IFB-452 KOMPUTASI AWAN AA

INTEGRASI BACKEND DENGAN FRONTEND MENGGUNAKAN NGINX

Disusun Oleh:

Hanifah Dwi Aprilianti (152022050)

INSTITUT TEKNOLOGI NASIONAL BANDUNG

FAKULTAS TEKNOLOGI INDUSTRI

INFORMATIKA

2025

Integrasi Backend dengan Frontend menggunakan Nginx

NGINX adalah web server yang juga berfungsi sebagai email proxy, reverse proxy, dan load balancer. Struktur software ini bersifat asinkron dan event-driven; yang memungkinkan banyak request atau permintaan diproses pada waktu bersamaan. Selain itu, NGINX juga bisa diskalakan.

Buatlah terlebih dahulu instance di ec2 pada AWS untuk instance frontend dan backend nya, Langkah pembuatannya telah ada di laporan sebelumnya. Lalu ikuti Langkah berikut :

1. Transfer file “backend.pem” dari komputer lokal (frontend) ke server EC2 (backend). File kunci ini diperlukan untuk autentikasi dalam mengelola server backend.

```
D:\KULIAH\Semester 6\IFB-452 KOMPUTASI AWAN AA>scp -i "modul5.pem"  "D:\KULIAH\Semester 6\IFB-452 KOMPUTASI AWAN AA\backend.pem
" ec2-user@13.229.75.119:~
backend.pem                                                   100% 1678    36.4KB/s   00:00

D:\KULIAH\Semester 6\IFB-452 KOMPUTASI AWAN AA>
```

2. Masuk ke instance Frontend dengan syntax : *ssh -I “modul5.pem” ec2-user@13.229.75.119* . lalu ketikan syntax berikut :

```
[ec2-user@ip-10-0-1-9 ~]$ sudo dnf install -y dnf-plugins-core
Last metadata expiration check: 11:26:40 ago on Mon Apr  7 02:41:00 2025.
Package dnf-plugins-core-4.3.0-13.amzn2023.0.5.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

3. Masukkan perintah berikut untuk memasuki folder nginx nya :

```
[ec2-user@ip-10-0-1-9 ~]$ mkdir -p ~/nginx_complete
[ec2-user@ip-10-0-1-9 ~]$ cd ~/nginx_complete
```

4. Jalankan perintah berikut pada folder nginx :

```
[ec2-user@ip-10-0-1-9 nginx_complete]$ sudo dnf download --resolve \
> nginx nginx-core nginx-mimetypes \
> gperftools-libs libunwind \
> nginx-filesystem system-logos-httpd
Last metadata expiration check: 11:28:19 ago on Mon Apr  7 02:41:00 2025.
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                528 kB/s |  19 kB     00:00
(2/7): nginx-core-1.26.3-1.amzn2023.0.1.x86_64.rpm                           12 MB/s | 670 kB     00:00
(3/7): nginx-filesystem-1.26.3-1.amzn2023.0.1.noarch.rpm                    433 kB/s | 9.6 kB     00:00
(4/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                      4.5 MB/s | 308 kB     00:00
(5/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                     1.0 MB/s |  21 kB     00:00
(6/7): nginx-1.26.3-1.amzn2023.0.1.x86_64.rpm                               1.6 MB/s |  33 kB     00:00
(7/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                            3.2 MB/s |  66 kB     00:00
[ec2-user@ip-10-0-1-9 nginx_complete]$
```

5. Membuat file arsip untuk seluruh isi folder nginx_complete menjadi satu file bernama nginx_complete.tar.gz dalam format kompresi .tar.gz.

