BAB 4
Hasil dan Pembahasan

4.1 Implementasi Sistem

Pada tahap ini, dibangun sistem *digital evidence storage* dengan metode *clustering storage* menggunakan *software defined storage*. Metode *clustering storage* adalah metode pengelompokan dua atau lebih *storage* yang berperilaku sebagai satu sistem yang bekerja sama untuk meningkatkan kinerja, kapasitas serta reliabilitas sistem. Keuntungan yang diharapkan dari penggunaan metode *clustering* adalah tingkat redundansi dan ketersediaan yang tinggi serta kemudahan penambahan kapasitas, mengingat kebutuhan akan tempat penyimpanan barang bukti digital yang harus dapat diakses dari lokasi yang jauh, memiliki tingkat availabilitas yang tinggi, serta kemudahan penambahan kapasitas.

Setiap sistem dalam sebuah *cluster* disebut node. Setiap node adalah sistem nyata dan lengkap dengan semua sumber daya yang dibutuhkan seperti prosesor, memori, perangkat I/O, dan sub sistem penyimpanan yang bekerja di bawah kendali salinan sistem operasinya sendiri.

Clustering storage salah satunya dapat diwujudkan dengan penggunan *software defined storage*. *Software defined storage* (SDS) adalah istilah untuk *software* penyimpanan data komputer yang memungkinkan pengelolaan penyimpanan data yang pembuatan dan pengelolaannya terlepas dari keterikatan terhadap perangkat keras yang mendasarinya. *Software defined storage* biasanya mencakup bentuk virtualisasi *storage* untuk memisahkan *storage hardware* dari *software* yang mengelolanya. Berbeda dengan teknologi penyimpanan tradisional seperti SAN dan NAS. SDS menawarkan hal yang lebih baik dari segi aspek biaya, kelincahan (*agility*), skalabilitas, reliabilitas, dan multi-tenancy. Dalam hal skalabilitas, penyimpanan tradisional hanya dapat diskalakan hingga maksimal 4 – 8 *controller*, sedangkan SDS dapat menskalakan hingga ratusan ribu node - yang artinya secara teoritis dapat diskalakan tanpa batas (Paul Chinnaraju et al., 2018).

Software defined storage yang digunakan dalam penelitian ini adalah CephFS. CephFS merupakan salah satu turunan Ceph yang bekerja sebagai filesystem. CephFS dirancang untuk digunakan pada perangkat *server*, dan tidak dimaksudkan untuk dipasang pada desktop *user*. Beberapa keuntungan penggunaan CephFS seperti yang dikutip dari website Ceph ("Ceph Ceph storage - Ceph," n.d.) adalah :

- Memberikan keamanan data yang lebih kuat.
- Menyediakan penyimpanan *filesystem* yang hampir tidak terbatas.
- Aplikasi yang menggunakan *filesystem* dapat menggunakan CephFS. Tidak diperlukan integrasi atau penyesuaian.
- Ceph secara otomatis menyeimbangkan *filesystem* untuk memberikan kinerja maksimum.
- Ceph secara unik mengirimkan objek, blok, dan penyimpanan file dalam satu sistem terpadu

Pada tahap instalasi sistem ini juga dilakukan instalasi dan konfigurasi *software* pendukung seperti OpenVPN dan ownCloud. OwnCloud digunakan dalam penelitian ini untuk memudahkan manajemen layanan dan sinkronisasi file barang bukti digital antara file hasil akuisisi dan Ceph *client*. OwnCloud dapat mengatur layanan *cloud storage* mulai dari *user*, *group user*, *limited* akses, *size quota user*, fitur sinkronisasi otomatis, melakukan *transfer* data menggunakan enkripsi SSL dan melakukan verifikasi *checksum* terhadap file yang ter-*upload* atau *download*. OwnCloud Desktop Sync digunakan untuk melakukan sinkronisasi file barang bukti digital.

Konfigurasi OpenVPN server digunakan agar sistem *storage* dapat diakses menggunakan jaringan internet. Dalam penelitian ini dibangun 2 jalur koneksi untuk melakukan akses ke dalam sistem yaitu melalui jaringan kabel dan jaringan nirkabel (wireless maupun GSM). Selain agar dapat diakses menggunakan jaringan internet, konfigurasi OpenVPN juga digunakan sebagai upaya pengamanan transmisi data. Open VPN server di install pada *virtual machine*.

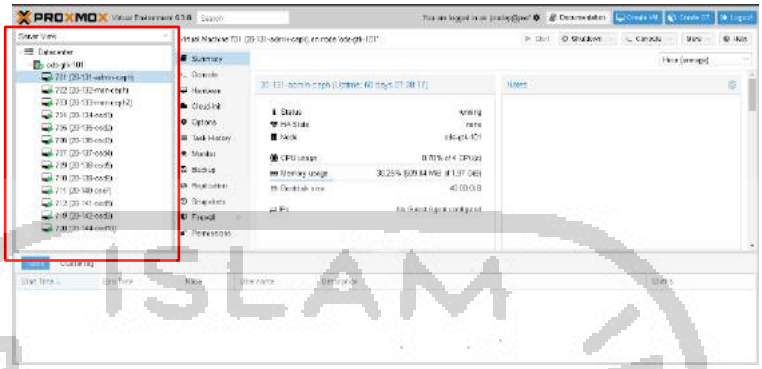
4.1.1 Persiapan Infrastruktur Server

Tahap persiapan infrastruktur server merupakan tahap *virtualisasi server* fisik. Dalam penelitian ini dibuat beberapa *Virtual Machine* (VM) untuk melakukan simulasi sistem. Ada 13 *virtual machine* yang dibuat dengan beberapa fungsi yaitu sebagai Server Ceph Admin, Server Ceph Monitor, Server Ceph OSD dan Server Ceph Client serta 1 *virtual machine* yang berfungsi sebagai OpenVPN Server.

- Membuat *Virtual Machine*

  Virtualisasi server fisik dilakukan menggunakan sistem operasi Proxmox *Virtual Environment* (Proxmox VE). Proxmox VE merupakan sistem operasi *Open Source* yang berfungsi untuk virtualisasi. Setelah sistem operasi Proxmox VE dikonfigurasi

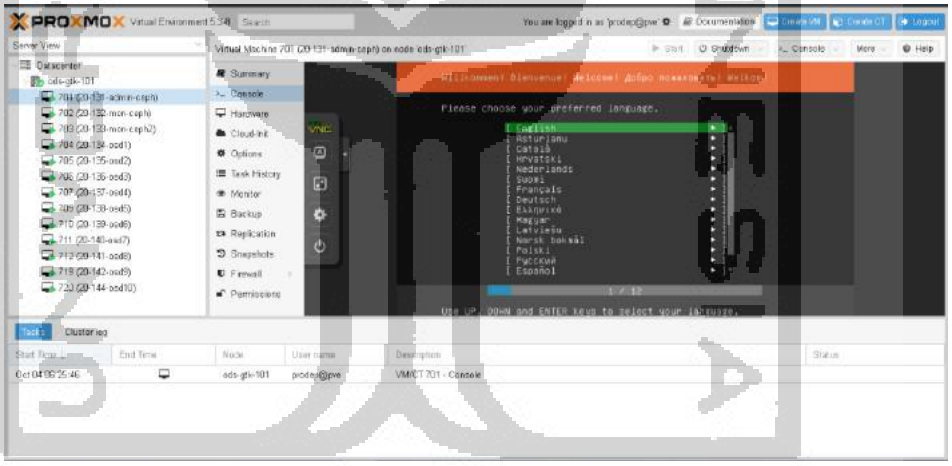
dengan baik, langkah selanjutnya adalah membuat 14 *virtual machine* sesuai dengan rancangan sebelumnya.



