

Nama: Ni Made Nadya Dewindra Wirata

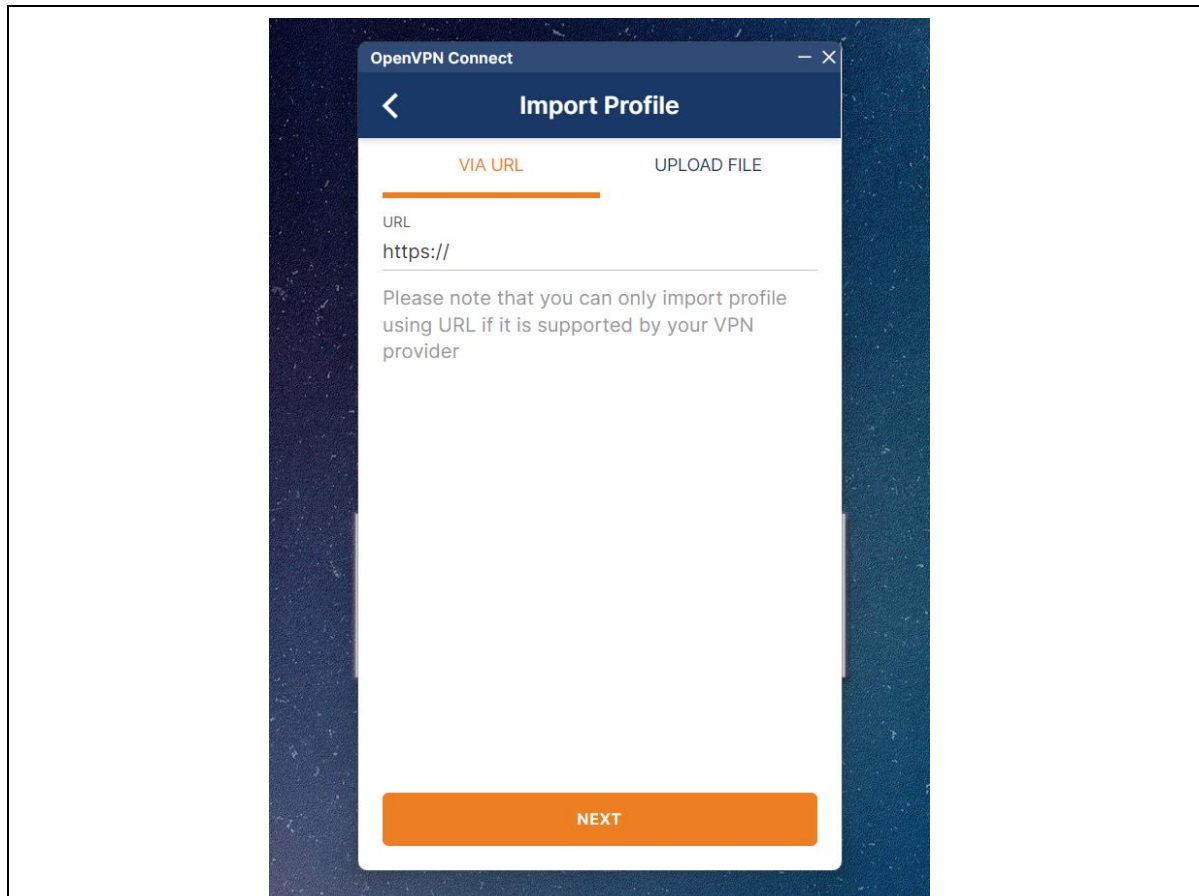
NIM: 2205551021

Kelas: Pemrograman Internet (A)

