
Nama : Josua Panggabean

NIM : 13320007



Laporan Praktikum Administrasi Jaringan

28 -12- 2022

**Institut Teknologi Del
Jl. Sisingamangaraja, Sitoluama, Laguboti
Kabupaten Toba, Sumatera Utara**

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR	ii
DAFTAR TABEL.....	ix
BAB 1. PENGELOLAAN USER & GROUP PADA SISTEM OPERASI LINUX	10
BAB 2. PROCESS CONTROL.....	17
BAB 3. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 1	26
BAB 4. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 2	40
BAB 5. FILESYSTEM AND LOGICAL VOLUME MANAGER	61
BAB 6. FILESYSTEM AND LOGICAL VOLUME MANAGER – PART II	71
BAB 7. SAMBA DOMAIN CONTROLLER.....	9
BAB 8. PENGELOLAAN OBJEK PADA ACTIVE DIRECTORY MENGGUNAKAN POWERSHELL	16
BAB 9. MANAGING GROUP POLICY & ADVANCED AD MANAGEMENT WITH POWERSHELL.....	26
BAB 10. AUDITING & MONITORING.....	34

DAFTAR GAMBAR

Gambar 1	12
Gambar 2	12
Gambar 3	12
Gambar 4	12
Gambar 5	12
Gambar 6	12
Gambar 7	13
Gambar 8	13
Gambar 9	13
Gambar 10	13
Gambar 11	13
Gambar 12	13
Gambar 13	17
Gambar 14	18
Gambar 15	18
Gambar 16	19
Gambar 17	19
Gambar 18	20
Gambar 19	20
Gambar 20	20
Gambar 21	21
Gambar 22	21
Gambar 23	22
Gambar 24	22
Gambar 25	25
Gambar 26	25
Gambar 27	25
Gambar 28	25
Gambar 29	26
Gambar 30	26
Gambar 31	27
Gambar 32	27
Gambar 33	27

Gambar 34	28
Gambar 35	28
Gambar 36	28
Gambar 37	29
Gambar 38	34
Gambar 39	34
Gambar 40	34
Gambar 41	35
Gambar 42	36
Gambar 43	36
Gambar 44	36
Gambar 45	37
Gambar 46	37
Gambar 47	37
Gambar 48	38
Gambar 49	38
Gambar 50	38
Gambar 51	39
Gambar 52	39
Gambar 53	41
Gambar 54	42
Gambar 55	42
Gambar 56	42
Gambar 57	42
Gambar 58	42
Gambar 59	44
Gambar 60	44
Gambar 61	44
Gambar 62	44
Gambar 63	44
Gambar 64	45
Gambar 65	45
Gambar 66	45
Gambar 67	45

Gambar 68	45
Gambar 69	46
Gambar 70	46
Gambar 71	47
Gambar 72	47
Gambar 73	47
Gambar 74	47
Gambar 75	47
Gambar 76	47
Gambar 77	48
Gambar 78	48
Gambar 79	49
Gambar 80	49
Gambar 81	49
Gambar 82	49
Gambar 83	49
Gambar 84	50
Gambar 85	50
Gambar 86	50
Gambar 87	53
Gambar 88	54
Gambar 89	54
Gambar 90	54
Gambar 91	54
Gambar 92	54
Gambar 93	54
Gambar 94	55
Gambar 95	55
Gambar 96	55
Gambar 97	56
Gambar 98	56
Gambar 99	57
Gambar 100	57
Gambar 101	57

Gambar 102	57
Gambar 103	58
Gambar 104	58
Gambar 105	58
Gambar 106	58
Gambar 107	58
Gambar 108	58
Gambar 109	59
Gambar 110	59
Gambar 111	59
Gambar 112	59
Gambar 113	59
Gambar 114	60
Gambar 115	60
Gambar 116	60
Gambar 117	61
Gambar 118	62
Gambar 119	62
Gambar 120	63
Gambar 121	63
Gambar 122	64
Gambar 123	64
Gambar 124	65
Gambar 125	65
Gambar 126	66
Gambar 127	66
Gambar 128	67
Gambar 129	67
Gambar 130	67
Gambar 131	67
Gambar 132	68
Gambar 133	68
Gambar 134	68
Gambar 135	68

Gambar 136	69
Gambar 137	69
Gambar 138	69
Gambar 139	69
Gambar 140	70
Gambar 141	71
Gambar 142	72
Gambar 143	72
Gambar 144	73
Gambar 145	73
Gambar 146	74
Gambar 147	1
Gambar 148	1
Gambar 149	2
Gambar 150	2
Gambar 151	3
Gambar 152	3
Gambar 153	3
Gambar 154	4
Gambar 155	4
Gambar 156	4
Gambar 157	4
Gambar 158	5
Gambar 159	5
Gambar 160	5
Gambar 161	5
Gambar 162	6
Gambar 163	6
Gambar 164	6
Gambar 165	7
Gambar 166	7
Gambar 167	7
Gambar 168	8
Gambar 169	8

Gambar 170	9
Gambar 171	9
Gambar 172	9
Gambar 173	10
Gambar 174	10
Gambar 175	10
Gambar 176	11
Gambar 177	11
Gambar 178	11
Gambar 179	12
Gambar 180	12
Gambar 181	12
Gambar 182	13
Gambar 183	13
Gambar 184	14
Gambar 185	14
Gambar 186	14
Gambar 187	15
Gambar 188	17
Gambar 189	18
Gambar 190	19
Gambar 191	19
Gambar 192	20
Gambar 193	20
Gambar 194	20
Gambar 195	22
Gambar 196	23
Gambar 197	23
Gambar 198	24
Gambar 199	24
Gambar 200	25
Gambar 201	25
Gambar 202	26
Gambar 203	27

Gambar 204	27
Gambar 205	28
Gambar 206	28
Gambar 207	29
Gambar 208	29
Gambar 209	30
Gambar 210	30
Gambar 211	30
Gambar 212	31
Gambar 213	31
Gambar 214	32
Gambar 215	32
Gambar 216	33
Gambar 217	33
Gambar 218	36
Gambar 219	36
Gambar 220	36
Gambar 221	37
Gambar 222	37
Gambar 223	37
Gambar 224	38
Gambar 225	38
Gambar 226	39
Gambar 227	39
Gambar 228	39
Gambar 229	40
Gambar 230	40
Gambar 231	40
Gambar 232	41
Gambar 233	41
Gambar 234	42
Gambar 235	42
Gambar 236	43
Gambar 237	43

Gambar 238	43
Gambar 239	43
Gambar 240	43
Gambar 241	44
Gambar 242	44
Gambar 243	44
Gambar 244	44
Gambar 245	44
Gambar 246	45
Gambar 247	45
Gambar 248	45
Gambar 249	46

DAFTAR TABEL

Table 1 Anggota Divisi.....	11
Table 2	35
Table 3	52
Table 4 Event 1 DS Access	35

BAB 1. PENGELOLAAN USER & GROUP PADA SISTEM OPERASI LINUX

A. Skenario

“Pengelolaan User dan Group pada Linux”

Perusahaan Scroll Royce bergerak dalam bidang pertambangan, dengan Head Office yang berlokasi di Jakarta. Pada perusahaan terdapat beberapa divisi kerja, yaitu divisi Finance, HRD & GA, CEO, dan Legal yang dipimpin oleh seorang Komisaris perusahaan yaitu Bapak Binsar, yang mempunyai wewenang penuh untuk tiap divisi.

Pada Head Office telah diimplementasikan pembangunan jaringan lokal maupun internet, tetapi jaringan belum dikelola dengan baik, terutama dalam hal file sharing untuk masing-masing divisi dan penerapan pembatasan hak akses. Berdasarkan requirement dari pihak HRD, maka administrator jaringan diminta untuk membuat folder kerja dengan nama sesuai dengan masing-masing divisi, folder share informasi public, dan folder kerja untuk komisaris. Divisi yang pekerjaannya saling berhubungan dengan divisi lainnya akan mendapatkan hak akses terhadap folder kerja masing-masing divisi tersebut, dengan ketentuan sebagai berikut.

- 1. Folder Finance dapat diakses oleh bagian Finance, bagian CEO, dan juga komisaris dengan mendapatkan akses penuh.**
- 2. Folder HRD & GA dapat diakses oleh bagian HRD & GA dan komisaris dengan hak akses penuh, sedangkan bagian Finance dan Legal dengan hak akses read dan execute.**
- 3. Folder CEO hanya dapat diakses oleh CEO dan Komisaris dengan hak akses penuh.**
- 4. Folder Legal dapat diakses oleh bagian Legal, CEO, dan Komisaris dengan hak akses penuh, serta dapat diakses oleh user Bylardo bagian Finance dengan hak akses read dan execute.**
- 5. Folder Komisaris hanya dapat diakses oleh komisaris perusahaan.**
- 6. Folder Info Publik dapat diakses oleh semua user.**

Anggota setiap divisi dapat dilihat pada

Tabel 1 berikut.

Table 1 Anggota Divisi

Divisi	Anggota
Finance	Bylardo
	Leo
HRD & GA	Sahat
	Alexander
CEO	Christine
	Ruth
Legal	Ruhut
	Togar
Komisaris	Binsar

Berdasarkan skenario diatas anda diposisikan menjadi seorang administrator yang akan memenuhi *requirement* dari HRD, dengan ketentuan berikut:

- 1. Tambahkan user sesuai dengan Tabel 1 pada server Anda, dan lengkapi dengan nama lengkap user.**
- 2. Home direktori untuk setiap user terletak di /home/ScrollRoyce.**
- 3. Buatlah tabel yang berisi hak akses antara folder masing-masing divisi dengan usernya.**
- 4. Folder yang dibuat sesuai dengan nama divisi terletak di /data/nama_folder.**
- 5. Implementasikan pembuatan folder dan pengaturan hak aksesnya, sesuai dengan *requirement*.**
- 6. Anda dapat melakukan penambahan user, pengubahan informasi user, pembuatan folder, pembuatan group, dan pembatasan hak akses atau permission.**
- 7. Tampilkan hasil screenshot dari eksekusi:**
 - a. perintah untuk menambahkan setiap user beserta home direktorinya,**
 - b. perintah untuk membuat group serta registrasi user ke group**
 - c. dan perintah untuk memberikan hak akses pada folder serta kepemilikan folder.**

B. Jawaban

1. Membuat direktori /home/ScrollRoyce

```
[root@localhost data]# mkdir /home/ScrollRoyce  
[root@localhost data]#
```

Gambar 1

2. Membuat user baru pada direktori /home/ScrollRoyce

```
[root@localhost data]# sudo useradd Leo -d /home/ScrollRoyce  
useradd: user 'Leo' already exists  
[root@localhost data]# sudo passwd Leo  
Changing password for user Leo.  
New password:  
BAD PASSWORD: The password contains the user name in some form  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@localhost data]#
```

Gambar 2

```
[root@localhost data]# sudo useradd Bylardo -d /home/ScrollRoyce  
useradd: user 'Bylardo' already exists  
[root@localhost data]#
```

Gambar 3

```
[root@localhost data]# useradd Bylardo  
[root@localhost data]# useradd Leo  
[root@localhost data]# useradd Sahat  
[root@localhost data]# useradd Alexander  
[root@localhost data]# useradd Christine  
[root@localhost data]# useradd Ruth  
[root@localhost data]# useradd Ruhut  
[root@localhost data]# useradd Togar  
[root@localhost data]# useradd Binsar
```

Gambar 4

3. Membuat group user

```
[root@localhost data]# groupadd Finance  
[root@localhost data]# groupadd HRDGA  
[root@localhost data]# groupadd CEO  
[root@localhost data]# groupadd Legal
```

Gambar 5

4. Mensest user kedalam group

```
[root@localhost data]# usermod -a -G Finance Bylardo  
[root@localhost data]# usermod -a -G Finance Leo  
[root@localhost data]# usermod -a -G HRD GA Sahat | usermod -a -G HRD GA Alexander  
[root@localhost data]# usermod -a -G CEO Christine | usermod -a -G CEO Ruth  
[root@localhost data]# usermod -a -G Legal Ruhut | usermod -a -G Togar  
[root@localhost data]# usermod -a -G Komisaris Binsar
```

Gambar 6

5. Kemudian men-create direktori /data/nama_folder dan membuat direktori

Kemudian kita melakukan set permission:

1. Folder Finance dapat diakses oleh bagian Finance, bagian CEO, dan juga komisaris dengan mendapatkan akses penuh.

```
|||[root@localhost ~]# cd data
|[root@localhost data]# mkdir Finance
|[root@localhost data]# mkdir HRD_GA
|[root@localhost data]# mkdir CEO
|[root@localhost data]# mkdir Legal
|[root@localhost data]# mkdir Komisaris
|[root@localhost data]# ls
```

Gambar 7

```
[root@localhost data]# setfacl -m g:Finance:rwx -R Finance
[root@localhost data]# setfacl -m g:CEO:rwx -R Finance
[root@localhost data]# setfacl -m g:Komisaris:rwx -R Finance
```

Gambar 8

2. Folder HRD & GA dapat diakses oleh bagian HRD & GA dan komisaris dengan hak akses penuh, sedangkan bagian Finance dan Legal dengan hak akses read dan execute.

```
[root@localhost data]# setfacl -m g:HRD_GA:rwx -R HRD_GA
[root@localhost data]# setfacl -m g:Komisaris:rwx -R HRD_GA
[root@localhost data]# setfacl -m g:Finance:rwx -R HRD_GA
[root@localhost data]# setfacl -m g:Legal:rwx -R HRD_GA
```

Gambar 9

3. Folder CEO hanya dapat diakses oleh CEO dan Komisaris dengan hak akses penuh.

```
[root@localhost data]# setfacl -m g:CEO:rwx -R CEO
[root@localhost data]# setfacl -m g:Komisaris:rwx -R CEO
```

Gambar 10

4. Folder Legal dapat diakses oleh bagian Legal, CEO, dan Komisaris dengan hak akses penuh, serta dapat diakses oleh user Bylardo bagian Finance dengan hak akses read dan execute.

```
[root@localhost data]# setfacl -m g:Legal:rwx -R Legal
[root@localhost data]# setfacl -m g:Komisaris:rwx -R Legal
[root@localhost data]# setfacl -m g:CEO:rwx -R Legal
[root@localhost data]# setfacl -m g:Komisaris:rwx -R Komisaris
```

Gambar 11

5. Folder Komisaris hanya dapat diakses oleh komisaris perusahaan.

```
[root@localhost data]# setfacl -m g:Komisaris:rwx -R Komisaris
```

