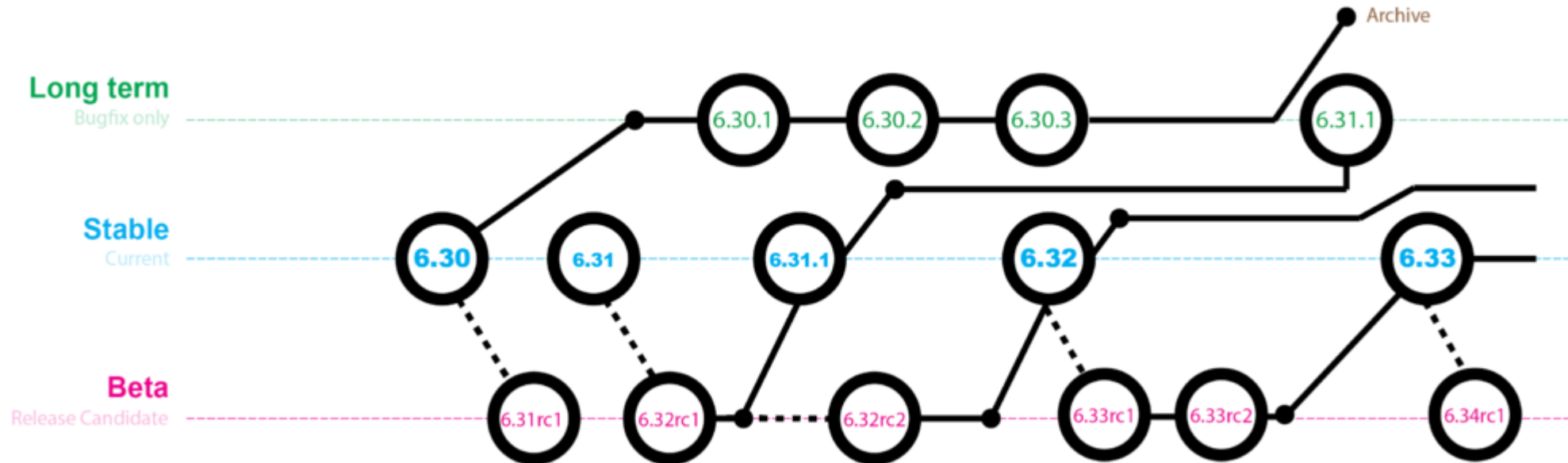


PENGAMANAN ROUTER MIKROTIK

VERSI ROUTEROS

- Disarankan untuk menggunakan versi *RouterOS* terkini.
- Beberapa rilis *RouterOS* versi lama memiliki kelemahan atau kerentanan yang telah diperbaiki.
- MikroTik secara berkala menambahkan fungsionalitas baru dan meningkatkan unjuk kerja serta stabilitas dari *RouterOS* dengan merilis pembaharuan.
- Lakukan pembaharuan versi *RouterOS* dari perangkat yang digunakan untuk memastikan keamanannya.

PENOMORAN VERSI ROUTEROS



(Image Source: <https://wiki.mikrotik.com/wiki/File:Metro-systemv2.png>)

Versi RouterOS memiliki 3 (tiga) *release chains*:

- **Beta:** dirilis setiap beberapa hari dan hanya menjalani pengujian internal dasar.
- **Stable:** dirilis setiap beberapa minggu dan mencakup seluruh fitur serta perbaikan yang teruji.
- **Long Term:** jarang dirilis dan mencakup hanya perbaikan yang sangat penting serta peningkatan dalam satu nomor percabangan namun tidak menambah fitur baru.

PEMBAHARUAN ROUTEROS

- Terdapat 2 (dua) metode **Upgrade** yaitu **Automatic Upgrade** dan **Manual Upgrade**.
- Fitur **Automatic Upgrade** akan menghubungkan ke **MikroTik download servers** dan mengecek apakah terdapat versi *RouterOS* yang lebih baru untuk perangkat yang digunakan saat ini.
- Terdapat beberapa cara yang dapat digunakan untuk melakukan pembaharuan *RouterOS* secara otomatis yaitu melalui:
 - **Quickset**
 - **System Packages**

METODE PEMBAHARUAN OTOMATIS



admin@00:0C:29:68:EA:F6 (R1) - WinBox v6.43.8 on CHR (x86_64)

Session Settings Dashboard

Safe Mode Session: 00:0C:29:68:EA:F6

Quick Set

Ethernet Quick Set

Configuration

Mode: Router Bridge

Internet

Address Acquisition: Static Automatic PPPoE

IP Address: Renew Release

Netmask:

Gateway:

MAC Address: 00:0C:29:68:EA:F6

Local Network

IP Address: 0.0.0.0

Netmask: 255.255.255.0 (/24)

VPN

VPN Address: 192.168.169.1

System

Router Identity: R1

Check For Updates Reset Configuration

Password...

active

- Quickset
- System Packages

Package List

Check For Updates Enable Disable Uninstall Unschedule Downgrade Check Installation Find

Name	Version	Build Time	Scheduled
dude	6.43.8	Dec/21/2018 07:10:42	
rou			

Check For Updates

Channel: stable

OK

Installed Version: 6.43.8

Latest Version: 6.47.1

Download

Download&Install

What's new in 6.47.1 (2020-Jul-08 12:34):

- *) crs3xx - fixed HW offloading for netPower 15FR and netPower 16P devices (introduced in v6.47);
- *) crs3xx - fixed increased CPU temperature for CRS354-48G-4S+2Q+ device (introduced in v6.47);
- *) crs3xx - improved Ethernet port group traffic forwarding for CRS354 devices;
- *) defconf - fixed default configuration generation on devices without "wireless" package installed;

13 items

ROUTEROS AUTO-UPGRADE

- Tersedia mulai *RouterOS* versi 6.
- Otomatisasi proses pembaharuan dapat dilakukan dengan menjalankan skrip pada **System Scheduler**.
- Skrip untuk *RouterOS* **setelah versi 6.31**.

```
/system package update
check-for-updates once
:delay 3s;
:if ( [get status] = "New version is available") do={ install }
```

- Skrip untuk *RouterOS* **sampai versi 6.31**.

```
/system package update
check-for-updates
:delay 3s;
:if ( [get current-version] != [get latest-version]) do={ upgrade }
```

DEMO ROUTEROS AUTO-GRADE



METODE PEMBAHARUAN MANUAL

Terdapat beberapa cara yang dapat digunakan untuk melakukan pembaharuan *RouterOS* secara manual yaitu melalui:

- **Winbox**, dengan melakukan *drag & drop file routers-*.npk* ke menu **Files**.
- **WebFig**, dengan mengunggah *file routers-*.npk* melalui menu **Files**.
- **FTP**, dengan mengunggah *file routers-*.npk* ke *root directory*.
- **The Dude**

PROSES PEMBAHARUAN SECARA MANUAL



Awali dengan mengunduh *file upgrade package* **routeros-*.npk** sesuai dengan jenis sistem dari perangkat yang digunakan pada situs MikroTik <https://www.mikrotik.com/download>

The screenshot shows a web browser window with the address bar containing 'mikrotik.com/download'. The page header includes the MikroTik logo and navigation links: Home, About, Buy, Jobs, Hardware, Software (underlined), Support, Training, and Account. A blue navigation bar below the header contains the following items: Software, Downloads, Changelogs, Download archive, RouterOS, The Dude, and Mobile app. The main content area is titled 'Upgrading RouterOS' and contains the following text:

If you are already running RouterOS, upgrading to the latest version can be done by clicking on "Check For Updates" in **QuickSet** or **System > Packages** menu in WebFig or WinBox.

See the [documentation](#) for more information about upgrading and release types.