Gambar 4.1 *Virtual machine* dalam penelitian ini

- Instalasi sistem operasi semua node Ceph

  Operating system yang digunakan dalam sistem Ceph *Cluster Storage* adalah Ubuntu 18.04 LTS dan mengunakan aplikasi Ceph Mimic Stable Version 13.2.5.



Gambar 4.2 Instalasi sistem operasi pada semua node Ceph

4.1.2 Konfigurasi *Ceph Storage Cluster* dan Komponen Pendukung

Proses instalasi dan konfigurasi pada semua node dapat dijalankan dari node ceph-admin menggunakan *tools* ceph-deploy. Ceph-deploy adalah *tools* yang memungkinkan penyebaran Ceph cluster dengan mudah dan cepat tanpa melibatkan konfigurasi manual yang rumit dan terperinci. Ceph-deploy bekerja dengan menggunakan SSH untuk melakukan konfigurasi ke node lain. Ceph-deploy digunakan untuk menginstal paket Ceph pada host jarak jauh, membuat cluster, menambahkan monitor, mengumpulkan atau

menghapus kunci, menambahkan OSD dan server metadata, mengkonfigurasi host admin, dan melakukan penghapusan cluster.

- Konfigurasi host pada node ceph-admin

  Sebelum melakuan instalasi Ceph dengan ceph-deploy kita melakukan konfigurasi host pada node ceph-admin dengan mendaftarkan node ceph pada directory `/etc/hosts` di ceph-admin. File hosts file komputer adalah file sistem operasi yang memetakan nama host ke alamat IP.

```
ceph-admin@admin-ceph:~$ cat /etc/hosts
127.0.0.1       localhost.localdomain   localhost
::1             localhost6.localdomain6 localhost6

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
192.168.20.131 admin-ceph
192.168.20.132 mon-ceph1
192.168.20.133 mon-ceph2
192.168.20.134 osd1
192.168.20.135 osd2
192.168.20.136 osd3
192.168.20.137 osd4
192.168.20.138 osd5
192.168.20.139 osd6
192.168.20.140 osd7
192.168.20.141 osd8
192.168.20.142 osd9
192.168.20.143 osd10

192.168.20.147 ceph-client
ceph-admin@admin-ceph:~$
```

Gambar 4.3 Konfigurasi file *hosts* pada ceph-admin

Untuk menginstal paket ke semua *node* dengan menggunakan ceph-deploy, diperlukan *user* yang mempunyai hak *sudo* tanpa kata sandi. Selanjutnya diperlukan akses SSH tanpa kata sandi dari ceph admin ke semua node yang dilakukan dengan menyalin SSH *key* ke semua node.

Tabel 4.1 Daftar IP Address *Virtual Machine*

No	Jenis	Hostname	IP
1	Ceph Admin	ceph-admin	192.168.20.131/24
2	Ceph Monitor	ceph-mon1	192.168.20.132/24
3	Ceph Monitor	ceph-mon2	192.168.20.133/24
4	OSD	node-osd1	192.168.20.134/24
5	OSD	node-osd2	192.168.20.135/24
6	OSD	node-osd3	192.168.20.136/24
7	OSD	node-osd4	192.168.20.137/24
8	OSD	node-osd5	192.168.20.138/24
9	OSD	node-osd6	192.168.20.139/24
10	OSD	node-osd7	192.168.20.140/24
11	OSD	node-osd8	192.168.20.141/24
12	OSD	node-osd9	192.168.20.142/24
13	OSD	node-osd10	192.168.20.143/24

- Instalasi Ceph cluster

  Dalam proses instalasi Ceph cluster langkah pertama adalah melakukan *create cluster* dengan melakukan inisialisasi Ceph Monitor Daemon (ceph-mon) pada node Ceph Monitor. Server ini berfungsi sebagai node yang menyimpan informasi konfigurasi *cluster storage*.

```
$ ceph-deploy new ceph-mon1 ceph-mon2
```

Gambar 4.4 Inisialisasi node ceph-monitor dengan ceph-deploy

```
ceph-admin@ceph-admin:~/ceph-deploy$ ceph-deploy new mon-ceph1 mon-ceph2
[ceph_deploy.conf][DEBUG ] found configuration file at: /home/ceph-admin/.cephdeploy.conf
[ceph_deploy.cli][INFO  ] Invoked (2.0.1): /usr/bin/ceph-deploy new mon-ceph1
[ceph_deploy.cli][INFO  ] ceph-deploy options:
[ceph_deploy.cli][INFO  ]  username                      : None
[ceph_deploy.cli][INFO  ]  verbose                       : False
[ceph_deploy.cli][INFO  ]  overwrite_conf                : False
[ceph_deploy.cli][INFO  ]  quiet                         : False
[ceph_deploy.cli][INFO  ]  cd_conf                       : <ceph_deploy.conf.cephdeploy.Conf instance at 0x7f4c0720a950>
[ceph_deploy.cli][INFO  ]  cluster                       : ceph
[ceph_deploy.cli][INFO  ]  ssh_copykey                   : True
[ceph_deploy.cli][INFO  ]  mon                           : ['mon-ceph1']
[ceph_deploy.cli][INFO  ]  func                          : <function new at 0x7f4c07456d70>
[ceph_deploy.cli][INFO  ]  public_network                : None
[ceph_deploy.cli][INFO  ]  ceph_conf                     : None
[ceph_deploy.cli][INFO  ]  cluster_network               : None
[ceph_deploy.cli][INFO  ]  default_release               : False
[ceph_deploy.cli][INFO  ]  fsid                          : None
[ceph_deploy.new][DEBUG ] Creating new cluster named ceph
[ceph_deploy.new][INFO  ] making sure passwordless SSH succeeds
[mon-ceph1][DEBUG ] connected to host: ceph-admin
[mon-ceph1][INFO  ] Running command: ssh -CT -o BatchMode=yes mon-ceph1
[mon-ceph1][DEBUG ] connection detected need for sudo
[mon-ceph1][DEBUG ] connected to host: mon-ceph1
[mon-ceph1][DEBUG ] detect platform information from remote host
[mon-ceph1][DEBUG ] detect machine type
[mon-ceph1][DEBUG ] find the location of an executable
[mon-ceph1][INFO  ] Running command: sudo /bin/ip link show
[mon-ceph1][INFO  ] Running command: sudo /bin/ip addr show
[mon-ceph1][DEBUG ] IP addresses found: [u'192.168.20.132']
[ceph_deploy.new][DEBUG ] Resolving host mon-ceph1
[ceph_deploy.new][DEBUG ] Monitor mon-ceph1 at 192.168.20.132
[ceph_deploy.new][DEBUG ] Monitor initial members are ['mon-ceph1']
[ceph_deploy.new][DEBUG ] Monitor addrs are ['192.168.20.132']
[ceph_deploy.new][DEBUG ] Creating a random mon key...
[ceph_deploy.new][DEBUG ] Writing monitor keyring to ceph.mon.keyring...
[ceph_deploy.new][DEBUG ] Writing initial config to ceph.conf...
```