```
[ec2-user@ip-10-0-1-9 nginx_complete]$ cd ~
[ec2-user@ip-10-0-1-9 ~]$ tar -czvf nginx_complete.tar.gz nginx_complete/
nginx_complete/
nginx_complete/generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm
nginx_complete/gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm
nginx_complete/nginx-core-1.26.3-1.amzn2023.0.1.x86_64.rpm
nginx_complete/nginx-filesystem-1.26.3-1.amzn2023.0.1.noarch.rpm
nginx_complete/nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm
nginx_complete/nginx-1.26.3-1.amzn2023.0.1.x86_64.rpm
nginx_complete/libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm
[ec2-user@ip-10-0-1-9 ~]$
```

Cek apakah file sudah masuk

```
[ec2-user@ip-10-0-1-9 ~]$ dir
backend.pem  nginx_complete  nginx_complete.tar.gz
[ec2-user@ip-10-0-1-9 ~]$
```

6. Transfer file arsip nginx_complete.tar.gz dari server asal ke server tujuan IP 10.0.2.87. File backend.pem digunakan sebagai kunci autentikasi untuk mengamankan proses transfer.

```
[ec2-user@ip-10-0-1-9 ~]$ scp -i backend.pem nginx_complete.tar.gz ec2-user@10.0.2.87:~
nginx_complete.tar.gz                                         100% 1074KB  43.1MB/s   00:00
[ec2-user@ip-10-0-1-9 ~]$
```

7. Pindah ke instance backend dan masuk ke folder nginx lalu ketikan perintah berikut:
   sudo dnf install -y --disablerepo="" *.rpm*

8. Kemudian buatlah projek backend sederhananya, disini saya menggunakan Node.Js

9. Kembali dulu ke instance frontend untuk membuat folder, lalu ketikan perintah ini :

```
[ec2-user@ip-10-0-1-9 ~]$ mkdir tugasVPChani
[ec2-user@ip-10-0-1-9 ~]$ cd tugasVPChani
[ec2-user@ip-10-0-1-9 tugasVPChani]$ npm init -y
-bash: npm: command not found
[ec2-user@ip-10-0-1-9 tugasVPChani]$
```

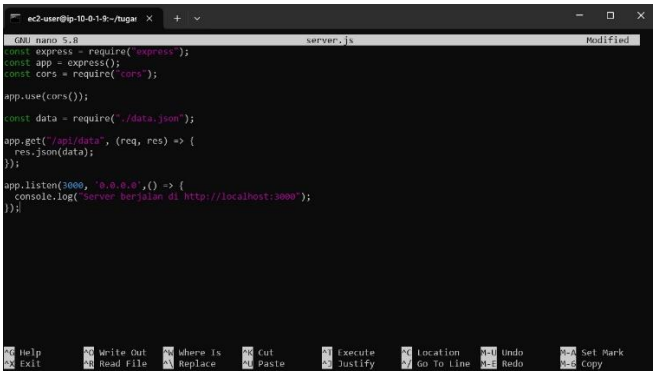
10. Pada folder yg sudah dibuat tadi install Node.Js nya, ketikan perintah berikut :
    sudo yum install -y nodejs

11. Masukkan perintah ini:

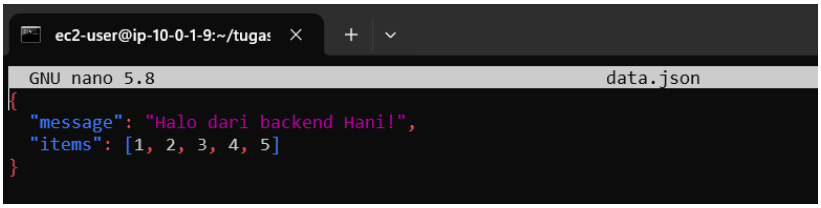
```
[ec2-user@ip-10-0-1-9 tugasVPChani]$ npm init -y
Wrote to /home/ec2-user/tugasVPChani/package.json:

{
  "name": "tugasvpchani",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}