To manage your router, use the web interface, or download the maintenance utilities. Winbox to connect to your device, Dude to monitor your network and Netinstall for recovery and re-installation.

DEMO METODE PEMBAHARUAN SECARA MANUAL





PENGAMANAN AKSES KE ROUTER

1. Mengubah *default username* **admin** menggunakan **username** berbeda, sebagai contoh menggunakan **ansible**.

```
/user add name=ansible password=ansiblesecret group=full
```

```
/user remove admin
```

Atau

```
/user set name=ansible admin
```

2. Mengatur sandi (*password*) dari *user* dan menggunakan sandi yang aman, sebagai contoh menggunakan **ansiblesecret**.

```
/user set password=ansiblesecret admin
```

Atau

```
/password
```



PENGAMANAN AKSES KE ROUTER

3. Membatasi akses *username* agar hanya dapat digunakan login dari alamat IP sumber tertentu.

Sebagai contoh *username* *ansible* hanya dapat digunakan untuk mengakses *RouterOS* dari IP **192.168.169.254**.

```
/user set address=192.168.169.254 ansible
```

Name	Group	Allowed Address	Last Logged In
system default user			
ansible	full	192.168.169.254	Aug/02/2020 13:15:45

DEMO VERIFIKASI HASIL PENGAMANAN AKSES KE ROUTER



PENGAMANAN IP SERVICES DARI ROUTER



1. Menonaktifkan *IP Service* yang tidak aman, seperti **ftp**, **telnet**, **www**.

```
/ip service disable ftp,telnet,www
```

2. Menonaktifkan *IP Service* yang tidak digunakan, seperti **api**, **api-ssl**.

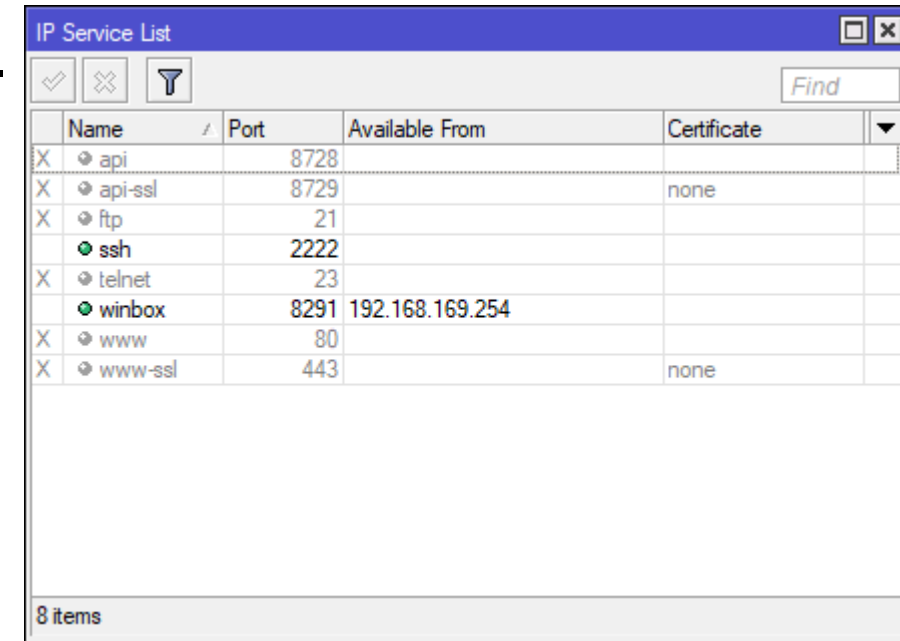
```
/ip service disable api,api-ssl
```

3. Mengubah nomor port dan membatasi akses pada *IP Service*.

```
/ip service set port=2222 ssh
```

```
/ip service set address=192.168.169.254 winbox
```

```
/ip service print
```



	Name	Port	Available From	Certificate
X	api	8728		
X	api-ssl	8729		none
X	ftp	21		
	ssh	2222		
X	telnet	23		
	winbox	8291	192.168.169.254	
X	www	80		
X	www-ssl	443		none

8 items

DEMO VERIFIKASI PENGAMANAN IP SERVICES DARI ROUTER



MENONAKTIFKAN ROUTEROS MAC-ACCESS



1. Menonaktifkan layanan **MAC-Telnet**.

```
[admin@R1] > /tool mac-server set allowed-interface-list=none
[admin@R1] > /tool mac-server print
  allowed-interface-list: none
```

2. Menonaktifkan layanan **MAC-Winbox**.

```
[admin@R1] > /tool mac-server mac-winbox set allowed-interface-list=none
[admin@R1] > /tool mac-server mac-winbox print
  allowed-interface-list: none
```

3. Menonaktifkan layanan **MAC-Ping**.

```
[admin@R1] > /tool mac-server ping set enabled=no
[admin@R1] > /tool mac-server ping print
  enabled: no_
```

DEMO VERIFIKASI PENONAKTIFAN ROUTER MAC-ACCESS





MENONAKTIFKAN NEIGHBOR DISCOVERY

- **Mikrotik Neighbor Discovery Protocol (MNDP)** digunakan untuk menemukan dan mengetahui perangkat Mikrotik lainnya yang terdapat di jaringan.
- Menonaktifkan *neighbor discovery* pada semua *interface*.

```
[admin@R1] > /ip neighbor discovery-settings set discover-interface-list=none  
[admin@R1] > /ip neighbor discovery-settings print  
discover-interface-list: none
```

DEMO VERIFIKASI PENONAKTIFAN MINDP





MENONAKTIFKAN BANDWIDTH SERVER

- **Bandwidth Server** digunakan untuk menguji **throughput** diantara 2 (dua) *router MikroTik*.
- Disarankan untuk menonaktifkan **Bandwidth Test (Btest) Server** pada lingkungan *production*.

```
[admin@R1] > /tool bandwidth-server set enabled=no
[admin@R1] > /tool bandwidth-server print
                enabled: no
                authenticate: yes
allocate-udp-ports-from: 2000
                max-sessions: 100
```

DEMO VERIFIKASI PENONAKTIFAN BTEST SERVER





MENONAKTIFKAN DNS CACHE

- **Domain Name System (DNS) Cache** digunakan untuk meminimalkan permintaan DNS ke server DNS eksternal sehingga meminimalkan waktu resolusi atau pemetaan DNS.
- Apabila tidak diperlukan maka **DNS Cache** dapat dinonaktifkan.

```
[admin@R1] > /ip dns set allow-remote-requests=no
[admin@R1] > /ip dns print
        servers: 8.8.8.8,8.8.4.4
        dynamic-servers:
        allow-remote-requests: no
        max-udp-packet-size: 4096
        query-server-timeout: 2s
        query-total-timeout: 10s
        max-concurrent-queries: 100
max-concurrent-tcp-sessions: 20
        cache-size: 2048KiB
        cache-max-ttl: 1w
        cache-used: 17KiB
_
```



MENONAKTIFKAN CLIENT SERVICES

- Secara *default service* MikroTik Caching Proxy, MikroTik Socks Proxy, MikroTik (UPNP) Service dan MikroTik Dynamic Name Service tidak aktif. Namun apabila service tersebut telah dikonfigurasi dan tidak diperlukan atau tidak digunakan maka sangat disarankan untuk dinonaktifkan.
- Menonaktifkan MikroTik Caching Proxy.

```
[admin@R1] > /ip proxy set enabled=no
[admin@R1] > /ip proxy print
      enabled: no
      src-address: ::
          port: 8080
      anonymous: no
      parent-proxy: ::
      parent-proxy-port: 0
      cache-administrator: webmaster
      max-cache-size: unlimited
      max-cache-object-size: 2048KiB
      cache-on-disk: no
      max-client-connections: 600
      max-server-connections: 600
          max-fresh-time: 3d
      serialize-connections: no
      always-from-cache: no
      cache-hit-dscp: 4
      _cache-path: web-proxy
```