Gambar 12

Script :

```
999 sudo useradd Bylardo -d /home/ScrollRoyce
1000 ls
1001 sudo passwd Bylardo
1002 ls
1003*
1004 sudo passwd Leo
1005 sudo useradd Sahat -d /home/ScrollRoyce
1006 sudo passwd Sahat
1007 sudo useradd Alexander -d /home/ScrollRoyce
1008 sudo passwd Alexander
1009 sudo adduser Christine -d /home/ScrollRoyce
1010 sudo passwd Christine
1011 sudo adduser Ruth -d /home/ScrollRoyce
1012 sudo passwd Ruth
1013 sudo adduser Ruhut -d /home/ScrollRoyce
1014 sudo passwd Ruhut
1015 sudo adduser Togar -d /home/ScrollRoyce
1016 sudo passwd Togar
1017 sudo adduser Binsar -d /home/ScrollRoyce
1018 sudo passwd Binsar
1019 cd /data
1020 cd
1021 ls
1022 mkdir data
1023 ls data
1024 cd data
1025 mkdir Finance
1026 mkdir HRD & GA
1027 mkdir HRD_&_GA
1028 mkdir HRD&GA
1029 mkdir hrd&ga
1030 rmdir Finance
1031 ls
```

```
1032 rmdir nrđ
1033 rm HRD
1034 rmdir HRD
1035 rmdir HRD_
1036 rmdir hrd
1037 mkdir Finance
1038 mkdir CEO
1039 mkdir Legal
1040 mkdir Komisaris
1041 mkdir Info
1042 mkdir HRDGA
1043 ls
1044 groupadd Finance
1045 groupadd HRDGA
1046 groupadd CEO
1047 groupadd Legal
1048 groupadd Komisaris
1049 usermod -a -G Bylardo
1050 usermod -a -G Bylardo Leo
1051 usermod -a -G Finance Bylardo | usermod -a -G Finance Leo
1052 usermod -a -G HRDGA Sahat | usermod -a -G HRDGA Alexander
1053 usermod -a -G CEO Christine | usermod -a -G CEO Christine
1054 usermod -a -G Legal Ruhut | usermod -a -G Legal Togar
1055 usermod -a -G Komisaris Binsar
1056 ls
1057 setfacl -m g:Finance:rwx -R Finance
1058 setfacl -m g:CEO:rwx -R Finance
1059 setfacl -m g:Komisaris:rwx -R Finance
1060 setfacl -m g:HRDGA:rwx -R HRDGA
1061 setfacl -m g:Komisaris:rwx -R HRDGA
1062 setfacl -m g:Finance:rx -R HRDGA
1063 setfacl -m g:Legal:rx -R HRDGA
1064 setfacl -m g:CEO:rwx -R CEO
1065 setfacl -m g:Komisaris:rwx -R CEO
```

```
1066 setfacl -m g:Legal:rwx -R Legal
1067 setfacl -m g:CEO:rwx -R Legal
1068 setfacl -m g:Komusaris:rwx -R Legal
1069 setfacl -m g:Komisaris:rwx -R Legal
1070 setfacl -m u:Bylardo:rx -R Legal
1071 setfacl -m g:Komisaris:rwx -R Komusaris
1072 setfacl -m g:Komisaris:rwx -R Komisaris
1073 ls
1074 chmod 777 -R InfoPublik
1075 sudo chmod 777 InfoPublik
1076 sudo chmod 777 -R InfoPublik
```

BAB 2. PROCESS CONTROL

A. Process Control

A.1 Atribut Proses

Sebuah proses memiliki karakteristik yang dapat dilihat dengan menggunakan perintah

ps, sebagai berikut:

- **ID proses atau PID:** nomor identifikasi unik yang digunakan untuk merujuk kepada sebuah proses.
- **ID Proses induk atau PPID:** nomor identifikasi unik yang dimiliki oleh proses induk.
- **Nice number:** derajat prioritas sebuah proses.
- **Terminal atau TTY:** terminal dimana proses tersebut terhubung.
- **Username dari real dan effective user (RUID dan EUID):** Pemilik proses. Pemilik sebenarnya sebuah proses adalah pengguna yang mengeluarkan perintah (command), pengguna yang efektif (effective user) adalah pengguna yang menentukan akses ke sumber daya sistem. RUID dan EUID biasanya sama, dan prosesnya memiliki hak akses yang sama dengan pengguna yang mengeksekusi perintah.
- **Real and effective group owner (RGID and EGID):** pemilik sebenarnya dari sebuah proses adalah grup utama pengguna yang memulai proses. Pemilik group yang efektif biasanya sama, kecuali jika mode akses SGID telah diterapkan ke file.

A.2 Menampilkan Informasi Proses Menggunakan Perintah ps

Perintah ps adalah perintah utama dari seorang sistem administrator untuk memonitor proses. Perintah ini memiliki opsi-opsi untuk menampilkan atribut dari sebuah proses seperti PID, UID, prioritas dan terminal dari proses tersebut dieksekusi (seperti yang telah dijelaskan di A.1). Tanpa menentukan opsi, perintah ps hanya memberikan informasi mengenai shell yang sedang digunakan dan daftar proses yang sedang berjalan.

```
[eka@localhost ~]$ ps
  PID TTY      TIME CMD
 1241 pts/0    00:00:00 bash
 1261 pts/0    00:00:00 bash
 1282 pts/0    00:00:00 ps
```

Gambar 13

Namun informasi tersebut tidak cukup, dibutuhkan informasi yang berguna mengenai proses yang sedang berjalan pada sistem. Oleh karena itu, perintah ps dapat dipasangkan dengan argumen lainnya (lihat man ps), misalnya ps aux. Dengan penjelasan, argumen a menampilkan seluruh proses, argumen x menampilkan proses yang tidak memiliki kontrol pada terminal, sedangkan argumen u menampilkan format keluaran yang memilih “user oriented”.

```
[eka@localhost ~]$ ps -aux | more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root        1  0.0  0.1 125260  3720 ?        Ss  23:19  0:01 /usr/lib/systemd --switched-root --system --deserialize 21
root        2  0.0  0.0      0     0 ?        S   23:19  0:00 [kthreadd]
root        3  0.0  0.0      0     0 ?        S   23:19  0:00 [ksoftirqd/0]
root        5  0.0  0.0      0     0 ?        S<  23:19  0:00 [kworker/0:0H]
root        6  0.0  0.0      0     0 ?        S   23:19  0:00 [kworker/u2:0]
root        7  0.0  0.0      0     0 ?        S   23:19  0:00 [migration/0]
root        8  0.0  0.0      0     0 ?        S   23:19  0:00 [rcu_bh]
root        9  0.0  0.0      0     0 ?        R   23:19  0:00 [rcu_sched]
root       10  0.0  0.0      0     0 ?        S   23:19  0:00 [watchdog/0]
```

Gambar 14

Berikan hasil dari eksekusi perintah ps dengan argumen lax, kemudian berikan penjelasan dari luaran perintah tersebut.

Perintah dengan argumen yang panjang dapat menyebabkan perintah yang akan digunakan terpotong. Untuk mengatasi ini dapat menggunakan argumen w untuk menampilkan beberapa kolom dari luaran. Tambahkan w dua kali untuk lebar kolom yang tidak terbatas, argumen ini sangat membantu untuk process yang menggunakan argumen berbasis perintah yang panjang.

Seorang administrator seringkali membutuhkan informasi yang spesifik dari seluruh proses yang sedang berjalan pada sistem. Bayangkan jika anda harus mencari proses yang spesifik dari ratusan proses lainnya. Untuk mempermudah hal ini, dapat menggunakan perintah grep dan juga menggunakan pipe. Sebagai contoh perintah untuk menemukan PID dari proses sshd.

```
[eka@localhost ~]$ ps aux | grep sshd
root      892  0.0  0.2 106048  4104 ?        Ss  23:19  0:00 /usr/sbin/sshd -D
root     1230  0.0  0.2 148368  5452 ?        Ss  23:47  0:00 sshd: eka [priv]
eka      1240  0.0  0.1 148368  2256 ?        S   23:48  0:00 sshd: eka@pts/0
eka      1285  0.0  0.0 112812   972 pts/0     R+  23:52  0:00 grep --color=auto sshd
```

Gambar 15

Luaran dari perintah ps juga menampilkan perintah grep, dikarenakan grep aktif pada daftar proses yang sedang berjalan. Anda dapat menghapus baris ini dengan menggunakan perintah grep -v, seperti contoh berikut.

```
[eka@localhost ~]$ ps aux | grep -v grep | grep sshd
root      892  0.0  0.2 106048  4104 ?          Ss   23:19    0:00 /usr/sbin/sshd -D
root     1230  0.0  0.2 148368  5452 ?          Ss   23:47    0:00 sshd: eka [priv]
eka      1240  0.0  0.1 148368  2256 ?          S    23:48    0:00 sshd: eka@pts/0
```

Gambar 16

A.3 Menampilkan Informasi Proses Menggunakan Perintah top

Perintah ps menampilkan potret dari sistem pada waktu perintah ps dijalankan. Seringkali, informasi dari perintah ps tidak memberikan gambaran besar dari apa yang sedang terjadi pada sistem secara real time. Top adalah perintah untuk mengatasi kekurangan dari perintah ps tersebut. Top adalah versi real time dari ps yang memperbaharui informasi sistem secara berkala, memberikan kesimpulan dari proses dengan interaktif, serta memberikan informasi mengenai sumber daya yang digunakan oleh proses-proses yang berjalan pada sistem.

```
[eka@localhost ~]$ top
top - 23:57:17 up 37 min,  2 users,  load average: 0.00, 0.01, 0.05
Tasks: 105 total,   1 running, 104 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.3 us,  0.3 sy,  0.0 ni, 99.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 1883560 total, 1668220 free,   92168 used, 123172 buff/cache
KiB Swap: 499708 total, 499708 free,       0 used. 1645144 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 1296 eka      20   0 157736  2164  1532 R  1.0  0.1  0:00.09 top
 1286 root     20   0      0      0      0 S  0.3  0.0  0:00.42 kworker/0:1
    1 root     20   0 125260  3720  2508 S  0.0  0.2  0:01.54 systemd
    2 root     20   0      0      0      0 S  0.0  0.0  0:00.00 kthreadd
    3 root     20   0      0      0      0 S  0.0  0.0  0:00.09 ksoftirqd/0
    5 root      0 -20      0      0      0 S  0.0  0.0  0:00.00 kworker/0:0H
    6 root     20   0      0      0      0 S  0.0  0.0  0:00.01 kworker/u2:0
    7 root      rt   0      0      0      0 S  0.0  0.0  0:00.00 migration/0
```

Gambar 17

Secara default, tampilan informasi dari top diperbaharui setiap 1-2 detik. Proses yang paling banyak mengonsumsi sumberdaya CPU akan ditampilkan paling atas dari seluruh informasi proses yang ada. Namun demikian, kesimpulan informasi dari beberapa baris hasil luaran perintah top adalah informasi yang memerlukan prioritas untuk dianalisis dikarenakan hasil tersebut dapat memberikan informasi dari keadaan sistem apakah dalam keadaan yang baik atau tidak. Alasannya adalah dikarenakan informasi tersebut menunjukkan beban dari sistem, penggunaan memori, jumlah proses, dan uraian bagaimana CPU sedang digunakan.

A.4 Prioritas Penjadwalan Menggunakan Perintah nice dan renice

“Niceness” adalah petunjuk untuk kernel untuk menjadwalkan eksekusi sebuah proses oleh CPU. “high niceness” berarti sebuah proses diset dengan prioritas rendah, sedangkan “low niceness” berarti sebuah proses diset

dengan prioritas tinggi. Pada sistem operasi Linux, rentang dari -20 (prioritas tinggi, “not nice”) sampai dengan +19 (prioritas rendah, “very nice”), sedangkan 0 adalah nilai default. Kecuali pengguna memberikan aksi yang spesial, sebuah proses yang baru diciptakan akan mewarisi “niceness” dari proses induknya. Pemilik proses dapat meningkatkan “niceness” tetapi tidak dapat menurunkannya, bahkan menurunkannya menjadi nilai default. Tujuannya adalah untuk mencegah proses-proses yang berjalan pada prioritas rendah menanggung proses anak dengan prioritas tinggi. Meskipun demikian, superuser dapat mengatur nilai “nice” secara berubah-ubah.

Perintah nice dapat digunakan ketika membuat sebuah proses, namun dapat disesuaikan kemudian menggunakan perintah renice. Nice membutuhkan argument, sedangkan nice membutuhkan PID atau kadang-kadang username.

```
[eka@localhost ~]$ ps -lu eka | grep hello
0 S 1005 1439 1394 0 80 0 - 1054 hrtme pts/1 00:00:00 hello
```

Gambar 18

```
[eka@localhost ~]$ ps -lu eka | grep hello
0 S 1005 1444 1261 0 90 10 - 1054 hrtme pts/0 00:00:00 hello
```

Gambar 19

```
[root@localhost eka]# renice -n 5 -p 1444
1444 (process ID) old priority 10, new priority 5
```

Gambar 20

Berilah contoh penggunaan nice dan renice.

B. Siklus Hidup sebuah Proses

Untuk menciptakan sebuah proses baru, sebuah proses menyalin dirinya sendiri dengan menggunakan system call fork. Fork menciptakan salinan proses baru yang identik dengan proses induknya. proses yang baru memiliki PID yang berbeda dan memiliki informasi sendiri. Setelah menjalankan system call fork, proses anak menggunakan perintah exec untuk memulai eksekusi program.

Ketika sistem di-boot, kernel secara otomatis menciptakan dan menginstal beberapa proses, maka proses yang selalu menjadi nomor 1 adalah init atau systemd. Seluruh proses selain init akan dijadikan kernel sebagai keturunan dari proses init.

Init atau systemd menjalankan peran yang penting dalam mengelola proses. Ketika sebuah proses telah selesai dijalankan, maka sebuah system call

`_exit` akan dipanggil untuk memberitahu kernel bahwa proses tersebut sudah dapat diakhiri. Nilai 0 merupakan indikasi bahwa proses telah sukses diterminasi.

Sebelum proses dapat diizinkan untuk diterminasi dan hilang, kernel perlu memastikan bahwa induk proses telah mengetahui bahwa proses anak akan diterminasi. Dengan demikian, proses induk harus memanggil system call **wait**. Jika sebuah proses anak selesai, tetapi proses induk tidak memanggil wait, maka proses anak disebut sebagai **zombie**. Sementara, jika proses induk terlebih dahulu selesai dan diterminasi sebelum proses anak, maka proses anak disebut sebagai **orphan**.

B.1 Signals

Proses diterminasi karena menerima signal. Ada beberapa signal yang dapat dikirimkan untuk menterminasi proses sebagai contoh adalah perintah **kill**. Perintah kill dapat mengirimkan signal apapun, namun secara default adalah signal TERM. **kill** dapat digunakan oleh normal users pada prosesnya sendiri atau root pada seluruh proses. Signal yang harus diketahui oleh administrator sistem, seperti pada tabel berikut.