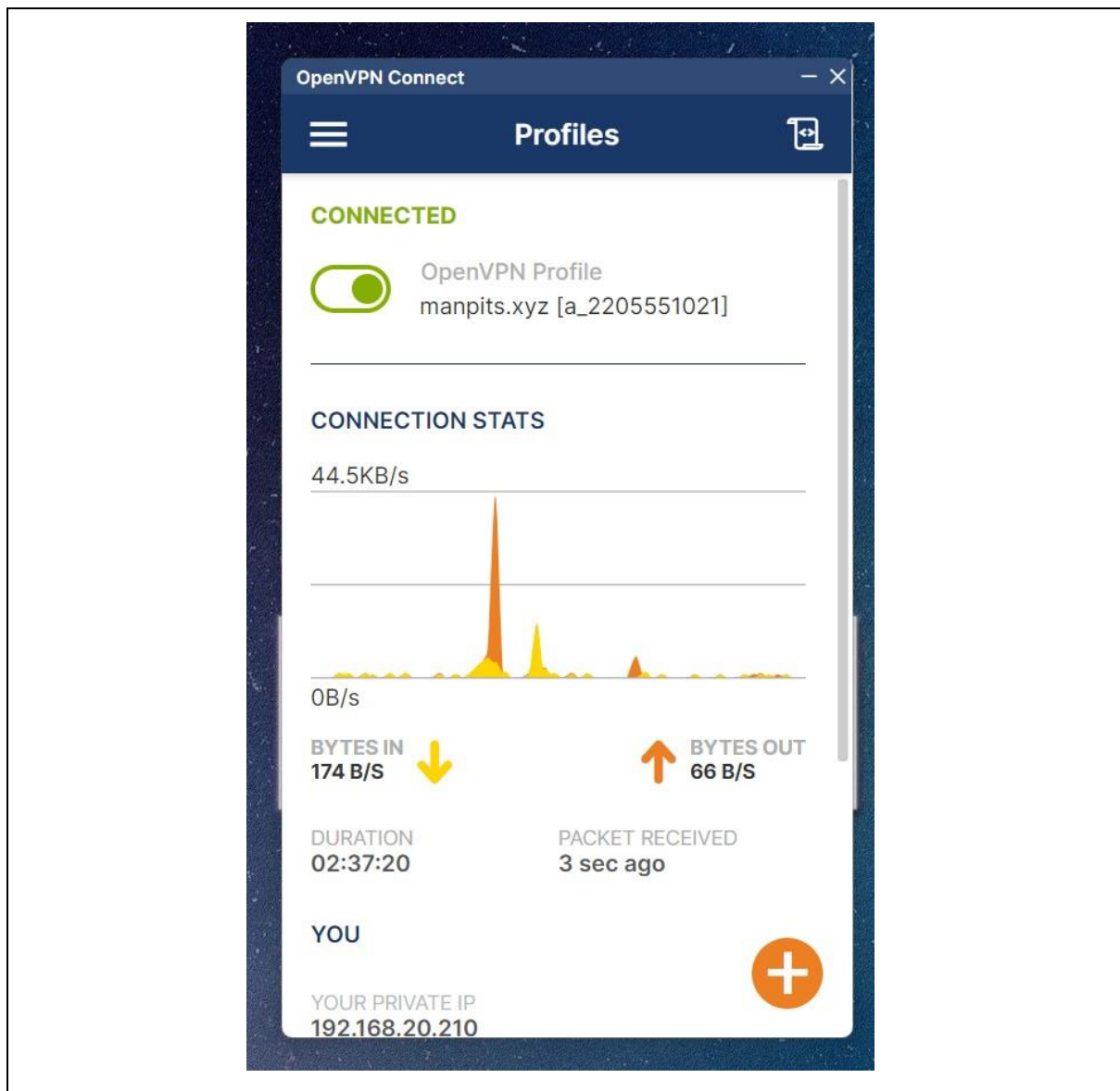
## **Dokumentasi Akses VPN dan SSH serta Deploy Aplikasi Sederhana**

### **1. Akses VPN**

Akses VPN adalah sumber informasi yang memberikan instruksi lengkap dan rinci tentang cara mengonfigurasi, menghubungkan, serta menggunakan jaringan pribadi virtual (VPN) dengan keamanan maksimal. Dokumentasi ini bertujuan untuk membimbing pengguna melalui setiap langkah yang diperlukan untuk memastikan bahwa mereka dapat memanfaatkan layanan VPN dengan efisien dan menjaga tingkat keamanan yang tinggi. Dalam dokumentasi ini, pengguna akan menemukan petunjuk langkah demi langkah tentang bagaimana mengatur koneksi VPN, mengkonfigurasi perangkat, dan mengakses jaringan pribadi dengan aman.



