



VERSI 1.0

APRIL, 2023

PRAKTIKUM SISTEM OPERASI

MODUL 4 - MANAGING BASIC STORAGE

TIM PENYUSUN: - MAHAR FAIQURAHMAN, S.KOM., M.T.
- SYAHRUL PANGESTU
- MUH. RIDHA AGAM

PRESENTED BY: LAB. INFORMATIKA

UNIVERSITAS MUHAMMADIYAH MALANG

SISTEM OPERASI

CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu menjelaskan Mounting dan Unmounting File Systems
2. Mahasiswa mampu menambahkan Partisi, Sistem File, dan Persistent Mount
3. Mahasiswa mampu membuat tautan antara file di Linux
4. Mahasiswa mampu Mengelola Swap Space

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu memberikan contoh langsung mengenai Mounting dan Unmounting File system
2. Mahasiswa mampu mempraktekkan langsung penambahan partisi, sistem file, dan Persistent Mount
3. Mahasiswa mampu mempraktekkan cara mengelola swap space

KEBUTUHAN HARDWARE & SOFTWARE

- Laptop/PC
- Virtual Machine (VMware, Virtual Box, VPC bila ekonomi diatas rata-rata :")
- Sistem Operasi CentOS, download [image](#) OVA (Wajib)

MATERI POKOK**Mounting and Unmounting File Systems****Mounting File Systems Manually**

File system yang berada pada perangkat penyimpanan yang dapat dilepas perlu dipasang agar dapat diakses. Dalam hal ini, Perintah mount memungkinkan pengguna root untuk memasang file system secara manual. Argumen pertama dari perintah mount menentukan file system yang akan dipasang. Argumen kedua menentukan direktori yang akan digunakan sebagai titik pasang dalam hierarki file system.

Ada dua cara umum untuk menentukan file system pada partisi disk ke perintah mount:

- Dengan nama file perangkat di `/dev` yang berisi file system.
- Dengan UUID yang ditulis ke file system, yaitu pengenal unik yang universal (*universally-unique identifier*).

Mounting cukup sederhana untuk dilakukan. Anda perlu mengidentifikasi perangkat yang ingin dipasang, memastikan titik pasang ada, dan memasang perangkat pada titik pasang tersebut.

Identifying the Block Device

Sebuah perangkat penyimpanan yang dapat dicolokkan saat beroperasi, entah itu merupakan hard disk drive (HDD) ataupun perangkat solid-state (SSD) yang terdapat pada wadah server, atau bahkan perangkat penyimpanan USB, kemungkinan akan terhubung ke port yang berbeda setiap kali perangkat tersebut dipasang pada suatu sistem.

Anda dapat menggunakan perintah **lsblk** untuk mencantumkan rincian mengenai perangkat blok yang spesifik atau semua perangkat yang ada pada sistem Anda. Perintah ini akan memberikan informasi yang akurat dan dapat diandalkan mengenai perangkat blok yang ingin Anda identifikasi, sehingga Anda dapat melanjutkan proses pemasangan file system dengan tepat.

```
[root@host ~]# lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
vda	253:0	0	12G	0	disk	
├─vda1	253:1	0	1G	0	part	/boot
├─vda2	253:2	0	1G	0	part	[SWAP]
└─vda3	253:3	0	11G	0	part	/
vdb	253:16	0	64G	0	disk	
└─vdb1	253:17	0	64G	0	part	

(Gambar 1. Penggunaan command **lsblk**)

Pada gambar diatas, terlihat bahwa partisi yang ingin ditambahkan terlihat jelas pada **/dev/vdb1**

Contoh berikut memasang sistem berkas di partisi **/dev/vdb1** pada direktori **/mnt/data**

```
[root@host ~]# mount /dev/vdb1 /mnt/data
```

(Gambar 2. Mounting)

Untuk memasang sistem berkas, direktori tujuan harus sudah ada. Direktori **/mnt** ada secara bawaan dan ditujukan untuk digunakan sebagai titik pasang sementara. Anda bisa menggunakan direktori **/mnt**, atau lebih baik lagi membuat subdirektori dari **/mnt** untuk digunakan sebagai titik pasang sementara, kecuali jika anda punya tempat lain untuk memasangnya di lokasi tertentu dalam *file hierarchy*.

Mounting by File-system UUID

Salah satu penanda yang stabil yang terkait dengan sistem berkas adalah UUID, sebuah angka heksadesimal yang panjang dapat berfungsi sebagai penanda unik. UUID ini ada di dalam sistem berkas dan tetap sama selama sistem berkasnya dan tidak dibuat ulang.

Perintah **lsblk -fp** mencantumkan jalur lengkap perangkat, bersama dengan UUID dan titik kait, serta jenis sistem berkas di partisi. Jika sistem berkas tidak terpasang, titik kait akan kosong.