Gambar 4.5 Proses inisialisasi *node ceph monitor* dengan *ceph-deploy*

Setelah prosess inisialisasi ceph monitor selesai, kemudian dilanjutkan dengan instalasi paket ceph ke semua node ceph cluster. Instalasi paket ceph ke semua node menggunakan ceph-deploy.

```
ceph-admin@ceph-admin:~/ceph-deploy$ ceph-deploy install ceph-mon1 ceph-mon2 osd1 osd2 osd3 osd4 osd5 osd6 osd7 osd8
```

Gambar 4.6 Instalasi paket *Ceph*

```
.......................
[osd8][DEBUG ] Setting up ceph-base (13.2.2-1bionic) ...
[osd8][DEBUG ] Setting up python-pecan (1.2.1-2) ...
[osd8][DEBUG ] update-alternatives: using /usr/bin/python2-pecan to provide /usr/bin/pecan (pecan) in auto mode
[osd8][DEBUG ] update-alternatives: using /usr/bin/python2-gunicorn_pecan to provide /usr/bin/gunicorn_pecan (gunicorn_pecan) in auto mode
[osd8][DEBUG ] Setting up ceph-osd (13.2.2-1bionic) ...
[osd8][DEBUG ] chown: cannot access '/var/lib/ceph/osd/*/block*': No such file or directory
[osd8][DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-osd.target -> /lib/systemd/system/ceph-osd.target.
[osd8][DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-osd.target -> /lib/systemd/system/ceph-osd.target.
[osd8][DEBUG ] Setting up ceph-mds (13.2.2-1bionic) ...
[osd8][DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mds.target -> /lib/systemd/system/ceph-mds.target.
[osd8][DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-mds.target -> /lib/systemd/system/ceph-mds.target.
[osd8][DEBUG ] Setting up ceph-mon (13.2.2-1bionic) ...
[osd8][DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mon.target -> /lib/systemd/system/ceph-mon.target.
[osd8][DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-mon.target -> /lib/systemd/system/ceph-mon.target.
[osd8][DEBUG ] Setting up ceph-mgr (13.2.2-1bionic) ...
[osd8][DEBUG ] Created symlink /etc/systemd/system/multi-user.target.wants/ceph-mgr.target -> /lib/systemd/system/ceph-mgr.target.
[osd8][DEBUG ] Created symlink /etc/systemd/system/ceph.target.wants/ceph-mgr.target -> /lib/systemd/system/ceph-mgr.target.
[osd8][DEBUG ] Setting up ceph (13.2.2-1bionic) ...
[osd8][DEBUG ] Processing triggers for libc-bin (2.27-3ubuntu1) ...
[osd8][DEBUG ] Processing triggers for ureadahead (0.100.0-20) ...
[osd8][DEBUG ] Processing triggers for systemd (237-3ubuntu10) ...
[osd8][INFO  ] Running command: sudo ceph --version
[osd8][DEBUG ] ceph version 13.2.5 (02899bfda814146b021136e9d8e80eba494e1126) mimic (stable)
```

Gambar 4.7 *Output* proses instalasi

- Menambahkan OSD ke dalam sistem

  Setelah proses instalasi paket selesai dilanjutkan dengan proses *deploy the initial monitor(s)* dan proses pengumpulan *keyring.* Selanjutnya, dilakukan proses peyalinan konfigurasi file dan *keyring* ke node-admin dan semua node OSD.

  Setiap node OSD terdapat 3 keping harddisk, 1 hardisk sebagai *operating system* dan 2 hardisk akan di tautkan kedalam Ceph *storage cluster*, 2 keping hardisk tersebut adalah `/dev/vdb` dan `/dev/vdc`.

```
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd1
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd1
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd2
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd2
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd3
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd3
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd4
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd4
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd5
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd5
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd6
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd6
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd7
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd7
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdb osd8
ceph-admin@ceph-admin:~/ceph-deploy$ceph-deploy osd create --data /dev/vdc osd8
```

Gambar 4.8 Menambahkan OSD ke dalam sistem

- Melakukan pengecekan status *storage cluster*

  Setelah semua harddisk ditambahkan dilakukan pengecekan status cluster, pengecekan status cluster dilakukan dari node ceph monitor. Gambar 4.9 menunjukkan status *storage cluster* yang telah dibangun. Hasilnya system storage dinyatakan HEALTH dengan kapasitas harddisk 320GB.

```
ceph-admin@mon-ceph2:~$ sudo ceph status
  cluster:
    id:     abc92a0e-1eac-4aa9-9633-af43a3ba5fad
    health: HEALTH_OK

  services:
    mon: 2 daemons, quorum mon-ceph1,mon-ceph2
    mgr: osd1(active)
    mds: cephfs-1/1/1 up  {0=mon-ceph2=up:active}, 1 up:standby
    osd: 16 osds: 16 up, 16 in

  data:
    pools:   0 pools, 0 pgs
    objects: 0 objects, 0 B
    usage:   0 GiB used, 320 GiB / 320 GiB avail
    pgs:

ceph-admin@mon-ceph2:~$
```

Gambar 4.9 Status Ceph *storage cluster*

- Implementasi CephFS

  Implementasi CephFS memerlukan satu *metadata server* (MDS) yang akan dikonfigurasi pada salah satu dari node *cluster*. *Cluster* MDS dapat memperluas dan dapat menyeimbangkan filesystem secara dinamis untuk mendistribusikan data secara merata di antara host *cluster*. *Cluster* MDS memastikan sistem memiliki kinerja tinggi dan mencegah beban berat pada host tertentu di dalam *cluster*.

```
ceph-admin@admin-ceph:~$ ceph-deploy mds create mon-ceph1 mon-ceph2
```

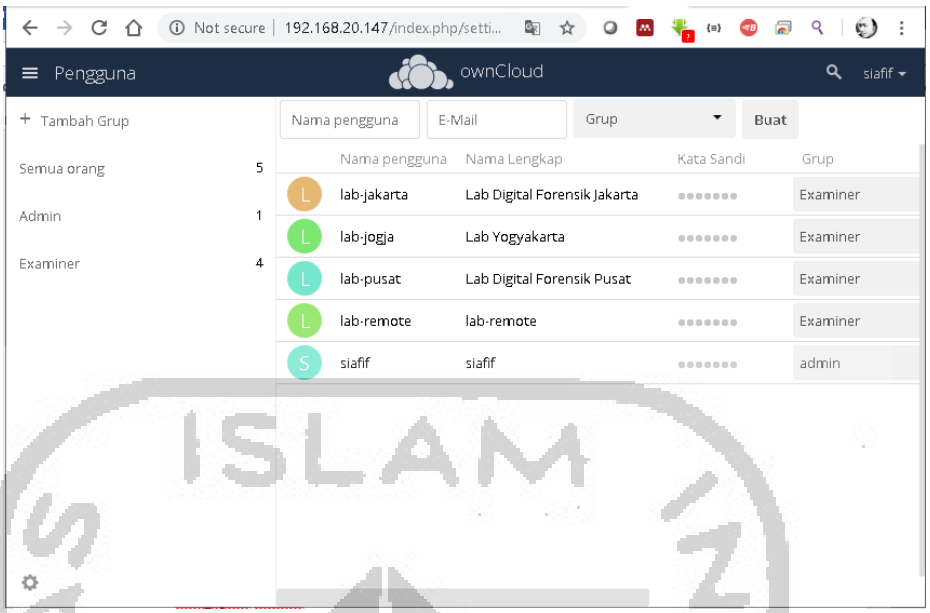
  Selain Ceph *metadata server* (MDS), Ceph juga membutuhkan minimal satu Ceph Monitor untuk dijalankan. Ceph Monitor juga bisa dijalankan pada lebih dari satu node supaya bisa menjadi *backup*.

```
ceph-admin@ceph-client:~$ df -h
Filesystem                      Size  Used Avail Use% Mounted on
udev                            1.9G     0  1.9G   0% /dev
tmpfs                           392M   26M  367M   7% /run
/dev/mapper/ceph--client--vg-root  38G  3.9G   32G  11% /
tmpfs                           2.0G     0  2.0G   0% /dev/shm
tmpfs                           5.0M     0  5.0M   0% /run/lock
tmpfs                           2.0G     0  2.0G   0% /sys/fs/cgroup
/dev/vda1                       720M  308M  376M  45% /boot
tmpfs                           392M     0  392M   0% /run/user/1000
ceph-fuse                       334G   16G  319G   5% /mnt/cephfs-data
tmpfs                           392M     0  392M   0% /run/user/1001
ceph-admin@ceph-client:~$
```

