Tugas 2 Ansible

Disusun oleh:
Muhammad Amir Al Aqwa

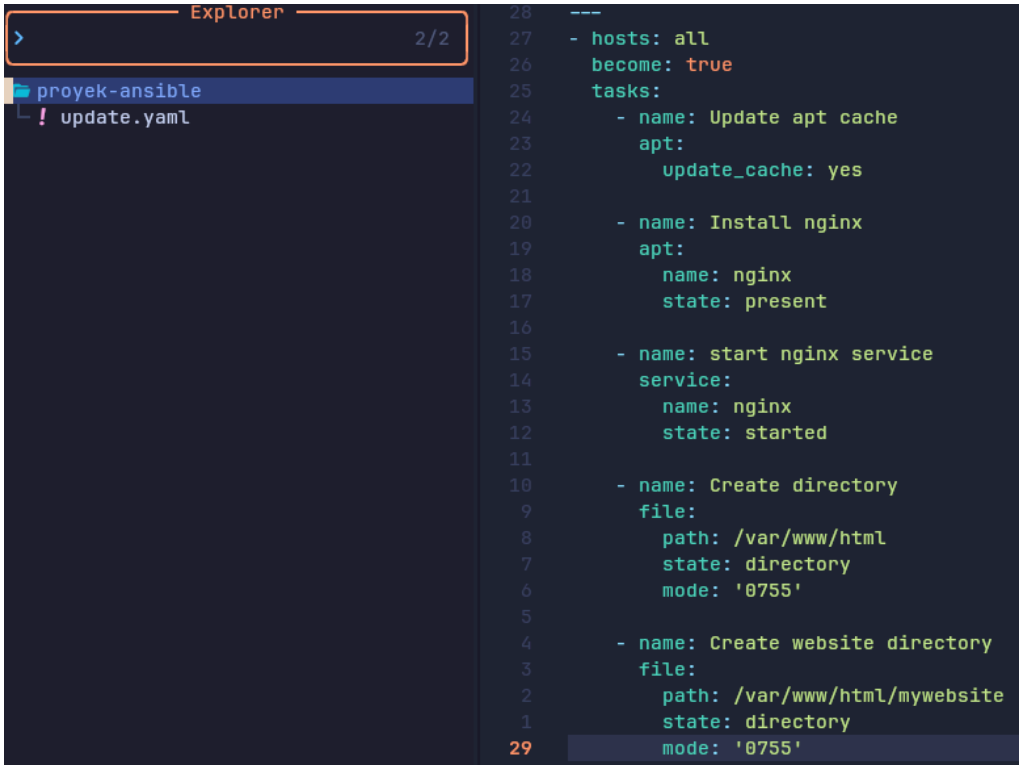
Pendahuluan

Laporan ini dibuat untuk memenuhi tugas mata kuliah Jaringan Komputer, khususnya pada materi konfigurasi server menggunakan Ansible. Dokumen ini berisi dua bagian utama: (1) pembuatan playbook sederhana untuk menginstal nginx dan membuat direktori, serta (2) implementasi Ansible Roles yang bersifat reusable untuk konfigurasi web server. Setiap tahapan akan dilengkapi dengan screenshot sebagai bukti verifikasi keberhasilan eksekusi.

Langkah Praktikum

1. Pembuatan playbook

1. Pada direktori pengerjaan playbook, buat file playbook dengan nama webserver.yaml dengan isi file sebagai berikut.



2. Jalankan playbook dengan perintah sebagai berikut.

```
ansible-playbook -i /etc/ansible/hosts webserver.yaml --ask-become-pass
```