```
[root@host ~]# lsblk -fp
NAME                FSTYPE LABEL UUID                                MOUNTPOINT
/dev/vda
├─/dev/vda1 xfs                23ea8803-a396-494a-8e95-1538a53b821c /boot
├─/dev/vda2 swap                cdf61ded-534c-4bd6-b458-cab18b1a72ea [SWAP]
└─/dev/vda3 xfs                44330f15-2f9d-4745-ae2e-20844f22762d /
/dev/vdb
└─/dev/vdb1 xfs                46f543fd-78c9-4526-a857-244811be2d88
```

(Gambar 3. Penggunaan **lsblk -fp**)

Melakukan Mounting berdasarkan **UUID**:

```
[root@host ~]# mount UUID="46f543fd-78c9-4526-a857-244811be2d88" /mnt/data
```

(Gambar 4. Mounting menggunakan UUID)

Automatic Mounting of Removable Storage Devices

Adapun, Jika Anda sudah masuk dan menggunakan lingkungan desktop grafis, maka media penyimpanan yang bisa dilepas akan otomatis dipasang saat dimasukkan.

Perangkat penyimpanan yang bisa dilepas dipasang di:

/run/media/USERNAME/LABEL

Dimana **USERNAME** adalah **nama pengguna** yang masuk ke lingkungan grafis dan **LABEL** adalah **penanda**, seringkali nama yang diberikan ke sistem berkas saat dibuat jika ada.

Unmounting File Systems

Prosedur shutdown dan reboot akan melepaskan semua sistem berkas secara otomatis. Sebagai bagian dari proses ini, data sistem berkas yang tersimpan di memori akan dikirim ke perangkat penyimpanan sehingga memastikan bahwa sistem berkas tidak mengalami kerusakan data.

Cara untuk unmount:

```
[root@host ~]# umount /mnt/data
```

(Gambar 5. Unmounting)

Unmounting tidak mungkin dilakukan jika sistem berkas yang dipasang sedang digunakan. Agar perintah `umount` berhasil, semua proses harus berhenti mengakses data di bawah titik pasang.

Berikut ini adalah contoh `umount` yang gagal dikarenakan sistem file sedang digunakan (Shell pada `/mnt/data` merupakan direktori yang sedang digunakan / direktori aktif)

```
[root@host ~]# cd /mnt/data
[root@host data]# umount /mnt/data
umount: /mnt/data: target is busy.
```

(Gambar 6. Fail Unmounting)

Selain dari unmounting kita juga dapat menggunakan command **lsof** yang memberikan list semua file yang sedang dibuka dan biasanya digunakan untuk mengidentifikasi proses yang membuat gagal dari *unmounting* itu sendiri

```
[root@host data]# lsof /mnt/data
```

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
bash	1593	root	cwd	DIR	253,17	6	128	/mnt/data
lsof	2532	root	cwd	DIR	253,17	19	128	/mnt/data
lsof	2533	root	cwd	DIR	253,17	19	128	/mnt/data

(Gambar 7. Penggunaan **lsof** ke direktori `/mnt/data`)

Setelah proses data yang sedang membuat gagal proses *unmount* itu sendiri teridentifikasi, maka kita dapat melakukan aksi berikutnya yakni dapat **menunggu** hingga proses tersebut selesai atau bisa melakukan perintah **SIGTERM** atau **SIGKILL** yang akan memberikan proses signal pemberhentian.