  Gambar 4.10 Mounting Ceph *cluster storage*

- Instalasi ownCloud

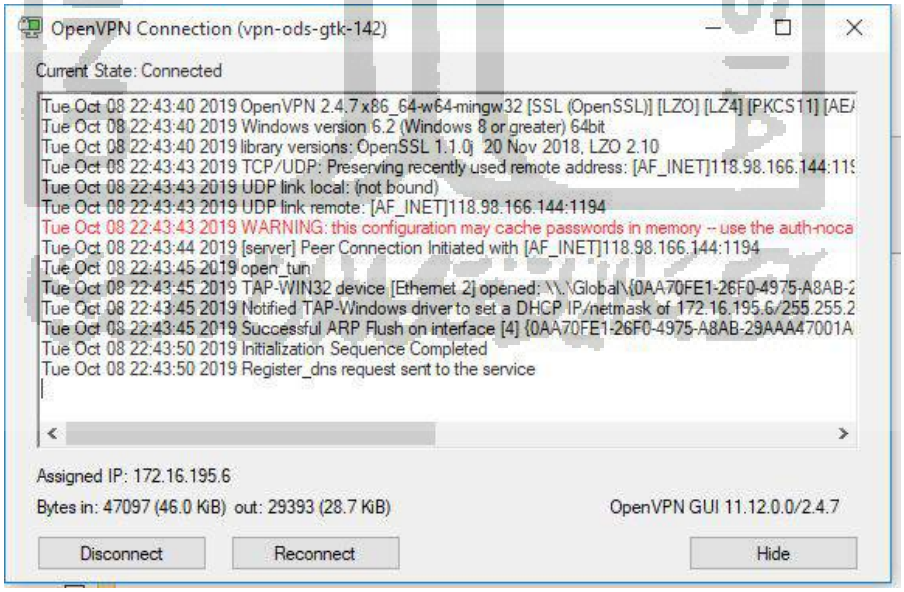
  Setelah Ceph *storage cluster* berhasil di-mount di Ceph client, selanjutnya untuk memudahkan manajemen layanan dan sinkronisasi file barang bukti digital antara file hasil akuisisi dan Ceph *client* dalam penelitian ini menggunakan ownCloud server. OwnCloud server di-install pada *directory* hasil mounting Ceph *cluster storage*.



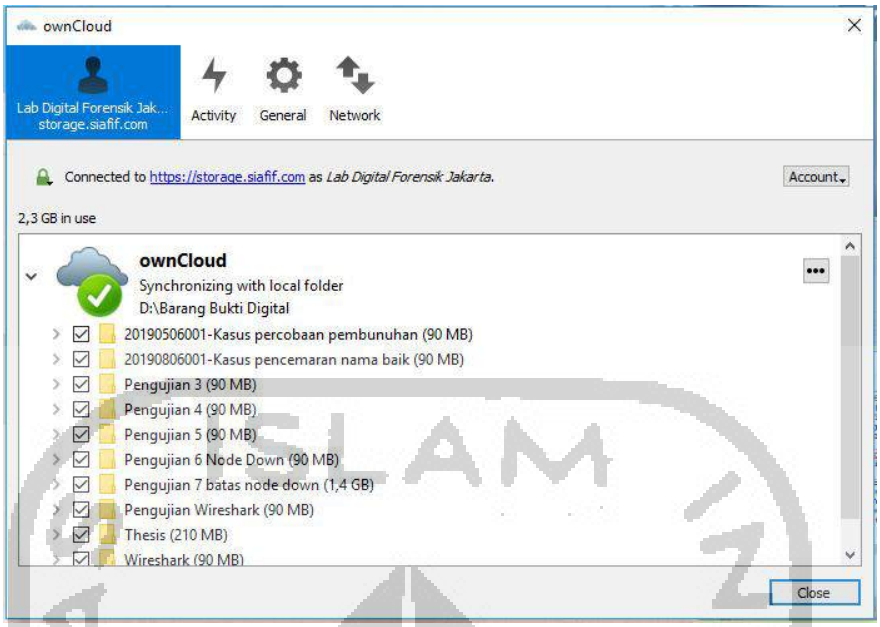
Gambar 4.11 Management user ownCloud

4.1.3 Uji coba Ceph *storage cluster*

Uji coba Ceph *storage cluster* dengan melakukan instalasi ownCloud Desktop Client dan Open VPN *Client* di *Examiner Workstation*. Setelah itu di lakukan uji coba sinkronisasi file dari *client* ke *storage cluster*. Hasil dari uji coba ini menunjukkan bahwa sistem siap digunakan.



Gambar 4.12 User melakukan koneksi OpenVPN ke server



Gambar 4.13 OwnCloud Client sebagai tools sinkronisasi barang bukti digital

4.2 Pengujian Sistem

Pada tahap ini dilakukan 4 skenario pengujian sistem yang diharapkan dapat memberikan hasil yang baik dari kriteria skalabilitas, serta dapat memenuhi kriteria keamanan informasi yang terdiri dari aspek *confidentiality*, *integrity*, dan *availability*. Hal ini berkaitan untuk menjaga agar bukti digital yang disimpan dalam *storage* dapat diterima dan digunakan sebagai barang bukti di persidangan.

4.2.1 Uji Skalabilitas

Pengujian dilakukan dengan menambahkan 2 node ke dalam sistem yang semula memiliki 8 node, sehingga total ada 10 node yang berjalan dalam sistem. Gambar 4.14 dan 4.15 menunjukkan keadaan sistem sebelum dan sesudah penambahan node baru. Total kapasitas disk sebelum dilakukan penambahan node adalah 320Gb, dan ssetelah penambahan 2 node baru dengan 4 OSD, total kapasitas disk menjadi 400Gb. Gambar 4.16 memperlihatkan detail keadaan masing-masing OSD setelah penambahan node baru. Ada total 20 OSD dengan masing-masing OSD memiliki kapasitas 20GB, dan telah digunakan sebesar 4,0 – 4,9 GB dengan total disk yang digunakan sebesar 90GB. Ketika OSD baru masuk ke dalam sistem, secara otomatis Ceph akan mereplikasi data yang telah ada sekaligus membaginya ke dalam OSD baru tersebut. Proses penambahan node ini berjalan dengan lancar tanpa ada gangguan sistem.

```
prodep@mon-ceph1:~$ sudo ceph df
[sudo] password for prodep:
GLOBAL:
    SIZE        AVAIL       RAW USED     %RAW USED
    320 GiB     251 GiB       69 GiB         21.44
```