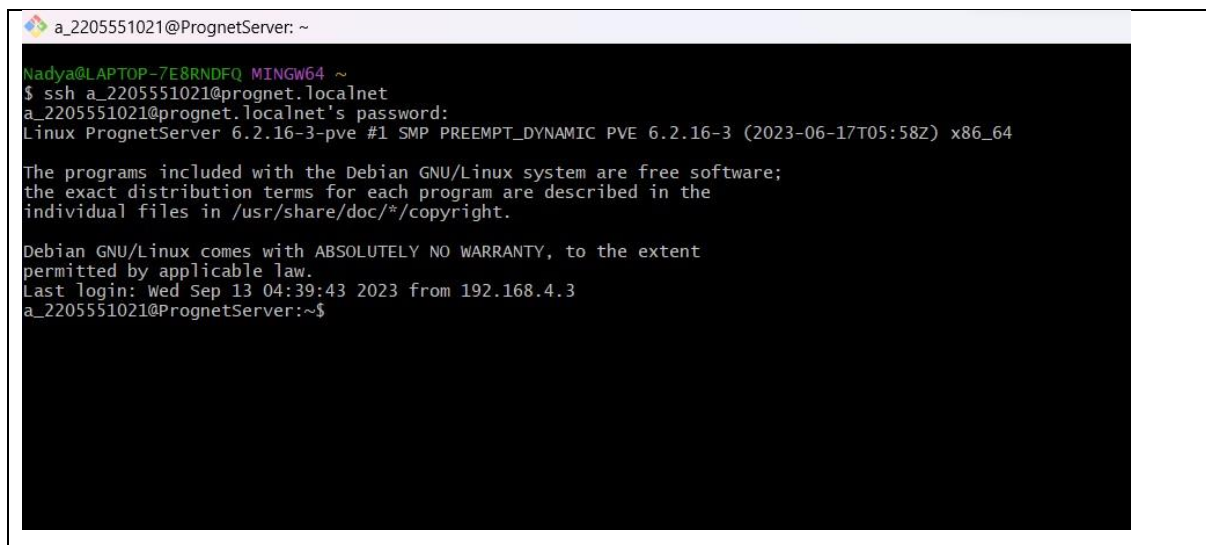
OpenVPN dapat diunduh dari tautan yang telah diberikan sebelumnya yaitu <https://openvpn.net/client/> di mana setelah mengunduh dan menginstal OpenVPN kalian memerlukan kredensial. Identifier ini diperlukan untuk dapat terhubung ke jaringan VE, *file* ini dapat diperoleh dari link yang disediakan oleh dosen pada mata kuliah pemrograman internet khususnya [http://manpits.xyz:1200/a\\_XXXXXXXXXX.ovpn](http://manpits.xyz:1200/a_XXXXXXXXXX.ovpn) kemudian ganti XXXXXXXXXXXX dengan nim yang sesuai.



Setelah memasukkan informasi login dan menekan tombol *connect* untuk menyambungkan, perangkat akan terhubung ke VPN. Perangkat yang terhubung ke VPN ini dapat digunakan untuk mengakses server SSH menggunakan Putty atau klien Git GUI, khususnya Git Bash.

## 2. SSH Access

Acces SSH pada Git Bash, mengacu pada penggunaan SSH untuk mengautentikasi dan mengamankan koneksi git antara komputer dan repositori git yang terletak di server remote. Untuk mengakses SSH pada git bash harus memasukkan *password* yang telah diberikan. Untuk memasukkan kata sandi memerlukan perintah, khususnya ssh a\_XXXXXXXX@prognnet.localnet kemudian ganti XXXX dengan nim anda masing-masing, kemudian masukkan password yang telah dibuatkan sebelumnya.



```
a_2205551021@PrognnetServer: ~  
Nadya@LAPTOP-7E8RNDFO MINGW64 ~  
$ ssh a_2205551021@prognnet.localnet  
a_2205551021@prognnet.localnet's password:  
Linux PrognnetServer 6.2.16-3-pve #1 SMP PREEMPT_DYNAMIC PVE 6.2.16-3 (2023-06-17T05:58Z) x86_64  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Wed Sep 13 04:39:43 2023 from 192.168.4.3  
a_2205551021@PrognnetServer:~$
```

Jika kalian telah login dengan perintah yang telah di jelaskan diatas kemudian setelah memasukkan passwordnya, Git Bash akan terhubung ke server.