```
[root@host data]# cd
[root@host ~]# umount /mnt/data
```

(Gambar 8. Unmount yang berhasil)

Adding Partitions, File Systems, and Persistent Mounts

Partitioning a Disk

Mempartisi disk memungkinkan administrator sistem membagi hard drive menjadi beberapa unit penyimpanan logis, yang disebut partisi. Dengan membagi disk menjadi beberapa partisi, administrator sistem dapat menggunakan partisi yang berbeda untuk menjalankan fungsi yang berbeda.

Contohnya, mempartisi disk diperlukan atau bermanfaat dalam situasi berikut:

- Membatasi ruang yang tersedia untuk aplikasi atau pengguna.
- Memisahkan file sistem operasi dan program dari file pengguna.
- Membuat area terpisah untuk pertukaran memori.
- Membatasi penggunaan ruang disk untuk meningkatkan kinerja alat diagnostik dan pencitraan cadangan.

MBR Partitioning Scheme

skema partisi Master Boot Record (MBR) sudah ada sejak 1982 dan telah mengatur cara disk dipartisi pada sistem yang menjalankan firmware BIOS. Skema ini mendukung maksimum empat partisi utama. Di sistem Linux, dengan menggunakan partisi yang diperluas dan logis, administrator dapat membuat maksimum 15 partisi. Karena data ukuran partisi disimpan sebagai nilai 32-bit, disk yang dipartisi dengan skema MBR memiliki ukuran disk dan partisi maksimum 2 TiB.



(Gambar 9. Skema partisi MBR pada /dev/vdb)

Karena ukuran disk fisik semakin besar, dan volume berbasis SAN bahkan lebih besar lagi, batasan ukuran disk dan partisi 2 TiB dari skema partisi MBR bukan lagi batasan teoritis, melainkan masalah dunia nyata yang semakin sering dihadapi administrator sistem di lingkungan produksi. Akibatnya, skema MBR lama sedang digantikan oleh Tabel Partisi GUID (GPT) yang baru untuk mempartisi disk.

GPT Partitioning Scheme

Untuk sistem yang menjalankan firmware *Unified Extensible Firmware Interface* (UEFI), GPT merupakan standar untuk menyusun tabel partisi pada hard disk fisik. GPT merupakan bagian dari standar UEFI dan mengatasi banyak keterbatasan yang diberlakukan oleh skema berbasis MBR lama.

GPT menyediakan maksimum 128 partisi. Berbeda dengan MBR, yang menggunakan 32 bit untuk menyimpan alamat blok logis dan informasi ukuran, GPT mengalokasikan 64 bit untuk alamat blok logis. Hal ini memungkinkan GPT untuk menampung partisi dan disk hingga delapan zebibytes (ZiB) atau delapan miliar tebibytes.

Selain mengatasi keterbatasan skema partisi MBR, GPT juga menawarkan beberapa fitur dan manfaat tambahan. GPT menggunakan pengidentifikasi unik global (GUID) untuk mengidentifikasi setiap disk dan partisi. Berbeda dengan MBR, yang memiliki satu titik kegagalan, GPT menawarkan redundansi informasi tabel partisinya. GPT utama berada di bagian awal disk, sementara salinan cadangan, GPT sekunder, berada di bagian akhir disk. GPT menggunakan checksum untuk mendeteksi kesalahan dan kerusakan pada header GPT dan tabel partisi.



(Gambar 10. Partisi GPT pada /dev/vdb)

Managing Partitions with Parted

Partition Editor adalah program yang memungkinkan administrator untuk melakukan perubahan pada partisi disk, seperti membuat partisi, menghapus partisi, dan mengubah tipe partisi. Untuk melakukan operasi ini, administrator dapat menggunakan editor partisi Parted untuk skema partisi MBR dan GPT.

Perintah **parted** menggunakan nama perangkat seluruh disk sebagai argumen pertama dan satu atau lebih subperintah. Contoh berikut menggunakan subperintah **print** untuk menampilkan tabel partisi pada disk **/dev/vda**.

```
[root@host ~]# parted /dev/vda print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system  Flags
  1      1049kB  10.7GB  10.7GB  primary  xfs          boot
  2      10.7GB  53.7GB  42.9GB  primary  xfs
```

(Gambar 11. Melakukan Partisi pada /dev/vda)

Jika Anda tidak memberikan subperintah, **parted** akan membuka sesi interaktif untuk mengeluarkan perintah.

Secara default, parted menampilkan semua ukuran dalam satuan pangkat 10 (KB, MB, GB). Anda dapat mengubah default tersebut dengan subperintah unit yang menerima parameter berikut:

Tipe	Penggunaan
s	Sektor
b	Byte
MiB, GiB, atau TiB	Pangkat 2
MB, GB, atau TB	Pangkat 10

Seperti yang ditunjukkan dalam contoh di atas, Anda juga dapat menentukan beberapa subperintah (di *here*, *unit* dan *print*) dalam satu baris yang sama.

Writing the Partition Table on a New Disk

Untuk mempartisi drive baru, Anda harus menulis label disk terlebih dahulu. Label disk menunjukkan skema partisi yang akan digunakan.

Sebagai pengguna root, gunakan perintah berikut untuk menulis label disk **MBR pada disk**:

```
[root@host ~]# parted /dev/vdb mklabel msdos
```

(Gambar 12. Melakukan konfigurasi untuk menjadi MBR pada disk)

Untuk menulis label disk GPT, gunakan perintah berikut:

```
[root@host ~]# parted /dev/vdb mklabel gpt
```

(Gambar 13. Menambahkan label GPT pada /dev/vdb)

Creating MBR Partitions

Membuat partisi disk MBR melibatkan beberapa langkah:

1. Tentukan perangkat disk tempat partisi akan dibuat.

Sebagai pengguna root, jalankan perintah `parted` dan tentukan nama perangkat disk sebagai argumen. Ini akan memulai perintah `parted` dalam mode interaktif dan menampilkan prompt perintah.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

(Gambar 14. Masuk pada partisi /dev/vdb)

2. Gunakan subperintah `mkpart` untuk membuat partisi utama atau partisi yang diperluas.

```
(parted) mkpart
Partition type? primary/extended? primary
```

(Gambar 15. Setting untuk partisi dengan command `mkpart`)

3. Tunjukkan jenis file system yang ingin Anda buat pada partisi, seperti `xfs` atau `ext4`. Ini tidak membuat file system pada partisi; ini hanya menandakan tipe partisi.

