

# **LAPORAN EVALUASI TENGAH SEMESTER TEKNOLOGI CLOUD COMPUTING IF-B**

## **PEMBUATAN DAN KONFIGURASI VIRTUAL MACHINE (VM) DI GOOGLE CLOUD PLATFORM UNTUK MENJALANKAN APLIKASI WEB DENGAN WEB SERVER**



Disusun oleh:

Ageng Sandar R. (123220011)  
Fahmi Kariem (123220028)  
Andrea Alfian S. P. (123220078)  
Fawwaz Rizaka A. (123220185)

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN INFORMATIKA  
FAKULTAS TEKNIK INDUSTRI  
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”  
YOGYAKARTA  
2025**

## **KATA PENGANTAR**

Puji Syukur kami panjatkan ke hadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan tugas ini dengan baik dan tepat waktu. Tugas ini disusun sebagai bagian dari Evaluasi Tengah Semester pada mata kuliah Teknologi Cloud Computing, yang bertujuan untuk memberikan pemahaman praktis mengenai pembuatan dan konfigurasi Virtual Machine (VM) serta penerapan web server di cloud, yang sangat relevan dengan materi yang telah dipelajari dalam mata kuliah ini.

Kami mengucapkan terima kasih kepada Dosen Pengampu Mata Kuliah, Bapak Dr. Awang Hendrianto P., S.T., M.T., atas materi yang telah diberikan dan arahan yang diberikan selama perkuliahan berlangsung. Terima kasih juga kepada semua pihak yang terlibat, baik secara langsung maupun tidak langsung, yang telah memberikan dukungan dalam penyelesaian tugas ini.

Kami menyadari bahwa laporan ini masih jauh dari sempurna, oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun untuk perbaikan di masa yang akan datang. Semoga tugas ini dapat memberikan manfaat bagi pembaca, khususnya dalam memahami lebih mengenai penerapan teknologi cloud dalam dunia profesional.

Yogyakarta, 12 April 2025

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>DAFTAR GAMBAR.....</b>	<b>v</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	3
<b>BAB II HASIL DAN PEMBAHASAN.....</b>	<b>4</b>
2.1 Pembuatan Virtual Machine (VM).....	4
2.2 Konfigurasi Akses SSH.....	6
2.3 Instalasi Web Server.....	13
2.4 Konfigurasi Firewall.....	14
<b>BAB III KESIMPULAN DAN SARAN.....</b>	<b>29</b>
3.1 Kesimpulan.....	29
3.2 Saran.....	29
<b>BAB IV PEMBAGIAN TUGAS.....</b>	<b>31</b>
3.1 Tabel Pembagian Tugas.....	31
<b>BAB V DAFTAR PUSTAKA.....</b>	<b>32</b>

## DAFTAR GAMBAR

Gambar 2.1 Tampilan Halaman VM Instances di Google Cloud.....	4
Gambar 2.2 Machine Configuration Virtual Machine di Google Cloud.....	5
Gambar 2.3 OS and Storage Configuration Virtual Machine.....	5
Gambar 2.4 Virtual Machine Berhasil Dibuat.....	6
Gambar 2.5 Dokumentasi Pembuatan SSH Key.....	7
Gambar 2.6 Command Melihat Username Pengguna.....	7
Gambar 2.7 URL Folder Ke SSH.....	8
Gambar 2.8 Ubah Alamat Dokumentasi.....	8
Gambar 2.9 Berhasil Membuat SSH Key.....	9
Gambar 2.10 Berhasil Membuat Dua File Kunci.....	9
Gambar 2.11 Kunci Publik SSH Key.....	10
Gambar 2.12 Konfigurasi Kunci Publik GCP.....	10
Gambar 2.13 Dokumentasi Connect to VM.....	11
Gambar 2.14 Hasil Penyesuaian Perintah SSH.....	12
Gambar 2.15 Pengujian Koneksi SSH antara Lokal dan Cloud.....	13
Gambar 2.16 Tampilan Update Paket di Terminal Ubuntu.....	13
Gambar 2.17 Tampilan Instalasi Apache2 pada Terminal.....	14
Gambar 2.18 Konfirmasi Instalasi Paket di Terminal Ubuntu.....	14
Gambar 2.19 Tampilan Daftar Aplikasi yang Terdaftar di UFW.....	15
Gambar 2.20 Tampilan Pemberian Akses Apache pada Firewall menggunakan UFW.....	16
Gambar 2.21 Tampilan Status UFW yang Masih Inaktif.....	16
Gambar 2.22 Tampilan Pemberian Akses SSH pada Firewall menggunakan UFW	16
Gambar 2.23 Tampilan Aktivasi Firewall dengan UFW.....	17
Gambar 2.24 Tampilan Status UFW yang Aktif dan Aturan Akses.....	17
Gambar 2.25 Tampilan Status Apache2 Menggunakan systemctl.....	18

Gambar 2.26 Tampilan Perintah untuk Menampilkan IP Internal VM.....	18
Gambar 2.27 Tampilan Halaman Default Apache2 di Browser.....	19
Gambar 2.28 Tampilan Perintah untuk Menghentikan Layanan Apache2.....	19
Gambar 2.29 Halaman Error Saat Mengakses Website Apache yang Dihentikan..	20
Gambar 2.30 Tampilan Perintah untuk Memulai Layanan Apache2.....	20
Gambar 2.31 Tampilan Pembuatan Direktori Baru untuk Web di /home/www.....	21
Gambar 2.32 Tampilan Pembuatan dan Pengeditan File index.html di Direktori Web.....	21
Gambar 2.33 Tampilan Daftar Izin Akses Direktori di /home.....	25
Gambar 2.34 Tampilan Perubahan Kepemilikan Direktori www ke www-data.....	25
Gambar 2.35 Tampilan Pemberian Izin Akses Direktori www dengan chmod.....	26
Gambar 2.36 Tampilan Akses dan Pengeditan File Konfigurasi Apache2.....	26
Gambar 2.37 Tampilan Akses dan Pengeditan File Konfigurasi Apache2.....	27
Gambar 2.38 Tampilan Perintah untuk Merestart Layanan Apache2.....	27
Gambar 2.39 Tampilan Halaman Web yang Berhasil Dijalankan pada Browser.	28

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

*Cloud computing* adalah salah satu teknologi yang sangat berkembang dan memberikan banyak kemudahan, terutama dalam hal penggunaan sumber daya komputasi tanpa harus memiliki perangkat fisik secara langsung. Dengan menggunakan *Google Cloud Platform (GCP)*, pengguna dapat membuat dan menjalankan *Virtual Machine (VM)* yang disesuaikan dengan kebutuhan mereka, baik dari segi performa, kapasitas, maupun biaya. Layanan yang digunakan dalam implementasi ini adalah *Google Compute Engine*, yaitu layanan dari GCP yang memungkinkan pengguna untuk membuat *instance* mesin virtual dengan berbagai spesifikasi, seperti pemilihan *region*, jenis mesin, sistem operasi, serta konfigurasi penyimpanan dan estimasi biaya. Tujuan utama dari pembuatan VM ini adalah untuk keperluan pembelajaran dan eksperimen sistem berbasis *cloud*, yang memungkinkan pengguna untuk mengelola dan menjalankan sistem tanpa harus menyediakan perangkat keras sendiri.

Dalam era digital yang terus berkembang, aplikasi berbasis web menjadi solusi penting untuk mendukung keberlangsungan bisnis dan organisasi. Namun, tantangan utama yang sering dihadapi adalah bagaimana mengelola infrastruktur yang dapat mengakomodasi peningkatan jumlah pengguna dan lalu lintas data. Salah satu solusi yang efektif adalah dengan memanfaatkan layanan *cloud computing*, yang memungkinkan penyediaan sumber daya komputasi yang elastis dan *scalable*. *Google Cloud Platform (GCP)*, dengan layanan *Compute Engine*, memberikan fleksibilitas bagi perusahaan untuk mengoptimalkan kinerja aplikasi mereka dengan cara yang lebih efisien, tanpa perlu mengelola server fisik secara langsung (Bagaskara & Agustina, 2023).

Untuk memastikan aplikasi berbasis web berjalan dengan baik di lingkungan *cloud*, aspek konfigurasi VM dan aksesibilitas juga harus

diperhatikan. *SSH (Secure Shell)* menjadi protokol penting dalam memastikan bahwa pengelolaan VM dapat dilakukan dengan aman dari lokasi yang berbeda. Selain itu, penggunaan *web server* seperti *Apache* memungkinkan aplikasi berbasis web untuk dapat diakses dengan baik dan efisien di berbagai platform *cloud*. Dengan mengadopsi pendekatan *cloud computing* ini, perusahaan dapat mengelola aplikasi mereka secara optimal, dengan meningkatkan kinerja, skalabilitas, dan keamanan yang dibutuhkan dalam menghadapi beban trafik yang tinggi (Fuzes, 2018; Sulistyono et al., 2017).

## 1.2 Rumusan Masalah

Berikut adalah rumusan masalah dari tugas yang dikerjakan terkait dengan pembuatan dan konfigurasi *Virtual Machine (VM)* di layanan *cloud*:

1. Bagaimana cara membuat dan mengkonfigurasi *Virtual Machine (VM)* di *Google Cloud Platform (GCP)* untuk mendukung aplikasi berbasis web?
2. Bagaimana cara mengkonfigurasi akses *SSH* untuk dapat mengelola VM dari komputer lokal dengan aman?
3. Bagaimana cara instal dan mengkonfigurasi *web server* seperti *Apache* pada VM untuk menjalankan aplikasi berbasis web?
4. Bagaimana cara membuat halaman HTML sederhana yang dapat diakses melalui web server pada VM yang telah dikonfigurasi?
5. Bagaimana cara mengkonfigurasi dan mengelola izin akses file dan direktori pada VM agar *web server* dapat mengaksesnya dengan benar?
6. Bagaimana cara mengkonfigurasi dan mengelola *firewall* untuk mengizinkan akses ke *port* yang digunakan oleh web server pada VM?