3. Berikut tanda kalau sudah berhasil.

```
 ------------
< PLAY RECAP >
 ------------
       \   ^__^
        \  (oo)\_______
           (__)\       )\/\
               ||----w |
               ||     ||

100.109.201.60             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued=0    ignored=0
192.168.100.215            : ok=1    changed=0    unreachable=0    failed=1    skipped=0    rescued=0    ignored=0
```

```
minkwaq@minkwaq-thinkpad ~> sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-09-24 13:17:33 WIB; 10s ago
 Invocation: 1ab520dcf04544e7a973fbcb1dbbba3e
       Docs: man:nginx(8)
    Process: 86283 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exite>
    Process: 86285 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status>
   Main PID: 86354 (nginx)
      Tasks: 5 (limit: 8580)
     Memory: 4.5M (peak: 11.1M)
        CPU: 61ms
     CGroup: /system.slice/nginx.service
             ├─86354 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─86356 "nginx: worker process"
             ├─86357 "nginx: worker process"
             ├─86358 "nginx: worker process"
             └─86359 "nginx: worker process"

Sep 24 13:17:33 minkwaq-thinkpad systemd[1]: Starting nginx.service - A high performance web serve>
Sep 24 13:17:33 minkwaq-thinkpad systemd[1]: Started nginx.service - A high performance web server>
lines 1-20/20 (END)
```

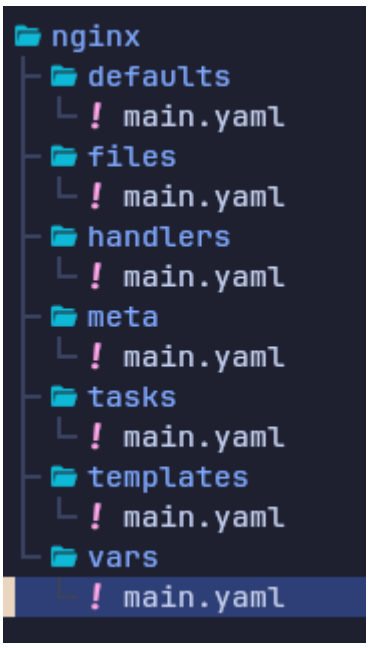
2. Implementasi ansible roles

1. Pada control nodes, buat folder kerja baru dan buat juga file dengan struktur berikut.

```
0s ~ / /Kuliah/SIB/8#/proyek-ansible
  ● ▶ mkdir -p roles/nginx/{tasks,handlers,templates,files,vars,meta,defaults}
0s ~ / /Kuliah/SIB/8#/proyek-ansible
  ● ▶ tree roles/
roles/
└── nginx
    ├── defaults
    ├── files
    ├── handlers
    ├── meta
    ├── tasks
    ├── templates
    └── vars
```

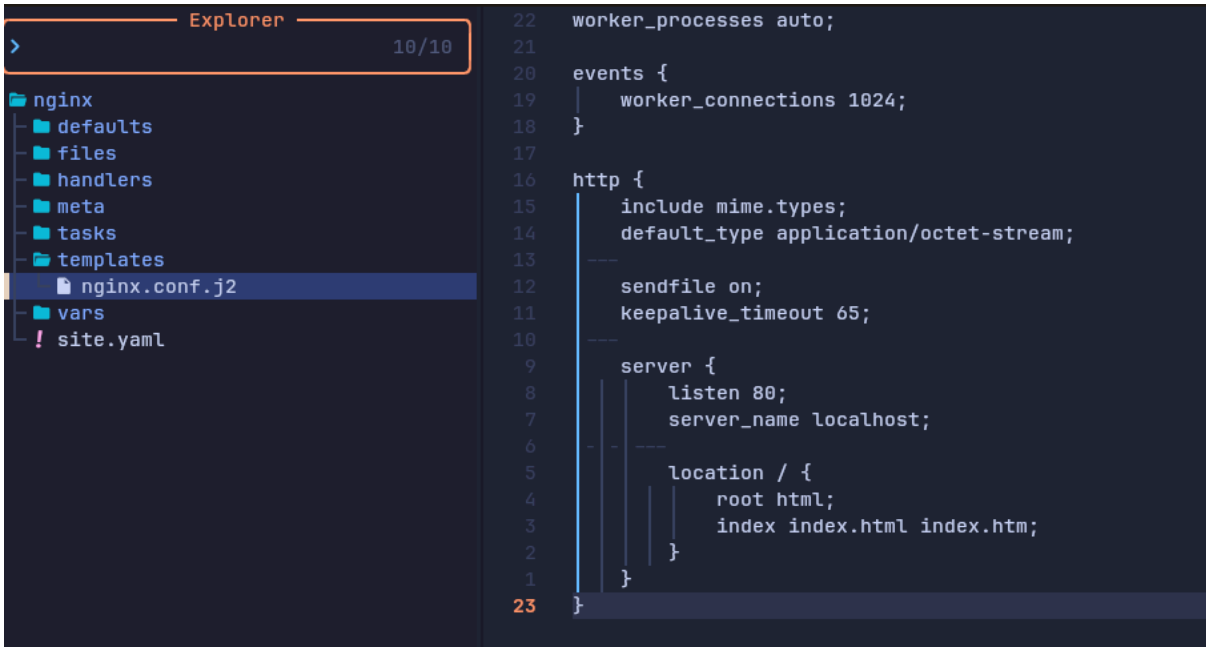
2. Lalu buat file seperti ini

```
● ▶ touch roles/nginx/tasks/main.yml roles/nginx/handlers/main.yml roles/nginx/templates/main.yml
roles/nginx/meta/main.yml
```