MENONAKTIFKAN CLIENT SERVICES

- Menonaktifkan **MikroTik Socks Proxy**.

```
[admin@R1] > /ip socks set enabled=no
[admin@R1] > /ip socks print
                enabled: no
                port: 1080
connection-idle-timeout: 2m
                max-connections: 200
```

- Menonaktifkan **MikroTik Universal Plug and Play (UPNP) Service**.

```
[admin@R1] > /ip upnp print
                enabled: no
allow-disable-external-interface: no
                show-dummy-rule: yes
```



MENONAKTIFKAN CLIENT SERVICES

- Menonaktifkan MikroTik Dynamic Name Service atau IP Cloud.

```
[admin@R1] > /ip cloud set ddns-enabled=no update-time=no  
[admin@R1] > /ip cloud print  
ddns-enabled: no  
update-time: no
```



MENGAKTIFKAN STRONG CRYPTO SSH

- RouterOS mendukung penggunaan **crypto** yang lebih tangguh untuk SSH namun secara *default* belum aktif.

`strong-crypto` (yes | no; Default: no)

Use stronger encryption, HMAC algorithms, use bigger DH primes and disallow weaker ones:

- prefer 256 and 192 bit encryption instead of 128 bits;
- disable null encryption;
- prefer sha256 for hashing instead of sha1;
- disable md5;
- use 2048bit prime for Diffie Hellman exchange instead of 1024bit.

- Mengaktifkan **Strong Crypto SSH**:

```
[admin@R1] > /ip ssh print
      forwarding-enabled: no
always-allow-password-login: no
      strong-crypto: yes
      host-key-size: 2048
```



MENONAKTIFKAN INTERFACE ROUTER

- Disarankan untuk menonaktifkan seluruh *interface* yang tidak digunakan pada *router* sehingga mengurangi akses yang tidak sah atau tidak terotorisasi.
- Sebagai contoh untuk menonaktifkan *interface* **ether2**, **ether3**, **ether4** dan **ether5** pada *router* maka dapat mengeksekusi perintah berikut:

```
[admin@R1] > /interface set disabled=yes ether2,ether3,ether4,ether5
```

```
[admin@R1] > /interface print
```

```
Flags: D - dynamic, X - disabled, R - running, S - slave
```

#	NAME	TYPE	ACTUAL-MTU	L2MTU	MAX-L2MTU
0	R ether1	ether	1500		
1	X ether2	ether	1500		
2	X ether3	ether	1500		
3	X ether4	ether	1500		
4	X ether5	ether	1500		



MANAJEMEN KONFIGURASI DAN OTOMATISASI DENGAN ANSIBLE





APA ITU ANSIBLE?

- Menurut situs [Ansible](#), **Ansible** merupakan mesin otomatisasi Teknologi Informasi (TI) sederhana yang dapat mengotomatisasi *cloud provisioning*, manajemen konfigurasi, penerapan aplikasi, *intra-service orchestration* dan kebutuhan TI lainnya.
- Menurut situs [Edureka](#), keuntungan menggunakan *Ansible* antara lain:
 1. **Simple**, menggunakan sintak penulisan yang ditulis menggunakan **YAML** yang disebut dengan **playbook**.
 2. **Agentless**, tidak diperlukan **agent** atau *software* khusus untuk diinstalasi pada host yang diautomasi.
 3. **Powerful & Flexible**, memiliki banyak modul untuk manajemen infrastruktur, jaringan, sistem operasi dan layanan.
 4. **Efficient**, tidak ada *software* yang diperlukan untuk instalasi pada server sehingga lebih banyak sumber daya yang dapat digunakan oleh aplikasi.



APA YANG DAPAT DILAKUKAN ANSIBLE?

1. Provisioning

Ansible memastikan paket-paket yang dibutuhkan akan diunduh dan diinstalasi sehingga aplikasi dapat diterapkan.

2. Configuration Management

Menetapkan dan mempertahankan konsistensi dari kinerja produk dengan mencatat dan memperbaharui informasi lengkap yang menjelaskan perangkat keras dan lunak perusahaan. Seperti versi dan pembaharuan yang telah diterapkan pada paket *software* yang terinstal dan lokasi serta alamat jaringan dari perangkat keras.

3. Application Deployment

Ansible dapat memajemen secara efektif keseluruhan *life cycle* dari aplikasi, mulai dari *development* sampai produksi.

4. Security and Compliance

Kebijakan keamanan dapat didefinisikan pada *Ansible* sehingga proses pemindaian dan pemulihan kebijakan ke lokasi dapat diintegrasikan ke dalam proses secara otomatis.



APA YANG DAPAT DILAKUKAN ANSIBLE?

5. Orchestration

Ansible menyediakan *orchestration* dalam arti menyelaraskan permintaan bisnis dengan aplikasi, data, dan infrastruktur. Ini mendefinisikan kebijakan dan tingkat layanan melalui alur kerja otomatis, penyediaan, dan manajemen perubahan sehingga menciptakan aplikasi selaras dengan infrastruktur yang dapat ditingkatkan dari atas ke bawah berdasarkan kebutuhan setiap aplikasi.



TERMINOLOGI PADA ANSIBLE

- **Controller Machine:** mesin dimana *ansible* diinstalasi dan bertanggungjawab untuk menjalankan *provisioning* pada *server* yang dikelola.
- **Inventory:** file inisialisasi yang memuat informasi tentang *server* yang dikelola.
- **Playbook:** Titik masuk untuk *Ansible provisioning*, dimana otomatisasi didefinisikan melalui tugas (*tasks*) menggunakan format YAML..
- **Task:** Blok yang mendefinisikan satu prosedur untuk dieksekusi, sebagai contoh instalasi *package* tertentu.
- **Module:** Modul merupakan abstraksi dari tugas sistem, seperti berkaitan dengan *package* atau membuat dan mengubah file. *Ansible* memiliki banyak modul *built-in* (bawaan), namun dapat juga dibuat modul khusus.