## 1.3 Tujuan

Tujuan dari tugas ini adalah untuk memahami dan mengimplementasikan pembuatan dan konfigurasi *Virtual Machine (VM)* di layanan *cloud* yang dapat digunakan untuk mendukung aplikasi berbasis web. Secara khusus, tujuan dari tugas ini adalah sebagai berikut:

1. Membuat dan mengkonfigurasi VM di *Google Cloud Platform (GCP)* yang dapat menjalankan aplikasi berbasis web dengan spesifikasi yang sesuai.
2. Mengkonfigurasi akses *SSH* untuk mengelola VM secara aman dari komputer lokal.
3. Menginstal dan mengkonfigurasi *web server Apache* pada VM untuk menjalankan aplikasi web dengan performa optimal.
4. Membuat halaman *HTML* sederhana dan menyimpannya di direktori *web server* yang telah dikonfigurasi pada VM.
5. Mengkonfigurasi *firewall* agar aplikasi berbasis web dapat diakses dengan aman melalui *HTTP* dan *HTTPS*.

#### **1.4 Manfaat**

Adanya penerapan konfigurasi *Virtual Machine (VM)* di *cloud* dan pengelolaan *web server* pada platform *cloud* dapat memberikan sejumlah manfaat sebagai berikut:

1. Pemahaman tentang *cloud computing* dan penggunaan GCP, khususnya dalam pembuatan dan konfigurasi VM untuk aplikasi berbasis web.
2. Peningkatan keterampilan dalam konfigurasi dan pengelolaan VM, termasuk pengaturan akses *SSH* yang aman untuk mengelola sistem di cloud.
3. Keterampilan dalam menginstal dan mengkonfigurasi *web server* seperti *Apache* untuk mendukung aplikasi berbasis web dalam lingkungan cloud.
4. Pengalaman dalam memastikan ketersediaan dan skalabilitas aplikasi web, serta penerapan konfigurasi *firewall* dan akses yang aman untuk aplikasi berbasis web.

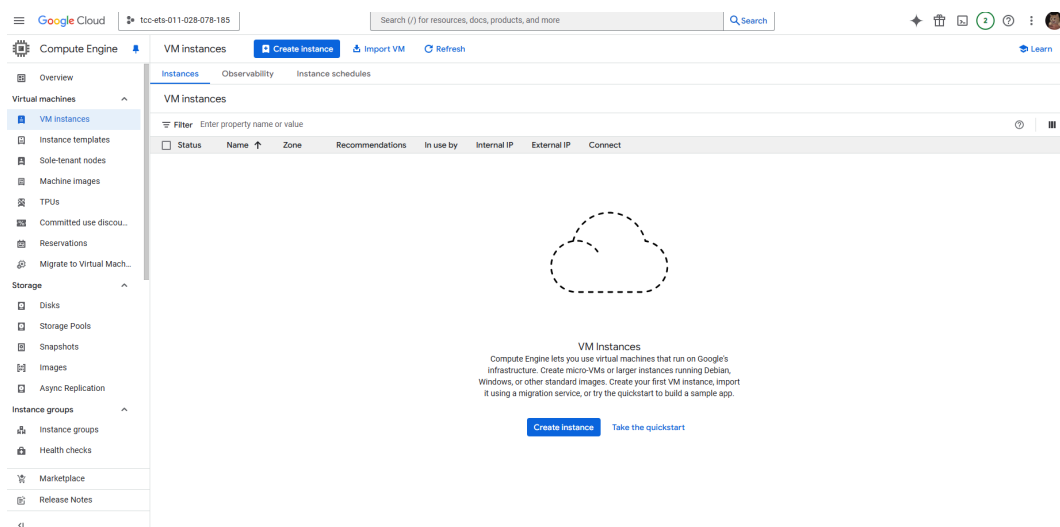


## BAB II

### HASIL DAN PEMBAHASAN

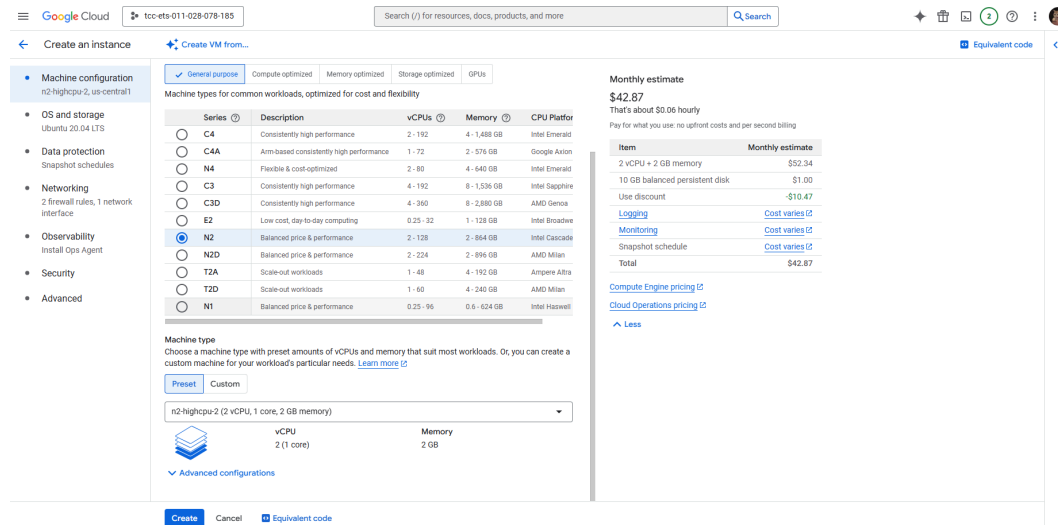
#### 2.1 Pembuatan Virtual Machine (VM)

Tahap pertama yang dilakukan yaitu dengan membuat *Virtual Machine* menggunakan Layanan *Google Cloud Platform*. Pada proses pembuatan *Virtual Machine (VM)* di *Google Cloud*, langkah pertama yang dilakukan adalah mengakses halaman *VM Instances* yang dapat ditemukan di *Google Cloud Console*. Pada **Gambar 2.1**, tampak tampilan halaman *VM Instances* yang menampilkan daftar semua *instance VM* yang telah dibuat di project yang sedang aktif. Di halaman ini, pengguna dapat memilih untuk membuat VM baru dengan mengklik tombol "Create Instance".



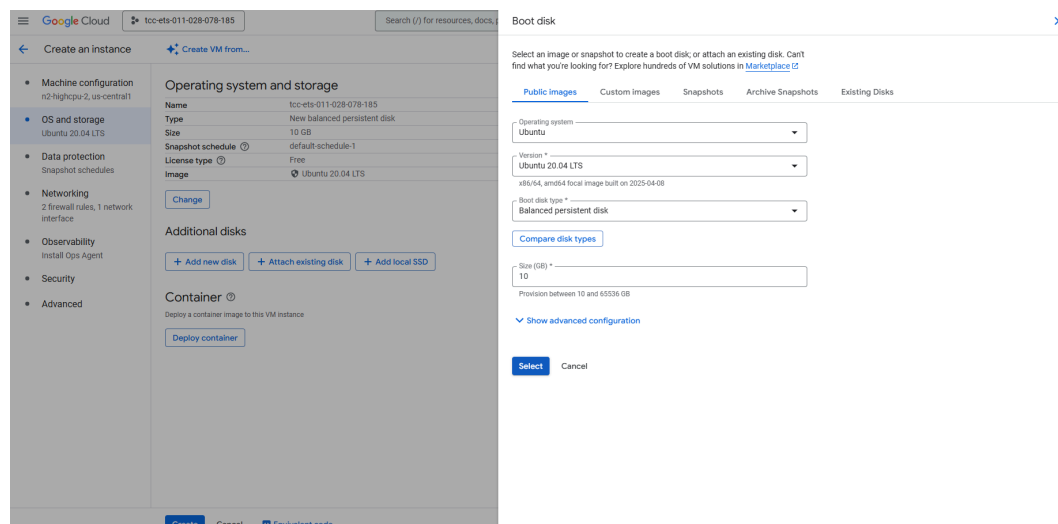
**Gambar 2.1** Tampilan Halaman *VM Instances* di *Google Cloud*

Selanjutnya, pada **Gambar 2.2**, terlihat tampilan *Machine Configuration* yang memungkinkan pengguna untuk mengkonfigurasi spesifikasi dari VM yang akan dibuat. Pada bagian ini, pengguna dapat menentukan jenis mesin (*machine type*), jumlah *vCPU*, dan memori yang dibutuhkan. Misalnya, untuk keperluan pengujian atau pengembangan aplikasi, dapat dipilih spesifikasi yang sesuai dengan kebutuhan, seperti 2 *vCPUs* dan 2GB RAM.