```
File system type? [ext2]? xfs
```

(Gambar 16. Menentukan tipe partisi)

Jika ingin melihat tipe partisi secara lengkap bisa dengan command dibawah:

```
[root@host ~]# parted /dev/vdb help mkpart
mkpart PART-TYPE [FS-TYPE] START END      make a partition

PART-TYPE is one of: primary, logical, extended
FS-TYPE is one of: btrfs, nilfs2, ext4, ext3, ext2, fat32, fat16, hfsx,
hfs+, hfs, jfs, swsusp, linux-swap(v1), linux-swap(v0), ntfs, reiserfs,
hp-ufs, sun-ufs, xfs, apfs2, apfs1, asfs, amufs5, amufs4, amufs3,
amufs2, amufs1, amufs0, amufs, affs7, affs6, affs5, affs4, affs3, affs2,
affs1, affs0, linux-swap, linux-swap(new), linux-swap(old)
```

(Gambar 17. Melihat FS-Type, Part-Type)

4. Tentukan sektor pada disk tempat partisi baru dimulai.

Start? 2048s

(Gambar 18. Membuat size awalan pada partisi baru)

Perhatikan akhiran s diatas, hal tersebut untuk memberikan nilai dalam sektor. Anda juga dapat menggunakan akhiran MiB, GiB, TiB, MB, GB, atau TB. Jika Anda tidak memberikan akhiran, MB adalah defaultnya. parted mungkin membulatkan nilai yang Anda berikan untuk memenuhi batasan disk.

Ketika parted dimulai, ia mengambil topologi disk dari perangkat. Misalnya, ukuran blok fisik disk biasanya merupakan parameter yang dikumpulkan parted. Dengan informasi tersebut, parted memastikan posisi awal yang Anda berikan sudah benar sehingga partisi sejajar dengan struktur disk. Penyelarasan partisi yang benar sangat penting untuk kinerja optimal. Jika posisi awal menghasilkan partisi yang tidak sejajar, parted akan menampilkan peringatan. Dengan kebanyakan disk, sektor awal yang merupakan kelipatan dari 2048 adalah asumsi yang aman.

5. Tentukan sektor disk di mana partisi baru harus berakhir.

End? 1000MB

(Gambar 19. Membuat akhiran size dari partisi baru)

Dengan *parted* kita tidak dapat secara langsung mengetahui size partisi tetapi, kita dapat menghitung dengan formula:

Size = End - Start

(Gambar 20. Formula menghitung size)

6. Keluar dari parted.

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

(Gambar 21. Keluar dari *parted* dengan **quit**)

7. Jalankan perintah `udevadm settle`. Perintah ini menunggu sistem untuk mendeteksi partisi baru dan membuat file perangkat yang terkait di bawah direktori `/dev`. Perintah ini hanya akan kembali saat selesai.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

(Gambar 22. Perintah untuk **deteksi partisi baru**)

Sebagai alternatif untuk mode interaktif, Anda juga dapat membuat partisi sebagai berikut:

```
[root@host ~]# parted /dev/vdb mkpart primary xfs 2048s 1000MB
```

(Gambar 23. Mode interaktif dalam pembuatan partisi)

Creating GPT Partitions

Hal ini sama seperti MBR, hanya saja disini kita membuat Skema GPT yang juga menggunakan perintah *parted* untuk membuat partisi baru:

1. Tentukan perangkat disk tempat partisi akan dibuat.

Sebagai pengguna **root**, jalankan perintah *parted* dengan perangkat disk sebagai satu-satunya argumen untuk memulai *parted* dalam mode interaktif dengan prompt perintah.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

(Gambar 24. Masuk menggunakan *parted* pada `/dev/vdb`)

2. Gunakan subperintah `mkpart` untuk mulai membuat partisi baru.

Dengan skema GPT, setiap partisi diberi nama.

```
(parted) mkpart
Partition name? []? usersdata
```

(Gambar 25. Memberi nama pada setiap partisi)

3. Tunjukkan jenis file system yang ingin Anda buat pada partisi, seperti xfs atau ext4. Ini tidak membuat file system pada partisi; ini hanya menandakan tipe partisi.

```
File system type? [ext2]? xfs
```

Catatan: Seperti pada sebelumnya tanda 's' merupakan nilai sektor

4. Tentukan sektor pada disk tempat partisi baru dimulai.

```
Start? 2048s
```

(Gambar 27. Membuat size awalan pada partisi baru)

5. Tentukan sektor disk di mana partisi baru harus berakhir.