# ^b	Name	Description	Default	Can catch?	Can block?	Dump core?
1	HUP	Hangup	Terminate	Yes	Yes	No
2	INT	Interrupt	Terminate	Yes	Yes	No
3	QUIT	Quit	Terminate	Yes	Yes	Yes
9	KILL	Kill	Terminate	No	No	No
10	BUS	Bus error	Terminate	Yes	Yes	Yes
11	SEGV	Segmentation fault	Terminate	Yes	Yes	Yes
15	TERM	Software termination	Terminate	Yes	Yes	No
17	STOP	Stop	Stop	No	No	No
18	TSTP	Keyboard stop	Stop	Yes	Yes	No
19	CONT	Continue after stop	Ignore	Yes	No	No
28	WINCH	Window changed	Ignore	Yes	Yes	No
30	USR1	User-defined #1	Terminate	Yes	Yes	No
31	USR2	User-defined #2	Terminate	Yes	Yes	No

Gambar 21

Sebuah perintah kill tanpa nomor signal tidak dapat menjamin bahwa proses akanditerminasi, sebab signal TERM bisa saja diblokir atau diabaikan. Contoh penggunaan kill sebagai berikut.

```
[eka@localhost ~]$ kill -9 1525
```

Gambar 22

Perintah diatas akan menjamin bahwa proses akan diterminasi dikarenakan adanya argumen signal 9.

Berikutnya perintah pkill akan mencari seluruh proses dengan menggunakan nama atau atribut proses lainnya dan mengirimkan signal.

Sebagai contoh, perintah berikut akan mengirimkan signal TERM ke seluruh proses yang dijalankan oleh pengguna xxx.

```
[root@localhost eka]# pkill -u eka
```

Gambar 23

C. Strace: Melacak signal dan system call

Terkadang, sangat sulit untuk melacak apa yang sebenarnya dilakukan oleh proses padasistem. Langkah pertama yang pada umumnya dilakukan adalah mengumpulkan data dari filesystem, log dan menggunakan alat seperti **ps**. Jika informasi yang telah dikumpulkan tidak mencukupi, maka proses dapat dilacak hingga ke level yang paling rendah (lower level) dengan menggunakan perintah **strace** (**di Linux perintah ini adalah paket opsional, harus diinstall**). Perintah ini menampilkan setiap system call yang dilakukan oleh proses, jadi yang dilakukan adalah perintah ini akan melacak system call dalam waktu tertentu, kemudian menyelesaikan pelacakannya tanpa mengganggu kerja proses. **strace** juga dilengkapi dengan argumen seperti **-f**, untuk informasi lebih lengkapnya dapat diketahui pada dokumentasi strace (**man strace**). Sebagai contoh perintah strace dijalankan untuk mengetahui aktivitas system call yang dilakukan oleh sebuah proses dengan PID 1968.

```
[eka@localhost ~]$ strace -p 1968
strace: Process 1968 attached
restart_syscall(<... resuming interrupted read ...>) = 0
write(1, "Hello, World! \n", 15)      = 15
rt_sigprocmask(SIG_BLOCK, [CHLD], [], 8) = 0
rt_sigaction(SIGCHLD, NULL, {sa_handler=SIG_DFL, sa_mask=[], sa_flags=0}, 8) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
nanosleep({tv_sec=10, tv_nsec=0}, [
```

Gambar 24

administrator. Cara ini dapat digunakan setelah menggunakan cara tradisional seperti memeriksa berkas log.

D. Penjadwalan Proses

Penggunaan skrip/program atau perintah sering digunakan untuk tanpa adanya interfensi manusia. Pada umumnya digunakan untuk menjadwalkan backup, pemeliharaan database, atau eksekusi terhadap batch job.

D.1 Cron: perintah untuk penjadwalan

Daemon cron adalah perangkat tradisional yang digunakan untuk menjalankan perintah yang jadwalnya telah ditentukan. Sebuah konfigurasi cron disebut sebagai “crontab” singkatan dari “cron table”. Ketika sebuah sistem mulai cron daemon mencari entri contab di /var/spool/cron, kemudian mencari /etc/cron.d dan /etc/crontab, selanjutnya menggunakan informasi

ini setiap menit untuk memeriksa apakah ada perintah yang harus dieksekusi berdasarkan perintah yang telah dituliskan oleh pengguna pada berkas crontab.

Format dari berkas crontab adalah *minute hour dom month weekday command*. Kelima kolom ini menginformasikan cron kapan harus menjalankan perintah yang telah dispesifikkkan pada beraks crontab.

Contoh spesifikasi waktu crontab *45 10 * * 1-5*, yang artinya 10.45 a.m., Senin sampai dengan Jumat. Petunjuk: jangan pernah menggunakan bintang (star) disetiap kolom, kecuali perintah tersebut harus dijalankan setiap menit, yang dapat digunakan untuk menjalankan skenario testing.

Perintah yang harus dieksekusi oleh cron adalah perintah baris **sh**. Untuk mengarahkan cron kepada perintah yang harus dieksekusi tersebut maka harus menggunakan path yang menuju berkas perintah sh tersebut, sebagai contoh:

0 4 * * Sun (/usr/bin/mysqlcheck -u maintenance -optimize -all-databases)

Entri ini akan menjalankan **mysqlcheck** program pemeliharaan database pada hari Minggu (Sundays) pukul 4:00 a.m.

20 1 * * * find /tmp -mtime +7 -type f -exec rm -f {} ;'

Entri ini akan menjalankan perintah **find** setiap pukul 1:20 setiap paginya. Perintah tersebut akan menghapus direktori **/tmp** yang tidak dimodifikasi selama 7 hari.

Jika cron job adalah sebuah skrip/program, pastikan dapat dieksekusi (executable) dengan menggunakan perintah **chmod +x**, jika tidak cron tidak akan dapat mengeksekusi skrip/program. Perintah cron yang dapat dituliskan untuk memanggil skrip/program sebagai contoh **bash -c ~/bin/myscript.sh**.

D.1.1 Pengelolaan Proses

Dengan menggunakan **crontab filename** maka menjadikan *filename* sebagai crontab. Perintah **crontab -e** digunakan untuk memodifikasi berkas crontab, dengan menggunakan perintah maka pengguna yang secara bersamaan membuka berkas dapat dihindari. Editor default adalah **vi**, tetapi anda dapat menggunakan editor **gvim** dan **gedit** (versi GUI). Untuk menampilkan seluruh daftar crontab yang ada di stdout (standard output) digunakan perintah **crontab -l**, sedangkan **crontab -r** digunakan untuk

menghapus seluruh berkas crontab.

Root dapat menspesifikasiusername sebagai argumen untuk memodifikasi atau menampilkan crontab dari user tersebut, sebagai contoh **crontab -r jsmith** akan menghapus crontab milik dari user jsmith atau **crontab -e jsmith** akan memodifikasi crontab milik jsmith. Selain itu, argumen lainnya dapat ditambahkan seperti username dan filename pada sebuah perintah, sehingga prefiks **-u** untuk menandai username harus digunakan. Misalnya, **crontab -u jsmith crontab.new**.

Jika crontab tidak diikuti dengan argumen, kebanyakan versi dari crontab akan mencoba untuk membaca crontab dari standard input (stdin). Jika terjebak dalam mode ini, maka jangan mencoba keluar dengan menggunakan **<Control-D>** karena perintah ini akan menghapus seluruh crontab, tetapi gunakanlah **<Control-C>**.

Distribusi Linux juga menyediakan entri crontab yang dibuatkan pada saat pre-install. berkas crontab ini sudah menyediakan skrip yang dapat dijalankan seperti skrip pada

/etc/cron.hourly, /etc/cron.daily, dan /etc/cron.weekly yang secara berurutan akandijalankan per jam, per hari, dan per minggu.

D.1.2 Kontrol Akses Cron

Pada Linux terdapat dua berkas untuk menentukan pengguna mana saja yang dapat membuat berkas crontab yaitu **/etc/cron.{allow, deny}**. Jika berkas **cron.allow** ada, maka pada berkas tersebut ditemukan daftar pengguna yang dapat membuat crontab. Tidak ada satu pun pengguna yang dapat menjalankan perintah crontab jika tidak terdaftar di berkas **cron.allow**, kemudian berkas **cron.deny** akan diperiksa. Jika pengguna tidak ditemukan pada kedua berkas, maka akan digunakan sistem default yaitu mengizinkan seluruh pengguna untuk menjalankan crontab atau membatasi crontab hanya dapat dijalankan oleh root.

Penting untuk dicatat bahwa kebanyakan pada sistem, kontrol akses diimplementasikanoleh crontab bukan oleh cron. Jika seorang pengguna dapat menyelinap untukmengakses sebuah berkas crontab, maka perintah cron dapat dijalankan untukmengeksekusi berkas crontab tersebut. Oleh karena itu, merupakan bagian yang sangat penting untuk memelihara kepemilikan (ownership) root terhadap direktori

/var/spool/cron dan **/var/cron/tabs**. Secara default sistem operasi telah mengatur hak akses ini dengan benar.

E. Tugas

1. Monitoring dan Pengaturan Proses

Proses mana yang berjalan pada sistem saya?

a. Perintah ps

i. Buka terminal dan jalankan perintah ps, tampilkan hasilnya.

```
[josa@localhost ~]$ ps
 PID TTY      TIME CMD
 5039 tty1    00:00:00 bash
 5082 tty1    00:00:00 ps
```

Gambar 25

ii. Buka terminal lain dan jalankan perintah sleep 1000 &, yang akan memulai proses tidur (sleeping process) pada proses background. Kemudian jalankan perintah ps dan tampilkan hasilnya.

```
[josa@localhost ~]$ sleep 1000 &
[1] 5089
```

Gambar 26

```
[josa@localhost ~]$ jobs
[1]+  Running                  sleep 1000 &
[josa@localhost ~]$ _
```

Gambar 27

iii. Pada terminal pertama jalankan perintah ps lagi. Ada yang perlu diperhatikan bahwa proses sleep tidak muncul, meskipun waktu 1000 detik belum kadaluwarsa.

```
[josa@localhost ~]$ ps
 PID TTY      TIME CMD
 5039 tty1    00:00:00 bash
 5089 tty1    00:00:00 sleep
 5105 tty1    00:00:00 ps
[josa@localhost ~]$ _
```

Gambar 28

iv. (Pertanyaan 1) Menurut anda mengapa terjadi demikian? (petunjuk: kolom TTY).

= ps menampilkan proses yang aktif yang sedang berjalan dan pada output ps, terdapat kolom tty yang dimana tty menampilkan nama file terminal yang terhubung ke input standar.

v. Untuk menampilkan seluruh proses pada terminal pertama dapat dijalankan dengan perintah ps -u. Perhatikan kolom PID, setiap proses memiliki ID yang unik diberikan oleh kernel. Sleep 1000 kurang tidak begitu berguna, sehingga perlu untuk diakhiri dengan cara mengirimkan signal ke proses sesuai dengan PID yang dituju.

```
[josa@localhost ~]$ ps -u
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
josa          5039  0.0  0.1 116172  2696  tty1      Ss  15:54  0:00 -bash
josa          6416  0.0  0.0 151064  1788  tty1      R+  17:06  0:00 ps -u
[1]+  Done                  sleep 1000
[josa@localhost ~]$
```

Gambar 29

vi. (Pertanyaan 2) Dalam kasus ini bagaimanakah cara untuk mengakhiri proses sleep tersebut? (petunjuk: signal dan PID).

= menggunakan command kill dan mengetahui PID dari proses yang akan diakhiri

vii. (Pertanyaan 3) Selain menggunakan perintah ps secara umum, perintah lain yang dapat digunakan adalah ps -e / ps -f / ps -ef. Tampilkan dan beri pembahasan dari hasil ketiga perintah tersebut.

=ps -e = Melihat proses yang tidak terkait dengan terminal

```
606 ?      00:00:00 kvm-irqfd-clean
639 ?      00:00:00 auditd
662 ?      00:00:00 rngd
663 ?      00:00:00 VGGAuthService
664 ?      00:00:27 vmtoolsd
666 ?      00:00:00 abrttd
667 ?      00:00:00 abrt-watch-log
669 ?      00:00:00 systemd-logind
673 ?      00:00:00 polkitd
680 ?      00:00:01 rsyslogd
681 ?      00:00:00 dbus-daemon
686 ?      00:00:00 gssproxy
702 ?      00:00:00 smartd
703 ?      00:00:00 lsmd
721 ?      00:00:00 ksmtuned
726 ?      00:00:00 firewalld
737 ?      00:00:01 NetworkManager
1052 ?      00:00:03 tuned
1055 ?      00:00:00 sshd
1060 ?      00:00:00 libvirtd
1067 ?      00:00:00 atd
1070 ?      00:00:00 crond
1345 ?      00:00:00 master
1348 ?      00:00:00 qmgr
1383 ?      00:00:00 dnsmasq
1387 ?      00:00:00 dnsmasq
4746 ?      00:00:00 kworker/u256:2
5032 ?      00:00:00 login
5039 tty1   00:00:00 bash
5657 ?      00:00:01 kworker/0:1
6093 ?      00:00:00 kworker/0:2
6349 ?      00:00:00 dhclient
6415 ?      00:00:00 pickup
6434 ?      00:00:00 kworker/0:0
6442 ?      00:00:00 sleep
6449 tty1   00:00:00 ps
[josa@localhost ~]$
```

Gambar 30

```

~ ps -f = Menampilkan proses berdasarkan field tertentu
[josa@localhost ~]$ ps -f
UID      PID  PPID  C STIME TTY          TIME CMD
josa      5039  5032  0 15:54 tty1        00:00:00 -bash
josa      6459  5039  0 17:10 tty1        00:00:00 ps -f
[josa@localhost ~]$