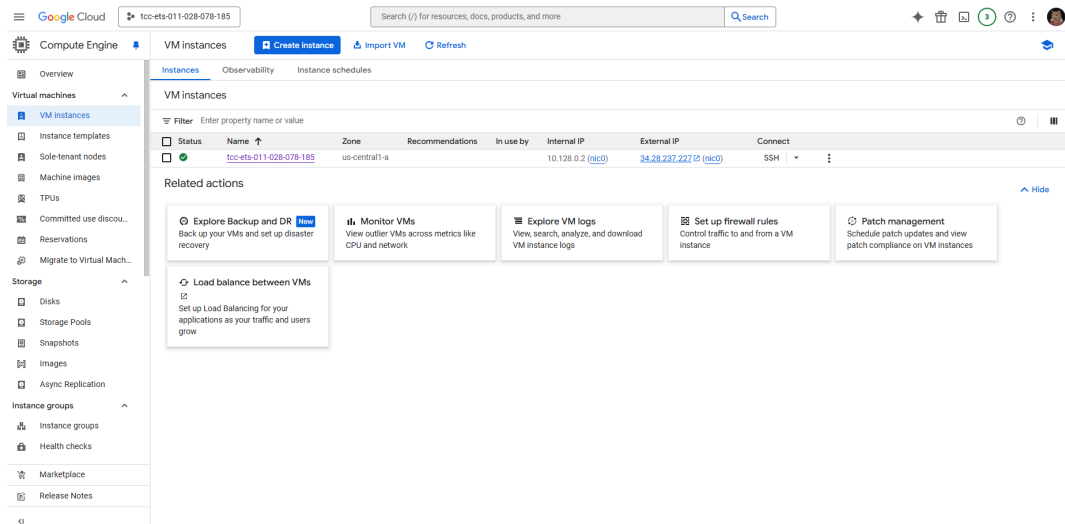
**Gambar 2.2** Machine Configuration Virtual Machine di Google Cloud

Pada langkah berikutnya, di **Gambar 2.3**, terlihat tampilan *OS and Storage Configuration* di mana pengguna memilih sistem operasi (OS) yang akan dipasang pada VM. Pada contoh ini, pengguna memilih *Ubuntu 20.04* sebagai OS yang diinginkan. Selain itu, bagian ini juga mencakup pengaturan penyimpanan yang menentukan ukuran dan tipe storage, seperti memilih *10GB SSD* yang akan digunakan sebagai *storage* untuk VM.



**Gambar 2.3** OS and Storage Configuration Virtual Machine

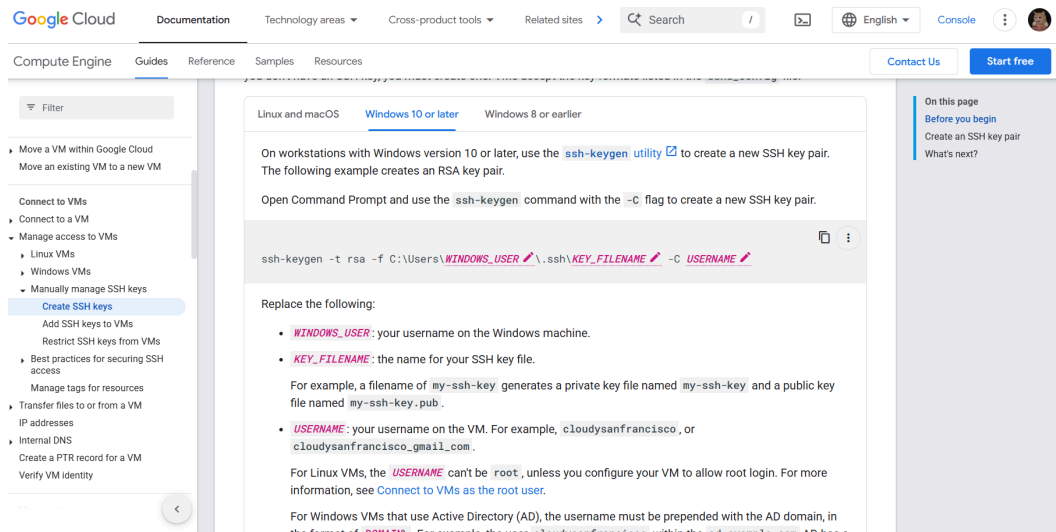
Setelah semua konfigurasi selesai dilakukan, pada **Gambar 2.4** tampak bahwa VM berhasil dibuat dan dapat dijalankan. Halaman ini menunjukkan informasi mengenai VM yang telah dibuat, seperti *External IP* dan status dari VM yang menunjukkan bahwa *instance* sudah siap untuk digunakan.



**Gambar 2.4** *Virtual Machine* Berhasil Dibuat

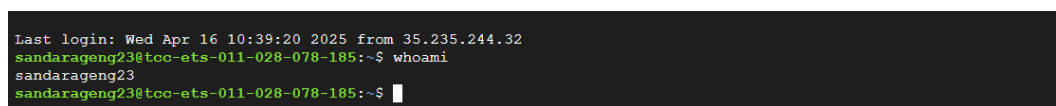
## 2.2 Konfigurasi Akses SSH

Setelah VM berhasil dibuat, tahapan selanjutnya adalah mengkonfigurasi akses *SSH* antara *Google Cloud* dengan Terminal Lokal. Proses konfigurasi akses *SSH* dimulai pada **Gambar 2.5**, yang menunjukkan dokumentasi awal pembuatan *SSH Key*. Di sana ditampilkan perintah `ssh-keygen` yang harus dilengkapi terlebih dahulu, yaitu dengan memasukkan `WINDOWS_USER`, `KEY_FILENAME`, dan `USERNAME`.



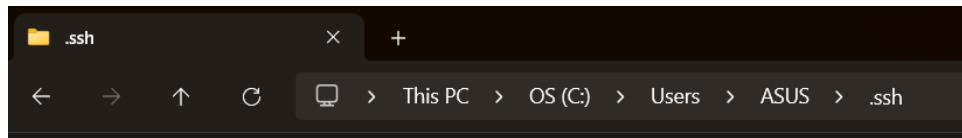
**Gambar 2.5** Dokumentasi Pembuatan SSH Key

Selanjutnya, tahapan awal dalam upaya melengkapi `ssh-keygen` yaitu dengan menemukan `USERNAME`. Untuk melihat username pengguna yang saat ini sedang aktif, dapat dilakukan dengan menjalankan perintah `whoami` melalui `SSH` pada VM yang telah dibuat sebelumnya. Penjelasan lebih jelasnya dapat dilihat pada **Gambar 2.6**, yang menunjukkan penggunaan perintah untuk melihat username pengguna yang sedang aktif.



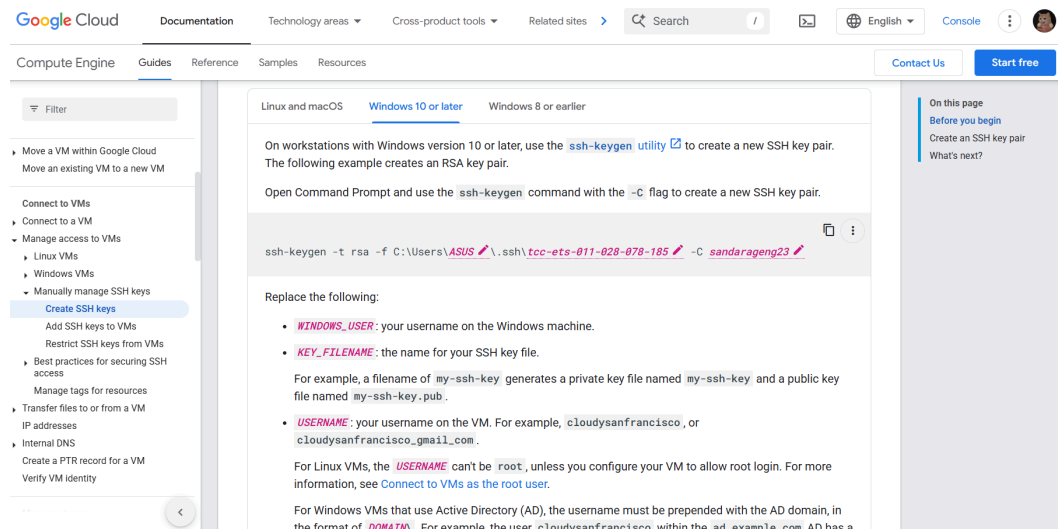
**Gambar 2.6** Command Melihat Username Pengguna

Selanjutnya, untuk melengkapi bagian `ssh-keygen` pada `WINDOWS_USER`, Dapat dilakukan dengan cara mencari nama perangkat device dengan mengakses folder `Users` dan mencari folder yang memiliki subfolder `.ssh`. Penjelasan lebih jelasnya dapat dilihat pada **Gambar 2.7**, yang menunjukkan penemuan nama folder yang dibutuhkan, yaitu `ASUS`



**Gambar 2.7** URL Folder Ke SSH

Setelah itu seperti yang terlihat pada **Gambar 2.8**, pengaturan `ssh-keygen` di halaman dokumentasi *Google Cloud - Create SSH Keys* telah selesai. Kode ini nantinya akan berguna untuk melakukan autentikasi dan mengamankan koneksi antara Terminal Lokal dengan VM di *Google Cloud*.



**Gambar 2.8** Ubah Alamat Dokumentasi



Setelah kunci *SSH* siap, selanjutnya buka *Command Prompt* pada perangkat lokal dan tempelkan `ssh-keygen` yang telah dibuat sebelumnya. Jika proses berhasil, hasilnya dapat dilihat pada **Gambar 2.9**, yang menunjukkan bahwa *SSH Key* telah berhasil dibuat.