```
End? 1000MB
```

(Gambar 28. Membuat akhiran size dari partisi baru)

6. Keluar dari parted.

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

(Gambar 29. Keluar dari *parted* dengan **quit**)

7. Jalankan perintah `udevadm settle`. Perintah ini menunggu sistem untuk mendeteksi partisi baru dan membuat file perangkat yang terkait di bawah direktori `/dev`. Perintah ini hanya akan kembali saat selesai.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

(Gambar 30. Perintah untuk **deteksi partisi baru**)

Sebagai alternatif untuk mode interaktif, Anda juga dapat membuat partisi sebagai berikut:

```
[root@host ~]# parted /dev/vdb mkpart usersdata xfs 2048s 1000MB
```

(Gambar 31. Mode interaktif dalam pembuatan partisi)

Catatan: Disini `parted` pada mode interaktif disesuaikan dengan nama partisi yang dibuat yakni `'usersdata'`.

Deleting Partitions

Langkah-langkah berikut berlaku untuk skema partisi MBR dan GPT:

1. Tentukan disk yang berisi partisi yang akan dihapus.

Sebagai pengguna root, jalankan perintah `parted` dengan perangkat disk sebagai satu-satunya argumen untuk memulai `parted` dalam mode interaktif dengan prompt perintah.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

(Gambar 32. Masuk menggunakan *parted* pada /dev/vdb)

2. Identifikasi nomor partisi dari partisi yang akan dihapus.

```
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name      Flags
  1      1049kB  1000MB  999MB   xfs          usersdata
```

(Gambar 33. *Print* untuk melihat nomor keberapa partisi yang ingin dihapus)

3. Hapus partisi.

```
(parted) rm 1
```

(Gambar 34. Remove partisi)

Subperintah `rm` langsung menghapus partisi dari tabel partisi pada disk.

4. Keluar dari `parted`.

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

(Gambar 35. Keluar dari *parted* dengan **quit**)

Creating File Systems

Setelah pembuatan partisi, langkah selanjutnya adalah menambahkan sistem file ke dalamnya. Ada dua yang umum dipakai yakni XFS dan ext4. Adapun, Anaconda disini dipakai sebagai manual instalasi untuk Red Hat Enterprise Linux yang menggunakan XFS secara default.

Sebagai **root**, gunakan perintah **mkfs.xfs** untuk menerapkan sistem file **XFS** ke *block device*. Untuk **ext4**, gunakan **mkfs.ext4**.

```
[root@host ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1            isize=512    agcount=4, agsize=60992 blks
      =                       sectsz=512    attr=2, projid32bit=1
      =                       crc=1        finobt=1, sparse=1, rmapbt=0
      =                       reflink=1
data      =                       bsize=4096   blocks=243968, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0, ftype=1
log       =internal log       bsize=4096   blocks=1566, version=2
      =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none               extsz=4096   blocks=0, rtextents=0
```

(Gambar 36. menerapkan file sistem **XFS** pada partisi)

Mounting File Systems

Setelah Anda menambahkan file sistem, langkah terakhir adalah memasang file sistem ke direktori dalam struktur direktori. Ketika Anda memasang sistem file ke dalam hierarki direktori, utilitas ruang pengguna dapat mengakses atau menulis file pada perangkat.

Manually Mounting File Systems

Kita dapat melakukan mounting secara manual dengan cara berikut:

```
[root@host ~]# mount /dev/vdb1 /mnt
```

(Gambar 37. Mounting)

Jika ingin melihat secara lengkap tempat kita sudah mount bisa seperti berikut:

```
[root@host ~]# mount | grep vdb1
/dev/vdb1 on /mnt type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

(Gambar 38. Melihat letak **vdb1** yang baru di mount)

Persistently Mounting File Systems

Memasang sistem file secara manual adalah cara yang baik untuk memastikan bahwa perangkat yang diformat dapat diakses dan berfungsi sesuai harapan. Namun, ketika server di-reboot, sistem tidak secara otomatis memasang sistem file ke direktori *tree* lagi; data utuh pada sistem file, tetapi pengguna tidak dapat mengaksesnya.

Untuk memastikan bahwa sistem secara otomatis memasang sistem file saat boot sistem, tambahkan entri ke file **/etc/fstab**. File konfigurasi ini mencantumkan sistem file yang akan dipasang saat boot sistem.

/etc/fstab adalah file yang dibatasi oleh spasi putih dengan enam bidang per baris.

```
[root@host ~]# cat /etc/fstab

#
# /etc/fstab
# Created by anaconda on Wed Feb 13 16:39:59 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
```

(Gambar 39. Penambahan letak untuk bagain /dev/vdb1)

Ketika Anda menambahkan atau menghapus entri di file **/etc/fstab**, jalankan perintah **systemctl daemon-reload**, atau reboot server, untuk sistem systemd mendaftarkan konfigurasi baru. (Dikarenakan menggunakan centos di *virtual machine* bisa dilakukan manual reboot juga)

```
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
UUID=a8063676-44dd-409a-b584-68be2c9f5570 / xfs defaults 0 0
UUID=7a20315d-ed8b-4e75-a5b6-24ff9e1f9838 /dbdata xfs defaults 0 0
```

(Gambar 40. Menambahkan vdb1 ke dalam file **/etc/fstab**)

Field	Penjelasan
Pertama	Bidang pertama menentukan perangkat. Contoh ini menggunakan UUID untuk menentukan perangkat. Sistem file menciptakan dan menyimpan UUID di blok super mereka pada saat pembuatan. Sebagai alternatif, Anda bisa menggunakan file perangkat, seperti /dev/vdb1.