```

Gambar 31

```

root      639  1  0 12:11 ?    00:00:00 /sbin/auditd
root      662  1  0 12:11 ?    00:00:00 /sbin/rngd -f
root      663  1  0 12:11 ?    00:00:00 /usr/bin/UGhutService -s
root      664  1  0 12:11 ?    00:00:27 /usr/bin/vmtoolsd
root      666  1  0 12:11 ?    00:00:00 /usr/sbin/abrtd -d -s
root      667  1  0 12:11 ?    00:00:00 /usr/bin/abrt-watch-log -F BUG: WARNING: at WARNIN
root      669  1  0 12:11 ?    00:00:00 /usr/lib/systemd/systemd-logind
polkitd   673  1  0 12:11 ?    00:00:00 /usr/lib/polkit-1/polkitd --no-debug
root      680  1  0 12:11 ?    00:00:01 /usr/sbin/rsyslogd -n
dbus     681  1  0 12:11 ?    00:00:00 /bin/dbus-daemon --system --address=systemd: --nof
root      686  1  0 12:11 ?    00:00:00 /usr/sbin/gssproxy -D
root      782  1  0 12:11 ?    00:00:00 /usr/sbin/smard -n -q never
libstor+ 783  1  0 12:11 ?    00:00:00 /usr/bin/lsmad -d
root      721  1  0 12:11 ?    00:00:00 /bin/bash /usr/sbin/ksmtuned
root      726  1  0 12:11 ?    00:00:00 /usr/bin/python -Es /usr/sbin/firewalld --nofork -
root      737  1  0 12:11 ?    00:00:01 /usr/sbin/NetworkManager --no-daemon
root      1052 1  0 12:11 ?    00:00:03 /usr/bin/python -Es /usr/sbin/tuned -l -P
root      1055 1  0 12:11 ?    00:00:00 /usr/sbin/sshd -D
root      1060 1  0 12:11 ?    00:00:00 /usr/sbin/libvirtd
root      1067 1  0 12:11 ?    00:00:00 /usr/sbin/atd -f
root      1078 1  0 12:11 ?    00:00:00 /usr/sbin/crond -n
root      1345 1  0 12:11 ?    00:00:00 /usr/libexec/postfix/master -w
postfix  1348 1345 0 12:11 ?    00:00:00 qmgr -l -t unix -u
nobody   1363 1  0 12:11 ?    00:00:00 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dns
root      1387 1383 0 12:11 ?    00:00:00 /usr/sbin/dnsmasq --conf-file=/var/lib/libvirt/dns
root      4746 2  0 15:34 ?    00:00:00 [kworker/u256:2]
root      5032 1  0 15:54 ?    00:00:00 login -- josa
josa      5039 5032 0 15:54 tty1  00:00:00 -bash
root      5657 2  0 16:46 ?    00:00:01 [kworker/0:1]
root      6093 2  0 17:04 ?    00:00:00 [kworker/0:2]
root      6349 737 0 17:06 ?    00:00:00 /sbin/dhclient -d -q -sf /usr/libexec/nm-dhcp-help
postfix  6415 1345 0 17:06 ?    00:00:00 pickup -l -t unix -u
root      6434 2  0 17:09 ?    00:00:00 [kworker/0:0]
root      6450 2  0 17:10 ?    00:00:00 [kworker/0:3]
root      6458 721 0 17:10 ?    00:00:00 sleep 60
josa      6460 5039 0 17:11 tty1  00:00:00 ps -ef
[josa@localhost ~]$

```

~ps -ef = Menampilkan proses yang sedang berjalan saat ini

Gambar 32

- Buatlah dua pengguna pada sistem anda (anda bebas menentukan nama pengguna).

```

[root@localhost josa]# useradd josaa
[root@localhost josa]# passwd josaa
Changing password for user josaa.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost josa]#

```

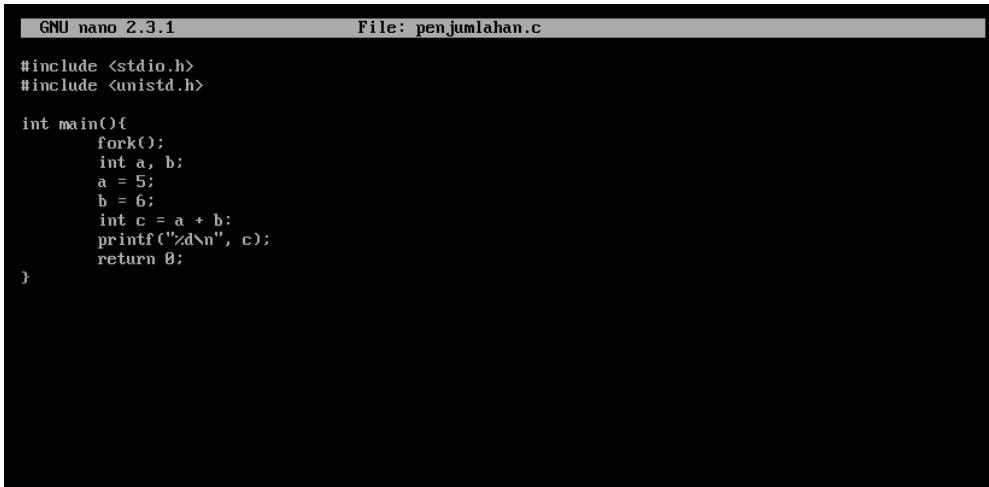
Gambar 33

- Perintah top dan htop
- Gunakanlah dua pengguna yang telah anda tambahkan di sistem pada bagian a.
- Berdasarkan pengetahuan anda pada matakuliah Sistem Operasi dan Sistem Programming, anda diharapkan untuk membuat program pengelolaan

proses menggunakan system call fork().

iii. Untuk masing-masing pengguna buatlah 2 program yang menggunakan system call fork() dengan ketentuan sebagai berikut:

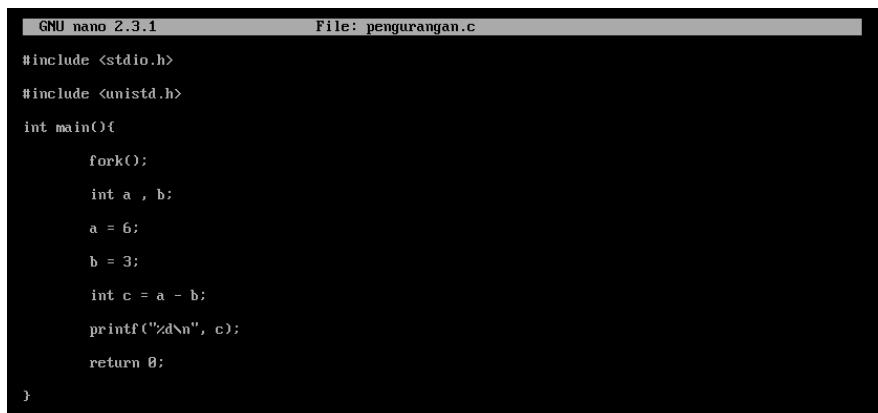
2. Program untuk pengguna pertama adalah penjumlahan dan pengurangan.



```
GNU nano 2.3.1          File: penjumlahan.c
#include <stdio.h>
#include <unistd.h>

int main(){
    fork();
    int a, b;
    a = 5;
    b = 6;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 34

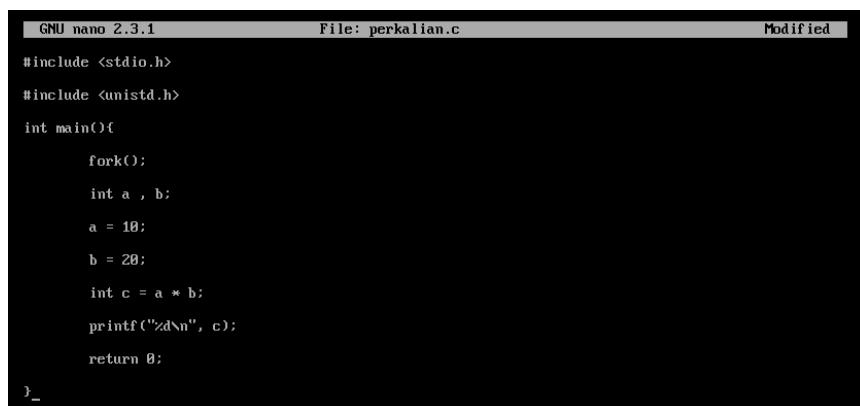


```
GNU nano 2.3.1          File: pengurangan.c
#include <stdio.h>
#include <unistd.h>

int main(){
    fork();
    int a , b;
    a = 6;
    b = 3;
    int c = a - b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 35

- Program untuk pengguna kedua adalah perkalian dan pembagian.



```
GNU nano 2.3.1          File: perkalian.c          Modified
#include <stdio.h>
#include <unistd.h>

int main(){
    fork();
    int a , b;
    a = 10;
    b = 20;
    int c = a * b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 36



```
GNU nano 2.3.1          File: pembagian.c

#include <stdio.h>
#include <unistd.h>
int main(){
    fork();
    int a , b;
    a = 200;
    b = 10;
    int c = a / b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 37

- **Program diletakkan di home folder masing-masing pengguna**
 - ii. Keempat program tersebut dibuatkan menjadi program zombie.
 - iii. Jalankan program yang telah ditentukan ke setiap pengguna di terminal masing-masing pengguna.
 - iv. Pada terminal utama jalankan perintah top, tampilkan dan berikan pembahasan hasilnya.
 - v. (Pertanyaan 1) Dalam kasus ini bagaimanakah cara untuk mengakhiri proses zombie tersebut?
 - vi. (Pertanyaan 2) Apakah anda menemukan perbedaan antara perintah ps dengan perintah top? Berikan penjelasan anda.
 - vii. Selain top terdapat perintah htop (open source software yang harus diinstall).
 - viii. Jalankan kembali program zombie yang telah anda ciptakan, kemudian jalankan perintah htop dan ps -ef.
 - ix. (Pertanyaan 3) Apakah perbedaan dari kedua perintah tersebut?
- b. Perintah nice dan renice
 - i. Buatlah sebuah program sederhana dengan nama prog_1.c, yang bertujuan untuk mencetak nilai dari 1 sampai dengan 50 dengan waktu sleep sepanjang 5 detik setiap 1 iterasi selesai. Jalankan program sehingga menghasilkan luaran seperti gambar berikut.
 - ii. Buka terminal yang baru, kemudian jalankan program top untuk melihat daftar proses yang sedang berjalan. Identifikasi nilai nice untuk proses prog_1.
 - iii. (Pertanyaan 1) Berapakah nilai nice? Mengapa demikian? Tampilkan hasil observasi anda.

iv. Hentikan proses dari prog_1. Kemudian jalankan program prog_1 kembali namun dengan menggunakan perintah nice dengan parameter prioriti 15.

v. Jalankan program top, kemudian identifikasi nilai nice untuk process proc_1.

vi. (Pertanyaan 2) Berapakah nilai nice? Mengapa demikian? Tampilkan hasil observasi anda.

vii. Buka terminal yang baru, kemudian dalam keadaan proses prog_1 masih berjalan, ubahlah nilai prioriti proses menggunakan perintah renice dengan nilai -20. Kemudian, perhatikan kembali program top yang sedang berjalan.

viii. (Pertanyaan 3) berapakah nilai nice? Mengapa demikian? Tampilkan hasil observasi anda.

c. Perintah Signal

i. Jalankanlah 3 program editor vi pada terminal yang berbeda. Kemudian buka terminal baru untuk menterminasi ketiga program tersebut.

ii. Sebelum menterminasi ketiga program tampilkan terlebih dahulu bahwa ketiga program tersebut adalah proses yang sedang berjalan pada sistem anda.

iii. Buka terminal yang baru kemudian terminasi seluruh proses vi.

iv. (Pertanyaan 1) Bagaimanakah cara untuk menterminasi seluruh proses vi? Tampilkan perintah yang anda gunakan dan hasil akhirnya.

v. Buatlah 2 akun pengguna dengan nama user_1 dan user_2.

vi. Jalankan 2 program apapun yang dieksekusi oleh masing-masing pengguna.

vii. (Pertanyaan 2) Bagaimanakah caranya untuk menterminasi seluruh proses yang dijalankan oleh masing-masing pengguna (petunjuk: pkill). Tampilkan perintah yang anda gunakan dan hasil akhirnya.

d. Perintah Strace

i. Jalankan sebuah program misalnya prog_1 yang telah anda buat sebelumnya.

ii. Kemudian gunakan perintah strace untuk menggali informasi lebih lengkap dari proses yang anda ekseskuasi pada poin i. Tampilkan perintah yang anda gunakan dan hasil akhirnya, kemudian berikan penjelasan anda terhadap hasil dari perintah strace.

3. Automasi menggunakan cron

a. Dari poin 1.b.iii., buatlah cron job yang akan menyalin berkas dari homefolder ke folder lainnya (anda dapat menentukan lokasi dan nama foldernya). Tambahkan 2 berkas lainnya dihome folder masing-masing

pengguna misalnya berkas .txt.

- b. File cron adalah unik untuk setiap pengguna dan hanya bisa dieksekusi oleh pengguna yang terkait saja dan juga root.
- c. Spesifikasi untuk cron-nya adalah cron dijalankan pada pukul 10:00 a.m. pada bulan 09, tahun 2021 setiap hari.
- d. Tampilkan seluruh proses, file crontab, dan juga hasil dari cron job.

BAB 3. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 1

A. Standard UNIX Access Control

Skema untuk kontrol akses pada UNIX memiliki beberapa peraturan yang digunakan juga pada banyak distribusi sistem operasi, sebagai berikut:

1. Objek (cth. File dan proses) mempunyai pemilik (owner). Pemilik objek memiliki kontrol terhadap objek yang mereka miliki.
2. Pengguna memiliki objek yang dibuat oleh pengguna tersebut.
3. Akun pengguna yang khusus (spesial) disebut sebagai “root” dapat berperan sebagai pemilik objek apapun.
4. Hanya root yang dapat melakukan aksi sensitif untuk seluruh proses administrasi.

System call tertentu misalnya `settimeofday` hanya bisa dijalankan oleh root. Pada implementasinya sistem akan melakukan pemeriksaan identitas dari pengguna dan menolak operasi jika pengguna tersebut bukanlah root.

A.1 AkunRoot

Akun root adalah pengguna administratif UNIX yang memiliki kuasa penuh pada sistem. Root juga dikenal sebagai akun superuser, meskipun nama sebenarnya adalah root. Akun root akan diberikan User Identifier (UID) 0. Dengan demikian, tidak akan ada pengguna lainnya yang dapat mencegah root untuk mengubah username-nya sendiri atau menambahkan pengguna lain yang juga memiliki UID 0. Meskipun, sebenarnya situasi ini merupakan ide yang buruk jika dipertimbangkan dari segi aspek keamanan sistem.

B. Pengaturan Akun Root

Akses root dibutuhkan pada seluruh administrasi sistem, dan juga merupakan titik penting untuk kemanan sistem. Pengaturan dari akun root merupakan keterampilan yang sangat penting.