[ec2-user@ip-10-0-1-9 tugasVPChani]$
```

    Jika sudah masukkan perintah ***npm install express***

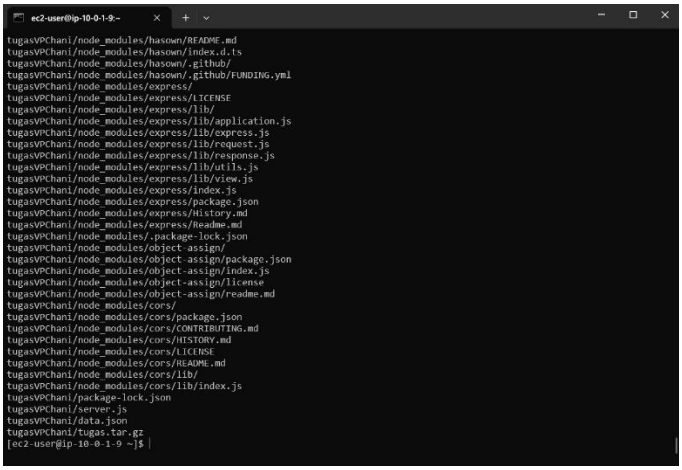
12. Ketikan perintah *nano server.js* : lalu isi seperti ini



13. Masukkan perintah *nano data.json* : lalu ketikan isi sepeerti ini

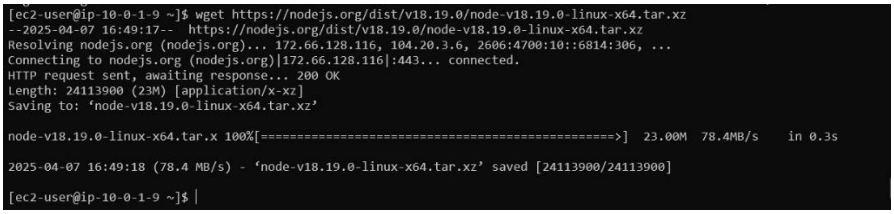


14. *tar -czvf tugas.tar.gz tugasVPChani/* ketikan perintah berikut untuk membuat file tugas yg akan di zip. Maka tampil seperti ini :



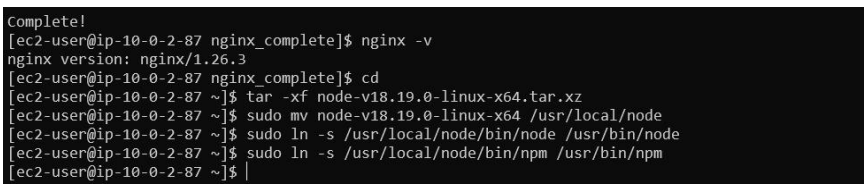
15. kirim file zip tadi ke backend

16. masukkan syntax ini : wget https://nodejs.org/dist/v18.19.0/node-v18.19.0-linux-x64.tar.xz



17. kirim Kembali file tadi ke backend : scp -i "backend.pem" node-v18.19.0-linux-x64.tar.xz ec2-user@10.0.2.87:~

18. masukan syntax berikut :



19. start Nginx di backend pada folder tugasVPChani

```
[ec2-user@ip-10-0-2-87 tugasVPChani]$ sudo systemctl start nginx
[ec2-user@ip-10-0-2-87 tugasVPChani]$ sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[ec2-user@ip-10-0-2-87 tugasVPChani]$
```

20. masukkan perintah : sudo nano /etc/nginx/nginx.conf

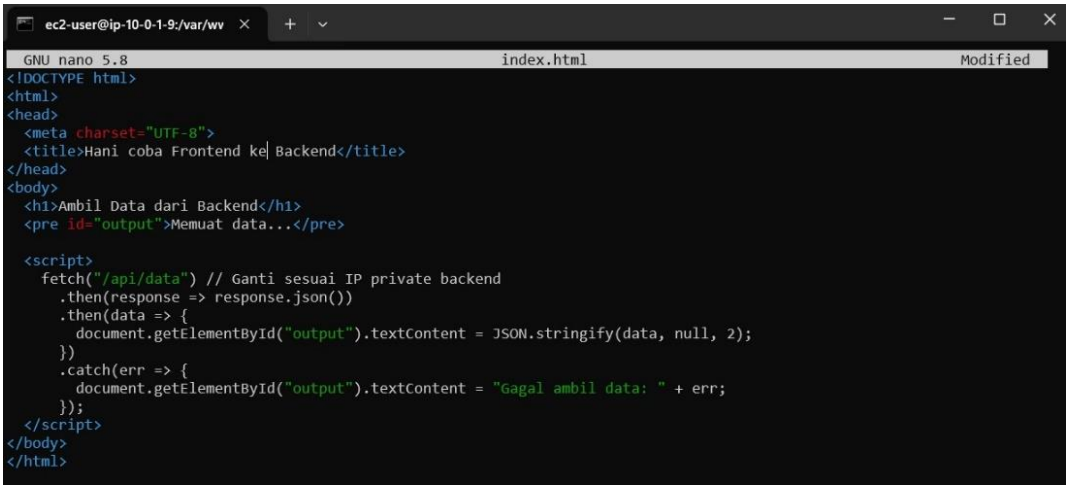
    ubah code server menjadi seperti ini :