Gambar 4.14 Keadaan sistem sebelum penambahan node baru

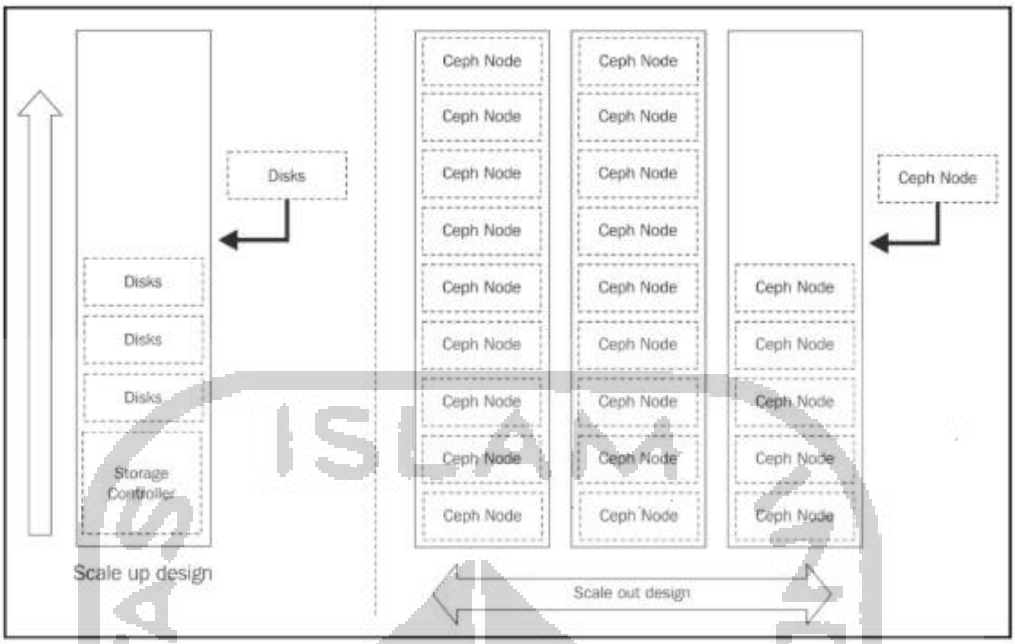
```
[sudo] password for prodep:
GLOBAL:
    SIZE        AVAIL       RAW USED     %RAW USED
    400 GiB     313 GiB       69 GiB         21.44
```

Gambar 4.15 Keadaan sistem setelah penambahan node baru

```
ceph-admin@mon-ceph1:~$ sudo ceph osd df
ID CLASS WEIGHT  REWEIGHT SIZE    USE     AVAIL   %USE  VAR  PGS
 0   hdd 0.01949  1.00000  20 GiB 4.6 GiB  15 GiB 23.23 1.04  14
 1   hdd 0.01949  1.00000  20 GiB 4.9 GiB  15 GiB 24.53 1.09  18
 4   hdd 0.01949  1.00000  20 GiB 4.7 GiB  15 GiB 23.73 1.06  16
 5   hdd 0.01949  1.00000  20 GiB 5.1 GiB  15 GiB 25.41 1.13  22
 6   hdd 0.01949  1.00000  20 GiB 4.8 GiB  15 GiB 23.87 1.06  23
 7   hdd 0.01949  1.00000  20 GiB 4.3 GiB  16 GiB 21.73 0.97  12
 8   hdd 0.01949  1.00000  20 GiB 4.7 GiB  15 GiB 23.33 1.04  18
 9   hdd 0.01949  1.00000  20 GiB 4.5 GiB  15 GiB 22.64 1.01  13
10   hdd 0.01949  1.00000  20 GiB 4.4 GiB  16 GiB 22.18 0.99  12
11   hdd 0.01949  1.00000  20 GiB 4.0 GiB  16 GiB 19.81 0.88  13
16   hdd 0.01949  1.00000  20 GiB 4.4 GiB  16 GiB 21.81 0.97  12
17   hdd 0.01949  1.00000  20 GiB 4.5 GiB  16 GiB 22.37 1.00  19
18   hdd 0.01949  1.00000  20 GiB 4.7 GiB  15 GiB 23.30 1.04  17
19   hdd 0.01949  1.00000  20 GiB 4.1 GiB  16 GiB 20.29 0.91   9
 2   hdd 0.01949  1.00000  20 GiB 4.6 GiB  15 GiB 22.79 1.02  27
 3   hdd 0.01949  1.00000  20 GiB 4.4 GiB  16 GiB 21.91 0.98  31
12   hdd 0.01949  1.00000  20 GiB 4.2 GiB  16 GiB 21.15 0.94  26
13   hdd 0.01949  1.00000  20 GiB 4.3 GiB  16 GiB 21.55 0.96  23
14   hdd 0.01949  1.00000  20 GiB 4.0 GiB  16 GiB 20.23 0.90  28
15   hdd 0.01949  1.00000  20 GiB 4.5 GiB  16 GiB 22.45 1.00  31
                   TOTAL 400 GiB  90 GiB 310 GiB 22.42
```

Gambar 4.16 Detail setiap OSD setelah penambahan node

Proses peningkatan *storage* yang mudah dapat terjadi dikarenakan Ceph dikembangkan dengan desain *scale-out*. Berbeda dengan penyimpanan tradisional yang dikembangkan dengan desain *scale-up* yang terbatas pada penambahan *storage*, desain *scale-out* fokus pada penambahan seluruh node baru, termasuk disk, CPU, dan memori ke dalam sistem yang ada. Dalam jenis desain ini, sistem tidak akan berakhir dengan penyimpanan terbatas, melainkan, akan diuntungkan oleh kinerja dan kekokohan sistem saat penambahan node baru sehingga saat ada penambahan node baru, kinerja Ceph juga mengalami peningkatan (Umrao, Singh, & Hackett, 2017). Sedangkan metode desain *scale-up* dalam hal penambahan *storage* melibatkan penambahan sumber daya disk ke sistem *controller* yang ada, yang dapat menjadi penghambat kinerja, kapasitas, dan pengelolaan ketika mencapai tingkat tertentu. Sehingga mungkin perlu kompromi agar kinerja, keandalan, dan availabilitas tetap terjaga saat penambahan sistem baru.



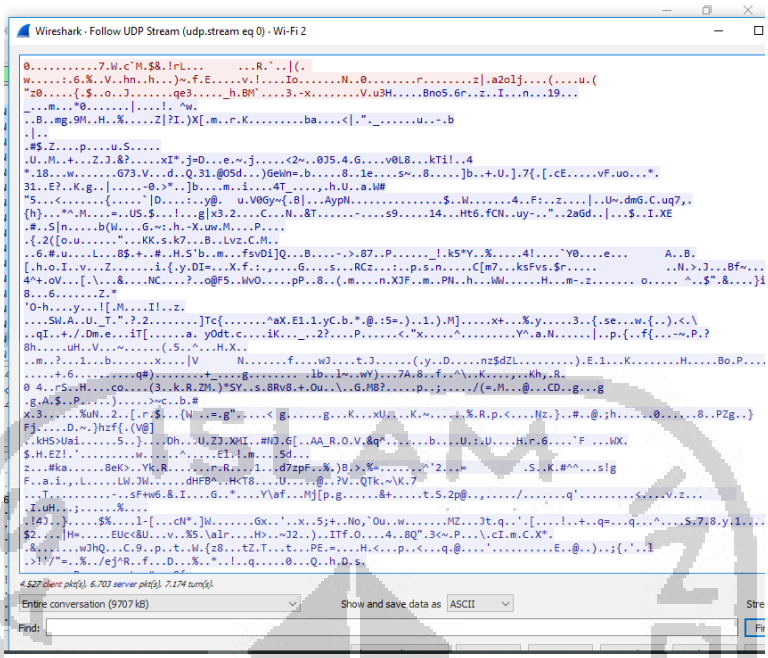
Gambar 4.17 Perbandingan antara desain *scale-up* dan desain *scale-out* dalam infrastruktur sistem *storage* (Umrao et al., 2017).