B.1 Login Akun Root

Ketika sistem diakses oleh root maka hak akses secara penuh akan diberikan kepada root. Namun demikian, praktek ini merupakan praktek yang tidak baik. Ketika pengguna root login ke sistem, maka pengguna ini tidak akan meninggalkan catatan (record) terhadap seluruh operasi yang telah dijalankannya. Misalnya, ketika seorang pengguna login dengan menggunakan akun root dan melakukan perubahan pada sistem

pada pukul 04.00 a.m., tetapi setelahnya tidak mengingat apa perubahan yang telah dilakukannya pada sistem. Kasus lainnya juga ketika pengguna lain menggunakan akun root namun sebenarnya pengguna tersebut adalah pengguna yang tidak sah, maka akan sangat sulit sekali untuk melacak apa yang telah dilakukan oleh intruder terhadap sistem. Kekurangan lainnya adalah skenario log-in-as-root tidak meninggalkan catatan apapun mengenai siapa yang sebenarnya melakukan aksi terhadap sistem. Jika beberapa pengguna memiliki akses terhadap akun root, maka tidak akan dapat dilacak siapa yang menggunakan akun root dan kapan menggunakannya.

Untuk alasan keamanan sistem seperti yang dijelaskan di atas, kebanyakan sistem menonaktifkan penggunaan root pada terminal, antar jaringan jarak jauh, dimanapun kecuali pada konsol sistem. Meskipun demikian, root juga harus diberikan password dengan kualitas yang tinggi.

B.2 su: Pengganti identitas pengguna

Cara yang sedikit lebih baik untuk mengakses akun root adalah dengan menggunakan perintah su. Jika perintah su dipanggil tanpa ada argumen, maka perintah su akan meminta password root dan kemudian memulai sebuah shell root. Hak istimewa root akan tetap berlaku sampai shell diterminasi dengan mengetikkan <Control-D> atau perintah exit. Perintah su tidak akan mencatat perintah yang dieksekusi sebagai root, tetapi akan menciptakan sebuah entri log yang menyatakan siapa pengguna yang menjadi root dan kapan waktu aksesnya.

Perintah su juga dapat menggantikan identitas selain root. Terkadang, cara yang tersedia untuk mengulang kembali atau debug persoalan yang dihadapi pengguna adalah dengan menggunakan perintah su ke akun mereka sehingga lingkungan kerja terjadinya persoalan dapat diulang kembali.

Jika anda mengetahui password seorang pengguna, anda dapat mengakses akun pengguna tersebut secara langsung dengan mengeksekusi perintah su - *username*. Seperti menggunakan perintah su untuk root, anda akan dimintakan untuk mengetikkan password sesuai dengan username. Opsi dash (-) membuat perintah su menghasilkan shell dalam mode login.

B.3 sudo: su terbatas

Pada situasi tertentu sulit untuk memampukan seorang pengguna untuk melakukan task seperti backup tanpa memberikan kebebasan kepada pengguna tersebut untuk menjalankan sistem. Jika akun root digunakan oleh beberapa administrator, maka anda akan kesulitan untuk menemukan siapa yang menggunakan akun root dan apa yang mereka kerjakan dengan akun tersebut.

Solusi yang paling banyak digunakan untuk persoalan di atas adalah dengan menggunakan perintah sudo. Perintah ini direkomendasikan sebagai metode utama untuk mengakses akun root.

Sudo mengambil argumennya sebagai baris perintah untuk dieksekusi sebagai root. Sudo akan memeriksa file /etc/sudoers, yang berisi daftar pengguna yang berwenang untuk melakukam perintah sudo dan perintah lainnya yang diizinkan untuk dijalankan oleh pengguna tertentu. Jika perintah yang dicoba dieksekusi oleh seorang pengguna diizinkan, maka perintah sudo akan meminta password dari pengguna tersebut dan mengeksekusi perintah yang dimaksudkan.

Sudo akan menyimpan perintah yang dieksekusi oleh pengguna, dan pengguna yang menjalankan perintah tersebut, direktori yang digunakan dan waktu ketika perintah dijalankan. Informasi ini dicatat oleh syslog. Pada sistem operasi Centos, histori perintah yang dieksekusi oleh pengguna yang menjalankan sudo akan tercatat di /var/log/secure.

Contoh konfigurasi file sudoers yang dieredit dengan menjalankan perintah visudo. Tambahkan user atas nama thor.

```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##       user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
eka    ALL=(ALL)        ALL
thor   ALL=(ALL)        ALL
```

Gambar 38

Pada contoh berikut thor melakukan perintah sudo kemudian menjalankan perintah untuk menambahkan akun pengguna baru atas nama spiderman. Untuk mengetahui perintah apa saja yang dijalankan oleh thor dengan wewenang sebagai root dapat dilihat pada gambar di bawah.

```
[root@localhost ~]# cat /var/log/secure | grep thor | more_
```

Gambar 39

```
Sep 29 06:05:00 localhost login: LOGIN ON ttym1 BY thor
Sep 29 06:05:24 localhost sudo:      thor : TTY=ttym1 : PWD=/home/thor : USER=root : COMMAND=/bin/bash
Sep 29 06:06:38 localhost sudo:      thor : TTY=ttym1 : PWD=/home/thor : USER=root : COMMAND=/sbin/user
del -r thanos
Sep 29 06:08:04 localhost sudo:      thor : TTY=ttym1 : PWD=/home/thor : USER=root : COMMAND=/sbin/addu
ser spiderman
Sep 29 06:08:24 localhost sudo:      thor : TTY=ttym1 : PWD=/home/thor : USER=root : COMMAND=/bin/cat /
etc/shadow
Sep 29 06:08:44 localhost login: pam_unix(login:session): session closed for user thor
```

Gambar 40

E. Tugas

Pada skenario ini terdapat pengguna dan group pengguna sebagai berikut:

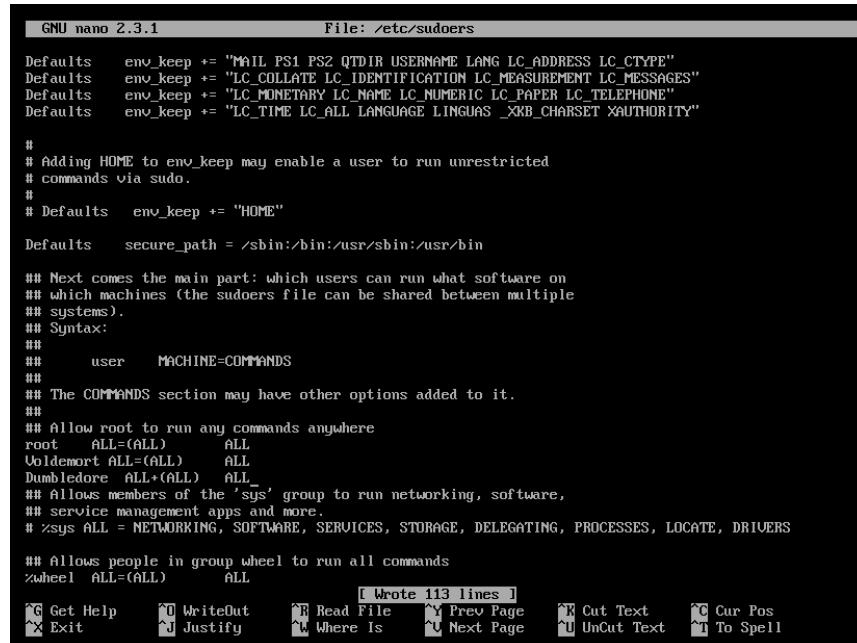
Table 2

Group: Master	Group: Master Bad
Dumbledore	Voldemort
Group: Dosen	Group: Dosen Bad
Snape	Lucius
Sirius	Bellatrix
Group: Siswa	Group: Siswa Bad
Harry	Malfoy
Hermione	Seamus
	Squirrel

Sebagai kondisi awal, Anda diperbolehkan untuk menggunakan akun root untuk melakukan operasi berikut:

1. Menciptakan pengguna Dumbledore dan Voldemort.
(lupa screenshoot)
2. Mendaftarkan kedua pengguna menjadi superuser dengan wewenang penuh pada sistem setara dengan root.

- Jalankan perintah: vi /etc/sudoers



```
GNU nano 2.3.1          File: /etc/sudoers

Defaults    env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults    env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults    env_keep += "HOME"

Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##       user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root      ALL=(ALL)        ALL
Voldemort ALL=(ALL)        ALL
Dumbledore ALL=(ALL)        ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
## zsys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
## Allows people in group wheel to run all commands
%wheel   ALL=(ALL)        ALL
[ Wrote 113 lines ]           ^G Get Help      ^O WriteOut     ^R Read File     ^Y Prev Page    ^K Cut Text      ^C Cur Pos
^X Exit        ^J Justify      ^W Where Is      ^U Next Page    ^U UnCut Text    ^T To Spell
```

Gambar 41

Setelah Dumbledore setara dengan root, maka dia dapat melakukan operasi berikut:

1. Dumbledore menciptakan pengguna pada group dosen yaitu Snape dan Sirius, lengkap dengan pengaturan password dan home directory dari kedua pengguna.

```
[Dumbledore@localhost root]$ sudo useradd Snape
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for Dumbledore:
[Dumbledore@localhost root]$ passwd Snape
passwd: Only root can specify a user name.
[Dumbledore@localhost root]$ sudo passwd Snape
Changing password for user Snape.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Dumbledore@localhost root]$ sudo adduser Sirius
[Dumbledore@localhost root]$ sudo passwd Sirius
Changing password for user Sirius.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Dumbledore@localhost root]$ _
```

Gambar 42

2. Mengelompokkan Snape dan Sirius pada group dosen.

```
[Dumbledore@localhost root]$ sudo groupadd Dosen_Good_Guys
[Dumbledore@localhost root]$ sudo usermod -a -G Dosen_Good_Guys Snape
[Dumbledore@localhost root]$ sudo usermod -a -G Dosen_Good_Guys Sirius
```

Gambar 43

3. Memberikan wewenang kepada mereka menjadi superuser yang hanya dapat melakukan penambahan akun pengguna lainnya.

```
## Command Aliases
## These are groups of related commands...
## Setting Dosen_Good_Guys
# Cmmd Alias TUGAS1 = /useradd, /bin/passwd, /sbin/groupadd, /sbin/usermod
%Dosen_Good_Guys      ALL=TUGAS1
```

Gambar 44

Setelah akun pengguna Snape dan Sirius tercipta, maka mereka dapat melakukan operasi berikut:

1. Sirius menciptakan akun pengguna pada group siswa yaitu Harry dan Hermione, lengkap dengan pengaturan password dan home directory dari kedua pengguna.

```
[root@localhost Sirius]# su -l Sirius
Last login: Fri Sep 23 15:44:03 WIB 2022 on tty1
[Sirius@localhost ~]$ sudo adduser -m -d /home/Harry Harry
[sudo] password for Sirius:
[Sirius@localhost ~]$ sudo passwd Harry
Changing password for user Harry.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Sirius@localhost ~]$ sudo adduser -m -d /home/Hermione Hermione
[Sirius@localhost ~]$ sudo passwd Hermione
Changing password for user Hermione.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Sirius@localhost ~]$
```

Gambar 45

2. Snape mengelompokkan Harry dan Hermione ke dalam group siswa.

```
[root@localhost Sirius]# su Snape
[Snape@localhost Sirius]$ sudo groupadd Siswa_Good_Guys

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for Snape:
[Snape@localhost Sirius]$ sudo usermod -a -G Siswa_Good_Guys Harry
[Snape@localhost Sirius]$ sudo usermod -a -G Siswa_Good_Guys Hermione
[Snape@localhost Sirius]$
```

Gambar 46

Pada sistem yang sama, Voldemort juga melakukan hal yang sama yaitu:

1. Menciptakan pengguna pada group dosen yaitu Lucius dan Bellatrix, lengkap dengan pengaturan password dan home directory dari kedua pengguna.

```
[Voldemort@localhost ~]$ sudo groupadd Dosen_Bad_Guys

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

[sudo] password for Voldemort:
[Voldemort@localhost ~]$ sudo useradd Lucius
[Voldemort@localhost ~]$ sudo passwd Lucius
Changing password for user Lucius.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Voldemort@localhost ~]$ sudo useradd Bellatrix
[Voldemort@localhost ~]$ sudo passwd Bellatrix
Changing password for user Bellatrix.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
Sorry, passwords do not match.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Voldemort@localhost ~]$ _
```

Gambar 47

2. Mengelompokkan Lucius dan Bellatrix pada group dosen.

```
[Voldemort@localhost ~]$
[Voldemort@localhost ~]$
[Voldemort@localhost ~]$ sudo usermod -a -G Dosen_Bad_Guys Lucius
[Voldemort@localhost ~]$
[Voldemort@localhost ~]$ sudo usermod -a -G Dosen_Bad_Guys Bellatrix
[Voldemort@localhost ~]$
```

Gambar 48

3. Memberikan wewenang kepada mereka menjadi superuser yang dapat menambah dan menghapus akun pengguna lainnya.

- Jalankan perintah sudo vi /etc/sudoers kembali, lalu isi dengan perintah berikut:

```
## Setting Dosen_Bad_Guys
# Cmnd Alias Tugas2 = /useradd, /bin/passwd, /sbin/groupadd, /sbin/usermod
%Dosen_Bad_Guys          ALL=TUGAS2
```

Gambar 49

Kemudian Lucius, meminta Bellatrix untuk melakukan operasi berikut:

1. Menciptakan akun Malfoy dan Seamus, lengkap dengan pengaturan password dan home directory dari kedua pengguna.

```
[root@localhost Sirius]# su Bellatrix
[Bellatrix@localhost Sirius]$
[Bellatrix@localhost ~]$ cd
[Bellatrix@localhost ~]$ sudo adduser -m -d /home/Malfoy Malfoy
[sudo] password for Bellatrix:
[Bellatrix@localhost ~]$ sudo pass Malfoy
sudo: pass: command not found
[Bellatrix@localhost ~]$ sudo passwd Malfoy
Changing password for user Malfoy.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Bellatrix@localhost ~]$ sudo adduser -m -d /home/Seamus Seamus

[Bellatrix@localhost ~]$
[Bellatrix@localhost ~]$ sudo adduser -m -d /home/Seamus Seamus

adduser: user 'Seamus' already exists
[Bellatrix@localhost ~]$
[Bellatrix@localhost ~]$ sudo passwd Seamus
Changing password for user Seamus.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[Bellatrix@localhost ~]$
```

Gambar 50

2. Mengelompokkan Malfoy dan Seamus pada group siswa.

```
[Bellatrix@localhost ~]$ sudo groupadd Siswa_Bad_Guys
[Bellatrix@localhost ~]$ sudo usermod -a -G Siswa_Bad_Guys Malfoy
usermod: user 'Malfoy' does not exist
[Bellatrix@localhost ~]$ sudo usermod -a -G Siswa_Bad_Guys Malfoy
[Bellatrix@localhost ~]$ sudo usermod -a -G Siswa_Bad_Guys Seamus
[Bellatrix@localhost ~]$ _
```

Gambar 51

3. Tim Voldemort melakukan kecurangan, dengan memberikan wewenang kepada Malfoy dan Seamus untuk dapat menghentikan seluruh proses yang dijalankan oleh pengguna Harry dan Hermione.

```
## Setting Siswa_Bad_Guys
# Cmnd Alias Tugas3 = /bin/kill, /bin/pkill
%Siswa_Bad_Guys ALL=(ALL:Siswa_Good_Guys) TUGAS3
```

Gambar 52

Lucius memberikan perintah kepada Malfoy dan Seamus untuk melakukan operasi berikut:

1. Malfoy harus menghentikan seluruh proses yang dijalankan oleh Harry.
2. Seamus harus menghentikan seluruh proses yang dijalankan oleh Hermione. Catatan: untuk melakukan instruksi 1 dan 2, anda dapat menjalankan program yang sudah anda buatkan pada praktikum mengenai proses untuk Harry dan Hermione.

