LAPORAN PRAKTIKUM VIRTUALISASI KOMPUTER

NETWORK ATTACHED STORAGE MENGGUNAKAN NFS

Agus Pranata Marpaung

13323033

DIII TEKNOLOGI KOMPUTER

INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI

Judul Praktikum

Minggu/Sesi	:	VI/2
Kode Mata Kuliah	:	4332103
Nama Mata Kuliah	:	VIRTUALISASI KOMPUTER
Setoran	:	Jawaban dalam bentuk *softcopy*
Batas Waktu Setoran	:	*11 Oktober 2024 Jam 21:30*
Tujuan	:	1. Mahasiswa mampu mengimplementasikan penyimpanan distribusi menggunakan NFS.

Petunjuk

Teori

A. Apa itu NFS?

NFS adalah singkatan dari **Network File System**, yang pertama kali dikembangkan oleh Sun Microsystems pada tahun 80-an, sebagai sarana untuk berbagi file pada lingkungan kerja diskless. NFS menyediakan sarana untuk berbagi file antar jaringan, sehingga suatu mesin dapat mengakses file-file di mesin lain seolah-olah mengakses file sistem lokal.

B. Protokol NFS

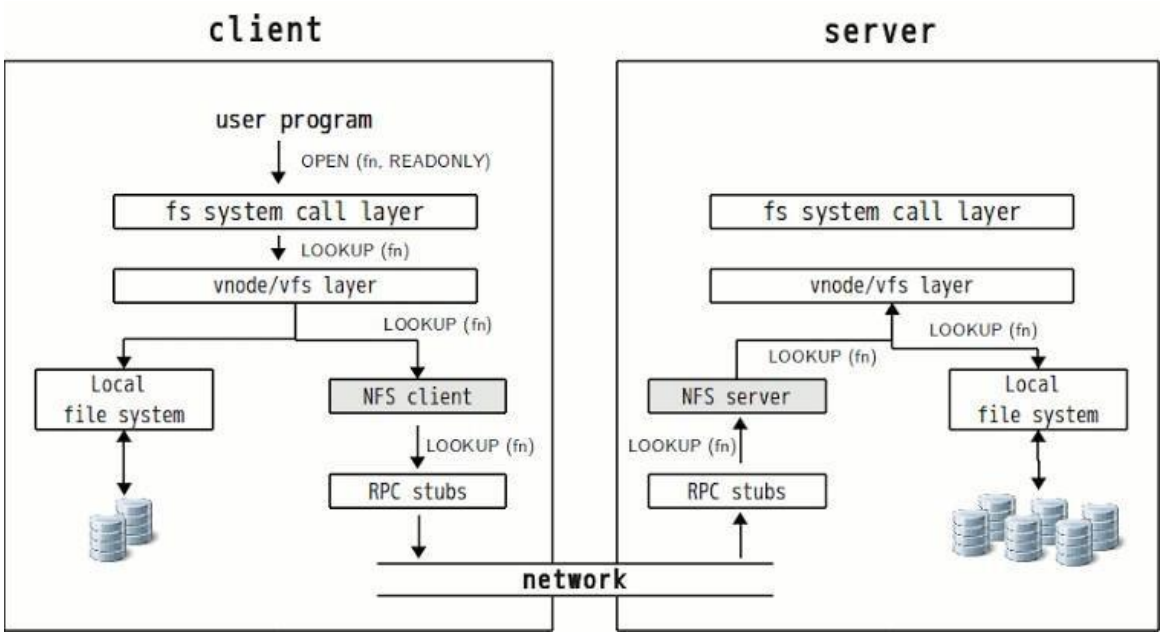
Protokol NFS adalah salah satu dari beberapa stkitar sistem file terdistribusi untuk penyimpanan yang terhubung dengan jaringan (network-attached storage — NAS). NFS umumnya menggunakan protokol Remote Procedure Call (RPC) yang berjalan di atas UDP dan membuka port UDP dengan port number 2049 untuk komunikasi antara client dan server di dalam jaringan.

Client NFS selanjutnya akan mengimpor sistem berkas remote dari server NFS, sementara server NFS mengekspor sistem berkas lokal kepada client.

C. Komponen NFS

a. **File System Model dan Architecture**: suatu mekanisme yang mendefinisikan sumber daya dan berkas yang akan digunakan secara bersama-sama dalam jaringan
b. **Resource Access Method**: tahapan-tahapan yang menggambarkan bagaimana pengguna melampirkan ataupun melepaskan sumber daya yang berasal dari tempat penyimpanan local mereka.
c. **Operation Set**: untuk mengatur operasi apa yang akan digunakan dan diperlukan pada saat pengguna menggunakan sumber daya yang digunakan bersama pada tempat penyimpanan local pengguna lain
d. **Messaging Protocols**: format pesan yang berisi operasi yang akan digunakan seperti informasi status dan protokol yang digunakan untuk bertukar pesan ini antar mesin pengguna
e. **Administrative Tool**: kumpulan fungsi yang dibutuhkan untuk mendukung operasi protokol dan penggunaan komponen lain yang mendukung

D. Diagram Architecture NFS



E. Tujuan NFS

Memungkinkan terjadinya pertukaran berkas antara mesin mesin secara transparan. Hubungannya yang terjadi disini adalah hubungan client — server dengan menggunakan perangkat lunak NFS Server dan NFS client yang berjalan diatas workstation. NFS didesain untuk dapat berjalan di platform yang heterogen. Adapun operasi-operasi yang didukung oleh NFS adalah sebagai berikut:

a. Mencari berkas didalam direktori
b. Membaca kumpulan direktori
c. Memanipulasi link dan direktori
d. Mengakses atribut berkas
e. Membaca dan menulis berkas

F. Manfaat NFS

a. Memungkinkan beberapa komputer menggunakan file yang sama, sehingga semua orang dapat mengakses file data yang sama
b. Mengurangi biaya penyimpanan.Memberikan konsistensi dan kekitalan data karena semua pengguna dapat membaca file data yang sama
c. Menjadikan sistem file secara transparan bagi pengguna
d. Dapat mengakses file jarak jauh bagi pengguna
e. Mendukung lingkungan yang heterogen
f. Mengurangi overhead administrasi system

G. Keamanan NFS

NFS sangat cocok untuk berbagi seluruh sistem file dengan sejumlah besar host yang dikenal secara transparan. Namun, dengan kemudahan penggunaan muncul berbagai potensi masalah keamanan.

H. File Permissions

Setelah sistem file NFS di-mount baca/tulis oleh host jarak jauh, satu-satunya perlindungan yang dimiliki setiap file bersama adalah izinnya. Jika dua pengguna yang berbagi nilai ID pengguna yang sama memasang sistem file NFS yang sama, mereka dapat saling memodifikasi file lainnya. Selain itu, siapa pun yang masuk sebagai root pada sistem klien dapat menggunakan perintah su- untuk menjadi pengguna yang dapat mengakses file tertentu melalui share NFS.

I. Contoh Implementasi NFS

a. FreeNFS
b. ProNFS
c. OpenTextNFS
d. Active *Directory*

Praktikum

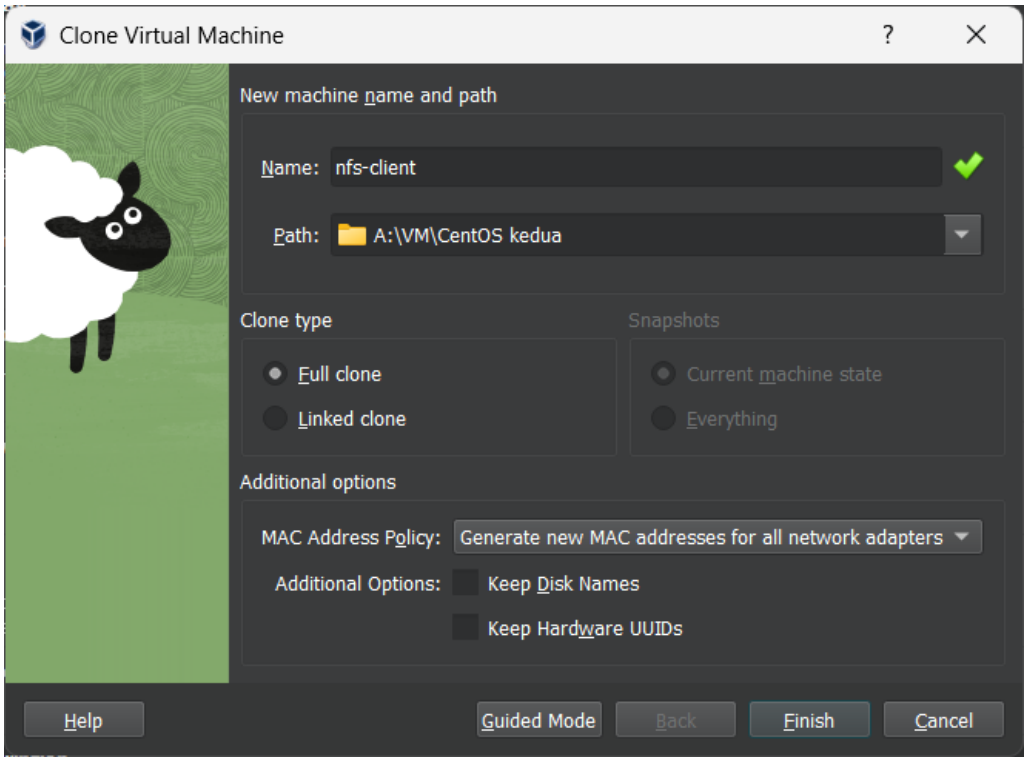
1. Konfigurasi NFS membutuhkan minimal 2 komputer, yaitu server dan client. Jadi, lakukan instalasi CentOS terlebih dahulu, setelah itu update CentOS yang kita instal. Kemudian, kita lakukan cloning CentOS dari VMware/Virtual Box, dengan metode **Full Clone**. Setelah itu, kita buat pembeda CentOS yang satu dengan yang lain.

   Jawab:

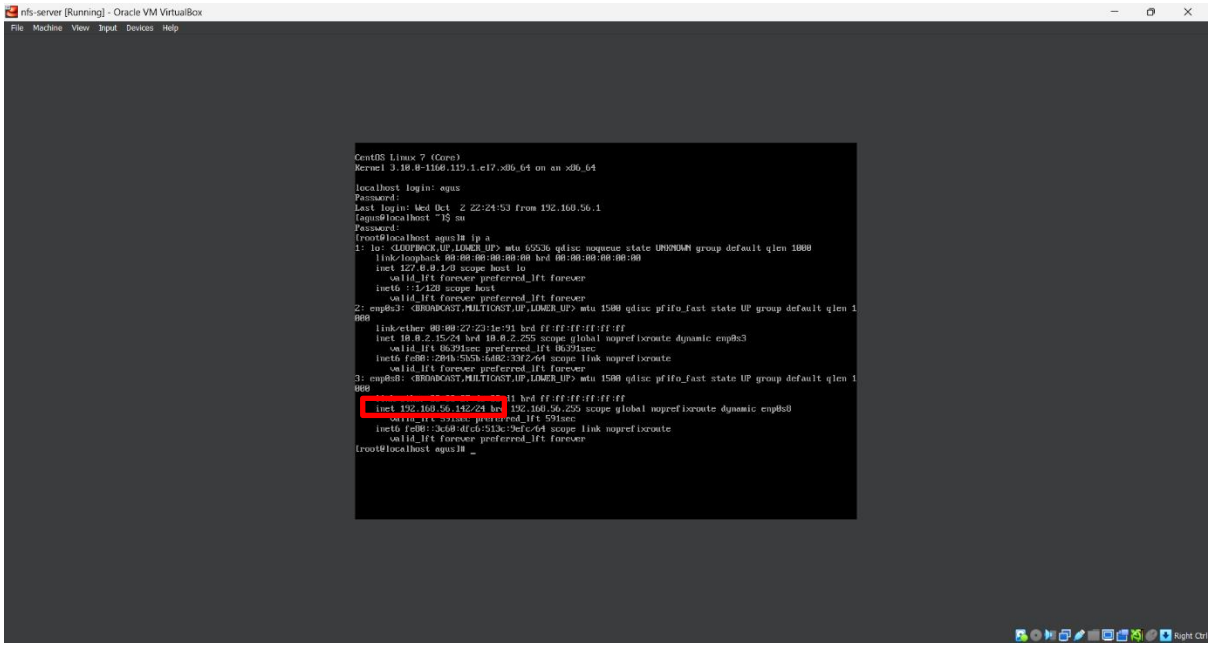
   Centos Server VM

Clone VM dari CentOS Server menjadi CentOS Client

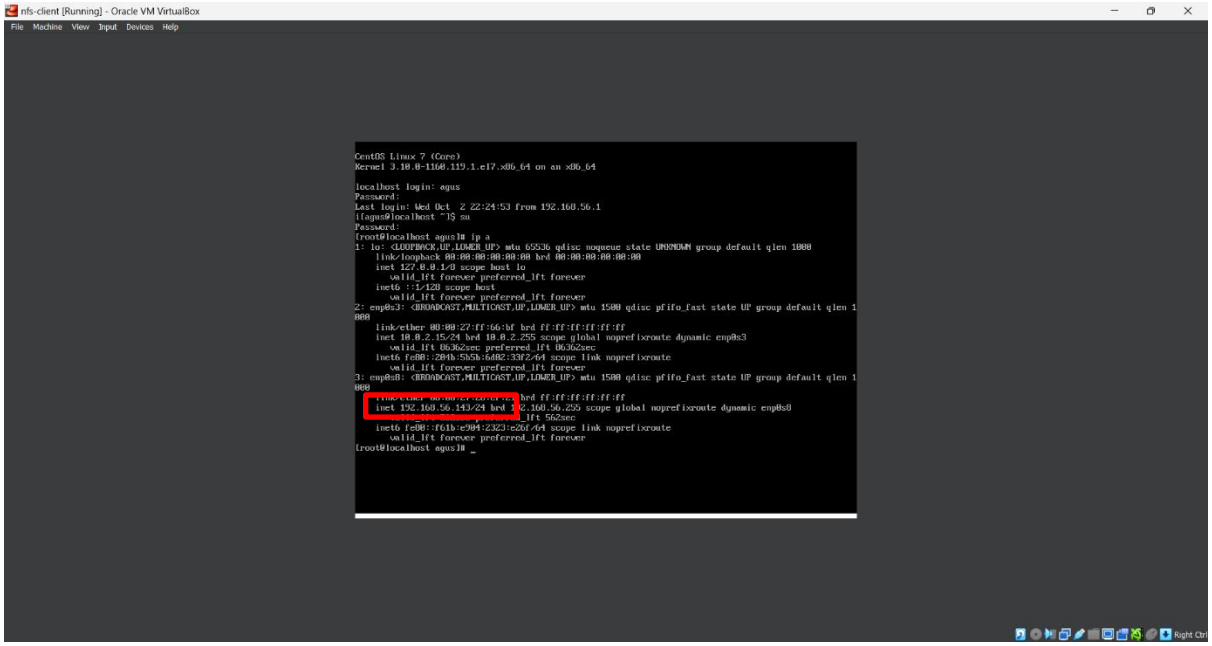
Karena memakai CentOS Server, jadi clone nya dipilih tipe full clone dan pilih Generate New MAC address for all network adapters.



2. Setelah CentOS sudah terinstal di VMware/Virtual Box, jalankan server dan clientnya. Setelah itu kita lihat terlebih dahulu IP dari kedua CentOS tersebut.
   a. IP address NFS-Server (**192.168.56.142**)



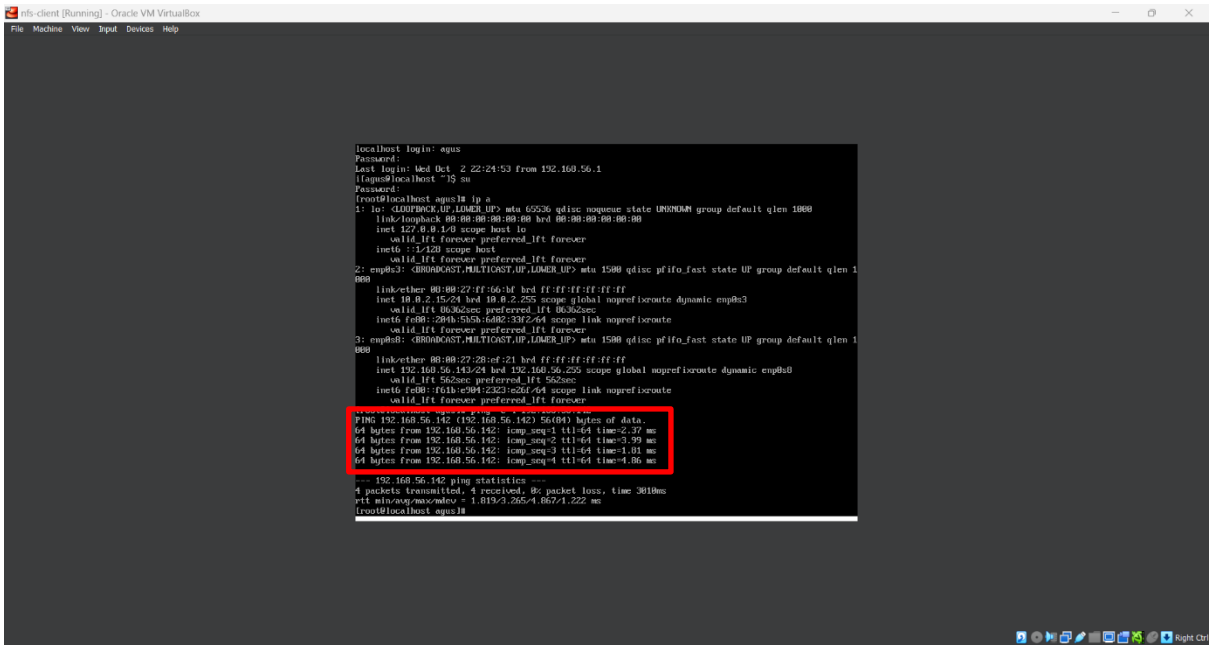
   b. IP address NFS-Client (**192.168.56.143**)



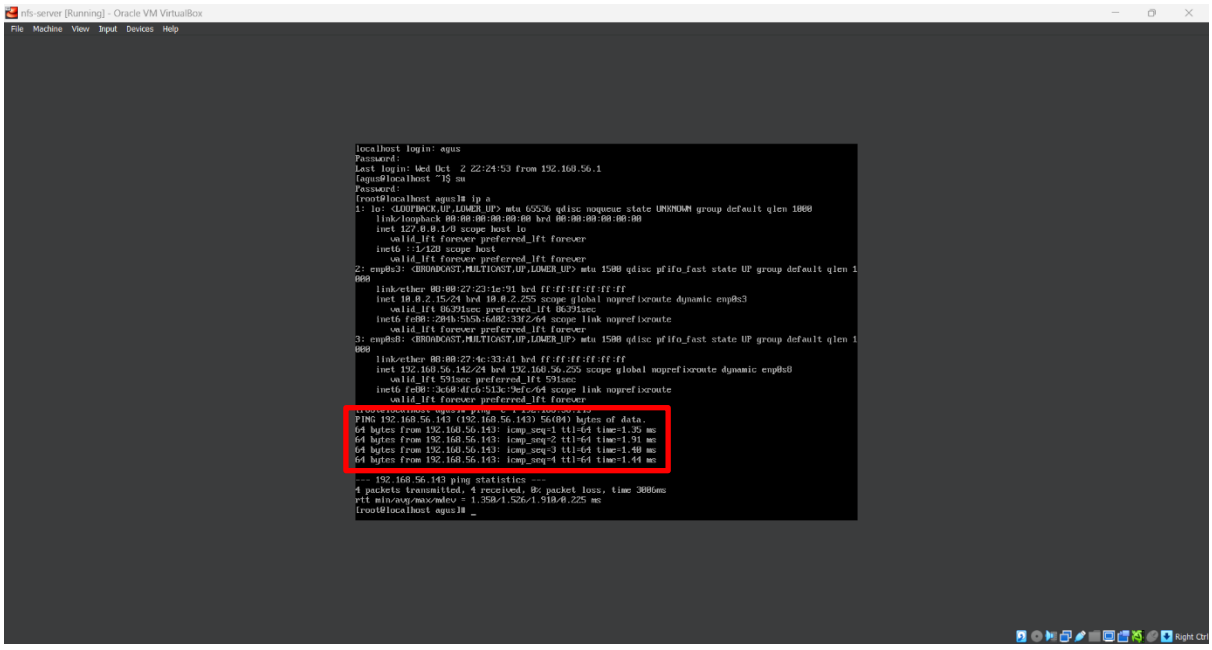
3. Kemudian pastikan server dan client bisa melakukan ping dengan menggunakan perintah berikut ini.

   Jawab:

   1. Ping Centos Client ke Centos Server

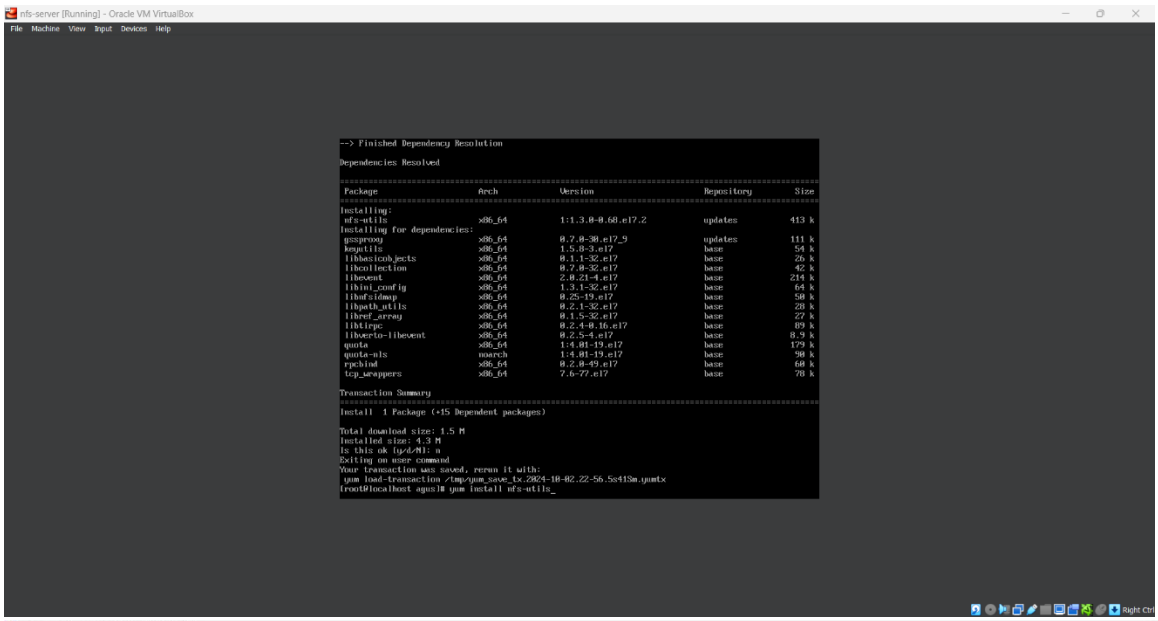


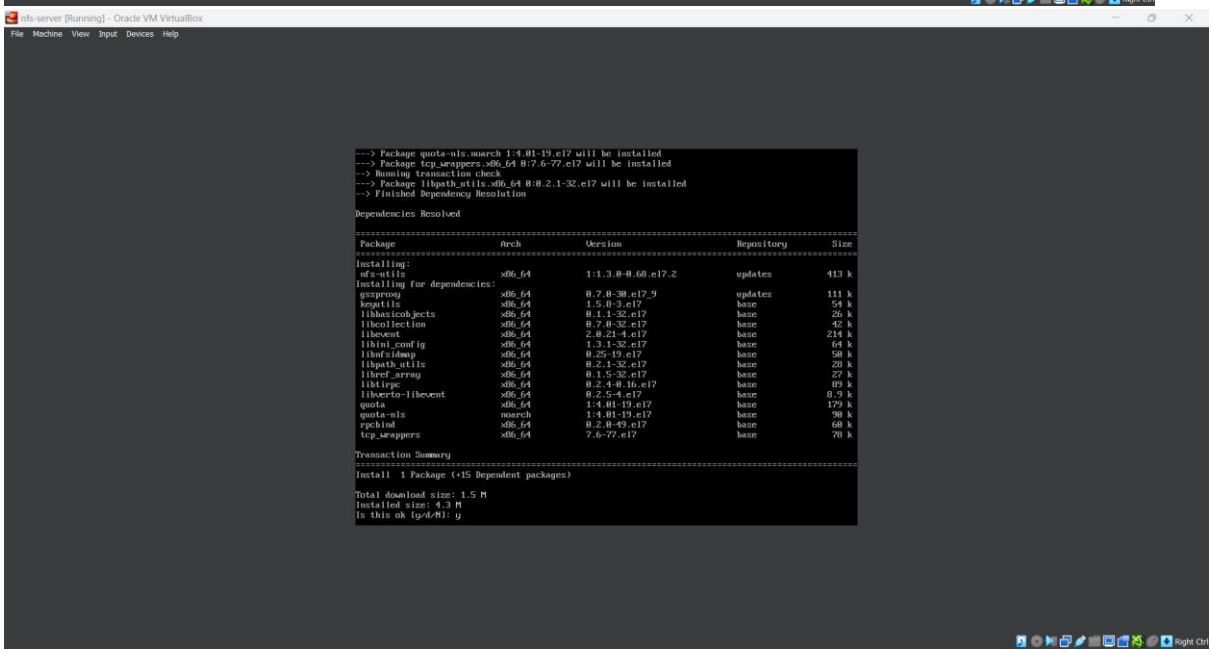
   2. Ping Centos Server ke Centos Client