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.22631.5189]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>ssh-keygen -t rsa -f C:\Users\ASUS\.ssh\tcc-ets-011-028-078-185 -C sandarageng23
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\ASUS\.ssh\tcc-ets-011-028-078-185
Your public key has been saved in C:\Users\ASUS\.ssh\tcc-ets-011-028-078-185.pub
The key fingerprint is:
SHA256:H+dJ0T6IEPQjGR4mI/FvkE04v0vZ/0eDxVixtx//uMI sandarageng23
The key's randomart image is:
+---[RSA 3072]-----+
|
| ooooB
| o+0 * ..
| +o* o .o.
| .o o oo+.
| .S o++..o.
| .. ..=o..o
| . + .+o .o
| o o . E .o
| ..o..oo.o|
+---[SHA256]-----+
C:\Users\ASUS>
```

**Gambar 2.9** Berhasil Membuat SSH Key

Setelah berhasil membuat *SSH Key*, selanjutnya kita bisa mengecek di folder tempat penyimpanan file *SSH*, seperti yang terlihat pada **Gambar 2.10**. Di sana, terdapat dua file: satu file tanpa ekstensi yang berisi kunci privat dan satu file dengan ekstensi `.pub` yang berisi kunci publik. File kunci publik ini akan digunakan untuk konfigurasi akses *SSH* ke server atau VM.

	tcc-ets-011-028-078-185	16/04/2025 17:49	File	3 KB
	tcc-ets-011-028-078-185.pub	16/04/2025 17:49	PUB File	1 KB

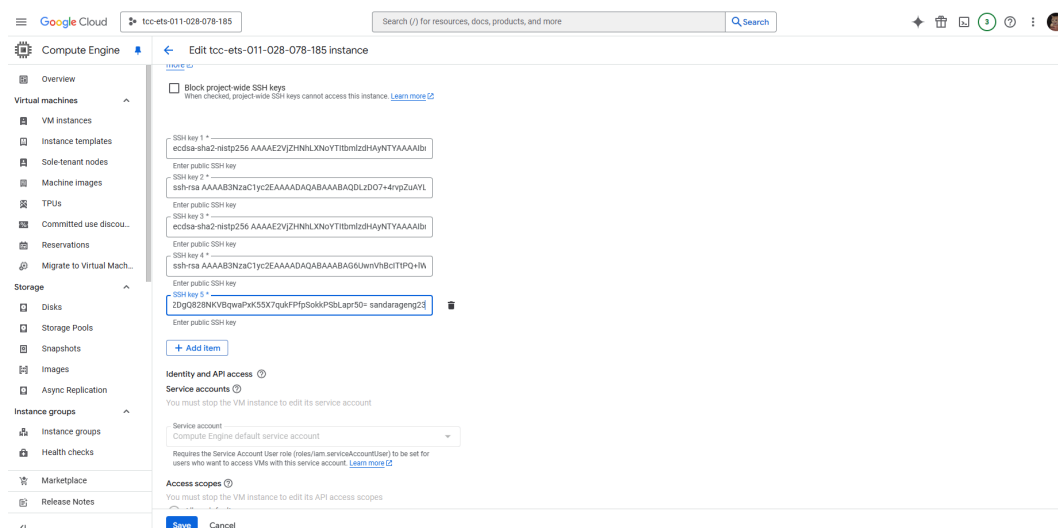
**Gambar 2.10** Berhasil Membuat Dua File Kunci

Selanjutnya, pergi ke *Command Prompt (CMD)* dan masukkan perintah seperti yang terlihat pada **Gambar 2.11** yaitu type [nama kunci dengan format `.pub`]. Perintah tersebut akan memunculkan output berupa kunci publik *SSH* yang dapat dilihat pada **Gambar 2.11**. Nantinya, kunci publik ini digunakan untuk mengkonfigurasi akses *SSH* ke server atau VM pada tahap selanjutnya.

```
C:\Users\ASUS>cd .ssh
C:\Users\ASUS\.ssh>type tcc-ets-011-028-078-185.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDEuRjvBg69YZMnO+L/5ouRgimYfsZPqouS4DpK5YoPAKsbHSnYSyFb7wMtBzrDDDDpmi03ldH13s8ohhrf
vvA7TR9nX/dUiOdH1gAk7Ropm2VLg7+DymYEM/mbyT6GNCOnCvYpYwEHduXRru9o3YKoMc8x881Dve8LIsh33f0vx/VEy073Zf25q/PQR6EYwdXCParuTny2
SPXieJMHQ5qerlyGcdQpeZj8x29WEF0a2N80cNtWz76IWzZFuef4Jg6VOVFpTDSebPMbLsbPhiZn1LznJ1r1lg78+HDCt8F3QH+e1yKNLH9zVtUEHtLGxxl
c0MqskXu5Pd4ENwanjRGSUC1o6pLSwcj278UmSF9NuyXHC9xmiCscXKyV+etTPbMuHqFmr+zw+bcTzxQeJadJ0xt7p+3PYhnMErab/RxrvnI1nxP+vdBxf
YrjYcrdHMY6SYgHb6qLKczcU2pcSH22DgQ828NKVBqwaPxK55X7qukFPfpSokkPSbLapr50= sandarageng23
```

**Gambar 2.11** Kunci Publik SSH Key

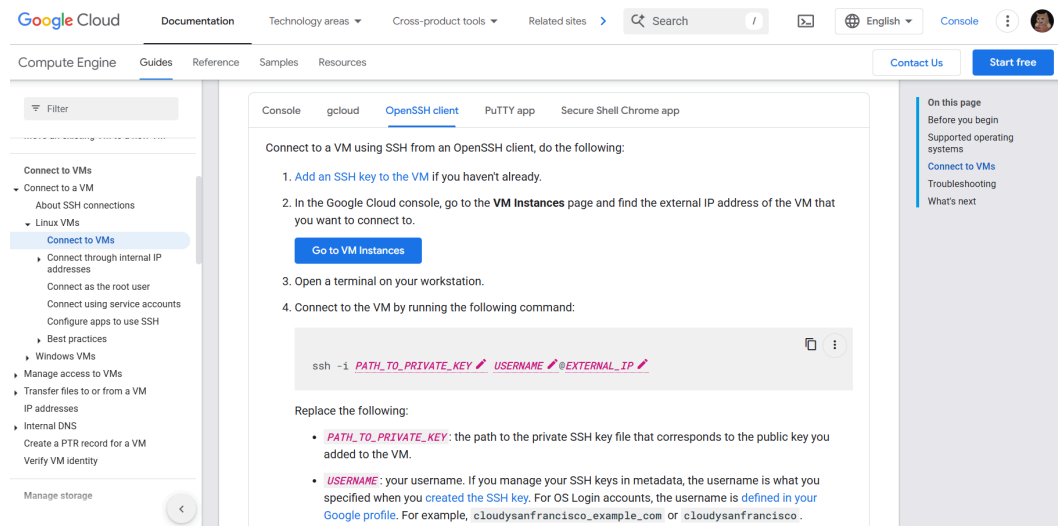
Setelah mendapatkan kunci publik, pergi ke *Google Cloud Console* dan buka halaman *instance* VM Anda. Lalu, lakukan edit pada *instance* tersebut dan cari bagian untuk menambahkan *SSH Keys*. Masukkan kunci publik yang telah didapat sebelumnya lalu save perubahan. Untuk tampilan yang lebih jelas, dapat dilihat pada **Gambar 2.12**.



**Gambar 2.12** Konfigurasi Kunci Publik GCP

Selanjutnya, kembali ke dokumentasi untuk mencari bagian '*Connect to VM*'. Untuk menemukannya, Anda bisa mencari '*Linux VMs*' dan klik '*Connect to VMs*'. Setelah mendapatkannya, tampilannya akan seperti pada **Gambar 2.13**, di mana Anda diminta untuk melengkapi informasi seperti `PATH_TO_PRIVATE_KEY` yaitu jalur atau *path* tempat file *private key* berada di perangkat lokal, alamat IP eksternal dan username. Langkah ini penting untuk mengkonfigurasi koneksi *SSH* dengan VM yang telah dibuat sebelumnya. Untuk

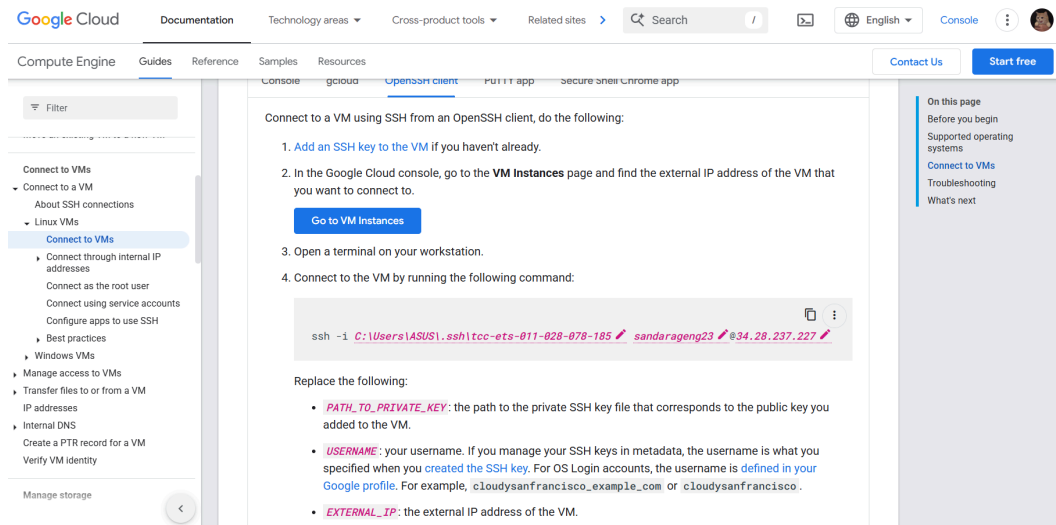
mendapatkan alamat *IP eksternal*, Anda bisa mencarikannya di halaman VM instances pada *Google Cloud Console*.



**Gambar 2.13** Dokumentasi *Connect to VM*

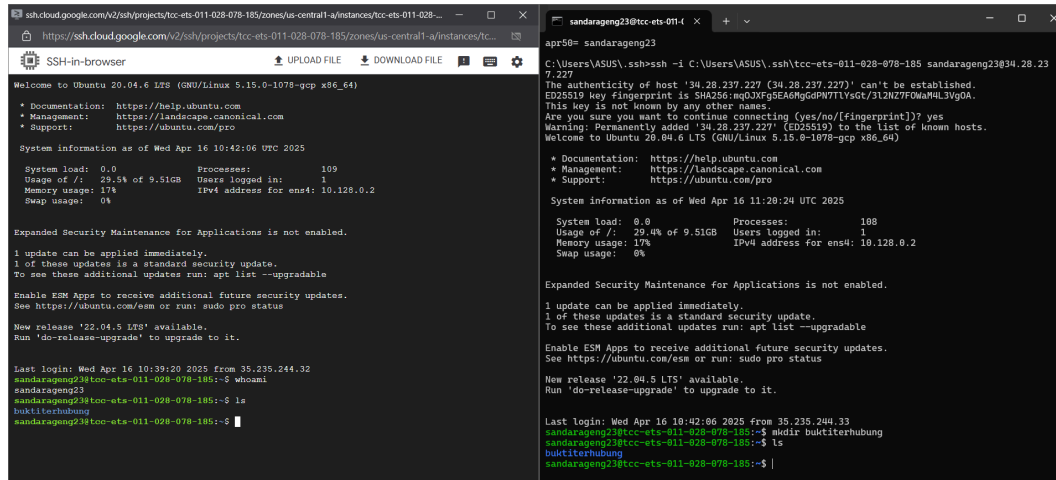
Di **Gambar 2.14**, tampak penyesuaian perintah *SSH* yang digunakan untuk mengakses VM telah selesai. Di sini, perintah *SSH* disesuaikan agar dapat bekerja sesuai dengan konfigurasi yang telah ditentukan, seperti menggunakan kunci privat yang sesuai dan alamat IP yang tepat.





**Gambar 2.14** Hasil Penyesuaian Perintah SSH

Setelah itu, salin dan tempelkan perintah yang telah dibuat ke *Command Prompt* di perangkat lokal, lalu lakukan pengujian dengan mencoba membuat sebuah folder di komputer lokal. Selanjutnya, cek apakah folder tersebut dapat diakses melalui *SSH di Google Cloud*. Akhirnya, pada **Gambar 2.15**, terlihat pengujian koneksi SSH antara perangkat lokal dan *cloud*. Pada tahap ini, pengguna menguji apakah koneksi SSH berhasil dilakukan antara komputer lokal dan VM di *cloud*.

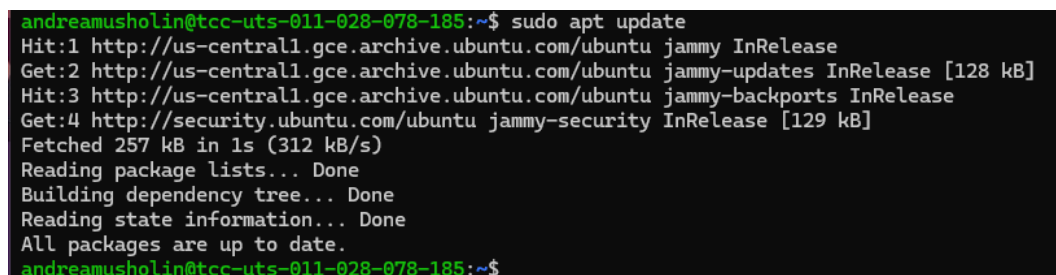


**Gambar 2.15** Pengujian Koneksi SSH antara Lokal dan Cloud

Dengan demikian, proses konfigurasi akses *SSH* ini memastikan bahwa pengguna dapat mengakses VM di *Google Cloud* secara aman dan efisien, menggunakan kunci *SSH* untuk autentikasi.

## 2.3 Instalasi Web Server

Setelah berhasil membuat *Virtual Machine (VM)* dan mengkonfigurasi akses *SSH*, langkah selanjutnya adalah menginstal *web server Apache2* pada VM yang telah dibuat. Proses ini dimulai dengan memperbarui paket sistem menggunakan perintah `sudo apt update`. Langkah ini penting untuk memastikan bahwa sistem memiliki paket-paket terbaru dan untuk menghindari masalah ketergantungan saat instalasi. Pada **Gambar 2.16**, terlihat proses pembaruan paket pada terminal Ubuntu.



**Gambar 2.16** Tampilan Update Paket di Terminal Ubuntu

Setelah pembaruan selesai, langkah selanjutnya adalah menginstal *Apache2* menggunakan perintah `sudo apt install apache2`. Pada **Gambar 2.17**, tampak tampilan terminal yang menunjukkan proses instalasi *Apache2*.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
  liblua5.3-0 mailcap mime-support ssl-cert
```