Kedua	Titik mount direktori, dari mana perangkat blok akan diakses dalam struktur direktori. Titik mount harus ada; jika tidak, buatlah dengan perintah <code>mkdir</code> .
Ketiga	Berisi tipe sistem file, seperti <code>xfs</code> atau <code>ext4</code> .
Keempat	Daftar opsi yang dipisahkan koma yang akan diterapkan pada perangkat. Defaults adalah seperangkat opsi yang umum digunakan.
Kelima	Digunakan oleh perintah <code>dump</code> untuk mencadangkan perangkat. Aplikasi pencadangan lainnya biasanya tidak menggunakan bidang ini.
Keenam	Bidang urutan fsck , menentukan apakah perintah <code>fsck</code> harus dijalankan saat boot sistem untuk memastikan sistem file bersih. Nilai dalam bidang ini menunjukkan urutan di mana <code>fsck</code> harus berjalan. Untuk sistem file XFS atur menjadi 0 karena XFS tidak menggunakan <code>fsck</code> untuk memeriksa status sistem file-nya. Untuk sistem file ext4 , atur ke 1 untuk sistem file root dan 2 untuk sistem file <code>ext4</code> lainnya. Dengan cara ini, <code>fsck</code> memproses sistem file root terlebih dahulu dan kemudian memeriksa sistem file pada disk terpisah secara bersamaan, dan sistem file pada disk yang sama secara berurutan.

Managing Swap Space

Swap Space Concepts (Konsep Dasar)

Swap space adalah area pada disk yang dikendalikan oleh subsistem manajemen memori kernel Linux. Kernel menggunakan *swap space* untuk melengkapi RAM sistem dengan menyimpan halaman memori yang tidak aktif. Gabungan RAM sistem dan *swap space* disebut memori virtual.

Ketika penggunaan memori pada sistem melebihi batas yang ditentukan, kernel mencari melalui RAM untuk menemukan halaman memori yang tidak aktif yang diberikan kepada proses. Kernel menulis halaman-halaman yang tidak aktif ke area swap dan menetapkan kembali halaman RAM ke proses lain. Jika program memerlukan akses ke halaman pada disk, kernel menemukan halaman memori lain yang tidak aktif, menuliskannya ke disk, lalu memanggil kembali halaman yang dibutuhkan dari area swap.

Karena area swap berada di disk, swap lebih lambat dibandingkan dengan RAM. Meskipun digunakan untuk melengkapi RAM sistem, Anda tidak boleh menganggap *swap space* sebagai solusi yang berkelanjutan untuk RAM yang tidak mencukupi bagi beban kerja Anda.

Sizing the Swap Space (Menentukan Ukuran Swap Space)

Administrator harus menentukan ukuran *swap space* berdasarkan beban kerja memori pada sistem. Vendor aplikasi terkadang memberikan rekomendasi mengenai hal tersebut. Tabel berikut ini memberikan beberapa panduan berdasarkan jumlah total memori fisik:

RAM	Swap Space	Swap Space pada saat Mengizinkan Hibernasi
2 GiB atau kurang	Dua kali RAM	Tiga kali RAM
Antara 2 GiB & 8 GiB	Sama dengan RAM	Dua kali RAM
Antara 8 GiB & 64 GiB	Minimal 4 GiB	1,5 kali RAM
Lebih dari 64 GiB	Minimal 4 GiB	Hibernasi tidak direkomendasikan

Fungsi dari hibernasi laptop dan desktop menggunakan space swap untuk menyimpan konten RAM sebelum mematikan sistem. Ketika Anda menghidupkan sistem kembali, kernel mengembalikan konten RAM dari space swap dan tidak memerlukan proses boot yang lengkap. Untuk sistem tersebut, *swap space* harus lebih besar dari jumlah RAM.

Creating a Swap Space

Untuk membuat *swap space*, Anda perlu melakukan hal berikut:

- Buat partisi dengan tipe sistem file **linux-swap**.
- Letakkan *signature* swap pada perangkat.

Creating a Swap Partition

Gunakan *parted* untuk membuat partisi dengan ukuran yang diinginkan dan atur tipe sistem file-nya menjadi linux-swap. Di masa lalu, alat melihat tipe sistem file partisi untuk menentukan apakah perangkat harus diaktifkan. Namun, hal itu tidak lagi berlaku. Meskipun utilitas tidak lagi menggunakan tipe sistem file partisi, pengaturan tipe tersebut memungkinkan administrator untuk dengan cepat menentukan tujuan partisi.