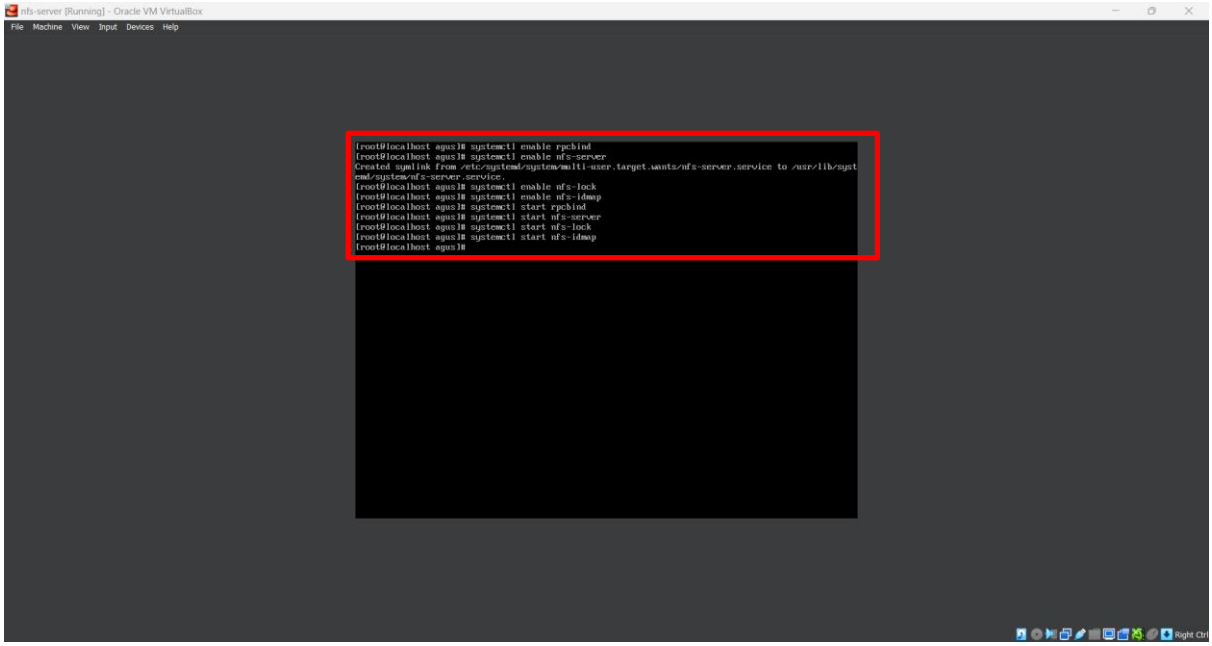
4. Setelah itu, maka kita bisa menginstal package NFS pada server dengan perintah dibawah ini.

```
yum install nfs-utils
```

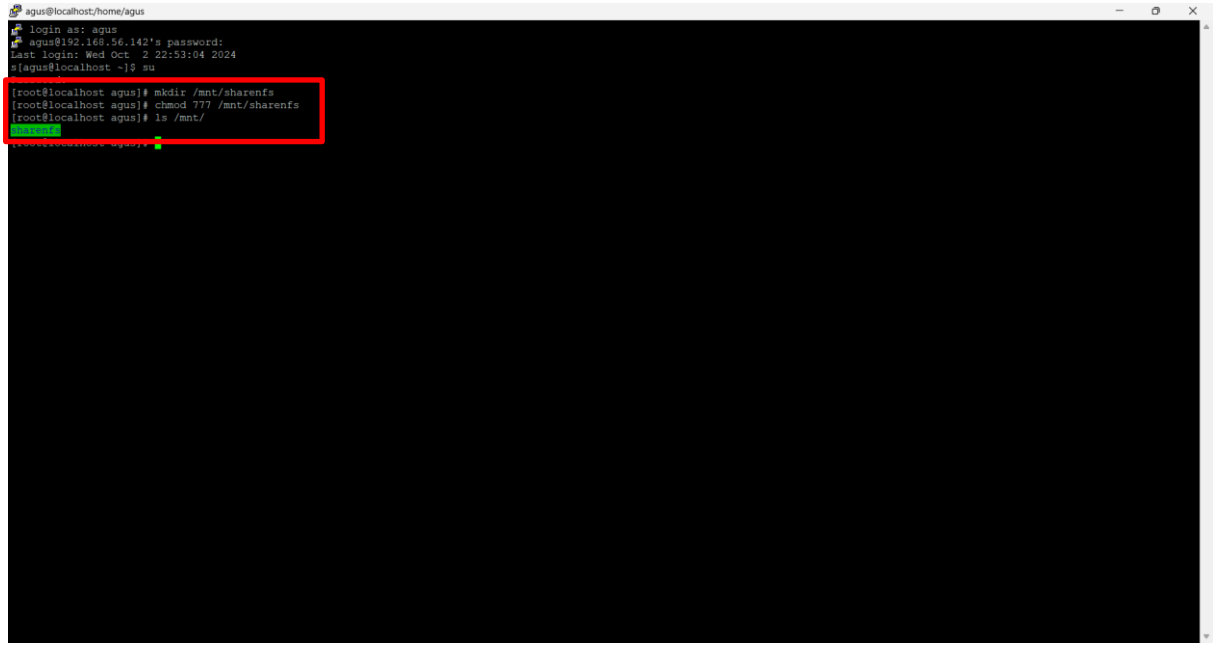




5. Aktifkan dan jalankan NFS service dengan menggunakan command berikut ini.



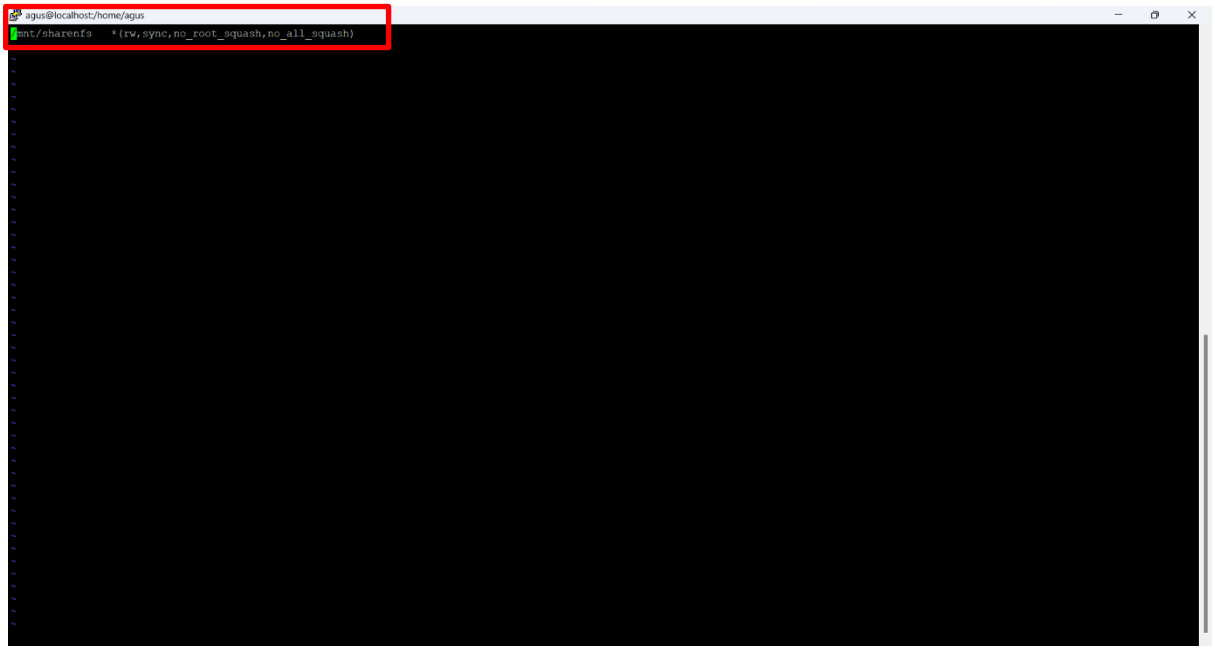
6. Kemudian, buatlah shared *directory* dengan nama **`/mnt/sharenfs`** di server dan izinkan client untuk membaca dan menulis *directory* yang dibuat itu.



7. Modifikasi file **`/etc/exports`** untuk membuat entri *directory* yang mau dibagikan, yaitu **`/mnt/sharenfs`** dengan menjalankan perintah berikut ini.

```
vi /etc/exports
```

8. Kemudian tambahkan line berikut ini.

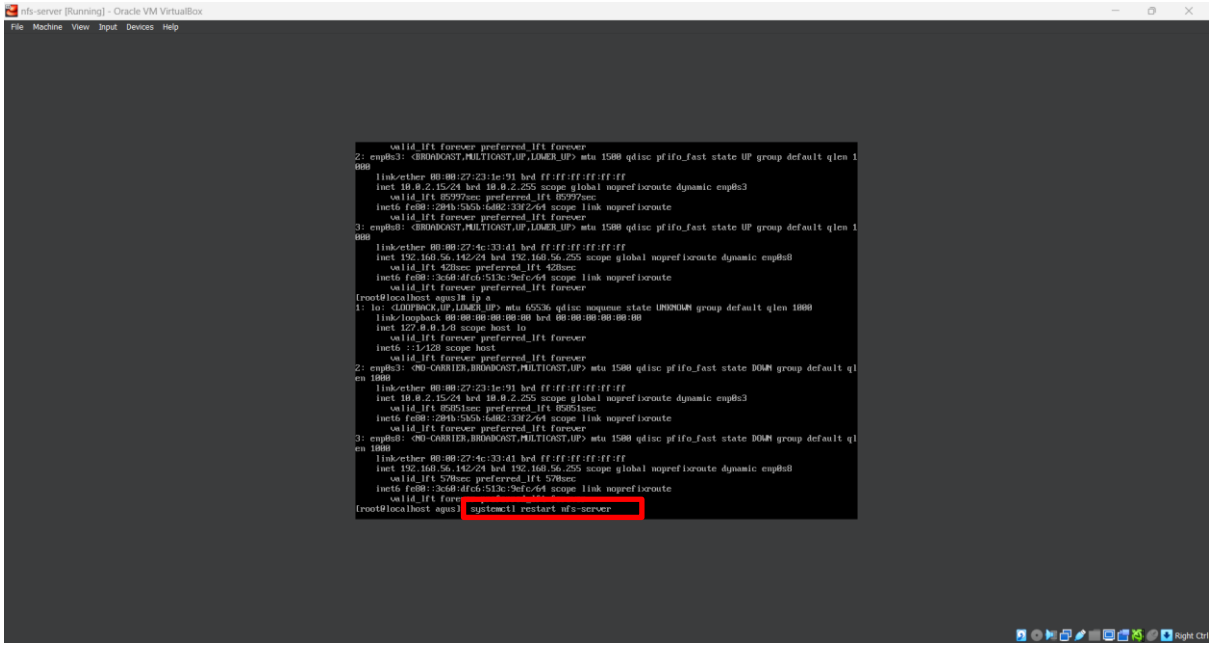


Keterangan:

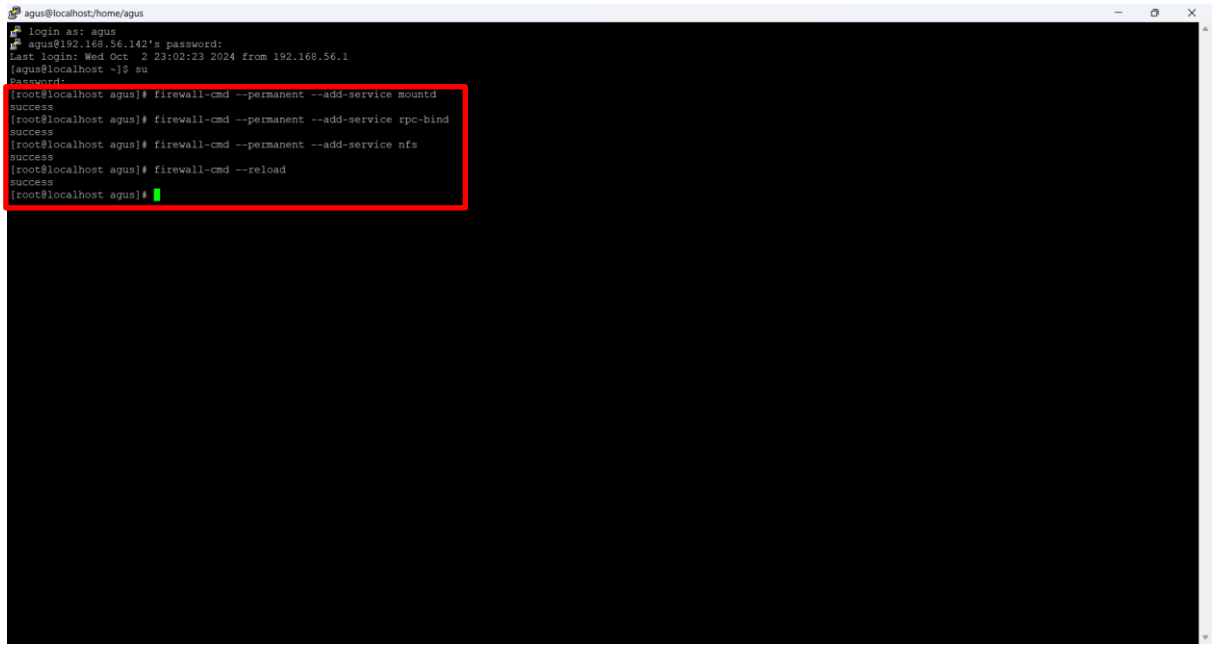
`/mnt/sharenfs`	:*Directory* yang dibagikan
`192.168.56.143`	:IP address client
`rw`	:Izin untuk menulis pada *directory* yang dibagikan
`sync`	:Sinkronisasi *directory* yang dibagikan
`no_root_squash`	:Mengaktifkan hak akses root
`no_all_squash`	:Mengaktifkan otoritas user

9. *Directory* yang dishare adalah *directory* **`/mnt/sharenfs`**. *Directory* itu hanya dishare ke client dengan IP address 192.168.56.143 (Sesuaikan dengan IP address client di laptop kita). Apabila kita ingin share ke semua IP address, maka silahkan ganti IP address dengan "*".

10. Setelah dimodifikasi, restart service **nfs-server** dengan perintah berikut ini.

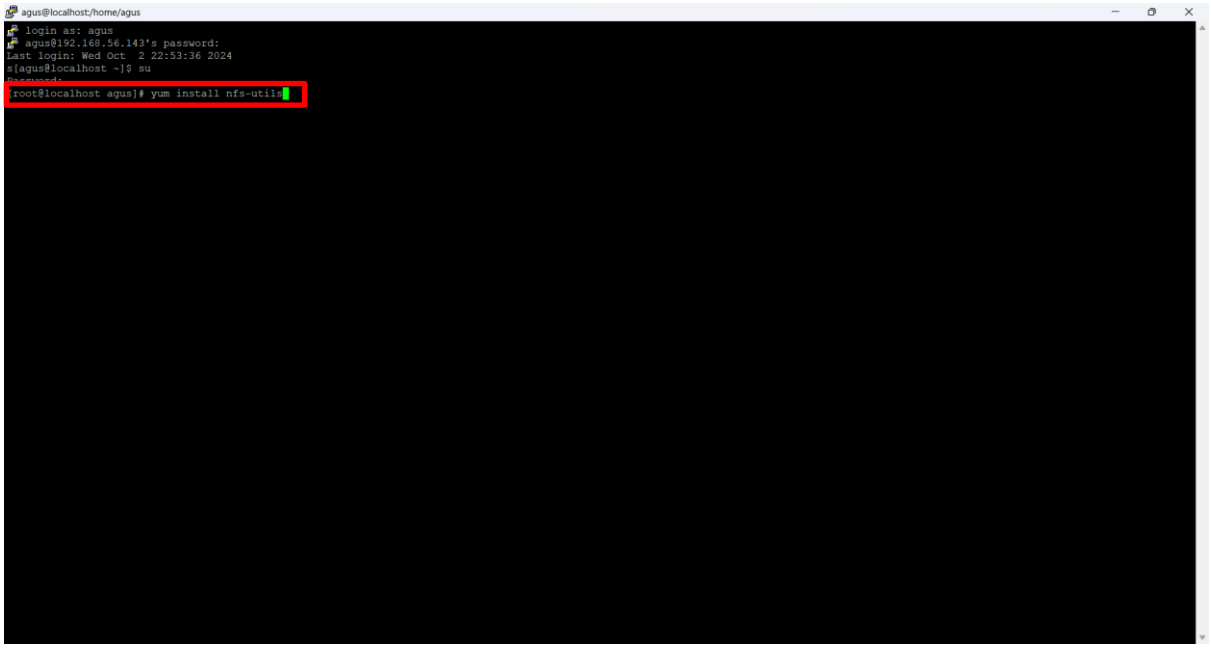


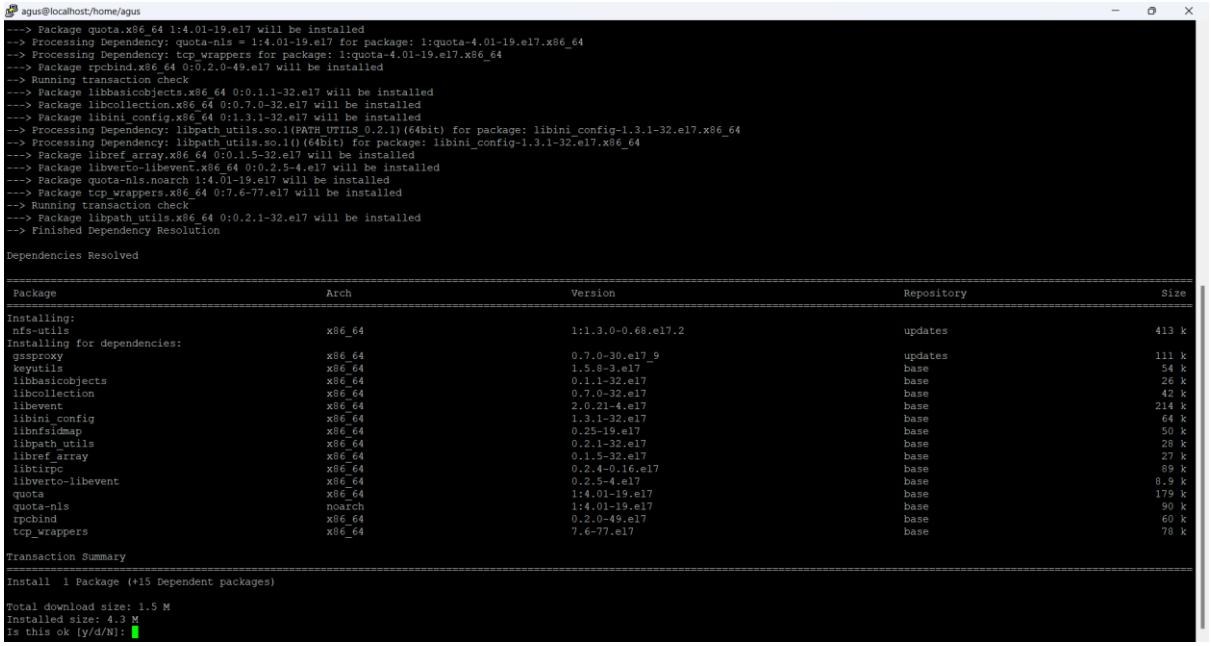
11. Kemudian kita harus mengkonfigurasi firewall pada NFS-Server agar NFS-Client dapat mengakses NFS share dengan menjalankan perintah berikut di NFS-Server.