- Menjalankan perintah vi pada user Harry
- Sudo vi berkas1.txt
- Menjalankan perintah vi pada user Harry.
- sudo vi berkas2.tx
- Hentikan seluruh proses berjalan pada kedua user

BAB 4. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 2

A. Filesystem

Tujuan dasar dari sebuah filesystem adalah untuk merepresentasikan dan mengatur sumberdaya penyimpanan sistem. Pada filesystem terdapat empat komponen utama, yaitu:

- a. **Namespace** – sebuah cara untuk memberi nama objek dan mengatur objek pada sebuah hirarki.
- b. **API** – sekumpulan system call untuk menavigasi dan memanipulasi objek.
- c. **Security model** – skema untuk memproteksi, menyembunyikan dan berbagi objek.
- d. **Implementasi** – software untuk mengikat model logis ke perangkat keras.

A.1 Pathnames

Filesystem ditunjukkan pada sebuah hirarki yang dimulai dari direktori / dan dilanjutkan kebawah melalui sejumlah subdirektori. / disebut juga sebagai direktori root. Pada sistem Linux, graphical user interfaces (GUI) direktori dirujuk sebagai folder. Folder dan direktori adalah benda yang sama; folder hanyalah bahasa yang sering digunakan oleh Windows dan macOS.

Daftar direktori yang harus dilalui untuk menemukan file tertentu ditambah dengan nama file tersebut akan membentuk sebuah pathname. Pathname dapat dikategorikan absolute seperti /tmp/foo atau relative seperti book4/filesystem. Pathname relative ditunjukkan dengan dimulai dari direktori saat ini (current directory). Setiap komponen dari sebuah pathname (direktori) harus memiliki nama dengan panjang tidak lebih dari 255 karakter.

A.2 Organization of the file tree

Pada filesystem root terdapat paling tidak direktori root dan sekumpulan file serta subdirektori. File yang mengandung kernel sistem operasi biasanya berada pada /boot.

/etc juga menjadi bagian dari filesistem root sebagai direktori untuk program sistem yang kritis dan file konfigurasi, /sbin dan /bin untuk utilitas penting, dan /tmp untuk file yang bersifat sementara, /dev untuk perangkat (devices), /lib untuk library berbagi, /usr untuk tempat program sistem yang standar tetapi bukan program yang kritis, /var untuk menyimpan file log, accounting information, dan berbagai item lainnya yang bertumbuh atau berubah dengan cepat dan yang berbeda-beda disetiap host.

Kedua

/usr dan /var harus tersedia untuk memampukan system dapat menjalankan mode multiuser. Pada table berikut merupakan daftar dari beberapa direktori standar yang penting.

Pathname	Contents
/bin	Core operating system commands
/boot	Boot loader, kernel, and files needed by the kernel
/compat	On FreeBSD, files and libraries for Linux binary compatibility
/dev	Device entries for disks, printers, pseudo-terminals, etc.
/etc	Critical startup and configuration files
/home	Default home directories for users
/lib	Libraries, shared libraries, and commands used by /bin and /sbin
/media	Mount points for filesystems on removable media
/mnt	Temporary mount points, mounts for removable media
/opt	Optional software packages (rarely used, for compatibility)
/proc	Information about all running processes
/root	Home directory of the superuser (sometimes just /)
/run	Rendezvous points for running programs (PIDs, sockets, etc.)
/sbin	Core operating system commands ^a
/srv	Files held for distribution through web or other servers
/sys	A plethora of different kernel interfaces (Linux)
/tmp	Temporary files that may disappear between reboots
/usr	Hierarchy of secondary files and commands
/usr/bin	Most commands and executable files
/usr/include	Header files for compiling C programs
/usr/lib	Libraries; also, support files for standard programs
/usr/local	Local software or configuration data; mirrors /usr
/usr/sbin	Less essential commands for administration and repair
/usr/share	Items that might be common to multiple systems
/usr/share/man	On-line manual pages
/usr/src	Source code for nonlocal software (not widely used)
/usr/tmp	More temporary space (preserved between reboots)
/var	System-specific data and a few configuration files
/var/adm	Varies: logs, setup records, strange administrative bits
/var/log	System log files
/var/run	Same function as /run; now often a symlink
/var/spool	Spooling (that is, storage) directories for printers, mail, etc.
/var/tmp	More temporary space (preserved between reboots)

- a. The distinguishing characteristic of /sbin was originally that its contents were statically linked and so had fewer dependencies on other parts of the system. These days, all binaries are dynamically linked and there is no real difference between /bin and /sbin.

Gambar 53

A.3 File Types

Pada kebanyakan filesystem mengimplementasikan tujuh tipe file, sebagai berikut:

- a. **Regular files**
- b. **Directories**
- c. **Character device files**
- d. **Block device files**

- e. Local domain sockets
- f. Named pipes (FIFOs)
- g. Symbolic links

Untuk menentukan tipe dari file dapat menggunakan perintah file. Perintah file akan memberitahu tipe standar dari file dan juga format umum yang digunakan pada regular file, seperti contoh berikut.

```
[root@localhost ~]# file /usr/include
/usr/include: directory
```

Gambar 54

```
[root@localhost ~]# file /etc/sysconfig/sshd
/etc/sysconfig/sshd: ASCII text
```

Gambar 55

Opsi lainnya untuk eksplorasi mengenai file dengan menggunakan perintah ls -ld. Flag -l menunjukkan detail informasi, dan flag -d hanya menampilkan informasi direktori daripada menampilkan seluruh isi dari direktori.

```
[root@localhost ~]# ls -l /home
total 0
drwx----- 2 desta      desta      62 Sep 14 04:59 desta
drwx----- 3 eka       eka      172 Sep 16 01:17 eka
drwx----- 4 ekastephani ekastephani 99 Mar  9 2021 ekastephani
drwx----- 2        1006     1007    62 Sep 29 05:52 harry
drwx----- 2 spiderman spiderman   62 Sep 29 06:08 spiderman
drwx----- 2 thor       thor      62 Sep 29 05:54 thanos
drwx----- 2 thor       thor      83 Sep 29 06:01 thor
drwx----- 2 user1     user1     83 Oct 30 2020 user1
drwx----- 2 user2     user2     62 Oct 30 2020 user2
drwx----- 2 vincent   vincent   83 Sep 14 05:09 vincent
```

Gambar 56

```
[root@localhost ~]# ls -ld /home
drwxr-xr-x. 12 root root 146 Sep 29 06:08 /home
```

Gambar 57

Karakter pertama dari luaran ls menunjukkan tipe. Seperti pada contoh diatas maka karakter pertama adalah “d” yang berarti /usr/include adalah sebuah direktori. Tipe file yang lainnya yang dihasilkan oleh perintah ls dapat dilihat pada tabel berikut.

File type	Symbol	Created by	Removed by
Regular file	-	editors, cp, etc.	rm
Directory	d	mkdir	rmdir, rm -r
Character device file	c	mknod	rm
Block device file	b	mknod	rm
Local domain socket	s	socket system call	rm
Named pipe	p	mknod	rm
Symbolic link	l	ln -s	rm

Gambar 58

Seperti yang ditunjukkan pada tabel diatas maka, perintah rm digunakan untuk menghapus file, sedangkan rmdir digunakan untuk menghapus direktori.

Regular Files

Regular files terdiri dari sekumpulan bytes; pada filesystems tidak terdapat struktur pada kontennya. File dalam bentuk teks, data, program yang dapat dieksekusi dan librari berbagi, semuanya disimpan dalam bentuk regular files.

Directories

Sebuah direktori berisi nama referensi yang merujuk ke file yang lain. Sebuah direktori diciptakan dengan menggunakan perintah mkdir dan menghapusnya dengan perintah rmdir (jika direktori kosong). Direktori yang tidak kosong dapat dihapus secara rekursif, beserta dengan seluruh isinya dengan menggunakan perintah rm -r.

Tanda khusus “.” Dan “..” merujuk kepada direktori itu sendiri dan direktori induknya, yang keduanya tidak dapat dihapus. Dikarenakan direktori root tidak memiliki direktori induk asli, maka path “/..” ekivalen dengan pat “/.” dan keduanya ekivalen dengan “/”.

Hard links

Sebuah nama file disimpan dalam induk direktorinya, bukan dengan file itu sendiri. Pada kenyataannya, lebih dari satu direktori dapat merujuk kepada sebuah file dalam satu waktu, dan referensinya dapat memiliki nama yang berbeda. Seperti ilusi bahwa sebuah file ada pada tempat yang berbeda dalam waktu yang sama.

Setiap file pada filesystem di Linux dimulai dari sebuah hard link tunggal. Link antara nama file dan data aktual disimpan pada filesystem. Filesystem memelihara sejumlah link yang merujuk pada setiap file yang dituju dan tidak akan melepaskan blok data file sampai link terakhirnya telah dihapus.

Hard link dapat diciptakan dengan menggunakan perintah ln dan menghapusnya dengan perintah rm. Cara mudah untuk menggunakan perintah ln adalah dengan mengingat cara menggunakan perintah cp. Perintah cp oldfile newfile akan menciptakan salinan dari oldfile yang disebut sebagai newfile, begitu juga dengan ln oldfile newfile menciptakan nama newfile sebagai referensi tambahan ke oldfile.

Implementasi hard link, dapat dilakukan sebagai berikut. Sebuah file dengan nama hard_link_test diciptakan pada direktori /home/eka/link, isi dari file tersebut adalah “This is a demo of hard link.”. Dengan demikian, akan diciptakan sebuah hard link tunggal antara nama file hard_link_test dengan isi file “This is a demo of hard link.”. Terdapat 1 link pada kasus ini.

```
[eka@localhost link]$ ls -l  
total 4  
-rw-rw-r-- 1 eka eka 29 Oct 26 23:50 hard_link_test
```

Gambar 59

Berikutnya, hard link yang baru akan diciptakan pada direktori /home/eka/tmp menuju ke file yang sama, maka perintah yang dijalankan sebagai berikut.

```
[eka@localhost link]$ ln /home/eka/link/hard_link_test /home/eka/tmp/hard_link_test_new
```

Gambar 60

Maka, ketika filesystem diperlihatkan akan ditampilkan dua hard link.

```
[eka@localhost link]$ ls -l hard_link_test /home/eka/tmp/hard_link_test_new  
-rw-rw-r-- 2 eka eka 29 Oct 26 23:50 hard_link_test  
-rw-rw-r-- 2 eka eka 29 Oct 26 23:50 /home/eka/tmp/hard_link_test_new
```

Gambar 61

Perbedaan yang utama adalah pada nama filenya. Jumlah link juga berubah menjadi 2. Jika isi file yang baru ditampilkan dengan menggunakan perintah cat maka, akan ditampilkan data originalnya.

```
[eka@localhost link]$ cat /home/eka/tmp/hard_link_test_new  
This is a demo of hard link.
```

Gambar 62

Ketika perubahan dilakukan pada satu nama file, maka file yang lain akan mencerminkan perubahan tersebut. Permission, jumlah link, ownership, timestamp, dan isi file akan sama persis. Jika data yang origin dihapus, maka data akan tetap ada pada hard link yang kedua. Data akan dihapus dari drive ketika, seluruh link ke data telah dihapus. Jika terdapat dua file dengan properti yang identik, tetapi belum dapat dipastikan apakah kedua file tersebut adalah hard link, maka perintah yang digunakan untuk konfirmasi adalah ls -i yang akan menampilkan nomor inode dari kedua file tersebut. File yang terhubung melalui hard-link akan berbagi nomor inode yang sama. Pada contoh dibawah, nomor inode kedua file adalah 1572936, yang berarti kedua file ini adalah data yang identik atau terhubung melalui hard link.

```
[eka@localhost link]$ ls -li hard_link_test /home/eka/tmp/hard_link_test_new  
1572936 -rw-rw-r-- 2 eka eka 29 Oct 26 23:50 hard_link_test  
1572936 -rw-rw-r-- 2 eka eka 29 Oct 26 23:50 /home/eka/tmp/hard_link_test_new
```

Gambar 63

Soft links/Symbolic links

Soft links atau symbolc links bertujuan untuk menciptakan link dari non-reguler ke file reguler. Sesuai dengan definisinya, sebuah soft link bukan merupakan sebuah file standar, tetapi sebuah file khusus yang menunjuk ke file yang sudah ada.

Sebagai contoh sebuah file diciptakan soft_link_test diciptakan pada direktori /home/eka/link, isi dari file tersebut adalah "This is a demo of soft link." .