Berikut ini adalah contoh pembuatan swap dengan memori 256mb:

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049kB  1001MB  1000MB                data

(parted) mkpart
Partition name? []? swap1
File system type? [ext2]? linux-swap
Start? 1001MB
End? 1257MB
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049kB  1001MB  1000MB                data
  2      1001MB  1257MB  256MB    linux-swap(v1) swap1

(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

(Gambar 41. Pembuatan Swap dengan nama *linux-swap*)

Setelah membuat partisi, jalankan perintah **udevadm settle**. Perintah ini menunggu sistem mendeteksi partisi baru dan membuat file perangkat yang terkait di **/dev**. Perintah ini hanya kembali ketika selesai.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

(Gambar 42. Penggunaan command **udevadm settle** untuk mendeteksi partisi yang baru dibuat)

Formatting the Device

Perintah **mkswap** menerapkan *signature swap* pada perangkat. Tidak seperti utilitas pemformatan lainnya, **mkswap** menulis blok data tunggal di awal perangkat, membuat sisa perangkat tidak terformat sehingga kernel dapat menggunakannya untuk menyimpan halaman memori.

```
[root@host ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 244 MiB (255848448 bytes)
no label, UUID=39e2667a-9458-42fe-9665-c5c854605881
```

(Gambar 43. Menghapus swap)

Activating a Swap Space

Anda dapat menggunakan perintah `swapon` untuk mengaktifkan ruang swap yang telah diformat.

Gunakan **swapon** dengan perangkat sebagai parameter, atau gunakan **swapon -a** untuk mengaktifkan semua ruang swap yang terdaftar dalam file `/etc/fstab`. Gunakan perintah **swapon --show** dan `free` untuk memeriksa ruang swap yang tersedia.

```
[root@host ~]# free
              total        used        free      shared  buff/cache   available
Mem:      1873036       134688       1536436        16748        201912       1576044
Swap:          0           0           0
[root@host ~]# swapon /dev/vdb2
[root@host ~]# free
              total        used        free      shared  buff/cache   available
Mem:      1873036       135044       1536040        16748        201952       1575680
Swap:     249852           0       249852
```

(Gambar 44. Mengaktifkan Swap space dan melihat jumlah memori dengan **free**)

Anda juga dapat menonaktifkan swap space menggunakan perintah `swapoff`. Jika *swap space* memiliki halaman yang ditulis, `swapoff` mencoba memindahkan halaman tersebut ke *swap space* aktif lainnya atau kembali ke memori. Jika tidak dapat menulis data ke tempat lain, perintah **swapoff** gagal dengan kesalahan, dan *swap space* tetap aktif.

Activating Swap Space Persistently

Untuk mengaktifkan ruang swap di setiap saat booting, letakkan entri di file `/etc/fstab`. Contoh di bawah ini menunjukkan baris tipikal dalam `/etc/fstab` berdasarkan ruang swap yang dibuat di atas.

```
UUID=39e2667a-9458-42fe-9665-c5c854605881    swap    swap    defaults    0 0
```

(Gambar 45. List baris pada `/etc/fstab`)

Contoh ini menggunakan UUID sebagai bidang pertama. Ketika Anda memformat perangkat, perintah **mkswap** menampilkan UUID tersebut.

Field	Penjelasan
Pertama	Jika Anda kehilangan keluaran mkswap, gunakan perintah lsblk --fs. Sebagai alternatif, Anda juga dapat menggunakan nama perangkat
Kedua	Digunakan sebagai titik kait. Namun, untuk perangkat swap, yang tidak dapat diakses melalui struktur direktori, bidang ini menggunakan nilai placeholder swap. Halaman fstab(5) man menggunakan nilai placeholder none , namun menggunakan nilai swap memungkinkan pesan kesalahan yang lebih informatif jika ada sesuatu yang salah.
Ketiga	Berisi tipe sistem file. Tipe sistem file untuk ruang swap adalah swap
Keempat	Merupakan Opsi. Pada conth digunakan opsi defaults. Opsi defaults mencakup opsi <i>mount auto</i> , yang berarti mengaktifkan ruang swap secara otomatis saat boot sistem.
Kelima dan Keenam	Bendera dump dan urutan fsck. Ruang swap tidak memerlukan pencadangan atau pengecekan sistem file, sehingga bidang ini harus diatur ke nol.

Ketika Sudah menambah atau menghapus entri di file **/etc/fstab**, jalankan perintah **systemctl daemon-reload**, atau reboot *virtual machine*, agar systemd mendaftarkan konfigurasi baru.

Setting the Swap Space Priority

Secara default, sistem menggunakan ruang *swap space series*, yang berarti kernel menggunakan *swap space* yang diaktifkan pertama kali sampai penuh, kemudian mulai menggunakan *swap space* kedua. Namun, Anda dapat menentukan prioritas untuk setiap *swap space* untuk memaksa urutan tersebut.

Untuk mengatur prioritas, gunakan opsi **pri** di **/etc/fstab**. Kernel menggunakan *swap space* dengan prioritas tertinggi terlebih dahulu. Prioritas default adalah -2.