**Gambar 2.17** Tampilan Instalasi Apache2 pada Terminal

Saat proses instalasi berlangsung, pengguna akan diminta untuk mengonfirmasi instalasi dengan menekan "y" dan menekan enter untuk melanjutkan. Hal ini terlihat pada **Gambar 2.18**, yang memperlihatkan tampilan konfirmasi instalasi paket *Apache2* pada terminal Ubuntu.

```
After this operation, 8528 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.2 [10
8 kB]
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.2
[92.8 kB]
Get:3 http://us-central1.gce.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-5ubu
```

**Gambar 2.18** Konfirmasi Instalasi Paket di Terminal Ubuntu

Setelah instalasi selesai, *Apache2* siap digunakan dan dapat diakses untuk memverifikasi bahwa *server web* berjalan dengan baik.

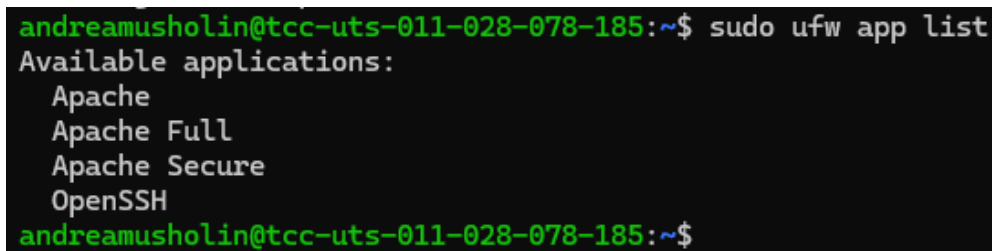
## 2.4 Konfigurasi Firewall

Sebelum melakukan pengujian terhadap *Apache2*, sangat penting untuk mengkonfigurasi *firewall* agar dapat membuka akses dari luar ke *port web default*. Biasanya, akses dari luar akan diblokir oleh *UFW (Uncomplicated Firewall)* secara *default*, sehingga kita perlu menyesuaikan pengaturannya.

Selama instalasi *Apache2*, server ini akan mendaftarkan dirinya ke *UFW* dan memberikan profil *Apache*. Profil ini nantinya akan berguna untuk mengelola

akses ke *Apache*, termasuk untuk mengaktifkan atau menonaktifkan akses ke *server web* melalui *firewall*.

Untuk mengecek profil *UFW* yang sudah terdaftar, kita bisa menggunakan perintah seperti pada **Gambar 2.19**, di mana aplikasi *Apache* dan profil-profil terkait (seperti *Apache Full*, *Apache Secure*, dan *OpenSSH*) dapat dilihat.



```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
andreamusholin@tcc-uts-011-028-078-185:~$
```

**Gambar 2.19** Tampilan Daftar Aplikasi yang Terdaftar di *UFW*

Berikut adalah penjelasan masing-masing profil:

1. *Apache*: Profil ini hanya membuka port 80, yang digunakan untuk lalu lintas web biasa (tanpa enkripsi). Port ini digunakan untuk mengakses situs web secara tidak terenkripsi.
2. *Apache Full*: Profil ini membuka dua port, yaitu port 80 (untuk lalu lintas web biasa tanpa enkripsi) dan port 443 (untuk lalu lintas web terenkripsi menggunakan TLS/SSL). Port 443 digunakan untuk mengakses situs web secara aman dengan enkripsi.
3. *Apache Secure*: Profil ini hanya membuka port 443, yang digunakan untuk lalu lintas web terenkripsi (TLS/SSL). Port ini memungkinkan komunikasi aman antara pengguna dan server web.

Untuk mengkonfigurasi *firewall* dan membuka akses ke server *Apache*, kita perlu menambahkan aturan di *UFW* yang memungkinkan akses ke *port* yang digunakan oleh *Apache*. Untuk melakukan ini, kita dapat menggunakan perintah ``sudo ufw allow 'Apache'``. Perintah ini akan mengizinkan akses ke *port* 80, yang digunakan oleh *Apache* untuk lalu lintas web biasa tanpa enkripsi. Pada **Gambar 2.20**, terlihat bahwa setelah perintah dijalankan, *rules di firewall* berhasil diupdate untuk memungkinkan akses ke *Apache*.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo ufw allow 'Apache'  
Rules updated  
Rules updated (v6)
```

**Gambar 2.20** Tampilan Pemberian Akses Apache pada Firewall menggunakan *UFW*

Setelah menambahkan aturan untuk mengizinkan akses *Apache* melalui *firewall*, langkah selanjutnya adalah memeriksa status *UFW* untuk memastikan bahwa *firewall* telah diaktifkan dan aturan yang telah diterapkan telah berjalan dengan baik. Untuk mengecek status *firewall*, kita dapat menggunakan perintah ``sudo ufw status``. Pada **Gambar 2.21**, terlihat bahwa status *UFW* masih dalam keadaan "inactive", yang berarti *firewall* belum diaktifkan.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo ufw status  
Status: inactive
```