TERMINOLOGI PADA ANSIBLE

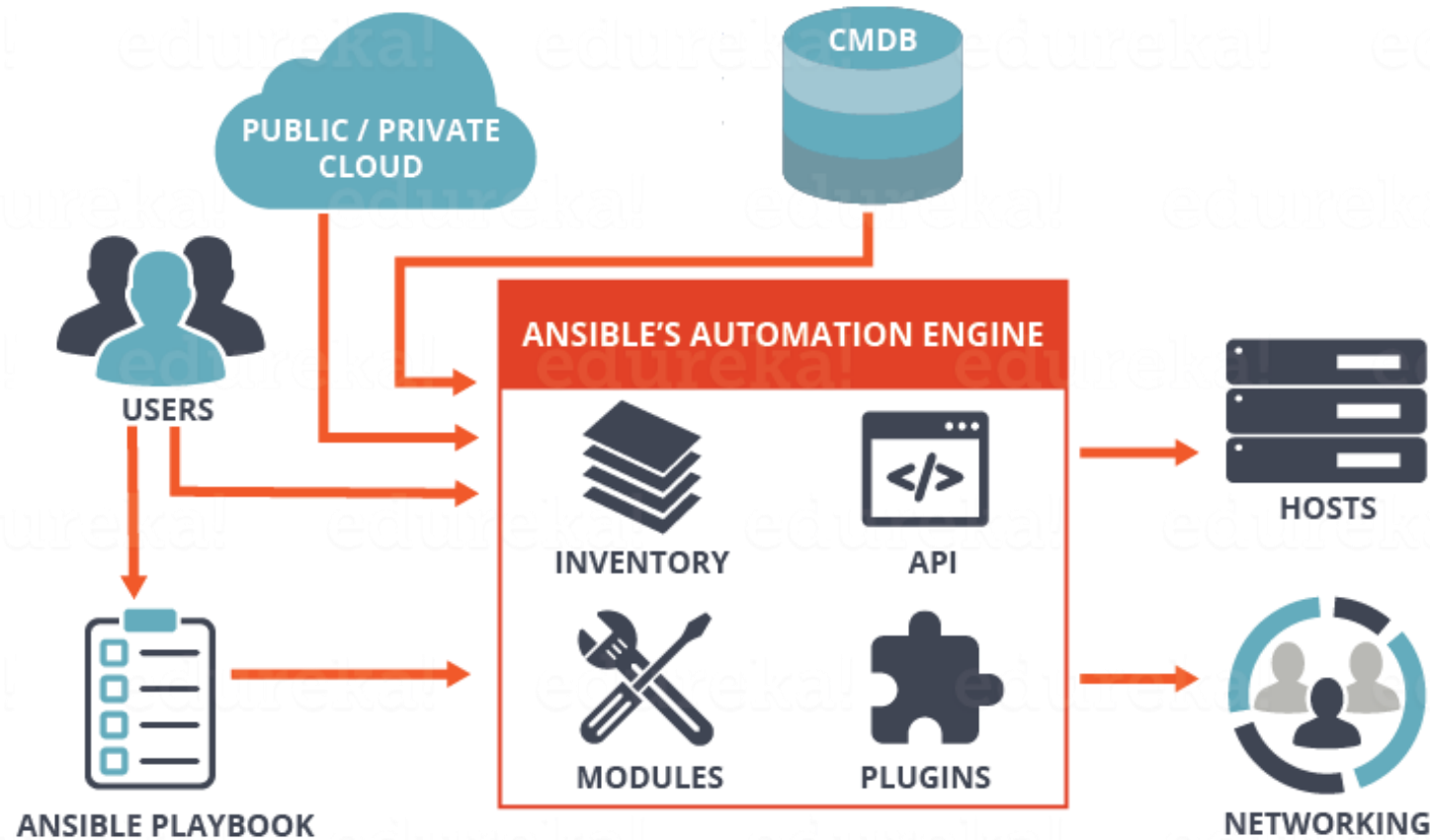
- **Role:** Cara yang telah ditentukan sebelumnya untuk mengatur *playbook* dan file lainnya untuk memfasilitasi berbagi pakai dan menggunakan kembali bagian dari *provisioning*.
- **Play:** *provisioning* yang dieksekusi mulai dari awal sampai akhir disebut dengan *play*. Dengan kata lain, eksekusi dari *playbook* disebut dengan *play*.
- **Facts:** *variable* global yang memuat informasi tentang sistem, seperti *interface* jaringan atau sistem operasi.
- **Handlers:** Digunakan untuk memicu perubahan status dari *service*, seperti *restart* atau menghentikan *service*.



ARSITEKTUR ANSIBLE

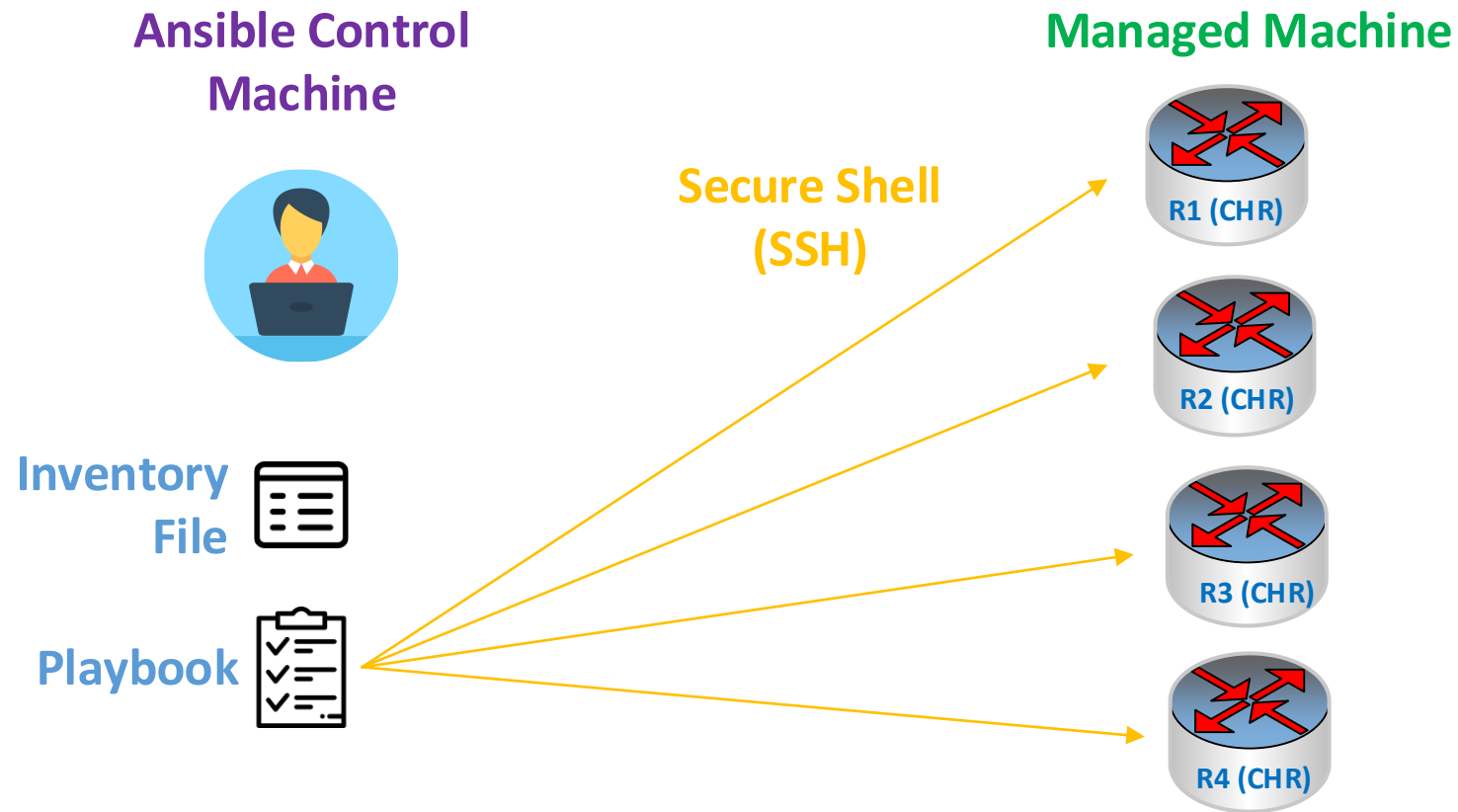
ANSIBLE ARCHITECTURE

edureka!



(Image source: <https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2016/11/Ansible-Architecture-What-Is-Ansible-Edureka-768x449.png>)

MIKROTIK SECURITY AUTOMATION WITH ANSIBLE





ANSIBLE MODULES (1)

- **raw**

Executes a low-down and dirty SSH command, not going through the module subsystem. (https://docs.ansible.com/ansible/latest/modules/raw_module.html)

The screenshot shows a web browser displaying the Ansible documentation page for the 'raw' module. The browser's address bar shows the URL 'docs.ansible.com/ansible/latest/modules/raw_module.html'. The page header includes 'Documentation' and navigation links for 'ANSIBLEFEST', 'PRODUCTS', 'COMMUNITY', 'WEBINARS & TRAINING', and 'BLOG'. The main content area features the title 'raw – Executes a low-down and dirty command' and a list of links for 'Synopsis', 'Parameters', 'Notes', 'See Also', 'Examples', and 'Status'. A 'Synopsis' section is visible at the bottom, containing the text: 'Executes a low-down and dirty SSH command, not going through the module subsystem.' The left sidebar contains a search bar and navigation links for 'INSTALLATION, UPGRADE & CONFIGURATION', 'USING ANSIBLE', and 'CONTRIBUTING TO ANSIBLE'.