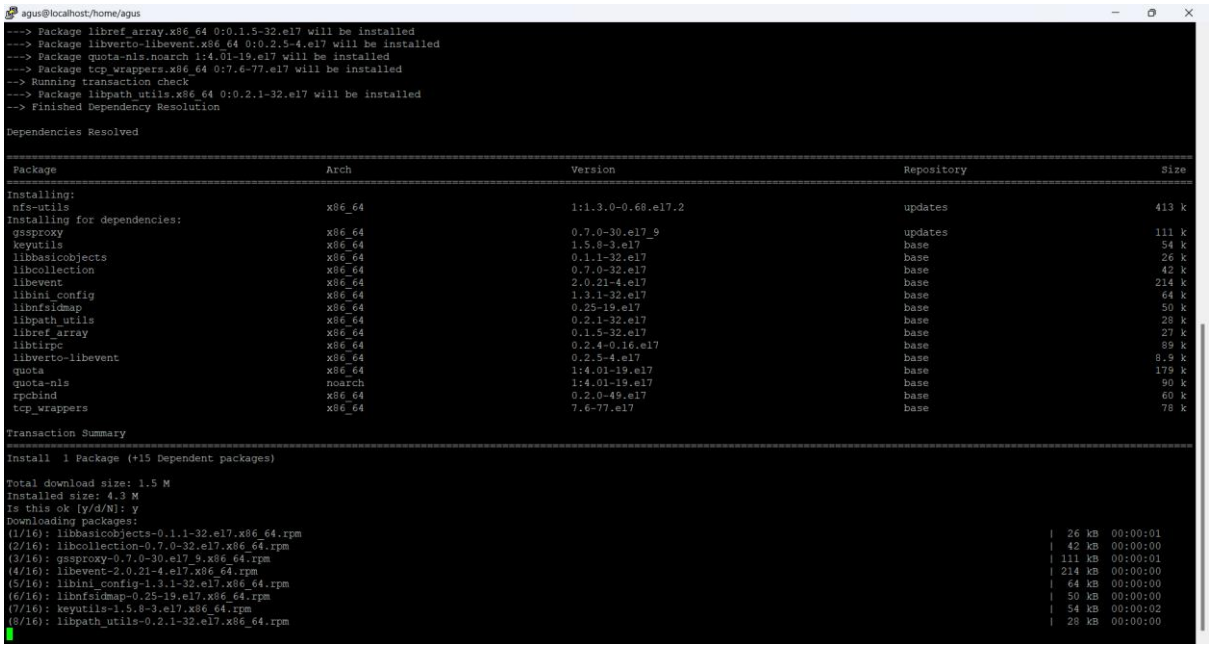


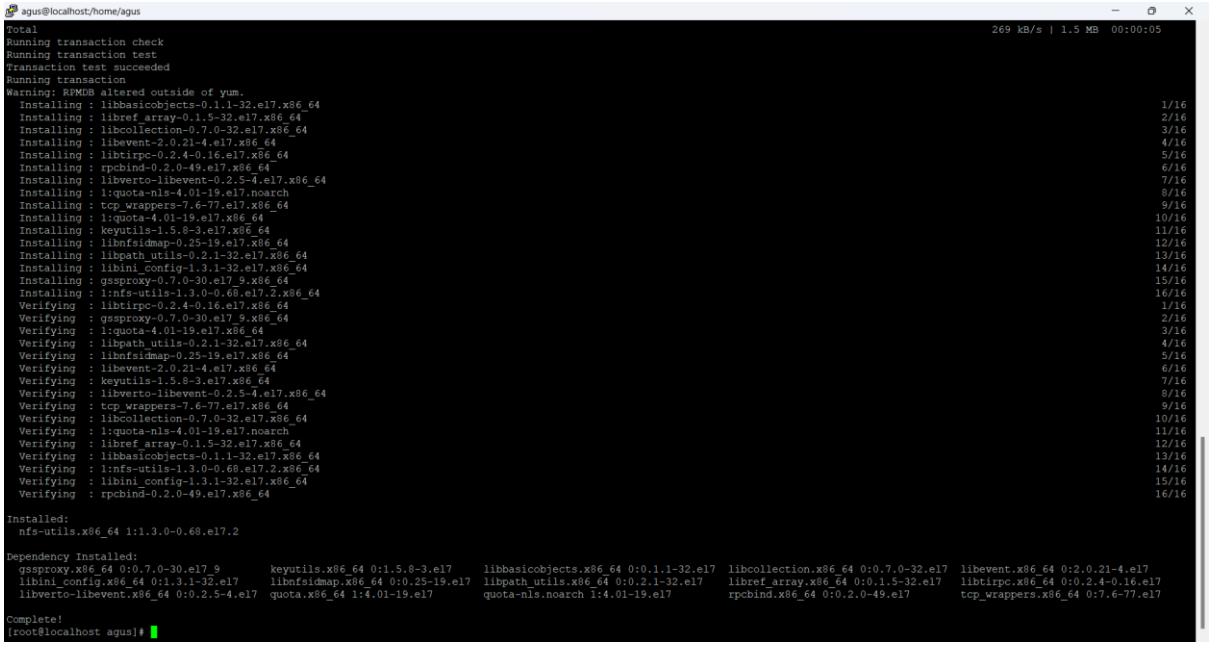
12. Pada sisi client, instal NFS packages dengan menggunakan perintah berikut ini.

```
yum install nfs-utils
```

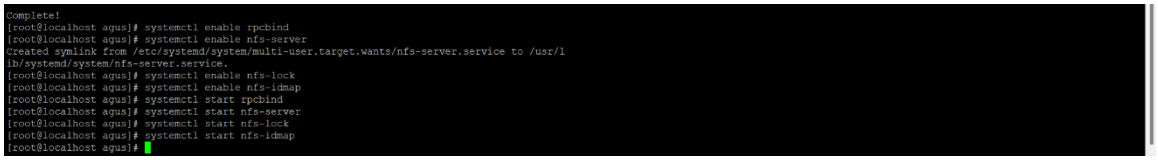



agus@localhost:/home/agus
---> Package quota.x86_64 1:4.01-19.el7 will be installed
--> Processing Dependency: quota-nls = 1:4.01-19.el7 for package: 1:quota-4.01-19.el7.x86_64
--> Processing Dependency: tcp_wrappers for package: 1:quota-4.01-19.el7.x86_64
---> Package rpcbind.x86_64 0:0.2.0-49.el7 will be installed
--> Running transaction check
---> Package libbasicobjects.x86_64 0:0.1.1-32.el7 will be installed
---> Package libcollection.x86_64 0:0.7.0-32.el7 will be installed
---> Package libini_config.x86_64 0:1.3.1-32.el7 will be installed
--> Processing Dependency: libpath_utils.so.1(PATH_UTILS_0.2.1)(64bit) for package: libini_config-1.3.1-32.el7.x86_64
--> Processing Dependency: libpath_utils.so.1()(64bit) for package: libini_config-1.3.1-32.el7.x86_64
---> Package libref_array.x86_64 0:0.1.5-32.el7 will be installed
---> Package libverto-libevent.x86_64 0:0.2.5-4.el7 will be installed
---> Package quota-nls.noarch 1:4.01-19.el7 will be installed
---> Package tcp_wrappers.x86_64 0:7.6-77.el7 will be installed
--> Running transaction check
---> Package libpath_utils.x86_64 0:0.2.1-32.el7 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
Package Arch Version Repository Size
Installing:
nfs-utils x86_64 1:1.3.0-0.68.el7.2 updates 413 k
Installing for dependencies:
gssproxy x86_64 0.7.0-30.el7_9 updates 111 k
keyutils x86_64 1.5.8-3.el7 base 54 k
libbasicobjects x86_64 0.1.1-32.el7 base 26 k
libcollection x86_64 0.7.0-32.el7 base 42 k
libevent x86_64 2.0.21-4.el7 base 214 k
libini_config x86_64 1.3.1-32.el7 base 64 k
libnfsidmap x86_64 0.25-19.el7 base 50 k
libpath_utils x86_64 0.2.1-32.el7 base 28 k
libref_array x86_64 0.1.5-32.el7 base 27 k
libtirpc x86_64 0.2.4-0.16.el7 base 89 k
libverto-libevent x86_64 0.2.5-4.el7 base 8.9 k
quota x86_64 1:4.01-19.el7 base 179 k
quota-nls noarch 1:4.01-19.el7 base 90 k
rpcbind x86_64 0.2.0-49.el7 base 60 k
tcp_wrappers x86_64 7.6-77.el7 base 78 k
Transaction Summary
Install 1 Package (+15 Dependent packages)
Total download size: 1.5 M
Installed size: 4.3 M
Is this ok [y/d/N]:

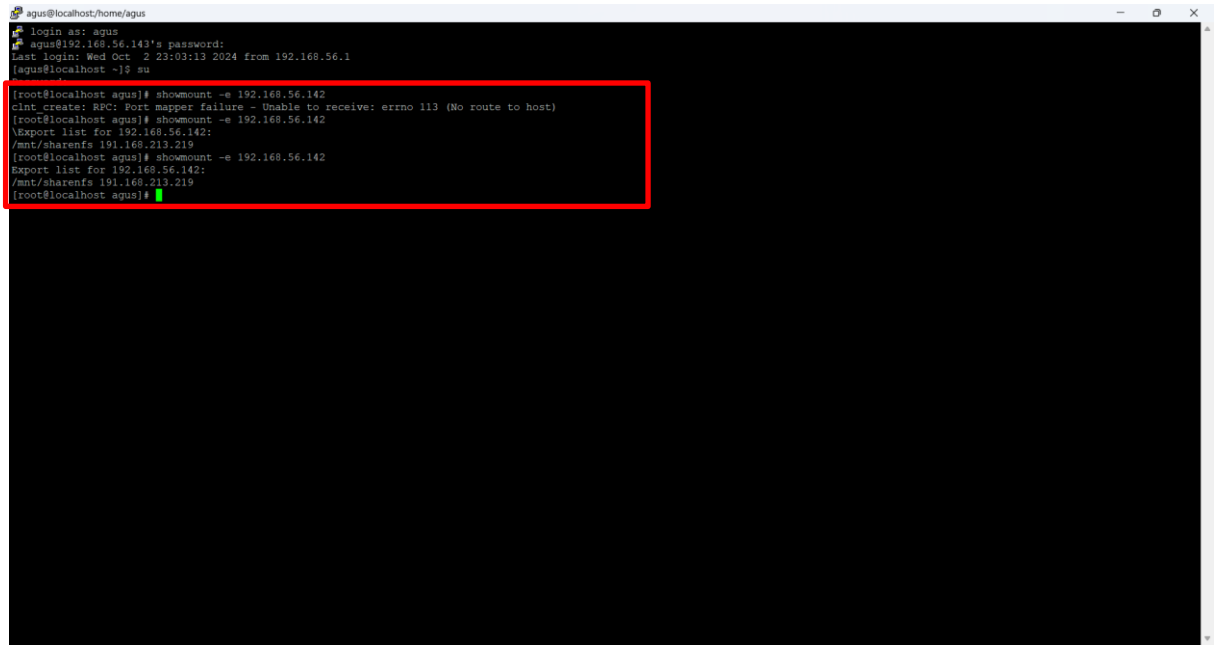

agus@localhost:/home/agus
---> Package libref_array.x86_64 0:0.1.5-32.el7 will be installed
---> Package libverto-libevent.x86_64 0:0.2.5-4.el7 will be installed
---> Package quota-nls.noarch 1:4.01-19.el7 will be installed
---> Package tcp_wrappers.x86_64 0:7.6-77.el7 will be installed
--> Running transaction check
---> Package libpath_utils.x86_64 0:0.2.1-32.el7 will be installed
--> Finished Dependency Resolution
Dependencies Resolved
Package Arch Version Repository Size
Installing:
nfs-utils x86_64 1:1.3.0-0.68.el7.2 updates 413 k
Installing for dependencies:
gssproxy x86_64 0.7.0-30.el7_9 updates 111 k
keyutils x86_64 1.5.8-3.el7 base 54 k
libbasicobjects x86_64 0.1.1-32.el7 base 26 k
libcollection x86_64 0.7.0-32.el7 base 42 k
libevent x86_64 2.0.21-4.el7 base 214 k
libini_config x86_64 1.3.1-32.el7 base 64 k
libnfsidmap x86_64 0.25-19.el7 base 50 k
libpath_utils x86_64 0.2.1-32.el7 base 28 k
libref_array x86_64 0.1.5-32.el7 base 27 k
libtirpc x86_64 0.2.4-0.16.el7 base 89 k
libverto-libevent x86_64 0.2.5-4.el7 base 8.9 k
quota x86_64 1:4.01-19.el7 base 179 k
quota-nls noarch 1:4.01-19.el7 base 90 k
rpcbind x86_64 0.2.0-49.el7 base 60 k
tcp_wrappers x86_64 7.6-77.el7 base 78 k
Transaction Summary
Install 1 Package (+15 Dependent packages)
Total download size: 1.5 M
Installed size: 4.3 M
Is this ok [y/d/N]: y
Downloading packages:
(1/16): libbasicobjects-0.1.1-32.el7.x86_64.rpm | 26 kB 00:00:01
(2/16): libcollection-0.7.0-32.el7.x86_64.rpm | 42 kB 00:00:00
(3/16): gssproxy-0.7.0-30.el7_9.x86_64.rpm | 111 kB 00:00:01
(4/16): libevent-2.0.21-4.el7.x86_64.rpm | 214 kB 00:00:00
(5/16): libini_config-1.3.1-32.el7.x86_64.rpm | 64 kB 00:00:00
(6/16): libnfsidmap-0.25-19.el7.x86_64.rpm | 50 kB 00:00:00
(7/16): keyutils-1.5.8-3.el7.x86_64.rpm | 54 kB 00:00:02
(8/16): libpath_utils-0.2.1-32.el7.x86_64.rpm | 28 kB 00:00:00