**Gambar 2.21** Tampilan Status *UFW* yang Masih Inaktif

Karena *firewall UFW* masih dalam keadaan tidak aktif (inactive), kita perlu mengaktifkannya. Namun, sebelum itu, pastikan untuk mengizinkan akses *SSH* agar tetap bisa mengontrol VM setelah *firewall* diaktifkan. Gunakan perintah `sudo ufw allow ssh` untuk membuka akses ke *port SSH* (port 22) di *firewall*. Perintah ini memastikan bahwa koneksi *SSH* dari perangkat lokal ke VM tetap dapat dilakukan meskipun *firewall* diaktifkan. Setelah perintah ini dijalankan, aturan *firewall* akan diperbarui untuk memungkinkan akses *SSH*, yang dapat dilihat pada Gambar 2.22.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo ufw allow ssh  
Rules updated  
Rules updated (v6)
```

**Gambar 2.22** Tampilan Pemberian Akses *SSH* pada Firewall menggunakan *UFW*

Setelah memastikan bahwa akses *SSH* telah diizinkan, langkah selanjutnya adalah mengaktifkan *firewall*. Jalankan perintah `sudo ufw enable` untuk mengaktifkan *UFW*. Perintah ini akan mengaktifkan *firewall* dan memastikan bahwa *UFW* akan berjalan secara otomatis setiap kali sistem dinyalakan, seperti yang terlihat pada **Gambar 2.23**.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

**Gambar 2.23** Tampilan Aktivasi Firewall dengan UFW

Setelah mengaktifkan *firewall*, langkah selanjutnya adalah memeriksa status *UFW* untuk memastikan bahwa *firewall* telah aktif dan aturan akses telah diterapkan dengan benar. Gunakan perintah `sudo ufw status` untuk memverifikasi statusnya. Pada **Gambar 2.24**, terlihat bahwa status *UFW* sudah aktif, dan aturan akses untuk *Apache* serta *SSH* telah berhasil diterapkan.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo ufw status
Status: active

To Action From
--
Apache ALLOW Anywhere
22/tcp ALLOW Anywhere
Apache (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)
```

**Gambar 2.24** Tampilan Status UFW yang Aktif dan Aturan Akses

Setelah *Apache* diinstal dan *firewall* dikonfigurasi, kita ingin memastikan bahwa web server berjalan dengan baik menggunakan perintah `sudo systemctl status apache2`

```

andreamusholin@tcc-uts-011-028-078-185:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2025-04-14 03:24:50 UTC; 18min ago
     Docs: https://httpd.apache.org/docs/2.4/
    Main PID: 14527 (apache2)
      Tasks: 55 (limit: 4687)
     Memory: 5.9M
        CPU: 142ms
    CGroup: /system.slice/apache2.service
            └─14527 /usr/sbin/apache2 -k start
              └─14529 /usr/sbin/apache2 -k start
                └─14530 /usr/sbin/apache2 -k start

Apr 14 03:24:50 tcc-uts-011-028-078-185 systemd[1]: Starting The Apache HTTP Server...
Apr 14 03:24:50 tcc-uts-011-028-078-185 systemd[1]: Started The Apache HTTP Server.

```

**Gambar 2.25** Tampilan Status Apache2 Menggunakan systemctl

Setelah memastikan *Apache2* berjalan (active), kita dapat mengakses halaman default *Apache* untuk memverifikasi bahwa *web server* berfungsi dengan baik menggunakan alamat IP. Untuk mengetahui *IP server*, kita menggunakan perintah `hostname -I`

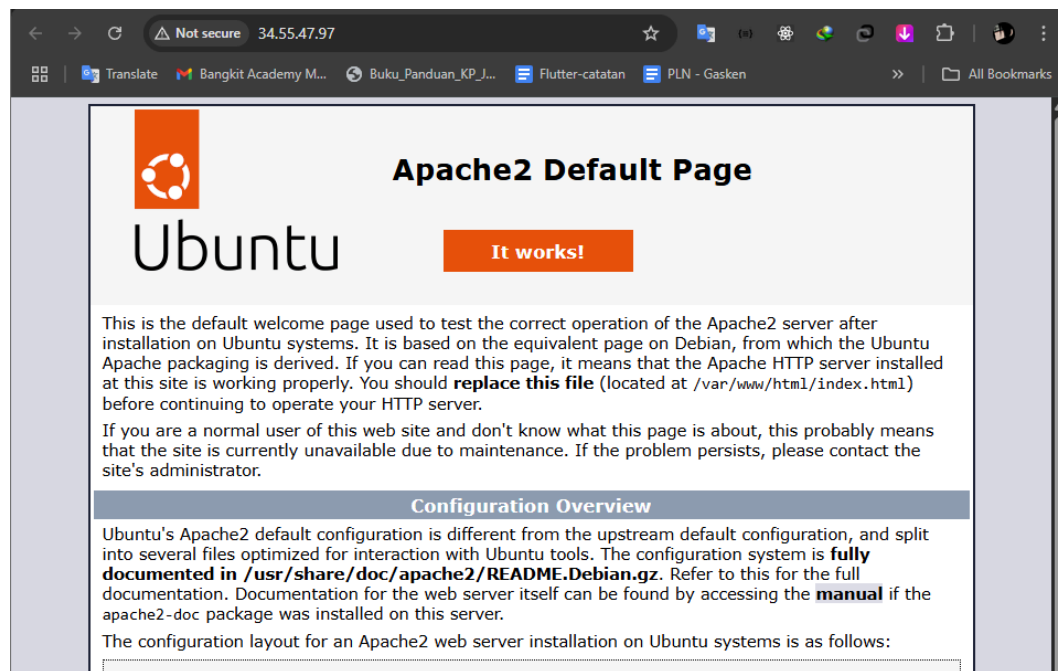
```

andreamusholin@tcc-uts-011-028-078-185:~$ hostname -I
10.128.0.5

```

**Gambar 2.26** Tampilan Perintah untuk Menampilkan IP Internal VM

Namun karena kita sekarang menggunakan VM dari GCP artinya IP dari *hostname* itu adalah *IP internal* dari jaringan VM GCP dan bukan *IP Public* dari VM kita. Untuk mengaksesnya webnya secara normal untuk sekadar testing kita tetap menggunakan *IP Public* dari VM kita yaitu <http://34.55.47.97> . Hasilnya bisa dilihat pada **Gambar 2.27**



**Gambar 2.27** Tampilan Halaman Default Apache2 di Browser

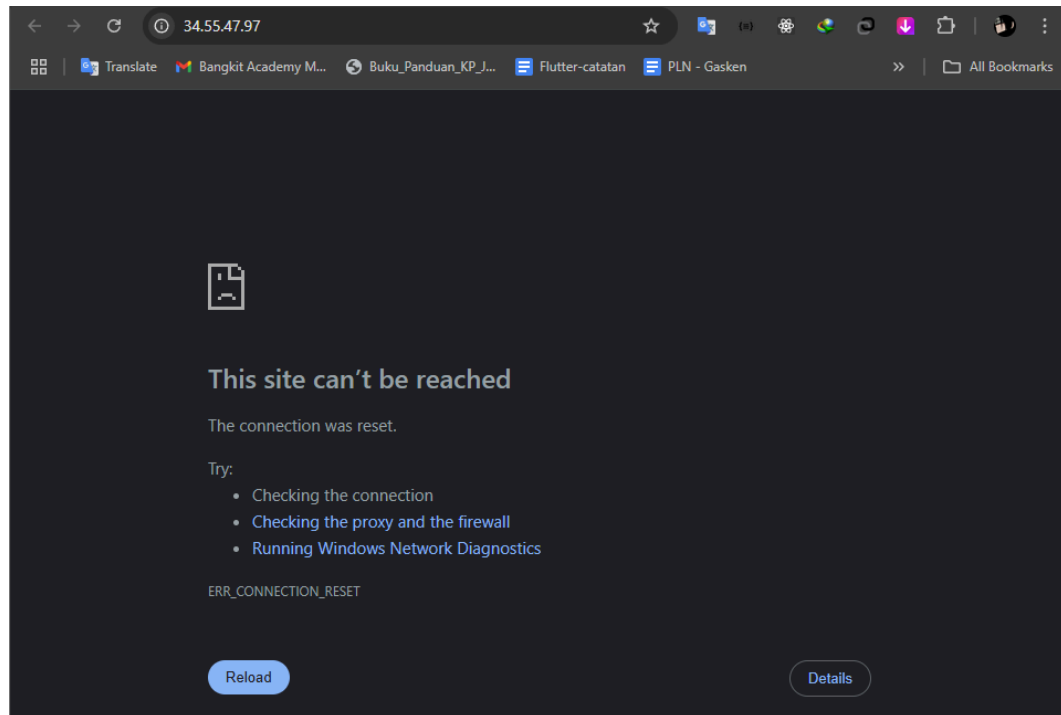
Setelah berhasil mengakses halaman web melalui IP publik, langkah selanjutnya adalah mengatur dan mengontrol proses di *Apache*. Kita dapat mengontrol proses *Apache* menggunakan `systemctl`, yang merupakan alat untuk mengelola layanan di sistem berbasis *Linux*. Untuk menghentikan web server, gunakan perintah `sudo systemctl stop apache2`. Perintah ini akan menghentikan layanan *Apache2* yang sedang berjalan di sistem.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo systemctl stop apache2
andreamusholin@tcc-uts-011-028-078-185:~$
```

**Gambar 2.28** Tampilan Perintah untuk Menghentikan Layanan Apache2

Setelah itu, lakukan kembali pengecekan IP publik untuk memastikan apakah layanan *Apache2* berhasil dihentikan. Jika layanan berhasil dihentikan, tampilan halaman akan menunjukkan error, seperti yang terlihat pada **Gambar 2.29**.





**Gambar 2.29** Halaman Error Saat Mengakses Website Apache yang Dihentikan

Untuk memulai kembali layanan *Apache2* setelah dihentikan, gunakan perintah `sudo systemctl start apache2`. Selain start dan stop, `systemctl` juga menyediakan perintah lainnya seperti `restart` untuk menghentikan dan memulai kembali layanan, `reload` untuk memuat ulang konfigurasi tanpa menghentikan layanan, serta `enable` dan `disable` untuk mengatur agar *Apache2* berjalan otomatis atau tidak saat sistem dinyalakan.