Berikutnya, soft link yang baru akan diciptakan pada direktori /tmp menuju ke file

yangsama, maka perintah yang dijalankan menggunakan perintah ln -s sebagai berikut.

```
[eka@localhost link]$ ln -s /home/eka/link/soft_link_test /tmp/soft_link_test_new
```

Gambar 64

Maka, ketika filesystem diperlihatkan akan ditampilkan sebuah soft link.

```
[eka@localhost link]$ ls -l soft_link_test /tmp/soft_link_test_new
-rw-rw-r-- 1 eka eka 29 Oct 27 00:55 soft_link_test
lrwxrwxrwx 1 eka eka 29 Oct 27 00:56 /tmp/soft_link_test_new -> /home/eka/link/soft_link_test
```

Gambar 65

Berikutnya jika isi data original dari file soft_link_test diubah, maka soft link juga akan berubah. Seperti contoh isi dari file soft_link_test ditambah “Try to discover the difference between hard link and soft link.”, maka isi dari soft link juga akan berubah, seperti berikut.

```
[eka@localhost link]$ cat /tmp/soft_link_test_new
This is a demo of soft link.
Try to discover the difference between hard link and soft link.
```

Gambar 66

Hal utama yang perlu untuk diperhatikan pada saat menggunakan soft link adalah data hilang dan kebingungan data. Jika file origin dihapus, maka soft link akan rusak.

```
[eka@localhost link]$ cat /tmp/soft_link_test_new
cat: /tmp/soft_link_test_new: No such file or directory
```

Gambar 67

Situasi ini akan disebut sebagai dangling soft link. Jika anda membuat file baru dengan nama yang sama seperti aslinya, maka akan terdapat kebingungan apabila isi file tidak lagi sama seperti yang telah diberikan soft link sebelumnya.

```
[eka@localhost link]$ vi soft_link_test
[eka@localhost link]$ cat /tmp/soft_link_test_new
This a new soft link demo after removed the first one.
```

Gambar 68

B File Attributes The permission bits

Pada model filesystem yang ada di Linux, setiap file memiliki 9 bit permission yang terdiri atas 3 bagian yang akan mengontrol siapa yang dapat melakukan read, write, dan execute file. Setiap bagian akan terdiri dari 3-digit yang nilai numeriknya bisa dari rentang 0-7. Bagian pertama untuk owner, bagian kedua untuk group, dan bagian ketiga untuk lainnya (others). Setiap bagian diberikan notasi yaitu r, w dan x dengan nilai r = 4, w = 2 dan x = 1. Permission untuk sebuah file merupakan gabungan dari owner, group dan others. Sebagai contoh, jika permission untuk owner sebuah direktori diberikan read write dan execution, group dan other diberikan hanya bisa read dan execution, maka nilai numeriknya sebagai berikut:

- **Owner: $rwx = 4 + 2 + 1 = 7$**
- **Group: $r-x = 4 + 0 + 1 = 5$**
- **Other: $r-x = 4 + 0 + 1 = 5$**

Chmod: change permission

Perintah chmod ditujukan untuk mengubah permission pada sebuah file. Hanya pemiliki (owner) dari sebuah file dan superuser yang dapat mengubah permission sebuah file. Argumen pertama yang digunakan adalah perintah chmod kemudian permission yang spesifik selanjutnya nama file.

```
[eka@localhost link]$ chmod 755 hard_link_test
```

Gambar 69

Pada tabel berikut diberikan delapan ilustrasi kombinasi untuk set tiga bit r,w,dan x.

Permission encoding for chmod

Octal	Binary	Perms	Octal	Binary	Perms
0	000	---	4	100	r--
1	001	--x	5	101	r-x
2	010	-w-	6	110	rW-
3	011	-wx	7	111	rwx

Gambar 70

Perintah chmod juga dapat dikombinasikan dengan menambahkan owner (o), group(g), dan other (o). Pada tabel berikut diberikan contoh sintaks mnemonic chmod.

Examples of chmod's mnemonic syntax

Spec	Meaning
u+w	Adds write permission for the owner of the file
ug=rw,o=r	Gives r/w permission to owner and group, and read permission to others
a-x	Removes execute permission for all categories (owner/group/other)
ug=srx,o=	Makes setuid/setgid and gives r/x permission to only owner and group
g=u	Makes the group permissions be the same as the owner permissions

Gambar 71

Contoh perintah yang dapat digunakan sebagai berikut.

```
[eka@localhost link]$ chmod g+w hard_link_test
```

Gambar 72

Perintah chmod juga dapat dieksekusi secara rekursif dengan menggunakan opsi -R untuk memperbarui permission file dalam sebuah direktori, sebagai contoh berikut.

```
[eka@localhost ~]$ chmod -R g+w link/
```

Gambar 73

Chown and chgrp: change ownership and group

Perintah chown digunakan untuk mengubah kepemilikan file (file ownership) dan chgrp merupakan perintah untuk mengubah kepemilikan group (group ownership).

Contoh pengubahan kepemilikan file seperti berikut.

```
[eka@localhost ~]$ sudo chown ekastephani matt
```

Gambar 74

Perubahan kepemilikan file seperti berikut.

```
[eka@localhost ~]$ ls -l
total 924
-rw-rw-r-- 1 eka          eka    131 Sep 16 00:20 1
-rwxrwxr-x 1 eka          eka   8568 Sep 16 00:25 hello
-rw-rw-r-- 1 eka          eka    156 Sep 16 00:25 hello.c
drwxrwxr-x 2 eka          eka     50 Oct 27 01:02 link
drwxrwxr-x 2 ekastephani eka      6 Oct 27 05:47 matt
```

Gambar 75

Contoh pengubahan kepemilikian file group seperti berikut.

```
[root@localhost eka]# chgrp staff matt
```

Gambar 76

Perubahan kepemilikan group file seperti berikut

```
[root@localhost eka]# ls -l
total 924
-rw-rw-r-- 1 eka      eka      131 Sep 16 00:20 1
-rwxrwxr-x 1 eka      eka     8568 Sep 16 00:25 hello
-rw-rw-r-- 1 eka      eka      156 Sep 16 00:25 hello.c
drwxrwxr-x 2 eka      eka      50 Oct 27 01:02 link
drwxrwxr-x 2 ekastephani staff    6 Oct 27 05:47 matt
```

Gambar 77

B.1 ACL

ACL menyediakan mekanisme permission yang lebih fleksibel pada filesystem. Secara tradisional objek file pada Linux diasosiasikan dengan tiga set permission yaitu read (r), write (w), dan execute (x) untuk tiga jenis grup pengguna file yaitu owner, group dan other. Namun demikian, pada situasi tertentu permission file secara tradisional tidak cukup untuk digunakan, sehingga dibutuhkan ACL. Dengan menggunakan ACL, permission dapat diberikan kepada user atau group secara individu meskipun tidak sesuai dengan owner atau owning group.

Struktur Entri ACL

Pada dasarnya ACL dapat dibagi menjadi dua kelas yaitu:

- Minimum ACL yang mengatur entri untuk tipe owner, owning group, dan other yang sesuai dengan bit permission secara konvensional untuk file dan direktori.**
- Extended ACL harus mengandung sebuah entri mask dan entri untuk tipe named user dan named group.**

Tabel berikut menyediakan ringkasan singkat mengenai tipe ACL entri yang beragam.

Type	Text Form
owner	user::rwx
named user	user:name:rwx
owning group	group::rwx
named group	group:name:rwx
mask	mask::rwx
other	other::rwx

Gambar 78

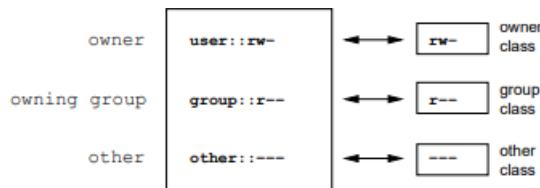
ACL Entries dan File mode permission bits

Gambar pada penjelasan berikut mengilustrasikan minimum ACL dan extended ACL. Pada gambar terdapat tiga blok, blok yang berada pada sebelah kiri menunjukkan spesifikasi dari entri ACL, blok pada bagian tengah menunjukkan contoh ACL, dan blok pada bagian kanan menunjukkan permission bit yang ditampilkan dengan perintah ls -l.

Pada kedua kelas ACL tersebut, permission dari owner class dipetakan kepada entri ACL yaitu owner. Pemetaan dari permission other class ke entri ACL juga konstan. Namun

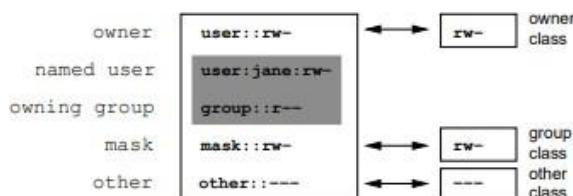
demikian, pemetaan permission group class beragam seperti:

- a. Pada kasus minimum ACL – tanpa mask – permission group class dipetakan ke ACL owning group, yang ditunjukkan pada gambar berikut.



Gambar 79

- b. Pada kasus extended ACL – dengan mask- permission group class dipetakan ke entri mask, yang ditunjukkan pada gambar berikut.



Gambar 80

Sebuah direktori dengan akses ACL

Penanganan akses dari ACL didemonstrasikan pada tiga langkah seperti contoh berikut:

- Menciptakan sebuah objek file system (dalam kasus ini adalah direktori)
- Memodifikasi ACL
- Menggunakan mask

Perhatikan langkah berikut.

- sebelum menciptakan direktori, gunakan perintah umask untuk menentukan akses permission yang mana yang harus di-masked.

```
[eka@localhost reymond]$ umask 027
```

Gambar 81

Owner akan memiliki akses dengan permission (read, write, execute) (0) dan akses write di-nonaktifkan untuk owning group (2). Seluruh akses oleh other user akan ditolak (7). Nilai dari umask dapat dibaca sebagai bit mask.

```
[eka@localhost reymond]$ mkdir mydir
```

Gambar 82

setelah menciptakan direktori mydir dengan permission yang diset menggunakan umask, gunakan perintah ls -dl untuk memeriksa apakah seluruh permission telah ditetapkan dengan benar.

```
[eka@localhost reymond]$ ls -dl mydir
drwxr-x--- 2 eka eka 6 Oct 28 04:43 mydir
```

Gambar 83

2. Periksa status awal dari ACL dan sisipkan user baru dan sebuah group.

```
[eka@localhost reymond]$ getfacl mydir
# file: mydir
# owner: eka
# group: eka
user::rwx
group::r-x
other::---
```

Gambar 84

```
[eka@localhost reymond]$ sudo chgrp staff mydir
[eka@localhost reymond]$ getfacl mydir
# file: mydir
# owner: eka
# group: staff
user::rwx
group::r-x
other::---
```

Gambar 85

Luaran dari perintah getfacl merefleksikan pemetaan bit permission dan entri ACL yang telah dijelaskan pada bagian ACL Entries dan File Mode Permission Bits sebelumnya. Pada kasus ini, yang ditampilkan oleh perintah getfacl adalah minimum ACL, perintah getfacl belum dapat menghasilkan informasi yang tidak dapat diperoleh dengan menggunakan perintah ls.

Oleh karena itu untuk pertama sekali perlu dilakukan modifikasi ACL untuk akses permission read, write, dan execution untuk user tambahan, sebagai contoh untuk user jane dan tambahan group djungle.

```
[eka@localhost reymond]$ setfacl -m user:thor:rwx,group:djungle:rwx mydir
[eka@localhost reymond]$ getfacl mydir
# file: mydir
# owner: eka
# group: staff
user::rwx
user:thor:rwx
group::r-x
group:djungle:rwx
mask::rwx
other::---
```

Gambar 86

B. Tugas

Dengan menggunakan panduan praktikum yang telah dijelaskan pada bagian A.1 sampai dengan A.5, lakukanlah pengaturan atau konfigurasi pada server Anda sesuai dengan skenario berikut. Anda harus menyediakan laporan dengan menunjukkan desain kerja Anda serta hasil screenshoot dari konfigurasi yang Anda terapkan serta konfirmasi hasil kebenaran konfigurasi.

Skenario

Anda seorang sistem administrator dari sebuah universitas yang bernama Beyond Imagination. Tugas Anda adalah untuk mendesain dan mengonfigurasi sebuah file server berbasis Linux untuk menentukan akses terhadap objek (file atau direktori) sesuai dengan ketentuan yang telah ditetapkan bersama oleh jajaran universitas. Pada file server universitas akan terdapat beberapa direktori kerja untuk masing-masing unit kerja, dan tugas Anda adalah menciptakan folder sesuai dengan hierarki direktori kerja dan menetapkan hak akses direktori kerja sesuai dengan yang telah ditetapkan.

Universitas Beyond Imagination memiliki struktur organisasi yang terdiri dari Rektor sebagai kepala universitas tertinggi yang memiliki tiga wakil rektor yaitu Wakil Rektor 1 (WR1), Wakil Rektor 2 (WR2), Wakil Rektor 3 (WR3). Setiap wakil rektor memiliki unit yang dipimpin. WR1 memimpin unit Kemahasiswaan dan Keasramaan, WR2 memimpin unit SDM dan Keuangan, WR3 memimpin unit Promosi dan Kerjasama.

Untuk unit yang berada dibawah setiap WR akan mengelola kegiatan yang berbeda-beda. Pada WR1 untuk unit Kemahasiswaan akan mengelola bagian kerja terkait Pembinaan Karakter dan Kegiatan Mahasiswa, sedangkan untuk unit Keasramaan akan mengelola bagian kerja yang berkaitan dengan Pengalokasian Asrama dan Kegiatan Asrama. Pada WR2 untuk unit SDM akan mengelola bagian kerja yang berkaitan dengan Artefak Dosen dan Artefak Staf, sedangkan unit Keuangan akan mengelola bagian kerja terkait dengan RKA Unit dan Bursar Mahasiswa. Pada WR3 untuk unit Promosi akan mengelola bagian kerja mengenai Strategi Promosi, sedangkan unit Kerjasama akan mengelola bagian kerja terkait dengan Kerjasama.

Untuk melancarkan kerja, setiap unit memerlukan direktori kerja. Oleh karena itu, Rektor akan memiliki direktori kerja sendiri, unit WR1, WR2 dan WR3 akan memiliki direktori kerja dengan masing-masing WR sebagai direktori induknya dengan subdirektori sesuai dengan nama masing-masing unit yang berada dibawah kepemimpinan masing-masing WR tersebut. Setiap unit yang berada dibawah WR akan memiliki direktori sesuai dengan bagian kerja dari masing-masing unit.

Akses terhadap untuk masing-masing direktori kerja unit perlu diatur sesuai dengan ketentuan yang ditetapkan. Sebelum hak akses ditetapkan ke setiap direktori kerja, diperlukan adanya pembentukan grup yang akan mengelompokkan seluruh anggota

unit. Nama grup harus disesuaikan dengan nama setiap unit kerja. Tabel berikut menampilkan data unit kerja dan pengguna pada setiap unit.