Tidak seperti solusi penyimpanan tradisional lainnya, Ceph sangat mudah beradaptasi dan tidak memerlukan *drive* penyimpanan atau *host* yang harus identik dalam hal jenis atau ukuran. *Cluster* yang dimulai dengan drive 4TB dapat dengan mudah dikembangkan dengan menambahkan drive 6TB atau 8TB, baik untuk menggantikan drive yang lebih kecil, maupun untuk menambahkan storage server secara bertahap. Cluster Ceph tunggal juga dapat berisi campuran jenis, ukuran, dan kecepatan drive penyimpanan, baik untuk beban kerja yang berbeda atau untuk menerapkan *tiering* guna memanfaatkan drive lambat yang hemat biaya untuk penyimpanan massal dan drive yang lebih cepat untuk membaca atau menyimpan *cache* (D'Atri et al., 2017)

Meskipun satu cluster yang terdiri dari set server dan drive yang seragam akan lebih mudah dalam hal pengelolaan dan lain sebagainya, namun beberapa model server, generasi, dan bahkan merek dalam suatu cluster juga dapat bekerja dengan baik dalam Ceph cluster.

4.2.2 Uji Keamanan Transkripsi Data

Pengujian dilakukan dengan menggunakan metode *sniffing* (penyadapan) pada setiap paket yang melewati jaringan, kemudian melihat dan menganalisa hasilnya. *Sniffing* adalah aktivitas menyadap paket data yang sedang berjalan pada *traffic* sebuah jaringan. Sedangkan alat yang digunakan untuk melakukan pengujian ini adalah Wireshark Network Protocol Analyzer.



Gambar 4.18 Hasil *sniffing* file menggunakan Wireshark menunjukkan bahwa koneksi OpenVPN telah terenkripsi.

Hasil pengujian yang ditunjukkan dalam Gambar 4.18 memperlihatkan data *random* atau acak dari hasil *sniffing*. Hal ini menunjukkan bahwa koneksi VPN dan penggunaan protokol HTTPS dalam penelitian ini membuat transmisi data terenkripsi, yang berarti bahwa kriteria *confidentiality* dalam sistem yang dibangun telah terpenuhi. Kriteria *confidentiality* dapat dipenuhi oleh teknologi OpenVPN karena OpenVPN menggunakan *library* OpenSSL untuk melakukan enkripsi. OpenSSL mendukung beberapa algoritma kriptografi yang berbeda seperti 3DES, AES, RC5, Blowfish. Seperti IPSec, VPN menerapkan algoritma AES yang sangat aman dengan kunci 256 bit. Dalam dunia enkripsi, AES merupakan teknologi terbaru dan dipertimbangkan sebagai 'standar emas' karena teknologi ini belum diketahui kelemahannya.

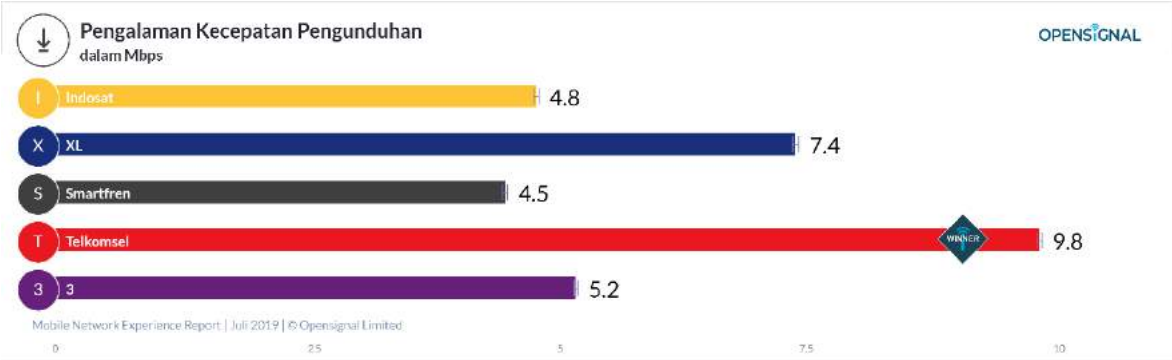
Selain penggunaan teknologi OpenVPN, penggunaan protokol HTTPS juga mendukung kriteria *confidentiality*. HTTPS dilengkapi dengan sistem keamanan (*security*) berupa SSL (*Secure Socket Layer*) yang berperan sebagai lapisan pelindung dari protokol jaringan suatu website dengan melakukan enkripsi data. Secara digital, cara kerja SSL adalah dengan mengunci *cryptographic key* ke informasi yang hendak diidentifikasi. Data pun akan terenkripsi dengan baik selama proses *transfer* sehingga pihak ketiga tidak bisa masuk dan mencuri informasi yang sensitif. Tak hanya *private key* dan *public key*, SSL juga memiliki *session key* untuk setiap *secure session* yang unik. Selama koneksi awal, *public key* dan *private key* akan digunakan untuk membuat *session key*, yang kemudian mengenkripsi dan

mendekripsi data yang sedang ditransfer. *Session key* ini akan tetap valid untuk waktu yang terbatas dan hanya digunakan di session tertentu.

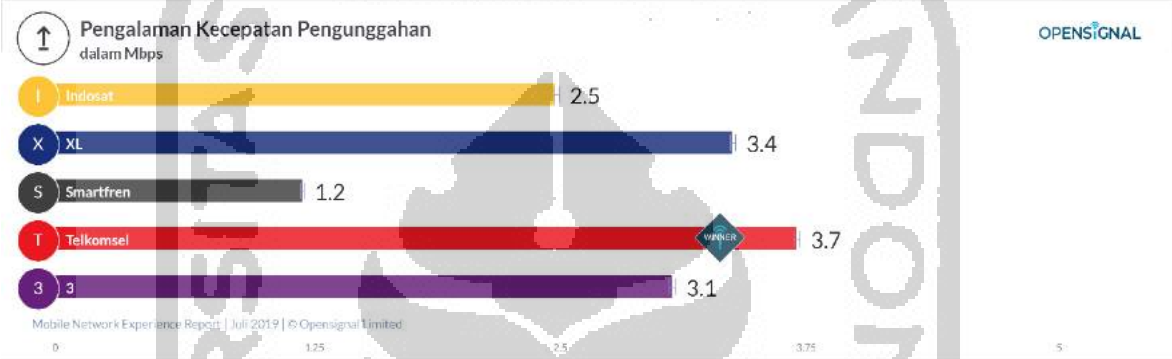
4.2.3 Uji Sinkronisasi File

Skenario pengujian berikutnya adalah pengujian sinkronisasi file dengan menggunakan jaringan 4G beberapa operator seluler dalam beberapa kualitas koneksi dengan parameter *ping time*, *jitter*, besaran *bandwidth*, serta *packet loss*. Pemilihan operator seluler dilakukan berdasarkan Laporan Pengalaman Jaringan Seluler di Indonesia Juli 2019 yang dirilis oleh opensignal.com dengan kriteria Pengalaman Kecepatan Pengunduhan dan Pengalaman Kecepatan Pengunggahan. Opensignal.com telah meneliti 5 operator seluler di Indonesia yaitu, Indosat, XL, Smartfren, Telkomsel dan 3. Dari kriteria Pengalaman Kecepatan Pengunduhan dan Pengalaman Kecepatan Pengunggahan dapat dillakukan pemeringkatan dengan urutan sebagai berikut: Telkomsel, XL, 3, Indosat serta Smartfren. Dalam Penelitian ini, akan dilakukan pengujian dengan jaringan 4G Telkomsel untuk memberikan gambaran kondisi jaringan reliable, serta Indosat dan Smartfren untuk memberikan gambaran kondisi jaringan yang unreliable.