```
andreamusholin@tcc-uts-011-028-078-185:~$ sudo systemctl start apache2
andreamusholin@tcc-uts-011-028-078-185:~$
```

**Gambar 2.30** Tampilan Perintah untuk Memulai Layanan Apache2

Karena kita menggunakan *VPS*, tidak diperlukan pembuatan *Virtual Host lokal*. Namun, jika kita ingin mengakses web menggunakan domain, maka kita perlu membeli domain terlebih dahulu.

Langkah selanjutnya adalah mengubah *root web server* agar mengarah ke folder web kita sendiri. Pertama-tama, kita perlu membuat direktori untuk

menyimpan halaman web kita menggunakan perintah `sudo mkdir -p /home/www`. Perintah `mkdir` digunakan untuk membuat direktori baru, dan parameter `-p` memastikan bahwa jika direktori induk `/home` belum ada, direktori `/home/www` akan dibuatkan secara otomatis.

```
andreamusholin@tcc-uts-011-028-078-185:/home$ cd /home
andreamusholin@tcc-uts-011-028-078-185:/home$ ls
andreamusholin fawazrizakarnp rafly271008 ubuntu
andreamusholin@tcc-uts-011-028-078-185:/home$ sudo mkdir -p /home/www
andreamusholin@tcc-uts-011-028-078-185:/home$ ls
andreamusholin fawazrizakarnp rafly271008 ubuntu www
andreamusholin@tcc-uts-011-028-078-185:/home$
```

**Gambar 2.31** Tampilan Pembuatan Direktori Baru untuk Web di `/home/www`

Selanjutnya, kita membuat file `index.html` yang akan digunakan sebagai halaman utama website. Gunakan perintah `sudo nano /home/www/index.html` untuk membuat dan mengedit file tersebut. Alternatifnya, kita bisa masuk terlebih dahulu ke direktori `/home/www` menggunakan perintah `cd` (*change directory*), kemudian menggunakan perintah `ls` untuk melihat daftar file dan direktori yang ada di dalam folder tersebut. Setelah itu, masuk ke direktori `www` dan gunakan perintah `sudo nano` untuk membuka editor kode dan membuat file `index.html`.

```
andreamusholin@tcc-uts-011-028-078-185:/home$ ls
andreamusholin fawazrizakarnp rafly271008 ubuntu www
andreamusholin@tcc-uts-011-028-078-185:/home$ cd www
andreamusholin@tcc-uts-011-028-078-185:/home/www$ sudo nano index.html |
```

**Gambar 2.32** Tampilan Pembuatan dan Pengeditan File `index.html` di Direktori Web

Kemudian, buatlah halaman web sederhana untuk menampilkan daftar anggota dan NIM.

```

<!DOCTYPE html>
<html lang="en">

    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <title>Selamat Datang</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body,
        html {
            height: 100%;
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana,
sans-serif;
            overflow: hidden;
        }

        body {
            background: linear-gradient(135deg, #alc4fd, #eaf7fd);
            position: relative;
            display: flex;
            align-items: center;
            justify-content: center;
            color: #333333;
        }

        .bokeh {
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            overflow: hidden;
            z-index: 0;
        }

        .bokeh span {
            position: absolute;
            display: block;
            width: 100px;
            height: 100px;
            background: rgba(255, 255, 255, 0.2);
            border-radius: 50%;
            animation: float 20s infinite ease-in-out;
        }

        .bokeh span:nth-child(1) {
            top: 10%;
            left: 20%;

```

```

        animation-duration: 25s;
    }

    .bokeh span:nth-child(2) {
        top: 30%;
        left: 70%;
        animation-duration: 18s;
    }

    .bokeh span:nth-child(3) {
        top: 60%;
        left: 40%;
        animation-duration: 22s;
    }

    .bokeh span:nth-child(4) {
        top: 80%;
        left: 80%;
        animation-duration: 27s;
    }

    .bokeh span:nth-child(5) {
        top: 50%;
        left: 10%;
        animation-duration: 20s;
    }

    @keyframes float {
        0% {
            transform: scale(1) translateY(0px);
            opacity: 0.3;
        }

        50% {
            transform: scale(1.2) translateY(-20px);
            opacity: 0.5;
        }

        100% {
            transform: scale(1) translateY(0px);
            opacity: 0.3;
        }
    }

    .content {
        text-align: center;
        position: relative;
        z-index: 1;
    }

    h1 {
        font-size: 3em;
        margin-bottom: 0.5em;
    }

```

```

        p {
            font-size: 1.2em;
            color: #000000;
        }
        .ok {
            position: relative;
            left: 50%;
            transform: translateX(-50%);
        }
    </style>
</head>

<body>
    <div class="bokeh">

<span></span><span></span><span></span><span></span><span></span>
    </div>
    <div class="content">
        <h1>Welcome to UTS Teknologi Cloud Computing IF-B👋</h1>
        <section class="ok" style="margin-top: 50px;">
            <h3 style="font-size: 1.8em; color: #535353;
margin-bottom: 0.5em;">Anggota Kelompok</h3>
            <p>Ageng Sandar R. (123220011)</p>
            <p>Fahmi Kariem (123220028)</p>
            <p>Andrea Alfian S. P. (123220078)</p>
            <p>Fawwaz Rizaka A. (123220185)</p>
        </section>
    </div>
</body>
</html>

```

Setelah selesai membuat konten halaman web, simpan dengan menekan Ctrl + S dan keluar dengan Ctrl + X. Selanjutnya, kita perlu mengubah *ownership* dan *permission* dari direktori /home/www/ agar *web server* dapat mengakses dan menampilkan isi folder tersebut. Pastikan bahwa folder /home memiliki izin eksekusi agar *web server* dapat membuka subfolder-nya (/home/www). Untuk mengecek *permission* dari direktori /home/www/ gunakan perintah `ls -l` Perintah ini akan menampilkan daftar file beserta izin aksesnya dalam format list.

```
andreamusholin@tcc-uts-011-028-078-185:/home$ ls -l
total 20
drwxr-x--- 5 andreamusholin andreamusholin 4096 Apr 14 03:19 andreamusholin
drwxr-x--- 4 fawazrizakarnp fawazrizakarnp 4096 Apr 12 03:27 fawazrizakarnp
drwxr-x--- 4 rafly271008 rafly271008 4096 Apr 12 03:46 rafly271008
drwxr-x--- 3 ubuntu ubuntu 4096 Apr 12 03:12 ubuntu
drwxr-xr-x 2 root root 4096 Apr 14 04:41 www
andreamusholin@tcc-uts-011-028-078-185:/home$ |
```

**Gambar 2.33** Tampilan Daftar Izin Akses Direktori di /home

Seperti yang dapat kita lihat pada **Gambar 2.33**, *permission folder* /home/www masih dimiliki oleh root. Untuk mengubah kepemilikan folder tersebut agar dapat diakses oleh *web server*; kita jalankan perintah `sudo chown -R www-data:www-data www/`. Setelah itu, kita dapat memeriksa kembali menggunakan perintah `ls -l` untuk memastikan bahwa kepemilikan sudah berubah. Seperti yang terlihat pada **Gambar 2.34**, sekarang direktori /home/www sudah menjadi milik `www-data`.

```
andreamusholin@tcc-uts-011-028-078-185:/home$ ls -l
total 20
drwxr-x--- 5 andreamusholin andreamusholin 4096 Apr 14 03:19 andreamusholin
drwxr-x--- 4 fawazrizakarnp fawazrizakarnp 4096 Apr 12 03:27 fawazrizakarnp
drwxr-x--- 4 rafly271008 rafly271008 4096 Apr 12 03:46 rafly271008
drwxr-x--- 3 ubuntu ubuntu 4096 Apr 12 03:12 ubuntu
drwxr-xr-x 2 www-data www-data 4096 Apr 14 04:41 www
andreamusholin@tcc-uts-011-028-078-185:/home$ |
```

**Gambar 2.34** Tampilan Perubahan Kepemilikan Direktori www ke www-data

Selanjutnya, kita menambahkan izin akses dengan perintah `sudo chmod -R 755 www/`. Perintah `chmod` digunakan untuk mengubah *permission* atau izin akses file dan direktori. Kode dalam `chmod` memiliki aturan sebagai berikut: angka pertama (7) mengatur izin untuk pemilik (*read, write, execute*), angka kedua (5) mengatur izin untuk grup (*read, execute*), dan angka ketiga (5) mengatur izin untuk pengguna lainnya (*read, execute*).

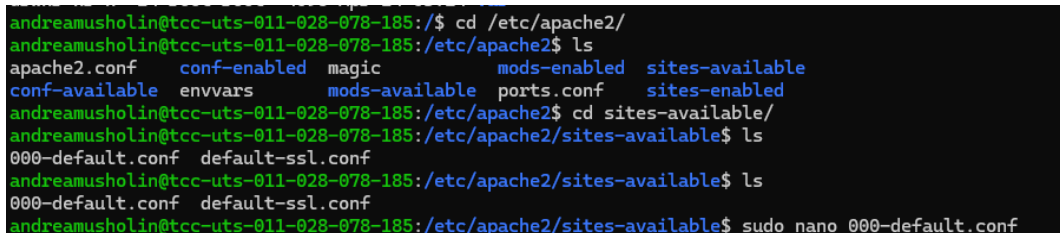
```
andreamusholin@tcc-uts-011-028-078-185:/home$ sudo chmod -R 755 www/
```

**Gambar 2.35** Tampilan Pemberian Izin Akses Direktori www dengan chmod

Selanjutnya, edit konfigurasi default *Apache*. File konfigurasi tersebut terletak di direktori: `/etc/apache2/sites-available/000-default.conf` Untuk mengeditnya, gunakan perintah:

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Dengan perintah ini, file konfigurasi *Apache2* yang ada di direktori `sites-available` dapat dibuka dan diedit.