Contoh berikut menunjukkan tiga ruang swap yang didefinisikan dalam. Kernel menggunakan entri terakhir terlebih dahulu, dengan **pri=10**. Ketika ruang itu penuh, kernel menggunakan entri kedua, dengan **pri=4**. Akhirnya, kernel menggunakan entri pertama, yang memiliki prioritas default -2.

```

UUID=af30cbb0-3866-466a-825a-58889a49ef33    swap    swap    defaults    0 0
UUID=39e2667a-9458-42fe-9665-c5c854605881    swap    swap    pri=4       0 0
UUID=fbd7fa60-b781-44a8-961b-37ac3ef572bf    swap    swap    pri=10      0 0

```

(Gambar 46. Deretan Swap space dengan memperlihatkan UUID menggunakan command **cat**)

Gunakan **swapon --show** untuk menampilkan prioritas ruang swap.

Catatan: Ketika ruang swap memiliki prioritas yang sama, kernel menulis ke dalam ruang swap secara bergantian (*round-robin*).

LEMBAR KERJA

KEGIATAN 1 (WAJIB DIKERJAKAN)

Mounting and Unmounting

1. Buatlah sebuah mount baru dengan menggunakan **fdisk**

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	20482047	10240000	83	Linux
/dev/sda2		20482048	24578047	2048000	82	Linux swap / Solaris
/dev/sda3		24578048	25316143	369048	83	Linux

(Expected output (Bisa bervariasi))

2. Jangan lupa di reboot dulu
3. Setelah itu gunakan command tertera diatas untuk melihat UUID dari device /dev/sda3 (disesuaikan) clue: -fp
4. Setelah itu mount ke /mnt/newspace direktori (bila belum ada, buat dulu direktorinya)
5. Kemudian pastikan kembali apakah sudah *mountpoint* di **/mnt/newspace** atau belum jika sudah, bisa lanjut
6. Tambahkan direktori baru pada /mnt/newspace bernama **newdir**
7. Buat sebuah file dengan nama **newfile.txt** letaknya di dalam **newdir** (/newdir/newfile.txt)
8. Kemudian unmount (Bila ada sesuatu terjadi disini, jelaskan kenapa bisa terjadi, bila tidak selamat kegiatan 1 selesai)
9. Kemungkinan perlu ada reboot, sehingga perlihatkan hasilnya terlebih dahulu

KEGIATAN 2

1. Melanjutkan dari kegiatan 1, gunakan command pada bagian **kedua** buat sebuah disk label baru dengan tipe '**msdos**' pada /dev/sda3
2. Dengan menggunakan partisi /dev/sda3 tambahkan sebuah partisi **primary** baru dengan size **1GB** dan sektor **2048** kemudian set tipe sistem file menjadi **XFS** dengan masuk sebagai *parted* (ada dua cara, bebas pilih yang mana)
3. Kemudian perlihatkan hasil yang sudah dibuat dengan **print**
4. Kemudian register hasil pembuatan partisi dengan command yang tersedia (hint: settle)
5. Setelah itu, format partisi baru tersebut menjadi sebuah **XFS** dengan cara **mkfs.xfs**
6. Lalu, konfigurasi agar **mount** pada /archive, bila belum ada direktori /archive maka anda perlu membuat terlebih dahulu
7. Perlihatkan hasil **mountpoint** beserta **UUID**
8. Update systemd agar bisa *register* konfigurasi baru pada /etc/fstab
9. Jelaskan **minimal 2** bagian dari entri di **/etc/fstab** (Lebih banyak lebih bagus yaa)
10. Gunakan **grep** untuk memperlihatkan hasil detail dari /archive
11. Selesai (Jika belum muncul atau ada error biasa harus reboot, silahkan konfirmasi ke asisten sebelum reboot)

KEGIATAN 3

1. Pada kegiatan 3 ini sama halnya, harus menyelesaikan kegiatan 1 dulu
2. Sekarang dengan menggunakan **parted** silahkan membuat sebuah partisi swap dengan menggunakan metode **GPT partitioning** dengan nama **myswap**
3. Kemudian verifikasi hasil pembuatan swap tersebut dengan **print**

Number	Start	End	Size	File system	Name	Flags
1	1049kB	1001MB	1000MB		data	
2	1001MB	1501MB	499MB		myswap	swap

(Expected Output)

4. Kemudian register partisi baru seperti di kegiatan 2 pakai yang hint diatas
5. Inisialisasi partisi baru tersebut dengan mk....
6. **Enable** swap space (hint: nyalakan)
7. Perlihatkan hasil enable (hint: pakai yang mirip **print**)
8. **Disable** swap space
9. Perlihatkan hasil disable
10. Pastikan bahwa untuk melakukan register pada konfigurasi sehingga kemungkinan perlu reload, sehingga perlihatkan terlebih dahulu hasil baru kemudian di reboot dan perlihatkan hasil yang sudah jadi.

CATATAN:

Sebelum mulai praktikum silahkan menghapus (format) terlebih dahulu **Mount** yang baru dibuat (/dev/sda3) baru kemudian dapat melaksanakan praktikum

RUBRIK PENILAIAN

Aspek Penilaian	Bobot Penilaian
Ketepatan menjawab semua kegiatan	65%
Pemahaman setiap aspek materi yang dibahas	20%
Quiz pertemuan materi minggu pertama	15%
Total	100%