## 3. Clone Github Repository

Pertama-tama membuat *repository* baru pada GitHub, *file* program yang telah dibuat bisa dilakukan *push* dengan menggunakan Git Bash. Hanya perlu membuka Git Bash pada direktori dengan *file* program yang kita ingin *push* didalamnya. Berikut adalah langkah-langkah untuk melakukan *push* menggunakan Git Bash.


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---


Required fields are marked with an asterisk (\*).

Owner \*

 Nadyadewindra

Repository name \*


tugas1progneth

 tugas1progneth is available.


Great repository names are short and memorable. Need inspiration? How about [legendary-umbrella](#) ?

Description (optional)

---

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

---

Initialize this repository with:

☐ Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Untuk membuat repositori baru di GitHub, masuk ke akun GitHub Anda, klik "New" di halaman Repositori, isi informasi seperti nama dan deskripsi, pilih visibilitas, dan klik "Create repository". Anda akan mendapatkan URL untuk menghubungkan repositori lokal dengan GitHub. Pada membuat repositori diatas menggunakan nama "tugas1progneth".

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Nadya\OneDrive\Documents\tugasprogneth> git add .
PS C:\Users\Nadya\OneDrive\Documents\tugasprogneth> git commit -m "first commit"
[master (root-commit) 93bc135] first commit
```

Kemudian buka terminal, setelah itu membuat langkah pertama yaitu Git init, karena percobaan pertama saya telah melakukan git init untuk melakukan inisialisasi pada repositori Git baru. Pada git init akan mulai melacak perubahan pada *file-file* dalam direktori tersebut dan membuat struktur dasar repositori. Kemudian setelah menjalankan perintah git init, selanjutnya melakukan perintah "Git add" yang digunakan untuk menambahkan semua perubahan pada

direktori kerja ke “*staging area*”. Kemudian Git commit -m “first commit” dimana jika anda telah melakukan perubahan ke staging area, perlu membuat commit untuk titik *check point* dalam Sejarah repositori yang berisi perubahan yang telah di tambah ke *staging area*. Pesan komit ("-m") adalah pesan singkat yang menggambarkan perubahan yang dilakukan dalam komit tersebut. Biasanya, pesan komit ini digunakan untuk memberikan deskripsi singkat tentang perubahan yang dilakukan dalam *commit*.

```
PS C:\Users\Nadya\OneDrive\Documents\tugasprogn> git remote add origin https://github.com/Nadyadewindra/tugaslprogn.git
PS C:\Users\Nadya\OneDrive\Documents\tugasprogn> git push -u origin master
Enumerating objects: 133, done.
Counting objects: 100% (133/133), done.
Delta compression using up to 8 threads
Compressing objects: 100% (129/129), done.
Writing objects: 100% (133/133), 4.89 MiB | 959.00 KiB/s, done.
Total 133 (delta 28), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (28/28), done.
To https://github.com/Nadyadewindra/tugaslprogn.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
PS C:\Users\Nadya\OneDrive\Documents\tugasprogn> |
```

Selanjutnya yaitu menjalankan perintah “Git remote add origin https://github.com/username/repomu: Perintah ini digunakan untuk menambahkan koneksi antara repository lokal Anda dengan repository di GitHub. "origin" adalah alias yang digunakan untuk merujuk ke repositori di GitHub tersebut. Kemudian yang terakhir adalah menjalankan perintah “Git push -u origin master” untuk mengunggah (*push*) semua perubahan yang ada dalam repositori local ke repositori GitHub.