ANSIBLE MODULES (2)



▪ ping

Try to connect to host, verify a usable python and return pong on success.
(https://docs.ansible.com/ansible/latest/modules/ping_module.html)

The screenshot shows a web browser displaying the Ansible documentation page for the 'ping' module. The browser's address bar shows the URL 'docs.ansible.com/ansible/latest/modules/ping_module.html'. The page header includes 'Documentation' and navigation links for 'ANSIBLEFEST', 'PRODUCTS', 'COMMUNITY', 'WEBINARS & TRAINING', and 'BLOG'. The main content area features a breadcrumb trail 'Docs » ping - Try to connect to host, verify a usable python and return pong on success' and an 'Edit on GitHub' link. The title of the page is 'ping - Try to connect to host, verify a usable python and return pong on success'. A table of contents lists 'Synopsis', 'Parameters', 'See Also', 'Examples', 'Return Values', and 'Status'. The 'Synopsis' section contains two bullet points: 'A trivial test module, this module always returns pong on successful contact. It does not make sense in playbooks, but it is useful from /usr/bin/ansible to verify the ability to login and that a usable Python is configured.' and 'This is NOT ICMP ping, this is just a trivial test module that requires Python on the remote-node.'

ANSIBLE MODULES (3)



▪ **routeros_command**

Run commands on remote devices running MikroTik RouterOS.

(https://docs.ansible.com/ansible/latest/modules/routeros_command_module.html)

The screenshot shows a web browser displaying the Ansible documentation page for the `routeros_command` module. The page title is "routeros_command - Run commands on remote devices running MikroTik RouterOS". The page content includes a search bar with "routeros" entered, a navigation menu on the left, and a main content area with the following sections:

- routeros_command - Run commands on remote devices running MikroTik RouterOS**
- New in version 2.7.*
- Synopsis**
- Parameters**
- Examples**
- Return Values**
- Status**

The **Synopsis** section contains the following text:

- Sends arbitrary commands to an RouterOS node and returns the results read from the device. This module includes an argument that will cause the module to wait for a specific condition before returning or timing out if the condition is not met.

ANSIBLE MODULES (4)



▪ **routeros_command** Parameters.

The screenshot shows the Ansible documentation page for the `routeros_command` module. The page is titled "Parameters" and contains a table with the following data:

Parameter	Choices/Defaults	Comments
<code>commands</code> - / required		List of commands to send to the remote RouterOS device over the configured provider. The resulting output from the command is returned. If the <code>wait_for</code> argument is provided, the module is not returned until the condition is satisfied or the number of retries has expired.
<code>interval</code> -	Default: 1	Configures the interval in seconds to wait between retries of the command. If the command does not pass the specified conditions, the interval indicates how long to wait before trying the command again.
<code>match</code> -	Choices: <ul style="list-style-type: none">anyall ←	The <code>match</code> argument is used in conjunction with the <code>wait_for</code> argument to specify the match policy. Valid values are <code>all</code> or <code>any</code> . If the value is set to <code>all</code> then all conditionals in the <code>wait_for</code> must be satisfied. If the value is set to <code>any</code> then only one of the values must be satisfied.
<code>retries</code> -	Default: 10	Specifies the number of retries a command should be tried before it is considered failed. The command is run on the target device every retry and evaluated against the <code>wait_for</code> conditions.
<code>wait_for</code> -		List of conditions to evaluate against the output of the command. The task will wait for each condition to be true before moving forward. If the conditional is not true within the configured number of retries, the task fails. See examples.

Below the table, there is an "Examples" section with the following code snippet:

```
tasks:  
  - name: run command on remote devices  
    routeros_command:  
      commands: /system routerboard print
```

ANSIBLE MODULES (5)



- **debug**

Print statements during execution.

(https://docs.ansible.com/ansible/latest/modules/debug_module.html)

The screenshot shows a web browser displaying the Ansible documentation page for the 'debug' module. The browser's address bar shows the URL 'docs.ansible.com/ansible/latest/modules/debug_module.html'. The page header includes 'Documentation' and navigation links for 'ANSIBLEFEST', 'PRODUCTS', 'COMMUNITY', 'WEBINARS & TRAINING', and 'BLOG'. The main content area features a breadcrumb trail 'Docs » debug - Print statements during execution' and an 'Edit on GitHub' link. The title of the page is 'debug - Print statements during execution'. A list of links is provided: 'Synopsis', 'Parameters', 'Notes', 'See Also', 'Examples', and 'Status'. The 'Synopsis' section contains a single bullet point: 'This module prints statements during execution and can be useful for debugging variables or expressions without necessarily halting the playbook.'



INSTALASI ANSIBLE PADA CENTOS 7 SEBAGAI CONTROL MACHINE

1. Menginstalasi *repository* **Extra Packages for Enterprise Linux (EPEL)** karena **Ansible** tidak secara *default* terdapat pada **Yum Repositories**.

```
# yum -y install epel-release
```

2. Menginstalasi **Ansible**

```
# yum -y install ansible
```

3. Menampilkan informasi versi **Ansible** yang telah terinstalasi.

```
# ansible --version
```

```
[root@ansible ~]# ansible --version
ansible 2.9.1
  config file = /etc/ansible/ansible.cfg
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plug
ins/modules']
  ansible python module location = /usr/lib/python2.7/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.7.5 (default, Oct 30 2018, 23:45:53) [GCC 4.8.5 20150623 (Red Hat 4.8.5-36
)]
```

MENGATUR ANSIBLE INVENTORY



- **Inventory** merupakan file inisialisasi yang digunakan oleh *Ansible* untuk mendaftarkan dan mengelompokkan mesin atau *host* yang akan dikelola. Lokasi dari file **inventory** secara *default* adalah “**/etc/ansible/hosts**”.
- Menambahkan **managed machines** ke file *inventory* menggunakan editor **nano**.
nano **/etc/ansible/hosts**

```
[routers]  
R1 ansible_host=192.168.169.1  
R2 ansible_host=192.168.169.2  
R3 ansible_host=192.168.169.3  
R4 ansible_host=192.168.169.4
```

Variable **ansible_host** digunakan untuk mengatur alamat IP dari target atau *managed machine* yaitu dalam hal ini adalah alamat IP dari setiap *router*. Sebagai contoh **R1** merupakan *inventory_hostname* atau alias bagi *router* dengan alamat **192.168.169.1**.

Simpan perubahan dengan menekan **CTRL+O** dan tekan **Enter**. Keluar dari editor *nano* dengan menekan **CTRL+X**.