```
andreamusholin@tcc-uts-011-028-078-185:/$ cd /etc/apache2/
andreamusholin@tcc-uts-011-028-078-185:/etc/apache2$ ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
andreamusholin@tcc-uts-011-028-078-185:/etc/apache2$ cd sites-available/
andreamusholin@tcc-uts-011-028-078-185:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
andreamusholin@tcc-uts-011-028-078-185:/etc/apache2/sites-available$ ls
000-default.conf  default-ssl.conf
andreamusholin@tcc-uts-011-028-078-185:/etc/apache2/sites-available$ sudo nano 000-default.conf
```

**Gambar 2.36** Tampilan Akses dan Pengeditan File Konfigurasi Apache2

Setelah membuka file di dalam editor `nano` atau kode editor, mulai mengedit bagian direktori yang akan digunakan. Selain itu, perlu juga menambahkan konfigurasi untuk direktori tersebut. Konfigurasi ini sangat penting karena, selain pengaturan permission dari sisi direktori, Apache juga memerlukan permission dari dirinya sendiri untuk mengakses folder tersebut. Oleh karena itu, baris `Require all granted` sangat penting untuk memastikan akses yang benar ke direktori tersebut.

```
GNU nano 6.2                                000-default.conf *
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /home/www/

<Directory /home/www>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
```

**Gambar 2.37** Tampilan Akses dan Pengeditan File Konfigurasi Apache2

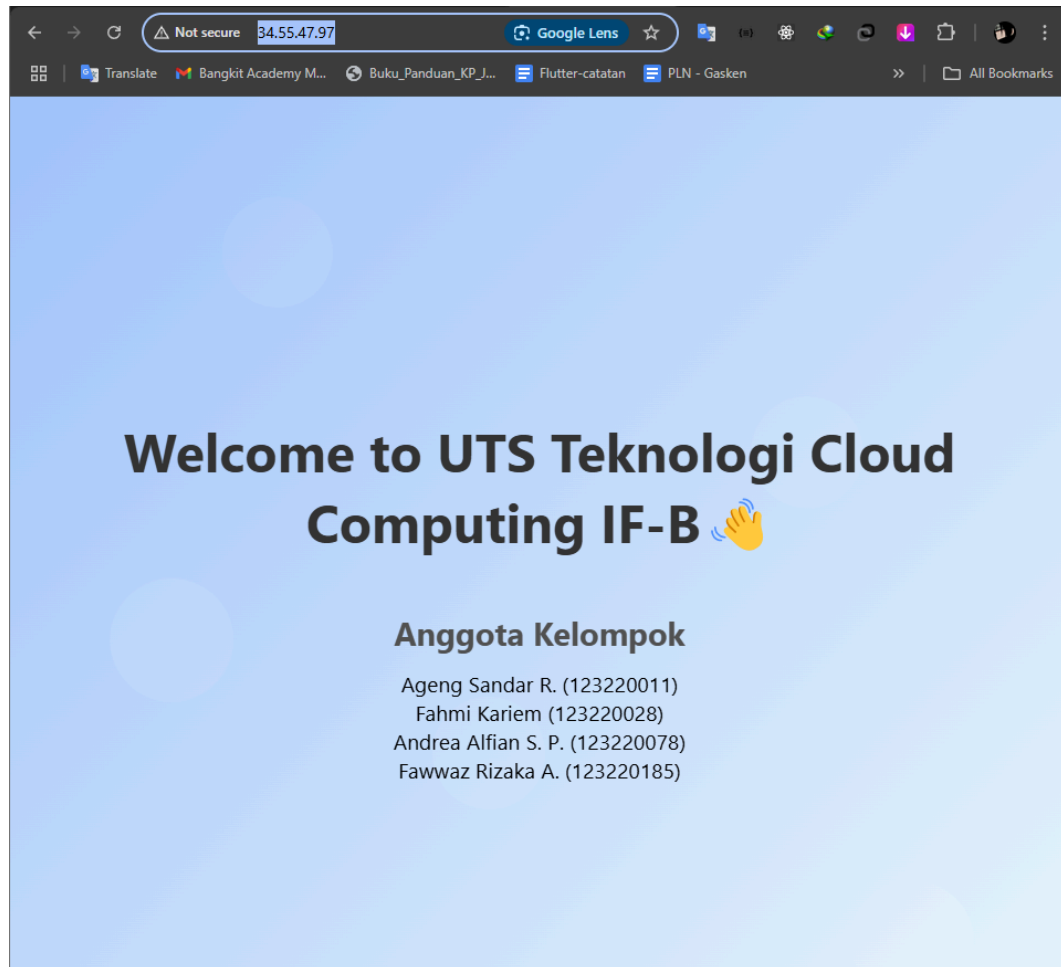
Setelah mengubah lokasi *root web server* dari `/var/www/html` menjadi `/home/www/` langkah selanjutnya adalah merestart layanan *Apache2* untuk menerapkan perubahan tersebut. Gunakan perintah `sudo systemctl restart apache2` untuk merestart layanan *Apache2*

```
andreamusholin@tcc-uts-011-028-078-185:/etc/apache2/sites-available$ sudo systemctl restart apache2
```

**Gambar 2.38** Tampilan Perintah untuk Merestart Layanan Apache2

Tahap terakhir yaitu mengecek kembali *browser* untuk memastikan apakah perubahan telah berhasil. Akses alamat berikut: <http://34.55.47.97> . Jika berhasil, tampilan yang muncul akan sesuai dengan yang terlihat pada **Gambar 2.39**.





**Gambar 2.39** Tampilan Halaman Web yang Berhasil Dijalankan pada Browser

## BAB III

### KESIMPULAN DAN SARAN

#### 3.1 Kesimpulan

Pelaksanaan tugas ini berhasil menunjukkan bahwa pembuatan dan konfigurasi *Virtual Machine (VM)* di *Google Cloud Platform (GCP)* dapat dilakukan dengan langkah-langkah yang jelas dan mudah diikuti. Konfigurasi *VM*, akses *SSH*, *web server Apache*, serta pembuatan halaman HTML sederhana yang dapat diakses melalui *server web* berjalan dengan lancar. Selain itu, pengaturan *firewall* juga berhasil diterapkan untuk memastikan akses yang aman ke aplikasi berbasis web.

Selama pengujian, aplikasi web dapat diakses dengan baik melalui browser, yang menunjukkan bahwa sistem berfungsi sebagaimana mestinya. Tugas ini juga berhasil mencapai tujuan untuk memberikan pemahaman yang lebih dalam tentang konsep *cloud computing*, serta memberikan wawasan praktis tentang pengelolaan *virtualisasi*, *server web*, dan *keamanan akses*. Melalui tugas ini, kami telah meningkatkan pemahaman tentang cara memanfaatkan cloud untuk aplikasi berbasis web dengan pengelolaan yang efektif dan aman.

#### 3.2 Saran

Berikut beberapa saran yang dapat dipertimbangkan untuk meningkatkan kualitas dan efektivitas pelaksanaan tugas serupa di masa mendatang:

1. Untuk meningkatkan proteksi, disarankan untuk membatasi akses *SSH* hanya pada alamat IP tertentu, guna mengurangi potensi celah keamanan.
2. Selain membuka port untuk *HTTP* dan *HTTPS*, sebaiknya pengaturan *firewall* lebih diperketat dengan membatasi akses hanya pada alamat IP yang diperlukan untuk menjaga keamanan lebih baik.

3. Agar lebih mudah dipahami dan dijadikan referensi, dokumentasi yang lebih sistematis tentang setiap langkah yang dilakukan perlu disusun dengan lebih rapi dan rinci.
4. Untuk meningkatkan performa aplikasi web yang dapat menangani lonjakan trafik, penggunaan auto-scaling dan load balancing perlu dipertimbangkan di masa depan.
5. Penggunaan alat otomatisasi seperti Cloud Build atau Terraform sangat dianjurkan untuk mempercepat dan menyederhanakan proses deployment dan pengelolaan infrastruktur.
6. Disarankan untuk memanfaatkan alat monitoring seperti untuk memantau kinerja VM dan aplikasi secara real-time, sehingga masalah dapat terdeteksi dan diatasi lebih cepat.

## **BAB IV**

### **PEMBAGIAN TUGAS**

#### **3.1 Tabel Pembagian Tugas**

Pembagian tugas pada kelompok kami:

<b>No</b>	<b>Nama</b>	<b>NIM</b>	<b>Tugas</b>
1	Ageng Sandar R.	123220011	Konfigurasi Akses SSH + Menyusun Laporan
2	Fahmi Kariem	123220028	Instalasi Web Server
3	Andrea Alfian S.P.	123220078	Konfigurasi Firewall + Pembuatan Website
4	Fawwaz Rizaka A.	123220185	Pembuatan VM di Google Cloud Platform

## BAB V

### DAFTAR PUSTAKA

- Google Cloud. (2023). *Cloud Run Documentation*. Google Cloud.  
<https://cloud.google.com/run/docs>
- Merkel, D. (2014). *Docker: Lightweight Linux containers for consistent development and deployment*. *Linux Journal*, 2014(239), 2.
- Turnbull, J. (2014). *The Docker Book: Containerization is the new virtualization*. James Turnbull Publishing.
- White, R., Parisi, A., & Walters, R. J. (2021). *Modern Frontend Development with Containers*. O'Reilly Media.
- Bagaskara, I., & Agustina, A. B. (2023). Penerapan Sistem Terdistribusi pada Database Website Forum Menggunakan Google Cloud Platform. *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*.
- Sulistyo, R., Erlansari, A., & Coastera, F. F. (2017). Aplikasi Cloud SQL Berbasis Web. *Jurnal Rekursif*, 5(1), 75-85.  
<https://ejournal.unib.ac.id/rekursif/article/view/2861>
- Fuzes, P. (2018). *How Does Cloud Computing Change the Strategic Alignment Between Business and IT?* Proceedings of the Fifth International Conference on Digital Information Processing, E-Business and Cloud Computing (DIPECC2018), Trabzon, Turkey, 2018. ISBN: 978-1-941968-48-2.