13. Aktifkan dan jalankan NFS service dengan menggunakan command berikut ini.



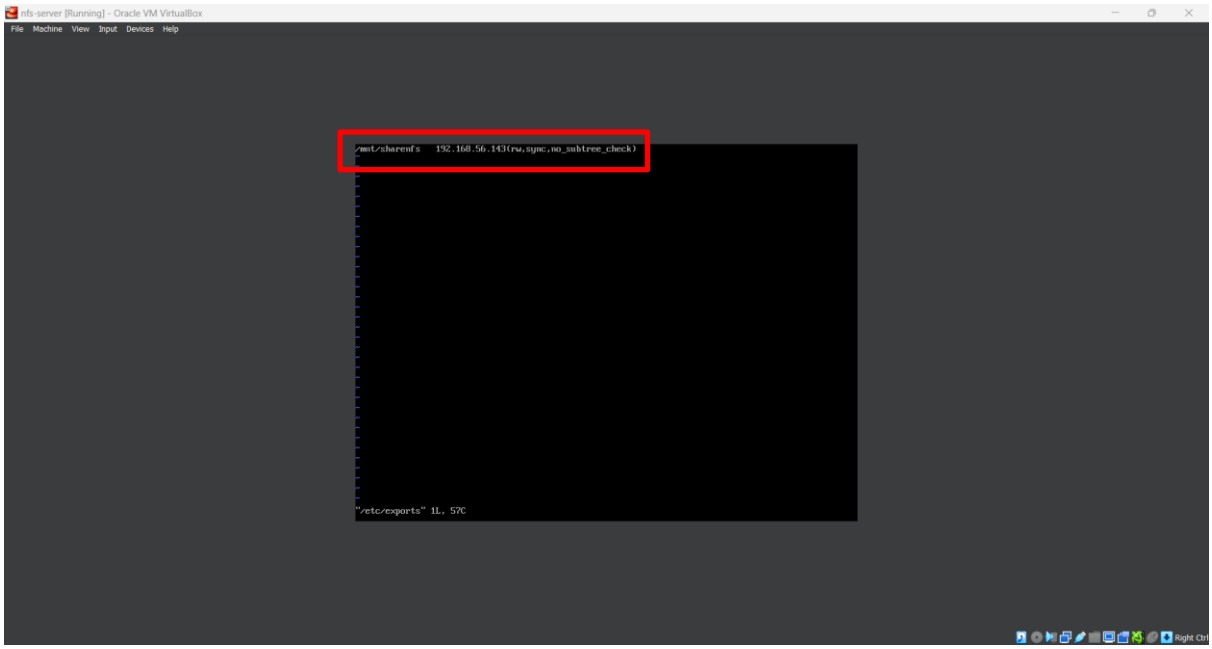
14. Sebelum mounting NFS, Kita harus memeriksa bagian NFS yang tersedia di NFS-Server dengan menjalankan perintah berikut pada NFS-Client.



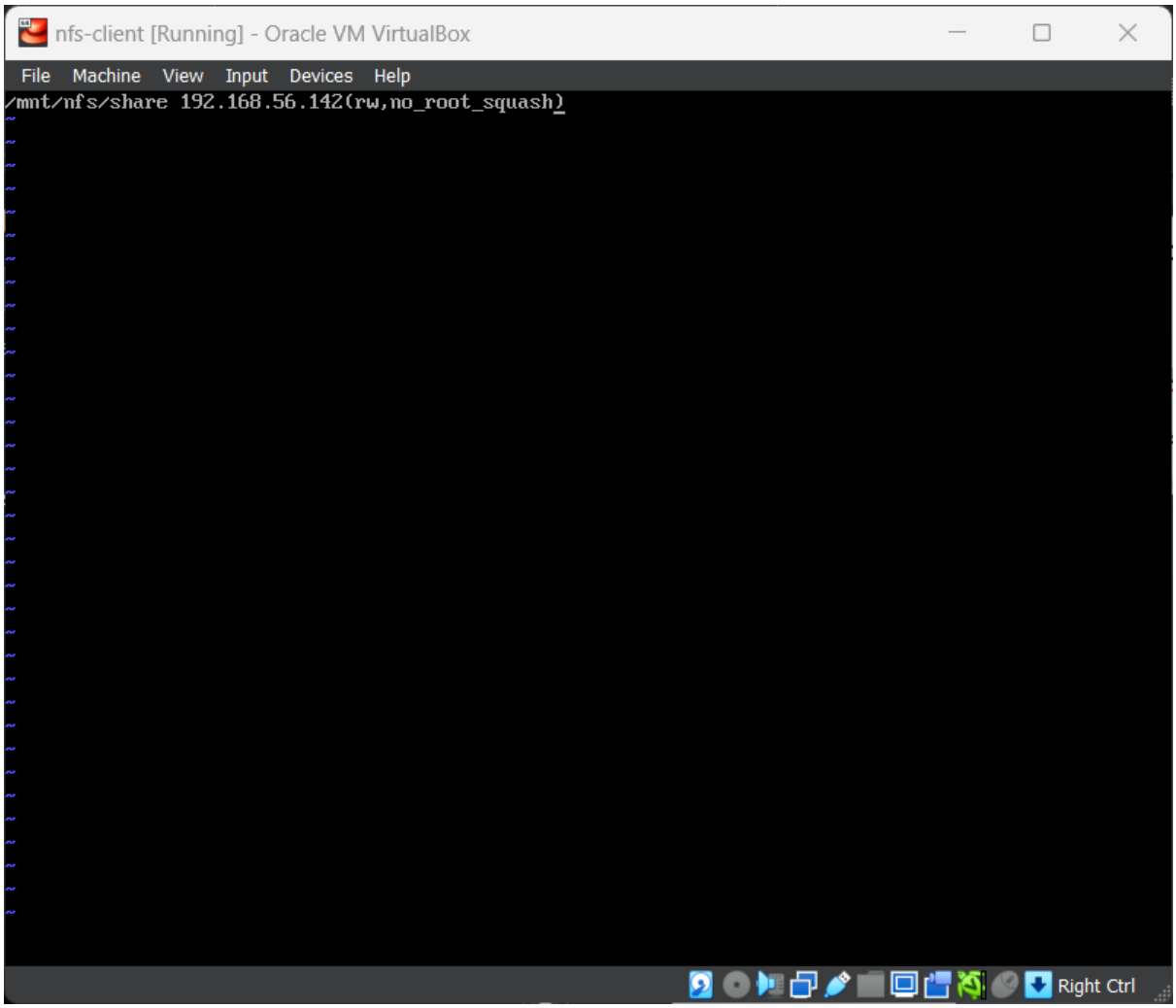
Sesuai output, **`/mnt/sharenfs`** tersedia di NFS-Server (192.168.56.142) untuk NFS Client (192.168.56.143)

15. Sekarang, buat *directory* pada NFS-Client untuk mount NFS share **`/mnt/sharenfs/`** yang telah kita buat di NFS-Server. Untuk *directory*-nya buat dengan nama **`/mnt/nfs/share`**.

16. Jalankan perintah dibawah ini untuk mount NFS share **`/mnt/sharenfs`** dari NFS-Server (192.168.56.142) di **`/mnt/nfs/share`** pada NFS-Client.



- Konfigurasikan IP Address server di client




```
/etc/exports" 1L, 57C written
[root@localhost agus]# sudo exportfs -ra
```

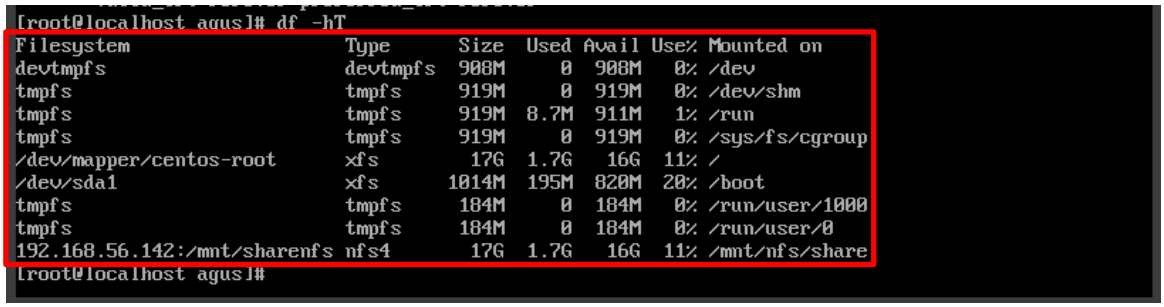
```
[root@localhost agus]# sudo systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor preset: disabled)
   Active: active (exited) since Wed 2024-10-02 23:11:47 WIB; 44min ago
  Process: 8861 ExecStopPost=/usr/sbin/exportfs -f (code=exited, status=0/SUCCESS)
  Process: 8859 ExecStopPost=/usr/sbin/exportfs -au (code=exited, status=0/SUCCESS)
  Process: 8858 ExecStop=/usr/sbin/rpc.nfsd 0 (code=exited, status=0/SUCCESS)
  Process: 8888 ExecStartPost=/bin/sh -c if systemctl -q is-active gssproxy; then systemctl reload g
ssproxy ; fi (code=exited, status=0/SUCCESS)
  Process: 8871 ExecStart=/usr/sbin/rpc.nfsd $RPCNFSDARGS (code=exited, status=0/SUCCESS)
  Process: 8870 ExecStartPre=/usr/sbin/exportfs -r (code=exited, status=0/SUCCESS)
 Main PID: 8871 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nfs-server.service

Oct 02 23:11:47 localhost.localdomain systemd[1]: Starting NFS server and services...
Oct 02 23:11:47 localhost.localdomain systemd[1]: Started NFS server and services.
[root@localhost agus]#
```

- Melakukan Mount dari IP Address server pada CentOS-Server

```
[root@localhost agus]# sudo mkdir -p /mnt/nfs/share
[root@localhost agus]# ls
[root@localhost agus]# dir
[root@localhost agus]# mkdir -p /mnt/nfs/share
[root@localhost agus]# ls
[root@localhost agus]# dir
[root@localhost agus]# mount 192.168.56.142:/mnt/sharenfs /mnt/nfs/share
[root@localhost agus]#
```

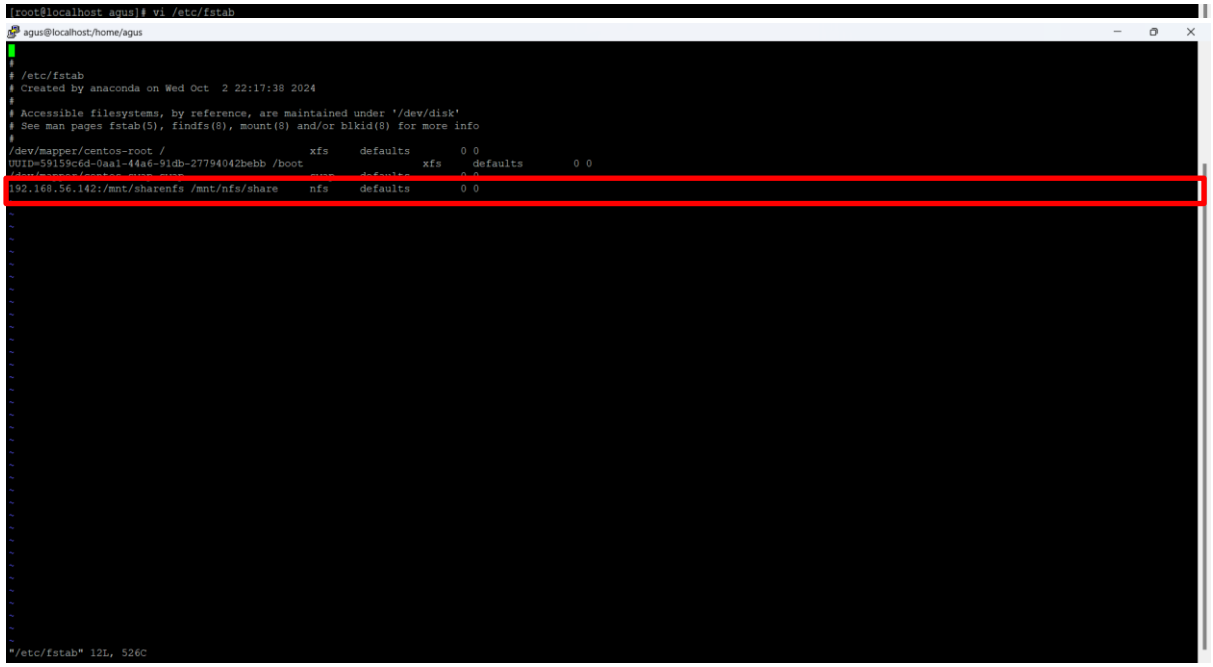
17. Verifikasi *directory* yang dishare pada NFS-Client menggunakan perintah **mount** dan juga bisa **`df -hT`**.



18. Setelah semua dilakukan, kita uji untuk membuat file di *directory* yang dishare untuk memverifikasi apakah client bisa memiliki hak akses membaca dan menulis pada NFS share.