VERIFIKASI ANSIBLE INVENTORY

- Memverifikasi hasil penambahan *inventory*.

```
# tail -n 5 /etc/ansible/hosts
```

```
[root@ansible ~]# tail -n 5 /etc/ansible/hosts  
[routers]  
R1 ansible_host=192.168.169.1  
R2 ansible_host=192.168.169.2  
R3 ansible_host=192.168.169.3  
R4 ansible_host=192.168.169.4
```

MENGATUR PUBLIC KEY AUTHENTICATION UNTUK SSH DENGAN MEMBUAT KEY PAIR



Membuat **key pair** menggunakan utilitas **ssh-keygen** pada **Ansible Control Machine**.

ssh-keygen

```
Generating public/private rsa key pair.  
Enter file in which to save the key (/root/.ssh/id_rsa):
```

Tekan **Enter** pada inputan **Enter file in which to save the key (/root/.ssh/id_rsa):** untuk menggunakan lokasi dan nama file *default* penyimpanan *key* yaitu **/root/.ssh/id_rsa**.

```
Created directory '/root/.ssh'.  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

Tekan **Enter** pada inputan “**Enter passphrase (empty for no passphrase):**” dan “**Enter same passphrase again:**” yang tampil untuk apabila ingin mengosongkan **passphrase**. Tampil proses pembuatan *key*. Tunggu hingga proses selesai dilakukan.

MENYALINKAN PUBLIC KEY DARI ANSIBLE CONTROL MACHINE KE SELURUH ROUTER



- Penyalinan **Public Key** dari **CentOS 7 Ansible Control Machine** ke *router* **R1** dapat dilakukan dengan menggunakan utilitas **scp** (*secure copy*).

```
# scp /root/.ssh/id_rsa.pub admin@R1:id_rsa.pub
```

Ketik **yes** dan tekan **Enter** pada pesan konfirmasi **Are you sure you want to continue connecting (yes/no)?**.

```
[root@ansible ~]# scp /root/.ssh/id_rsa.pub admin@R1:id_rsa.pub
The authenticity of host 'r1 (192.168.169.1)' can't be established.
RSA key fingerprint is SHA256:Ka/w5ECmXBI3zPZLxQ/gZP5BAZq2ch9KZfbWeMi59d8.
RSA key fingerprint is MD5:79:ab:65:3c:53:c6:23:68:cb:f2:5b:5f:1f:25:28:b0.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'r1,192.168.169.1' (RSA) to the list of known hosts.
admin@r1's password:
id_rsa.pub 100% 404 459.8KB/s 00:00
```

Selanjutnya tampil inputan **password** untuk *user* **admin** dari **managed machine R1** dengan alamat IP **192.168.169.1**. Masukkan sandi dari *user* **admin** dan tekan **Enter**.

MENYALINKAN PUBLIC KEY DARI ANSIBLE CONTROL MACHINE KE SELURUH ROUTER



- Dengan cara yang sama, lakukan untuk **managed machine R2, R3 dan R4**.
- Proses penyalinan **Public Key** ke **R2 dan R3**.

```
[root@ansible ~]# scp /root/.ssh/id_rsa.pub admin@R2:id_rsa.pub
The authenticity of host 'r2 (192.168.169.2)' can't be established.
RSA key fingerprint is SHA256:3Tvs1AY9n06jJ2Yjv6LsI2G4oIyh8q/gERsWw8nu+c.
RSA key fingerprint is MD5:6c:75:30:6b:a2:d4:a8:d3:e2:f3:23:19:42:9e:13:ba.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'r2,192.168.169.2' (RSA) to the list of known hosts.
admin@r2's password:
id_rsa.pub                               100% 404   482.6KB/s   00:00
[root@ansible ~]# scp /root/.ssh/id_rsa.pub admin@R3:id_rsa.pub
The authenticity of host 'r3 (192.168.169.3)' can't be established.
RSA key fingerprint is SHA256:3CwjIPB0EY0to8mVReA5ylAMLANemd5wy+v3Iw/8gew.
RSA key fingerprint is MD5:b1:3d:28:ad:95:42:ba:23:ef:97:f7:13:a7:27:03:3c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'r3,192.168.169.3' (RSA) to the list of known hosts.
admin@r3's password:
id_rsa.pub                               100% 404   488.4KB/s   00:00
```

MENYALINKAN PUBLIC KEY DARI ANSIBLE CONTROL MACHINE KE SELURUH ROUTER



- Proses penyalinan **Public Key** ke **R4**.

```
[root@ansible ~]# scp /root/.ssh/id_rsa.pub admin@R4:id_rsa.pub
The authenticity of host 'r4 (192.168.169.4)' can't be established.
RSA key fingerprint is SHA256:2ykbL40qP/M67tzkpPyUVs/kcLgyUrNQf+AwwN0maWQ.
RSA key fingerprint is MD5:57:3f:83:e8:ca:e5:be:8c:71:1b:14:9f:86:73:3f:4e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'r4,192.168.169.4' (RSA) to the list of known hosts.
admin@r4's password:
id_rsa.pub                                     100% 404 526.9KB/s 00:00
```

3 (TIGA) CARA MENJALANKAN ANSIBLE



1. Ad-Hoc

```
ansible <inventory> -m
```

Sebagai contoh eksekusi *ansible* untuk *group inventory* dengan nama “**routers**” dan modul **ping**.

```
# ansible routers -m ping
```

2. Playbooks

```
ansible-playbook filename.yml
```

Sebagai contoh eksekusi *ansible playbook* dengan nama file **security.yml**.

```
# ansible-playbook security.yml
```

3. Automation Framework

Menurut situs [Ansible](#), **Ansible Tower** merupakan solusi berbasis web yang membuat *Ansible* lebih mudah digunakan untuk tim TI dan dirancang untuk menjadi pusat semua tugas otomasi serta memungkinkan untuk mengontrol akses pengguna. **Inventory** dapat dikelola secara grafis atau disinkronisasikan dengan beragam sumber **Cloud**. **Tower** juga mencatat (*log*) semua pekerjaan, terintegrasi dengan **LDAP** dan memiliki **API REST**. Tersedia pula tool **Command Line** untuk memudahkan integrasi dengan **Jenkins** juga.

ANSIBLE AD-HOC MODUL RAW (1)



- Membuat *user* baru pada seluruh *router* dengan ketentuan:
 - a. Username: **ansible**
 - b. Password: **ansiblesecret**
 - c. Group: **full**
 - d. Allowed Address: **192.168.169.254** dan **192.168.169.253**

```
# ansible routers -m raw -a "/user add name=ansible group=full  
password=ansiblesecret address=192.168.169.253,192.168.169.254;quit;"  
-u admin -k
```

```
SSH password:  
R3 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.3 closed.  
  
R4 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.4 closed.  
  
R2 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.2 closed.  
  
R1 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.1 closed.
```

Pada inputan **SSH password:** yang tampil, masukkan sandi dari user **admin** dan tekan **Enter**.

ANSIBLE AD-HOC MODULE RAW (2)



- Hasil verifikasi melalui **Winbox** menunjukkan bahwa *username ansible* berhasil dibuat pada setiap *router*.

The image displays four screenshots of the WinBox User List interface, each showing the configuration for the 'ansible' user on a different router. The routers are identified as R1, R2, R3, and R4. Each screenshot shows the 'User List' table with columns 'Name', 'Group', and 'Allowed Address'. The 'ansible' user is listed with 'full' group and IP addresses 192.168.169.253 and 192.168.169.254. Red boxes highlight the user name and group, and red arrows point from the router ID in the title to the user name in the table.

Name	Group	Allowed Address
admin	full	
ansible	full	192.168.169.253, 192.168.169.254



ANSIBLE AD-HOC MODULE RAW (3)

- Meng-*import file SSH key id_rsa.pub* yang telah disalin ke setiap *router* pada tahap sebelumnya untuk user *ansible*.

```
# ansible routers -m raw -a "/user ssh-keys import  
public-key-file=id_rsa.pub user=ansible;quit;" -u admin -k
```

```
SSH password:  
R2 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.2 closed.  
  
R4 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.4 closed.  
  
R3 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.3 closed.  
  