Gambar 4.19 Ikhtisar perbandingan *mobile network experience* di Indonesia periode Juli 2019 (Khatri, 2019).



Gambar 4.20 Grafik Pengalaman Kecepatan Pengunduhan Operator Seluler di Indonesia Periode Juli 2019 (Khatri, 2019)



Gambar 4.21 Grafik Pengalaman Kecepatan Pengunggahan Operator Seluler di Indonesia Periode Juli 2019 (Khatri, 2019)

File yang digunakan dalam penelitian ini adalah file rekamanan CCTV dengan extensi .mp4 yang berukuran 91MB. Pengujian menggunakan *tool* ownCloud Desktop Client sehingga file bukti digital secara otomatis disinkronkan ke *server*. Metode penelitian yang dilakukan adalah mengubah kualitas koneksi dengan membatasi besaran *bandwidth*. Data diambil dengan membandingkan hasil *hashing* MD5 file sumber dan file yang telah terupload di *server*.

Tabel 4.2 Hasil Pengujian Sinkronisasi File Menggunakan Jaringan 4G Telkomsel

No	Jenis Koneksi	ping time (ms)			Jitter (ms)	Bandwitdh		Packet Loss (%)	MD5 Sumber	MD5 Server	Ket
		Min	Max	Ave		Transmit (Mbps)	Received (Mbps)				
1	4G	55	169	74	5.752	5.45	5.29	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
2	4G	45	122	64	12.427	3.04	2.88	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
3	4G	48	91	63	8.101	1.89	1.74	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
4	4G	47	100	59	21.246	1.05	0.905	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
5	4G	39	138	65	15.582	0.629	0.434	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok

Hasil pengujian yang ditunjukkan dalam Tabel 4.2 memperlihatkan bahwa dengan kondisi jaringan 4G Telkomsel yang buruk dimana kecepatan *received* 0.434 Mbps dan kecepatan *transmit* 0.629 Mbps, hasil pengecekan MD5 file sumber dan file yang telah terupload di server adalah sama. Dengan kata lain, transfer *file* dengan *bandwidth* yang kecil atau kualitas jaringan yang kurang baik tidak mempengaruhi *integrity* file.

Tabel 4.3 Hasil Pengujian Sinkronisasi Menggunakan Jaringan 4G Indosat

No	Jenis Koneksi	ping time (ms)			Jitter (ms)	Bandwitdh		Packet Loss (%)	MD5 Sumber	MD5 Server	Ket
		Min	Max	Ave		Transmit (Mbps)	Received (Mbps)				
1	4G	428	536	497	37.08	1.26	1.26	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
2	4G	722	762	742	46.17	0.829	0.822	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
3	4G	60	141	95	132.6	1.05	0.928	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
4	4G	50	54	51	21.78	0.839	0.668	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
5	4G	61	84	71	39.49	0.629	0.433	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok

Hasil pengujian yang ditunjukkan dalam Tabel 4.3 memperlihatkan bahwa dengan kondisi jaringan 4G Indosat yang buruk dimana kecepatan *received* 0.433 Mbps dan kecepatan *transmit* 0.629 Mbps, hasil pengecekan MD5 file sumber dan file yang telah terupload di server adalah sama. Pada pengujian ini, kita juga dapat melihat bahwa saat ping time buruk (742 ms), hasil pengecekan MD5 file sumber dan file yang telah terupload di server adalah sama. Dengan kata lain, transfer *file* dengan kualitas jaringan yang kurang baik (ping time yang buruk) tidak mempengaruhi *integrity* file.

Tabel 4.4 Hasil Pengujian Sinkronisasi Menggunakan Jaringan 4G Smartfren

No	Jenis Koneksi	ping time (ms)			Jitter (ms)	Bandwitdh		Packet Loss (%)	MD5 Sumber	MD5 Server	Ket
		Min	Max	Ave		Transmit (Mbps)	Received (Mbps)				
1	4G	57	163	94	10.38	2.41	2.20	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
2	4G	57	84	69	11.61	1.47	1.32	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
3	4G	43	86	66	33.82	1.26	1.06	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
4	4G	44	77	59	31.92	0.839	0.647	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok
5	4G	50	80	64	22.53	0.524	0.342	0	44847d4eb5a0d152cf65f533fe23ad74	44847d4eb5a0d152cf65f533fe23ad74	ok

Hasil pengujian yang ditunjukkan dalam Tabel 4.4 memperlihatkan bahwa dengan kondisi jaringan 4G Smartfren yang buruk dimana kecepatan *received* 0.524 Mbps dan kecepatan *transmit* 0.342 Mbps, hasil pengecekan MD5 file sumber dan file yang telah terupload di server adalah sama. Dengan kata lain, transfer *file* dengan kualitas jaringan yang kurang baik (ping time yang buruk) tidak mempengaruhi *integrity* file.

Skenario pengujian ini kami lakukan untuk memberikan gambaran bahwa saat melakukan sinkronisasi hasil akuisisi bukti digital dengan kondisi jaringan yang kurang baik pun masih dapat menjamin integritas barang bukti yang diakuisisi.

Integritas data dapat terjamin dikarenakan kemampuan ownCloud untuk melakukan sinkronisasi file antara berbagai sistem operasi dan penerapan fitur checksum yang memeriksa integritas file saat mengunggah dan mengunduh dengan mengkomputasi checksum setelah transfer file selesai.

Selain penggunaan ownCloud, kriteria *integrity* juga dipenuhi oleh penggunaan OpenVPN. OpenVPN yang digunakan dalam penelitian ini menggunakan port TCP, sehingga integritas data akan selalu terjaga karena setiap ada *packet* yang hilang, akan digunakan data *correction* untuk memperbaikinya.

Sampai dalam tahap ini, sistem *storage* telah menunjukkan kemampuannya dalam menjaga integritas data. Namun, hal ini saja tentu masih belum cukup agar barang bukti digital yang tersimpan dapat diterima pengadilan. Perlu adanya *chain of custody* yang berfungsi untuk menjamin integritas barang bukti. *Chain of custody* merupakan dokumen yang dapat menjamin bahwa informasi yang dipresentasikan adalah lengkap dan tidak mengalami perubahan dari sejak pertama kali ditemukan sampai akhir digunakan dalam proses persidangan. Karena keterbatasan waktu dan sumber daya yang dimiliki, penulis belum melakukan penelitian lebih lanjut mengenai aspek *chain of custody* dalam sistem ini. Diharapkan akan ada penelitian lanjutan mengenai *chain of custody* terutama dalam hal

pengembangan *chain of custody* yang *reliable* dan aman berdasarkan sistem *storage* yang dibangun ini.