```
server {
        listen       80;

        server_name  _;


        # Load configuration files for the default server block.
        include /etc/nginx/default.d/*.conf;

location /api/ {
            proxy_pass http://localhost:3000/;
            proxy_http_version 1.1;
            proxy_set_header Upgrade $http_upgrade;
            proxy_set_header Connection 'upgrade';
            proxy_set_header Host $host;
            proxy_cache_bypass $http_upgrade;
        }
        error_page 404 /404.html;
        location = /404.html {
        }

        error_page 500 502 503 504 /50x.html;
        location = /50x.html {
        }
    }
```

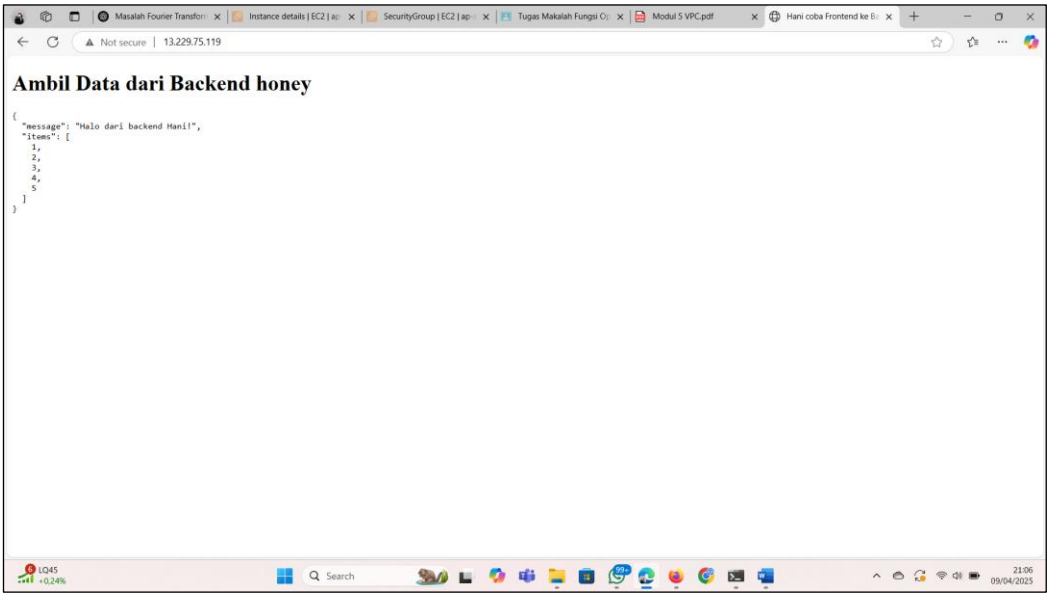
21. sudo yum install httpd

22. cd /var/www/html

23. lalu ketik perintah : sudo nano index.html



24. cobalah buka ip frontend pada browser



Jika tampil seperti ini maka frontend berhasil mengambil dan menampilkan data dari backend melalui koneksi IP public frontend. Data yang diterima berupa format JSON, yang kemudian ditampilkan di halaman browser. Hal ini menandakan bahwa integrasi antara frontend dan backend menggunakan Nginx berhasil.