R1 | CHANGED | rc=0 >>  
interrupted  
Shared connection to 192.168.169.1 closed.
```

Pada inputan **SSH password:** yang tampil, masukkan sandi dari user **admin** dan tekan **Enter**.



ANSIBLE AD-HOC MODULE RAW (4)

- Hasil verifikasi melalui **Winbox** menunjukkan bahwa **SSH key** untuk *user ansible* berhasil di *import* pada setiap *router*.

The image displays four screenshots of the WinBox interface, arranged in a 2x2 grid, showing the 'User List' tab for SSH Keys. Each screenshot corresponds to a different router (R1, R2, R3, R4) and shows the 'ansible' user with the key owner 'root@ansible.ubg.local'. The 'Import SSH Key' button is highlighted in each screenshot.

User	Key Owner
ansible	root@ansible.ubg.local

MENGATUR GROUP VARIABLES PADA FILE INVENTORY /ETC/ANSIBLE/HOSTS



```
# nano /etc/ansible/hosts
```

```
[routers]
R1 ansible_host=192.168.169.1
R2 ansible_host=192.168.169.2
R3 ansible_host=192.168.169.3
R4 ansible_host=192.168.169.4

[routers:vars]
ansible_user=ansible
ansible_connection=network_cli
ansible_network_os=routersos
```

- Variable `ansible_user` menentukan user yang digunakan untuk terkoneksi ke *router* yaitu `ansible`.
- Variable `ansible_connection` digunakan untuk mengatur agar *Ansible* memperlakukan *managed machine* sebagai perangkat jaringan dengan lingkungan eksekusi yang terbatas yaitu `network_cli`.
- Variable `ansible_network_os` digunakan untuk menginformasikan platform jaringan dari *managed machine* yaitu `routersos`.

Simpan perubahan dengan menekan **CTRL+O** dan tekan **Enter**. Keluar dari editor *nano* dengan menekan **CTRL+X**.



ANSIBLE AD-HOC MODULE RAW (6)

- Memverifikasi hasil penambahan *group variables* pada *file inventory*.
tail /etc/ansible/hosts

```
[root@ansible ~]# tail /etc/ansible/hosts
[routers]
R1 ansible_host=192.168.169.1
R2 ansible_host=192.168.169.2
R3 ansible_host=192.168.169.3
R4 ansible_host=192.168.169.4

[routers:vars]
ansible_user=ansible
ansible_connection=network_cli
ansible_network_os=routeros
```



ANSIBLE AD-HOC MODULE PING

- Memverifikasi koneksi ke seluruh **managed machines** untuk *group* “**routers**” menggunakan modul **ping**.

```
[root@ansible ~]# ansible routers -m ping
R2 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
R3 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
R1 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
R4 | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python"
  },
  "changed": false,
  "ping": "pong"
}
```

ANSIBLE PLAYBOOK (1)



- Membuat file *playbook* menggunakan editor **nano** dengan ekstensi “**yml**” untuk menghapus *username admin* di seluruh *router*.

```
# nano remove-admin.yml
```

```
1 ---
2 - name: Manajemen User RouterOS
3   hosts: routers
4   tasks:
5     - name: Menghapus user admin
6       routers_command:
7         commands: /user remove admin
```

Playbook

Play

Tasks

- Playbooks memuat Plays
 - Plays memuat Tasks
 - Tasks memanggil Modules

Simpan perubahan dengan menekan **CTRL+O** dan **Enter**.

Tekan **CTRL+X** untuk keluar dari editor *nano*.



ANSIBLE PLAYBOOK (2)

- Mengecek sintak dari *playbook* sebelum dieksekusi sehingga memastikan tidak terdapat permasalahan terkait sintak dengan mengeksekusi perintah **ansible-playbook** dengan flag **--syntax-check**.

```
# ansible-playbook remove-admin.yml --syntax-check
```

```
[root@ansible ~]# ansible-playbook remove-admin.yml --syntax-check  
playbook: remove-admin.yml
```

- Menampilkan informasi *host* yang terdampak oleh *playbook* sebelum dieksekusi dapat menggunakan perintah:

```
[root@ansible ~]# ansible-playbook remove-admin.yml --list-hosts  
playbook: remove-admin.yml  
  
play #1 (routers): Manajemen User RouterOS   TAGS: []  
  pattern: [u'routers']  
  hosts (4):  
    R4  
    R1  
    R2  
    R3
```



ANSIBLE PLAYBOOK (3)

- Mengeksekusi *file* *playbook* **remove-admin.yml** menggunakan **ansible-playbook**.

```
[root@ansible ~]# ansible-playbook remove-admin.yml

PLAY [Manajemen User RouterOS] *****

TASK [Gathering Facts] *****
ok: [R4]
ok: [R2]
ok: [R3]
ok: [R1]

TASK [Menghapus user admin] *****
ok: [R2]
ok: [R1]
ok: [R4]
ok: [R3]

PLAY RECAP *****
R1      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
         ignored=0
R2      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
         ignored=0
R3      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
         ignored=0
R4      : ok=2    changed=0    unreachable=0    failed=0    skipped=0    rescued=0
         ignored=0

[root@ansible ~]# █
```

ANSIBLE PLAYBOOK (4)



- Hasil verifikasi melalui **Winbox** menunjukkan bahwa *user admin* telah berhasil dihapus pada setiap *router*.

The image displays four screenshots of the WinBox User List interface, arranged in a 2x2 grid. Each screenshot shows the 'User List' table with a single entry for 'ansible' in the 'full' group, with allowed addresses 192.168.169.253 and 192.168.169.254. Red boxes highlight the user name and group, and red arrows point to the router ID in the title bar.

Name	Group	Allowed Address
ansible	full	192.168.169.253, 192.168.169.254

STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (1)



Kebijakan Keamanan *Mikrotik RouterOS* yang diotomatisasi menggunakan *Ansible Playbook* pada setiap *Router* adalah sebagai berikut:

1. Menonaktifkan **RouterOS MAC-Access**.
2. Menonaktifkan **Neighbor Discovery**.
3. Menonaktifkan **BTest Server**.
4. Menonaktifkan **DNS Cache**.
5. Menonaktifkan **Client Services**.
6. Mengaktifkan **Strong Crypto SSH**.
7. Menonaktifkan **Interface Router** untuk **ether4** dan **ether5**.
8. Mengaktifkan **RouterOS Auto-Upgrade** menggunakan *system scheduler* agar terjadwal untuk beroperasi setiap hari pada jam **01:00:00**.
9. Menonaktifkan **IP Service** **api**, **api-ssl**, **ftp**, **telnet**, **www** dan **www-ssl**.
10. Membatasi akses pada **IP Service** **Winbox** dan **SSH** agar hanya dapat diakses dari **192.168.169.253** dan **192.168.169.254**.

STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (2)



- Membuat file *playbook* menggunakan editor **nano** dengan ekstensi “**yml**” untuk mengotomatisasi pengamanan RouterOS berdasarkan kebijakan yang telah ditentukan pada seluruh *router*.