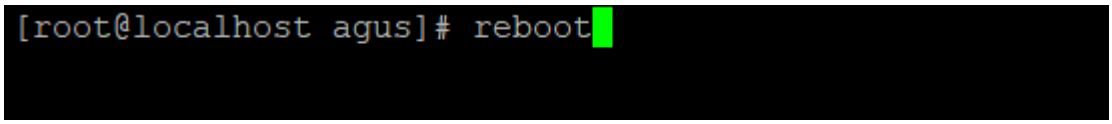
Disini terlihat outputnya tidak menampilkan error dan terdapat file test di *directory* yang dishare. Ini artinya NFS berhasil di konfigurasi.

19. Kemudian disini kita akan mount share secara otomatis setiap CentOS direboot, sehingga kita perlu memodifikasi file **`/etc/fstab`** pada NFS-Client kita dan menambahkan line berikut ini.

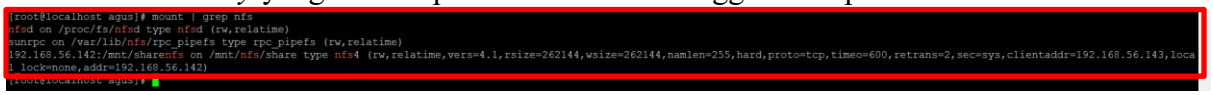


20. Kemudian lakukan reboot pada NFS-Client.

```
reboot
```



21. Verifikasi *directory* yang dishare pada NFS-Client menggunakan perintah mount.



Disini terlihat bahwa NFS bisa dipakai meskipun CentOS Client direstart karena sudah ditambahkan pada file **`/etc/fstab`**.

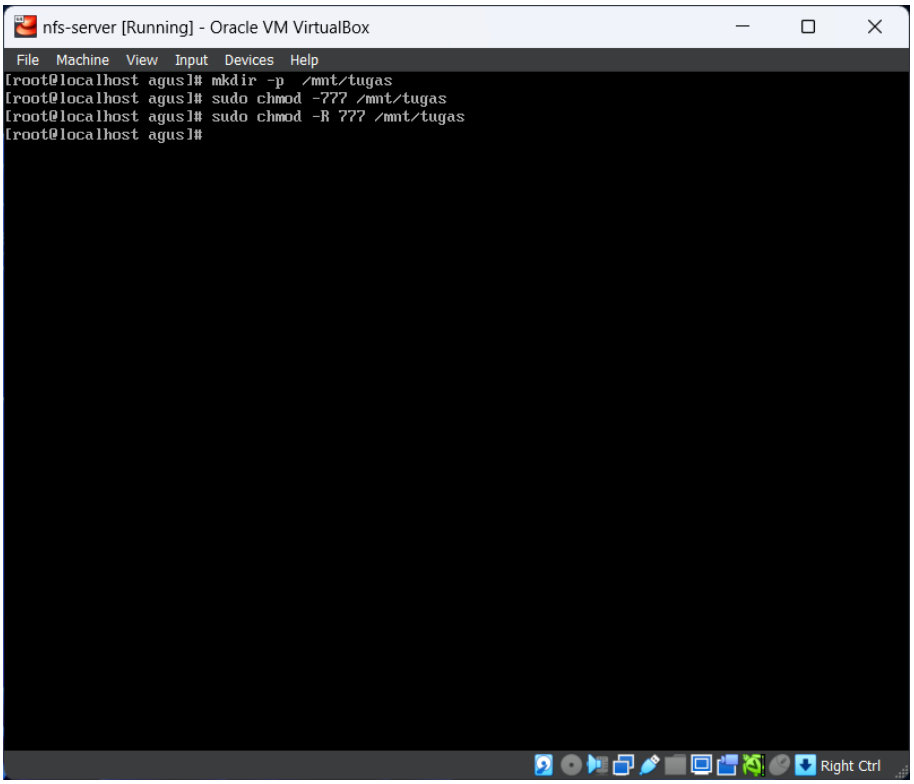
22. Selesai!

Tugas

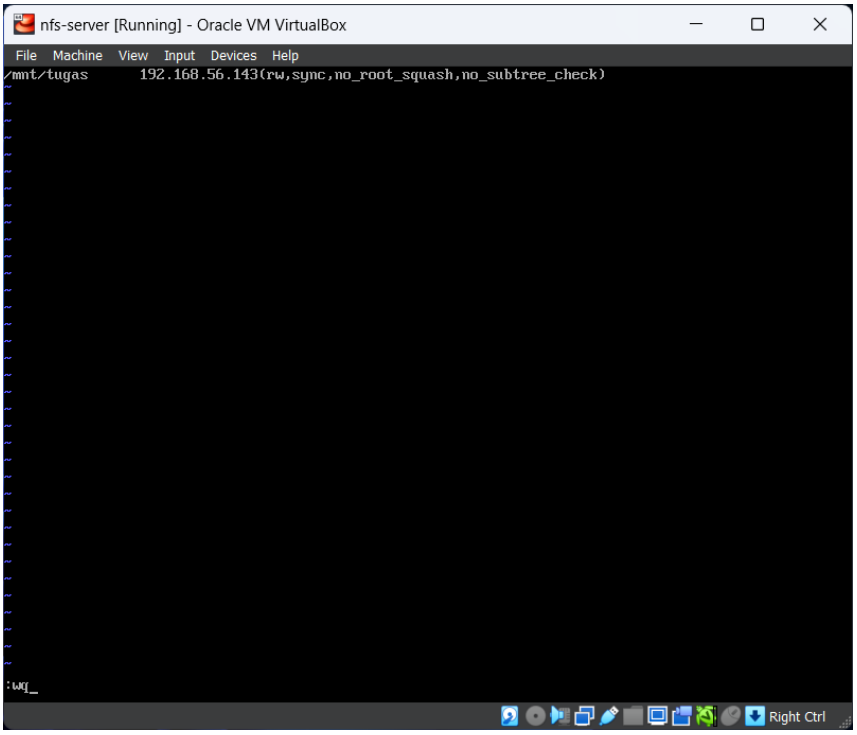
1. Tambahkan sebuah file pada *directory* yang dishare dan cek pada NFS-Server dan ujilah apakah file tersebut ada setelah berhasil ditambahkan dari NFS-Client kita.

   Jawab:

   1. Kita buat direktori baru pada NFS-Server dan memberikan akses pada file yang dibuat.



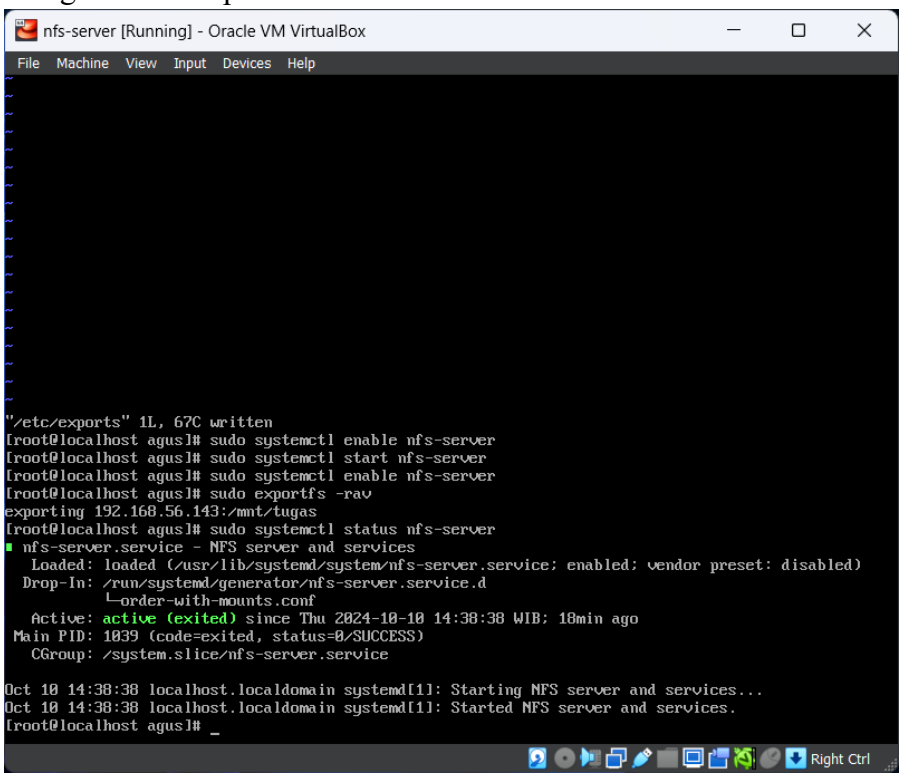
   2. Edit /etc/exports yang ada di NFS-Server. Ketika sudah selesai, ketik :wq untuk menyimpan



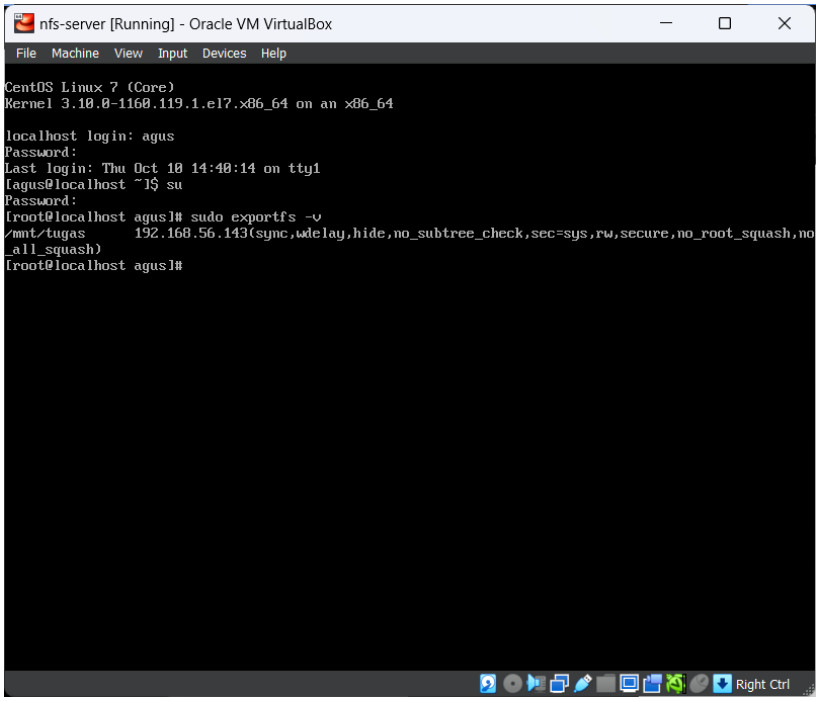
Keterangan:

- /mnt/tugas : Direktori pada server
- 192.168.56.143 : IP Server
- (rw,sync,no_root_squash,no_subtree_check) : Untuk memberikan akses pada direktori tersebut.

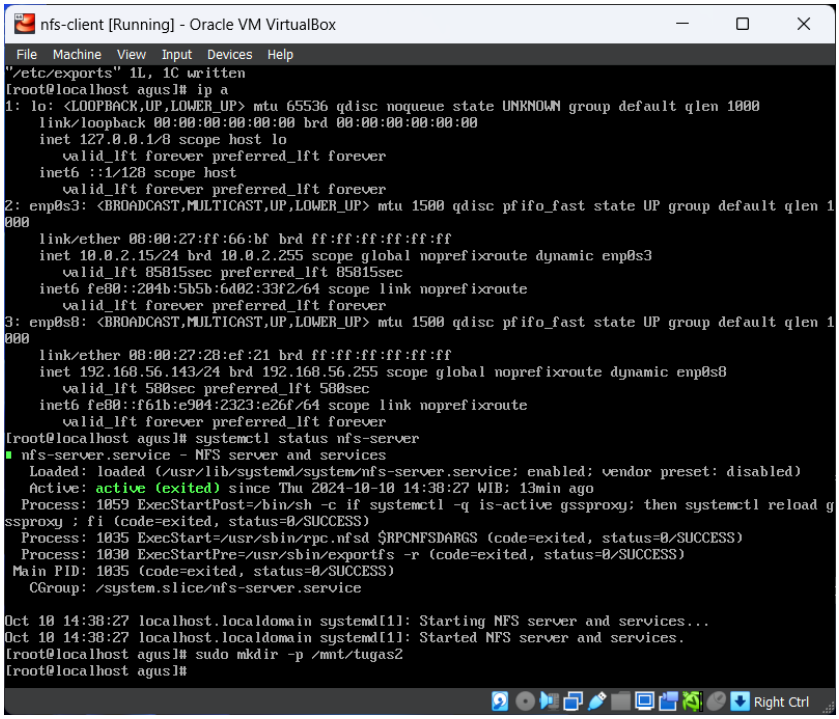
3. Kemudian kita nyalakan dan memulai nfs-server dan mengekspor nya dan juga mengecek status pada nfs-server.