```
a_2205551021@PrognServer:~/progn/public$ pwd
/home/a_2205551021/progn/public
```

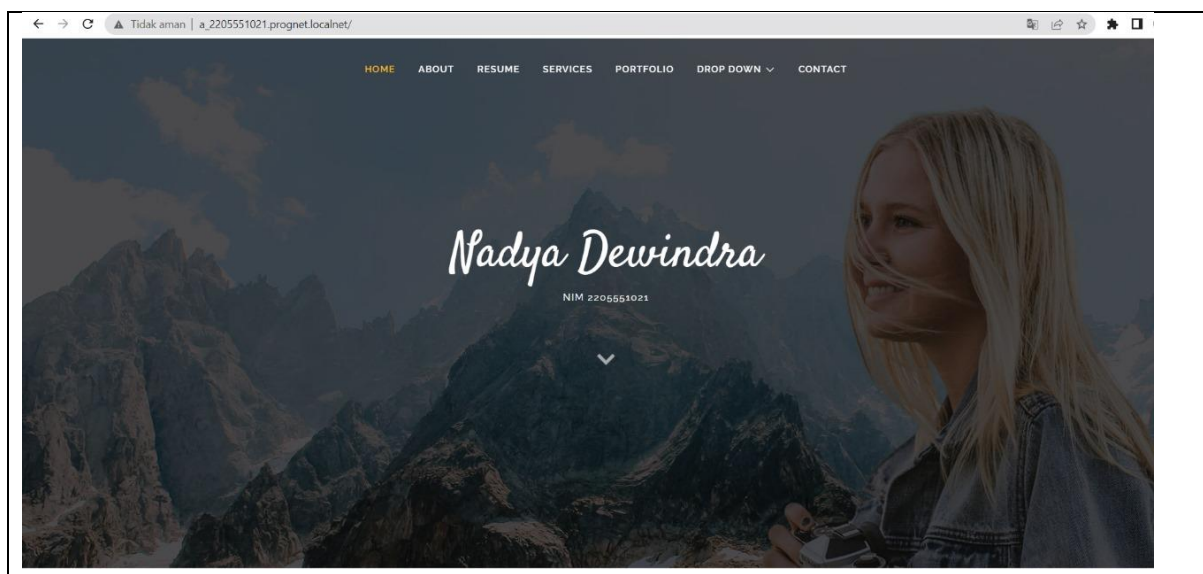
Setelah berhasil mengunggah proyek ke GitHub, Anda dapat melakukan masuk kembali ke server menggunakan koneksi SSH. Setelah berhasil masuk, Anda akan berada di direktori yang dapat Anda konfirmasi dengan perintah 'pwd', dan hasilnya adalah /home/a\_2205551021/progn/public.

```
a_2205551021@PrognServer:~/progn/public$ ls
tugaslprogn
```

Kemudian menjalankan perintah 'ls' untuk menampilkan daftar *file* dan direktori. Saat menjalankan perintah ini, anda akan melihat bahwa ada direktori yang disebut "prognnet". Selanjutnya, gunakan perintah 'ls prognnet/' untuk melihat isi dari direktori "prognnet". Di dalam direktori "prognnet", Anda akan menemukan sebuah direktori lagi yang bernama "public". Untuk mengecek isi dari direktori "public", jalankan perintah 'ls prognnet/public/', dan di dalamnya Anda akan menemukan sebuah file yang bernama "index.php".

```
a_2205551021@PrognnetServer:~$ pwd
/home/a_2205551021
a_2205551021@PrognnetServer:~$ ls
prognnet
a_2205551021@PrognnetServer:~$ ls prognnet/
public
a_2205551021@PrognnetServer:~$ ls prognnet/public/
a_2205551021@PrognnetServer:~$ cd prognnet/public/
a_2205551021@PrognnetServer:~/prognnet/public$ git clone https://github.com/Nadyadewindra/tugas1prognnet.git
Cloning into 'tugas1prognnet'...
remote: Enumerating objects: 133, done.
remote: Counting objects: 100% (133/133), done.
remote: Compressing objects: 100% (101/101), done.
remote: Total 133 (delta 28), reused 133 (delta 28), pack-reused 0
Receiving objects: 100% (133/133), 4.89 MiB | 3.23 MiB/s, done.
Resolving deltas: 100% (28/28), done.
```

Selanjutnya kita langsung menuju ke folder public setelah masuk ke folder public sekarang kita mengclone repo yg sudah kita upload ke github dengan cara 'git clone <https://github.com/Nadyadewindra/tugas1prognnet.git> dan ketika kita menjalankan perintah 'ls' lagi akan terdapat folder prognnet yg kita clone dari repo github tersebut.



Gambar di atas merupakan tampilan dari web program yang telah berhasil diunduh (*pull*) dari repositori GitHub. Dalam gambar tersebut, kita dapat lihat program dengan semua elemen dan konten yang telah diambil dari repositori tersebut, seperti halaman web atau aplikasi yang telah diimplementasikan.