```
# nano security.yml
```

```
1 ---
2 [- name: Mikrotik RouterOS Security Automation
3   hosts: routers
4   vars:
5     available_from: "192.168.169.253,192.168.169.254"
6     interfaces:
7     - ether4
8     - ether5
9   tasks:
10  [- name: Menonaktifkan RouterOS Mac-Access (MAC-Telnet, MAC-Winbox dan MAC-Ping)
11    routeros_command:
12      commands:
13      - /tool mac-server set allowed-interface-list=none
14      - /tool mac-server mac-winbox set allowed-interface-list=none
15      - /tool mac-server ping set enabled=no
16
```


STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (3)



```
17 - name: Menonaktifkan Neighbor Discovery pada seluruh interfaces
18   routers_command:
19     commands: /ip neighbor discovery-settings set discover-interface-list=none
20
21 - name: Menonaktifkan Bandwidth Test (Btest) Server
22   routers_command:
23     commands: /tool bandwidth-server set enabled=no
24
25 - name: Menonaktifkan DNS Cache
26   routers_command:
27     commands: /ip dns set allow-remote-requests=no
28
29 - name: Menonaktifkan Client Services
30   routers_command:
31     commands:
32       - /ip proxy set enabled=no
33       - /ip socks set enabled=no
34       - /ip cloud set ddns-enabled=no update-time=no
35
36 - name: Mengaktifkan Strong Crypto SSH
37   routers_command:
38     commands: /ip ssh set strong-crypto=yes
```

STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (4)



```
39
40 - name: Menonaktifkan Interface dari Router
41   routeros_command:
42     commands: /interface set [ find name={{ item }} ] disabled=yes
43   with_items: "{{ interfaces }}"
44
45 - name: Menambahkan Schedule RouterOS Auto-Upgrade
46   routeros_command:
47     commands: /system scheduler add name=UPGRADE_ROUTEROS start-date=Aug/01/2020 on-event="/system package update check-for-updates once::delay 3s::if
48               ([/system package update get status] = \"New version is available\") do={ /system package update install }" interval=1d start-time=01:00:00
49
50 - name: Menonaktifkan IP Service
51   routeros_command:
52     commands: /ip service disable {{ item }}
53   with_items:
54     - api
55     - api-ssl
56     - ftp
57     - telnet
58     - www
59     - www-ssl
60
61 - name: Mengatur available from untuk service SSH dan Winbox
62   routeros_command:
63     commands:
64       - /ip service set [ find name={{ item }} ] address="{{ available_from }}"
65   with_items:
66     - winbox
67     - ssh
```

Simpan perubahan dengan menekan **CTRL+O** dan **Enter**. Tekan **CTRL+X** untuk keluar dari editor *nano*.



STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (5)

- Mengecek sintak dari *playbook* sebelum dieksekusi sehingga memastikan tidak terdapat permasalahan terkait sintak dengan mengeksekusi perintah **ansible-playbook** dengan flag **--syntax-check**.

```
# ansible-playbook security.yml --syntax-check
```

```
[root@ansible ~]# ansible-playbook security.yml --syntax-check
playbook: security.yml
```



STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (6)

- Mengeksekusi *file playbook* **security.yml** menggunakan **ansible-playbook**.

```
[root@ansible ~]# ansible-playbook security.yml

PLAY [Mikrotik RouterOS Security Automation] *****

TASK [Gathering Facts] *****
ok: [R1]
ok: [R4]
ok: [R2]
ok: [R3]

TASK [Menonaktifkan RouterOS Mac-Access (MAC-Telnet, MAC-Winbox dan MAC-Ping)] *****
ok: [R1]
ok: [R2]
ok: [R3]
ok: [R4]

TASK [Menonaktifkan Neighbor Discovery pada seluruh interfaces] *****
ok: [R1]
ok: [R2]
ok: [R3]
ok: [R4]

TASK [Menonaktifkan Bandwidth Test (Btest) Server] *****
ok: [R1]
```



STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (7)

```
TASK [Menonaktifkan Bandwidth Test (Btest) Server] *****
ok: [R1]
ok: [R2]
ok: [R3]
ok: [R4]

TASK [Menonaktifkan DNS Cache] *****
ok: [R3]
ok: [R2]
ok: [R4]
ok: [R1]

TASK [Menonaktifkan Client Services] *****
ok: [R1]
ok: [R2]
ok: [R3]
ok: [R4]

TASK [Mengaktifkan Strong Crypto SSH] *****
ok: [R1]
ok: [R2]
ok: [R3]
ok: [R4]
```



STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (8)

```
TASK [Menonaktifkan Interface dari Router] *****
ok: [R2] => (item=ether4)
ok: [R3] => (item=ether4)
ok: [R1] => (item=ether4)
ok: [R4] => (item=ether4)
ok: [R3] => (item=ether5)
ok: [R2] => (item=ether5)
ok: [R4] => (item=ether5)
ok: [R1] => (item=ether5)

TASK [Menambahkan Schedule RouterOS Auto-Upgrade] *****
ok: [R1]
ok: [R2]
ok: [R3]
ok: [R4]

TASK [Menonaktifkan IP Service] *****
ok: [R1] => (item=api)
ok: [R2] => (item=api)
ok: [R3] => (item=api)
ok: [R4] => (item=api)
ok: [R1] => (item=api-ssl)
ok: [R2] => (item=api-ssl)
ok: [R3] => (item=api-ssl)
```



STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (9)

```
ok: [R4] => (item=api-ssl)
ok: [R1] => (item=ftp)
ok: [R2] => (item=ftp)
ok: [R4] => (item=ftp)
ok: [R3] => (item=ftp)
ok: [R1] => (item=telnet)
ok: [R2] => (item=telnet)
ok: [R4] => (item=telnet)
ok: [R3] => (item=telnet)
ok: [R1] => (item=www)
ok: [R2] => (item=www)
ok: [R4] => (item=www)
ok: [R3] => (item=www)
ok: [R1] => (item=www-ssl)
ok: [R2] => (item=www-ssl)
ok: [R4] => (item=www-ssl)
ok: [R3] => (item=www-ssl)

TASK [Mengatur available from untuk service SSH dan Winbox] *****
ok: [R1] => (item=winbox)
ok: [R2] => (item=winbox)
ok: [R3] => (item=winbox)
ok: [R4] => (item=winbox)
ok: [R1] => (item=ssh)
```

STUDI KASUS: ANSIBLE PLAYBOOK UNTUK MENGOTOMATISASI KEAMANAN ROUTEROS (10)



```
ok: [R4] => (item=www)
ok: [R3] => (item=www)
ok: [R1] => (item=www-ssl)
ok: [R2] => (item=www-ssl)
ok: [R4] => (item=www-ssl)
ok: [R3] => (item=www-ssl)

TASK [Mengatur available from untuk service SSH dan Winbox] *****
ok: [R1] => (item=winbox)
ok: [R2] => (item=winbox)
ok: [R3] => (item=winbox)
ok: [R4] => (item=winbox)
ok: [R1] => (item=ssh)
ok: [R2] => (item=ssh)
ok: [R3] => (item=ssh)
ok: [R4] => (item=ssh)

PLAY RECAP *****
R1      : ok=11   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
R2      : ok=11   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
R3      : ok=11   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
R4      : ok=11   changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

[root@ansible ~]#
```


DEMO VERIFIKASI HASIL OTOMATISASI MIKROTIK ROUTEROS SECURITY PADA SELURUH ROUTER



REFERENSI

- MikroTik, Manual: Security Your Router, 2019, [https://wiki.mikrotik.com/wiki/Manual:Securing Your Router](https://wiki.mikrotik.com/wiki/Manual:Securing_Your_Router)
- MikroTik, Manual: Upgrading RouterOS, 2019, [https://wiki.mikrotik.com/wiki/Manual:Upgrading RouterOS](https://wiki.mikrotik.com/wiki/Manual:Upgrading_RouterOS)
- MikroTik, Manual:IP/DNS, 2020, <https://wiki.mikrotik.com/wiki/Manual:IP/DNS>
- Ansible Documentation, <https://docs.ansible.com/>
- Edureka, What is Ansible? – Configuration Management And Automation With Ansible, 2020, <https://www.edureka.co/blog/what-is-ansible/>
- I Putu Hariyadi, Modul One Day Workshop “Proxmox Automation With Ansible”, Universitas Bumigora, 2019, <https://iputuhariyadi.net/2019/11/04/modul-one-day-workshop-proxmox-automation-with-ansible/>



TERIMAKASIH