3. Isi file main.yaml pada folder defaults,

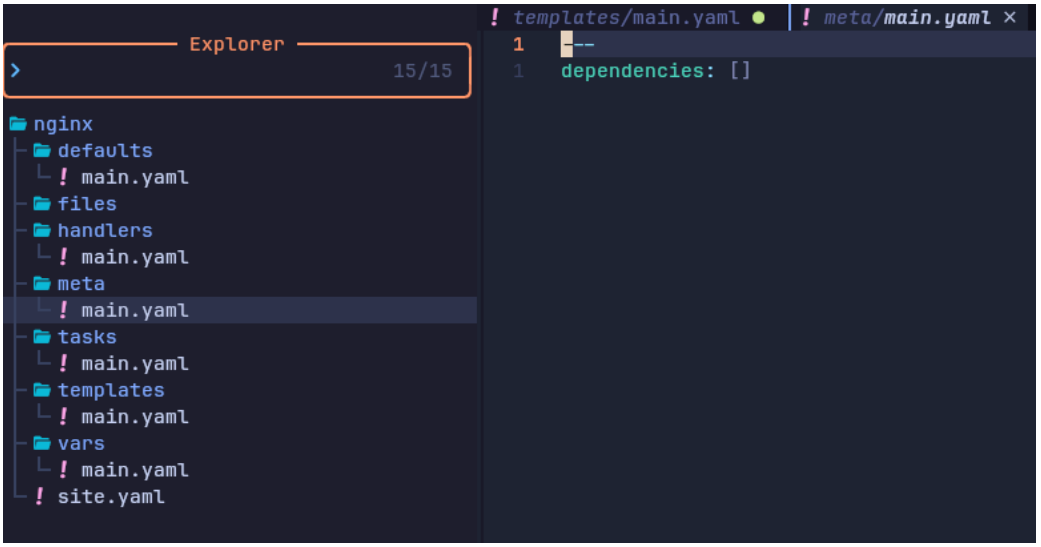
4. Folder templates, nginx.conf.j2,



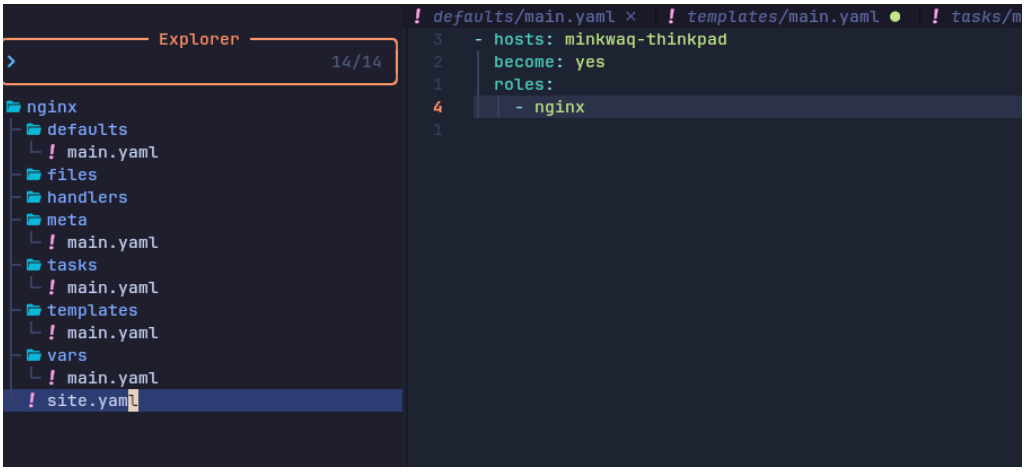
5. Folder tasks,

6. Folder handlers,

7. Folder meta,



8. Selanjutnya buat playbook dengan isi seperti ini.



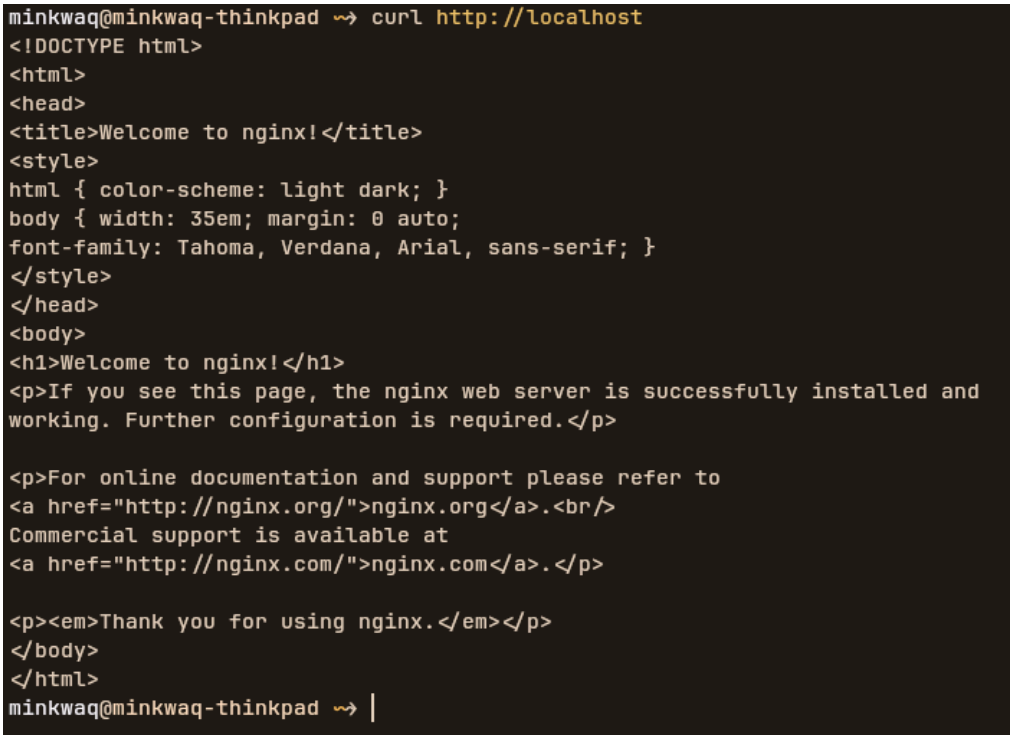
```
● ▶ tree
.
├── roles
│   └── nginx
│       ├── defaults
│       │   └── main.yaml
│       ├── files
│       ├── handlers
│       │   └── main.yaml
│       ├── meta
│       │   └── main.yaml
│       ├── tasks
│       │   └── main.yaml
│       ├── templates
│       │   └── nginx.conf.j2
│       └── vars
│           └── main.yaml
├── site.yaml
├── update.yaml
└── webserver.yaml
```

9. Lalu jalankan playbook.

```
● ▶ ansible-playbook -i /etc/ansible/hosts site.yaml --ask-become-pass
BECOME password:
```

```
100.109.201.60             : ok=0    changed=0    unreachable=1    failed=0    skipped=0    rescued
=0    ignored=0
192.168.100.215            : ok=4    changed=2    unreachable=0    failed=0    skipped=0    rescued
=0    ignored=0
```

10. Lakukan perintah berikut untuk verifikasi.



Penutup

Dengan demikian, laporan ini telah berhasil mendemonstrasikan kemampuan Ansible dalam mengotomatisasi konfigurasi server, mulai dari playbook sederhana hingga penggunaan Ansible Roles untuk pengelolaan yang lebih terstruktur dan efisien. Seluruh task yang diberikan telah berhasil dieksekusi dengan baik, sebagaimana dibuktikan oleh screenshot yang disertakan.