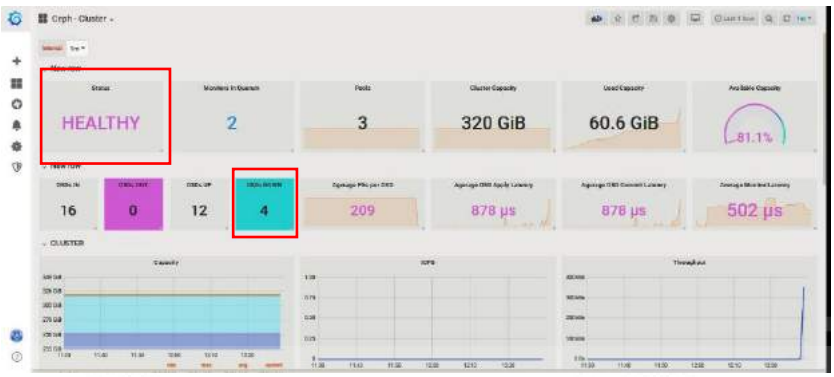
4.2.4 Uji Node Terputus

Skenario pengujian berikutnya yang dijalankan adalah dengan membuat kondisi salah satu node Ceph terputus ketika dilakukan transfer file. File yang ditransfer dalam hal ini berupa file rekaman CCTV dengan ekstensi .mp4 yang berukuran 91MB. Pengujian dilakukan dengan membuat 2 keadaan node terputus yaitu ketika 2 node terputus dan 4 node terputus lalu membandingkan hasilnya.

Ketika 2 node terputus, transfer file dari *client* ke *server* tetap berjalan dengan normal. Seperti yang terlihat dalam Gambar 4.20 keadaan sistem saat 2 node terputus masih dikategorikan sehat sehingga proses transfer masih berjalan dengan normal. 2 node terputus ini oleh sistem masih dikategorikan sebagai kegagalan minor. Setelah 2 node yang putus tersebut tersambung kembali, sistem akan melakukan sinkronisasi *file*. Pengecekan *hashing* MD5 terhadap file sumber dan file *server* memberikan hasil yang sama.

Gambar 4.21 memperlihatkan keadaan sistem saat 4 node terputus, transfer *file* dari *client* ke *server* berhenti. Ceph akan melakukan *blocking* ke dalam sistem *storage* sehingga file tertahan sampai hanya 3 node yang terputus. Saat node yang terputus hanya berjumlah 3, transfer *file* dapat berjalan kembali dengan normal. Dan saat sistem tersambung kembali sepenuhnya dan dinyatakan sehat, sistem akan melakukan sinkronisasi. Pengecekan hashing MD5 terhadap file sumber dan file *server* memberikan hasil tidak ada perubahan.

Proses perbaikan sistem berlangsung secara otomatis. Hal ini dapat terjadi dikarenakan fitur RADOS dari Ceph yang memungkinkan Ceph untuk dapat *self-healed* dan *self-manage*. RADOS juga membuat sistem menjadi fleksibel, dapat diandalkan dan mudah untuk di *manage*. Jika terjadi bencana, Ceph dapat memberikan reliabilitas terhadap *multiple failure*. Ceph mendeteksi dan memperbaiki kegagalan di setiap zona kegagalan sebagai disk, node, jaringan, rak, baris pusat data, pusat data, dan bahkan berbagai lokasi geografis. Ceph mencoba untuk mengelola situasi secara otomatis dan mengatasinya sebisa mungkin tanpa gangguan data (D'Atri et al., 2017).



Gambar 4.22 Pengujian dengan kondisi 2 *node* atau 4 OSD *down*, status *cluster healthy*

```
2019-08-08 05:58:10.809862 mon.mon-ceph1 [WRN] Health check update: 8 osds down (OSD_DOWN)
2019-08-08 05:58:10.809908 mon.mon-ceph1 [WRN] Health check failed: 1 host (2 osds) down (OSD_HOST_DOWN)
2019-08-08 05:58:31.672775 mon.mon-ceph1 [WRN] Health check failed: 1 MDSs report slow metadata IOs (MDS_SLOW_METADATA_IO)
2019-08-08 05:58:43.700918 mon.mon-ceph1 [WRN] Health check failed: 1 MDSs report slow requests (MDS_SLOW_REQUEST)
2019-08-08 05:58:40.140040 mds.mon-ceph2 [WRN] 1 slow requests, 1 included below; oldest blocked for > 34.559337 secs
2019-08-08 05:58:45.140151 mds.mon-ceph2 [WRN] 1 slow requests, 0 included below; oldest blocked for > 39.559473 secs
2019-08-08 05:58:50.140291 mds.mon-ceph2 [WRN] 1 slow requests, 0 included below; oldest blocked for > 44.559601 secs
```

Gambar 4.23 Pengujian dengan kondisi 4 *node* atau 8 OSD *down*, Ceph melakukan *blocking*

Dari pengujian sederhana ini, dapat disimpulkan bahwa sistem *storage* yang dibangun telah memenuhi aspek *availability* yaitu siap diakses kapanpun oleh *user / application/* sistem yang membutuhkannya. Kegagalan *minor* akan ditangani langsung oleh sistem, sehingga proses perbaikan tidak berlangsung lama dan sistem dapat segera digunakan. Hasil pengecekan *hashing* MD5 terhadap file sumber dan *file server* yang sama walaupun terjadi putus koneksi dalam pengujian ini juga menegaskan bahwa sistem *storage* ini telah memenuhi aspek *integrity*.

4.3 Penilaian Sistem oleh Ahli

Setelah melalui 4 skenario pengujian untuk menguji fungsi sistem *storage* dan mendapatkan hasil yang memuaskan dari aspek skalabilitas, *confidentiality*, *integrity* dan *availability*, kami membawa sistem *storage* ini untuk mendapatkan penilaian oleh ahli forensika digital. Penilaian dilakukan dengan kuisioner yang telah disiapkan sebelumnya. Responden atau ahli dalam hal ini adalah Fietyata Yudha, S.Kom., M.Kom., peneliti Pusat Studi Forensika (Pusfid) Digital Fakultas Teknologi Industri (FTI) Universitas Islam Indonesia (UII) yang juga berperan sebagai ahli professional dalam kasus kejahatan siber.

Berdasarkan penilaian yang dilakukan oleh ahli tentang keberfungsian sistem *storage* dalam membantu penyidik dalam hal penyimpanan bukti digital, sistem *storage* ini

mendapatkan nilai 4 dari skala 5 yang berarti sistem *storage* yang dibangun layak untuk membantu penyidik dalam hal penyimpanan bukti digital. Selain itu, sistem *storage* ini juga dinilai memudahkan penyidik dalam hal penyimpanan bukti digital.

Informasi lain yang kami gali dari kuisioner adalah keunggulan sistem *storage* ini yaitu metode *cluster storage* yang memungkinkan menambah media penyimpanan secara fleksibel, atau dengan kata lain memiliki kemampuan skalabilitas yang baik, dimana hal tersebut adalah salah satu yang ingin kami capai pada saat pembangunan sistem *storage* ini.

Bagaimanapun, sistem ini belum memenuhi standar *storage* penyimpanan barang bukti digital yang seharusnya, sehingga tidak bisa digunakan oleh pihak berwenang untuk diadopsi sebagai *storage* penyimpanan barang bukti digital. Hal ini sesuai dengan apa yang telah kami sampaikan sebelumnya pada halaman 54, bahwa sistem *storage* yang dibangun ini belum meng-*cover* kebutuhan perekaman *chain of custody*. Sebagai tambahan, sistem *storage* ini juga belum mendukung hierarki penyimpanan barang bukti berbasis kasus. Diharapkan akan ada penelitian lanjutan mengenai hal tersebut agar menambah wawasan keilmuan mengenai *storage* penyimpanan bukti digital.