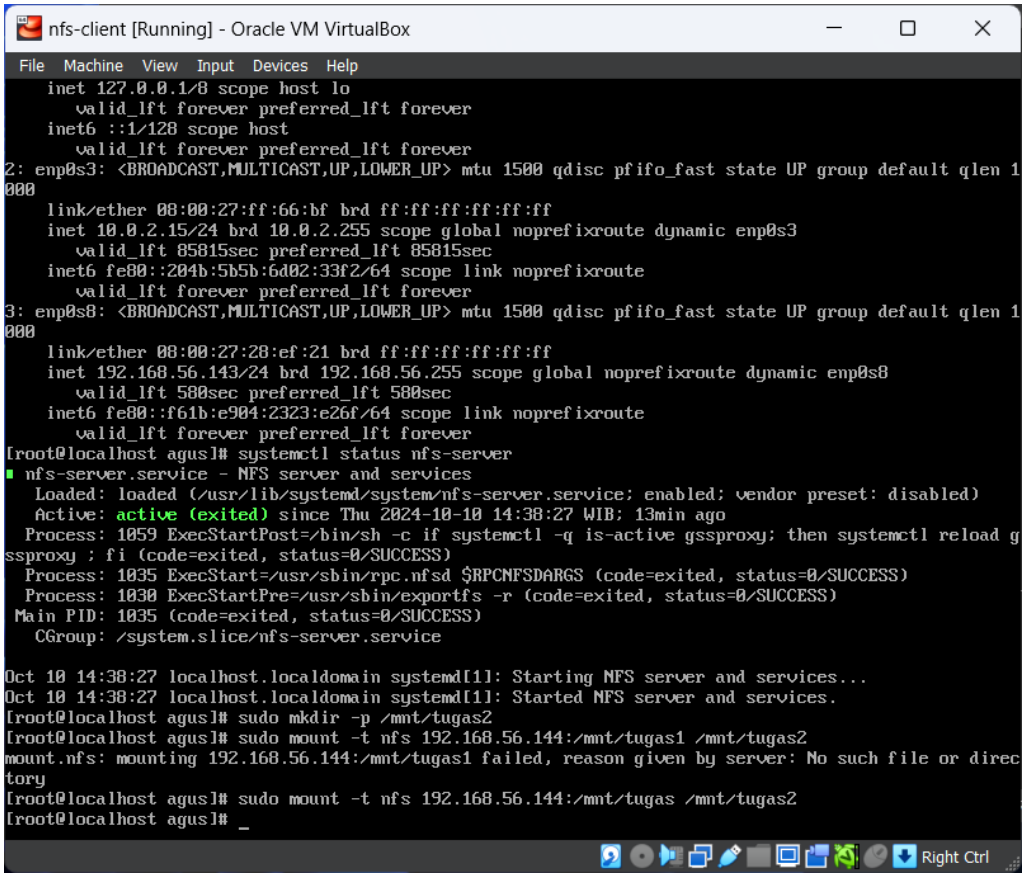
4. Kemudian, kita cek export directory pada NFS-Server untuk memastikan.



5. Kemudian kita buat direktori baru pada NFS-Client.



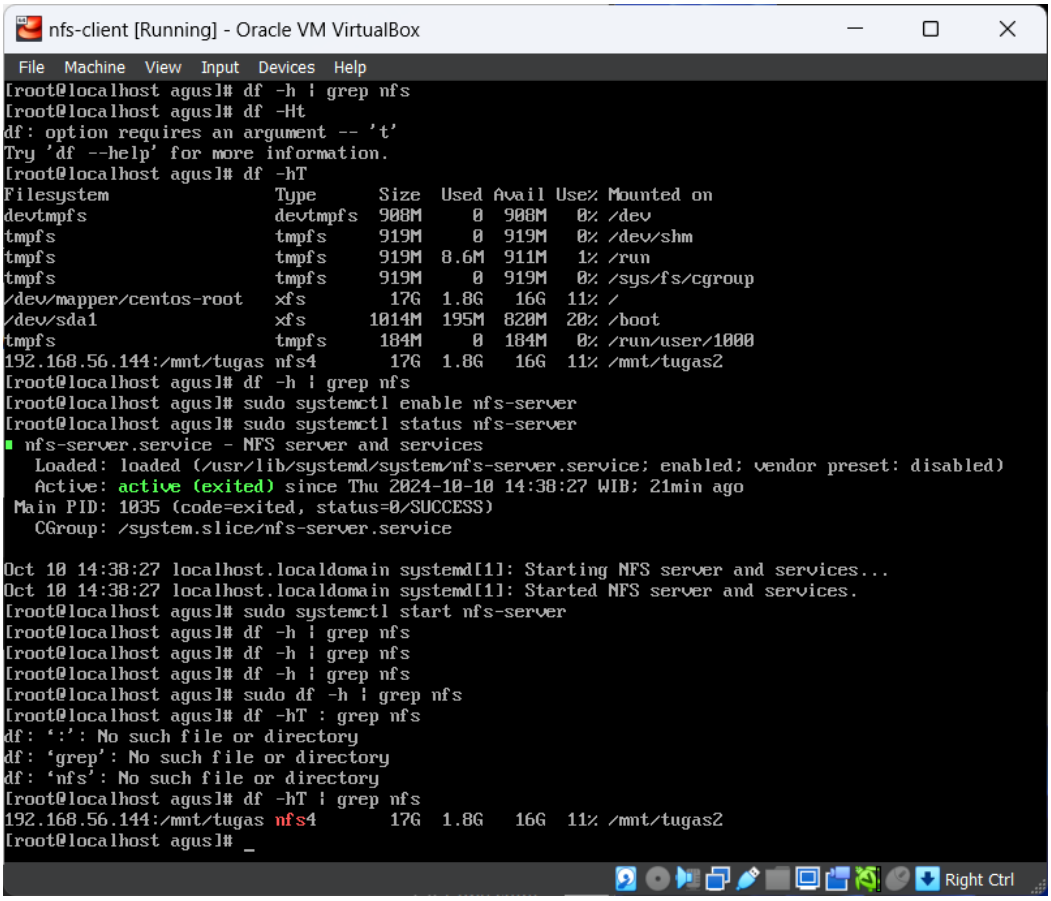
6. Kemudian, kita lakukan mount di NFS-Client



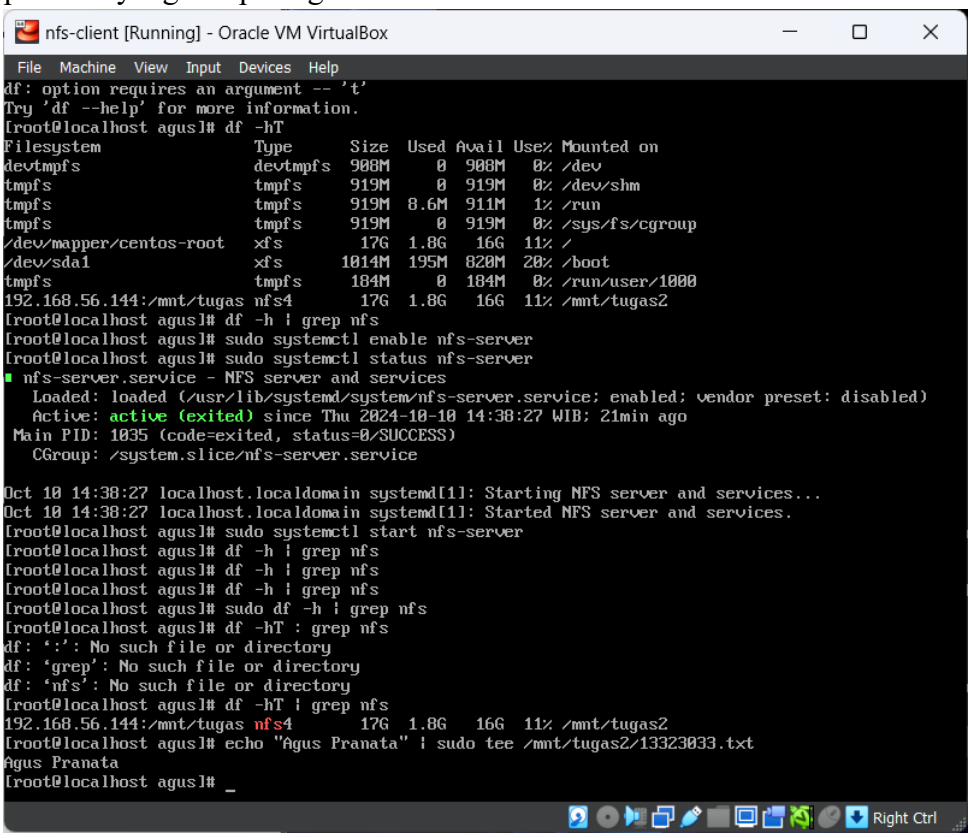
Keterangan:

- /mnt/tugas : Direktori pada server
- /mnt/tugas2 : Direktori pada client
- 192.168.56.144 : IP Server

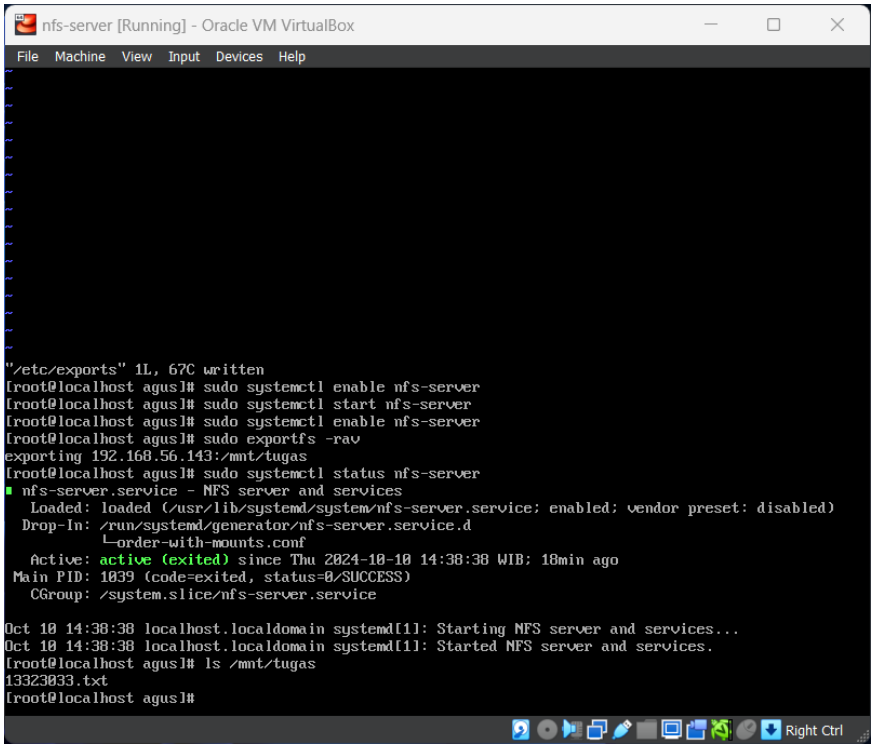
7. Kemudian verifikasi hasil mount pada di NFS-Client



8. Kemudian, kita akan membuat dan mengisi file tersebut dengan menggunakan perintah yang ada pada gambar ini.



9. Kemudian kita cek di NFS-Server apakah file tersebut ada.



10. Kita cek isi dalam file tersebut yang ada di NFS-Client pada NFS-Server.