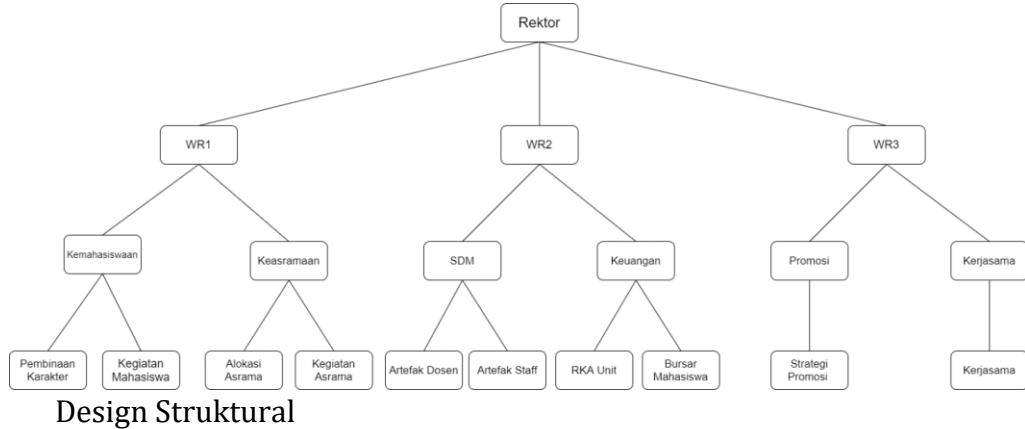
Table 3

Unit	Anggota
Rektor	Downey
WR1	Scarlett & Johhanson
Kemahasiswaan	Ruffalo & Mark
Keasramaan	Hermsworth & Chris
WR2	Evans & Chris
SDM	Hiddleston & Tom
Keuangan	Witwickly & Sam
WR3	Renner & Jeremy
Promosi	Fox & Meghan
Kerjasama	Mackie & Anthony

Khusus untuk anggota WR1, WR2 dan WR3 masing-masing WR memiliki sekretaris. Secara berurutan, nama sekretaris untuk setiap WR adalah Johhanson, Chris dan Jeremy.

Setelah setiap pengguna ditetapkan ke group unit masing-masing, ditetapkan mengenai kepemilikan dari setiap direktori kerja. Rektor akan sebagai pemilik mutlak direktori kerjanya, begitu juga dengan WR1, WR2 dan WR3 akan sebagai pemilik mutlak direktori kerjanya beserta dengan subdirektori didalamnya. Pada setiap direktori WR masing-masing WR dan sekretaris yang terkait akan memilki direktori kerja pribadi. Rektor dan seluruh WR beserta sekretarisnya akan memiliki hak akses penuh ke direktori kerja masing-masing.

Selanjutnya, direktori kerja akan dapat diakses secara penuh oleh setiap grup unit yang berada dibawah masing-masing WR. Anggota unit yang berada dari luar unit yang dipimpin oleh masing-masing WR, tidak diperbolehkan mengakses direktori kerja. Kemudian, untuk masing-masing direktori kerja yang berada dibawah pengelolaan masing-masing WR, ditetapkan bahwa hanya grup unit yang bersangkutan saja yang dapat mengakses direktori kerja tersebut. Unit lain, meskipun berada dibawah kepemimpinan WR yang sama, tidak boleh saling mengakses direktori kerja unit tersebut.



Gambar 87

Namun, terdapat beberapa ketentuan khusus yang perlu diperhatikan, sebagai berikut:

1. **Direktori kerja Rektor tidak dapat diakses oleh siapapun kecuali oleh Rektorschendiri.**
2. **WR mendapatkan hak akses penuh untuk folder sekretaris, tetapi sekretaris tidak dapat mengakses direktori kerja pribadi WR.**
3. **Dikarenakan, bagian kerja kemahasiswaan sering berkaitan dengan bagian keasramaan, maka khusus kepada ketua pembina asrama diberikan akses untuk membaca file (buatkan contoh file program_bina_karakter.txt) yang berada pada bagian kerja pembinaan karakter.**
4. **Begitu juga dengan bagian kerja RKA Unit harus dapat diakses oleh seluruh unit dengan hak akses penuh. Selanjutnya, dikarenakan WR1 juga memerlukan informasi strategi promosi dan juga kerjasama yang dimiliki oleh universitas, maka khusus untuk WR1 (tidak termasuk sekretaris) diberikan hak akses untuk membaca file yang berada pada direktori kerja bagian promosi dan kerjasama.**
5. **Kondisi berikutnya adalah, khusus untuk bagian kerja SDM sekretaris dari WR2 tidak diberikan hak akses sama sekali.**
6. **Khusus untuk unit keuangan bagian kerja Bursar Mahasiswa, Sam dan sekretaris WR2 tidak mendapatkan hak akses sama sekali pada direktori tersebut.**
7. **Dikarenakan masing-masing WR harus membutuhkan akses cepat untuk beberapa informasi tertentu maka dibutuhkan hard link atau soft link yang ada pada direktori kerja pribadi merujuk ke beberapa file berikut. Untuk WR1 dibutuhkan link ke file program_bina_karakter.txt pada direktori kerja pembinaan karakter, WR2 dibutuhkan link ke file RKA_unit_kemahasiswaan.txt pada direktori kerja RKA Unit, dan pada WR3 dibutuhkan link ke file kerjasama_dalam_dan_luar_negri.txt yang berada pada direktori kerja Kerjasama.**

Jawab:

1. **Membuat direktori kampus**

2. Kemudian masuk kedalam direktori dan membuat direktori Rektor, WR1, WR2, WR3.

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir Rektor | mkdir WR1 | mkdir WR2  
[root@serveransible Universitas_Beyond_Imagination]#
```

Gambar 88

3. Kemudian di dalam direktori WR1 kita membuat direktori Kemahasiswaan dan Keasramaan

```
[root@serveransible WR1]# mkdir Kemahasiswaan | mkdir Keasramaan  
[root@serveransible WR1]#
```

Gambar 89

4. Kemudian kita keluar dari direktori WR1 dan masuk ke direktori WR2 dan membuat direktori SDM dan Keuangan.

```
[root@serveransible Universitas_Beyond_Imagination]# cd WR2  
[root@serveransible WR2]# mkdir SDM | mkdir Keuangan  
[root@serveransible WR2]#
```

Gambar 90

5. Kemudian kita keluar dari WR2 dan masuk ke WR3 dan membuat direktori Promosi dan Kerjasama

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir WR3  
[root@serveransible Universitas_Beyond_Imagination]# cd WR3  
[root@serveransible WR3]# mkdir Promosi | mkdir Kerjasama  
[root@serveransible WR3]#
```

Gambar 91

6. Kemudian kita kembali ke direktori WR1 dan masuk ke direktori Kemahasiswaan dan Keasramaan dan membuat direktori Bina_Karakter dan Kegiatan_mahasiswa dalam direktori Kemahasiswaan dan direktori Alokasi_asrama dan Kegiatan_asrama di dalam Keasramaan

```
[root@serveransible Universitas_Beyond_Imagination]# cd WR1  
[root@serveransible WR1]# cd Kemahasiswaan  
[root@serveransible Kemahasiswaan]# mkdir Bina_karakter | mkdir Kegiatan_mahasiswa  
[root@serveransible Kemahasiswaan]# cd ..  
[root@serveransible WR1]# cd keasramaan  
bash: cd: keasramaan: No such file or directory  
[root@serveransible WR1]# cd Keasramaan  
[root@serveransible Keasramaan]# mkdir Alokasi_asrama | mkdir Kegiatan_asrama  
[root@serveransible Keasramaan]#
```

Gambar 92

7. Kemudian kita masuk ke direktori WR2 dan masuk ke direktori SDM dan Keuangan. didalam direktori SDM kita membuat 2 direktori baru yaitu artefak_dosen dan artefak_staff. di dalam direktori Keuangan kita membuat 2 direktori baru yaitu RKA_unit dan Bursar_mahasiswa.

```
[root@serveransible WR1]# cd ..  
[root@serveransible Universitas_Beyond_Imagination]# cd WR2  
[root@serveransible WR2]# cd SDM  
[root@serveransible SDM]# mkdir artefak_dosen | mkdir artefak_staff  
[root@serveransible SDM]# cd ..  
[root@serveransible WR2]# cd Keuangan  
[root@serveransible Keuangan]# mkdir RKA_unit | mkdir Bursar_mahasiswa
```

Gambar 93

8. Kemudian kita masuk ke WR3 dan masuk ke direktori Promosi dan Kerjasama. didalam direktori Promosi kita membuat direktori Strategi_promosi dan didalam direktori kerjasama kita membuat direktori kerjasama

```
[root@serveransible ~]# cd ..  
[root@serveransible Universitas_Beyond_Imagination]# cd WR3  
[root@serveransible WR3]# cd Promosi  
[root@serveransible Promosi]# mkdir Strategi_promosi  
[root@serveransible Promosi]# cd..  
bash: cd..: command not found...  
[root@serveransible Promosi]# cd ..  
[root@serveransible WR3]# cd Kerjasama  
[root@serveransible Kerjasama]# mkdir Kerjasama  
[root@serveransible Kerjasama]# █
```

Gambar 94

9. Sekarang kita dapat membuat grup yang sesuai dengan nama unit

```
[root@serveransible ~]# cd  
[root@serveransible ~]# groupadd Rektor | groupadd WR1 | groupadd WR2 | groupadd WR3  
[root@serveransible ~]# groupadd Kemahasiswaan | groupadd Keasramaan  
[root@serveransible ~]# groupadd SDM | groupadd Keuangan  
[root@serveransible ~]# groupadd Promosi | groupadd Kerjasama  
[root@serveransible ~]# █
```

Gambar 95

10. Kemudian kita membuat user sesuai dengan ketentuan.

```
[root@serveransible ~]# useradd -m -d /home/Downey Downey  
[root@serveransible ~]# passwd Downey  
Changing password for user Downey.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Scarlett Scarlett  
[root@serveransible ~]# passwd Scarlett  
Changing password for user Scarlett.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Johhanson Johhanson  
[root@serveransible ~]# passwd Johhanson  
Changing password for user Johhanson.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Ruffalo Ruffalo  
[root@serveransible ~]# passwd Ruffalo  
Changing password for user Ruffalo.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Mark Mark  
[root@serveransible ~]# passwd Mark  
Changing password for user Mark.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]#
```

Gambar 96

```
[root@serveransible ~]# useradd -m -d /home/Hermsworth Hermsworth
[root@serveransible ~]# passwd Hermsworth
Changing password for user Hermsworth.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Chris Chris
[root@serveransible ~]# passwd Chris
Changing password for user Chris.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Evans Evans
[root@serveransible ~]# passwd Evans
Changing password for user Evans.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Hiddleston Hiddleston
[root@serveransible ~]# passwd Hiddleston
Changing password for user Hiddleston.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Tom Tom
[root@serveransible ~]# passwd Tom
Changing password for user Tom.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

Gambar 97

```
[root@serveransible ~]# useradd -m -d /home/Witwicky Witwicky
[root@serveransible ~]# passwd Witwicky
Changing password for user Witwicky.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Sam Sam
[root@serveransible ~]# passwd Sam
Changing password for user Sam.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Renner Renner
[root@serveransible ~]# passwd Renner
Changing password for user Renner.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Jeremy Jeremy
[root@serveransible ~]# passwd Jeremy
Changing password for user Jeremy.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Fox Fox
[root@serveransible ~]# passwd Fox
Changing password for user Fox.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

Gambar 98

```
[root@serveransible ~]# useradd -m -d /home/Meghan Meghan
[root@serveransible ~]# passwd Meghan
Changing password for user Meghan.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Mackie Mackie
[root@serveransible ~]# passwd Mackie
Changing password for user Mackie.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Anthony Anthony
[root@serveransible ~]# passwd Anthony
Changing password for user Anthony.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

Gambar 99

11. Kemudian kita memasukkan user yang telah dibuat kedalam grup sesuai dengan ketentuan

```
[root@serveransible ~]# usermod -aG Rektor Downey
[root@serveransible ~]# usermod -aG WR1 Scarlett | usermod -aG WR1 Johhanson
[root@serveransible ~]# usermod -aG Kemahasiswaan Ruffalo | usermod -aG Kemahasiswaan Mark
[root@serveransible ~]# usermod -aG Keasramaan Ruffalo | usermod -aG Keasramaan Chris
[root@serveransible ~]# usermod -aG WR2 Evans | usermod -aG WR2 Chris
[root@serveransible ~]# usermod -aG SDM Hiddleston | usermod -aG SDM Tom
[root@serveransible ~]# usermod -aG Keuangan Witwicky | usermod -aG Keuangan Sam
[root@serveransible ~]# usermod -aG WR3 Renner | usermod -aG WR3 Jeremy
[root@serveransible ~]# usermod -aG Promosi Fox | usermod -aG Promosi Meghan
[root@serveransible ~]# usermod -aG Kerjasama Mackie | usermod -aG Kerjasama Anthony
[root@serveransible ~]#
```

Gambar 100

12. Direktori kerja Rektor tidak dapat diakses oleh siapapun kecuali oleh Rektor sendiri.

```
[root@serveransible Universitas_Beyond_Imagination]# chown Downey Rektor
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 Rektor
[root@serveransible Universitas_Beyond_Imagination]#
```

Gambar 101

Penjelasan: kita menyet Downey selaku rektor sebagai owner direktori Rektor dan memberikan akses penuh, tetapi grup dan other tidak dapat mengaksesnya

13. WR mendapatkan hak akses penuh untuk folder sekretaris, tetapi sekretaris tidak dapat mengakses direktori kerja pribadi WR.

```
[root@serveransible Universitas_Beyond_Imagination]# chown Scarlett:WR1 WR1
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR1
[root@serveransible Universitas_Beyond_Imagination]# chown Evans:WR2 WR2
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR2
[root@serveransible Universitas_Beyond_Imagination]# chown Renner:WR3 WR3
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Gambar 102

Penjelasan: kita menyet Scarlett, Evans, Renner sebagai owner dari direktori WR1, WR2, WR3 namun sekretaris tidak dapat mengakses.

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir Sekretaris_WR1 | mkdir Sekretaris_WR2 | mkdir Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Gambar 103

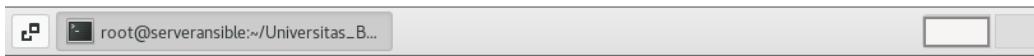
```
[root@serveransible Universitas_Beyond_Imagination]# chown Johhanson:WR1 Sekretaris_WR1
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR1
[root@serveransible Universitas_Beyond_Imagination]# chown Chris:WR2 Sekretaris_WR2
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR2
[root@serveransible Universitas_Beyond_Imagination]# chown Jeremy:WR3 Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Gambar 104

Penjelasan: Johhanson, Chris, dan Jeremy merupakan sekretaris dari WR sehingga mereka menjadi owner dari direktori mereka dan menset grup dapat mengakses direktori yang sama.

14. Dikarenakan, bagian kerja kemahasiswaan sering berkaitan dengan bagian keasramaan, maka khusus kepada ketua pembina asrama diberikan akses untuk membaca file (buatkan contoh file program_bina_karakter.txt) yang berada pada bagian kerja pembinaan karakter.

```
[root@serveransible Bina_karakter]# setfacl -m g:Keasramaan:r program_bina_karakter.txt
[root@serveransible Bina_karakter]#
```



Gambar 105

15. Begitu juga dengan bagian kerja RKA Unit harus dapat diakses oleh seluruh unit dengan hak akses penuh. Selanjutnya, dikarenakan WR1 juga memerlukan informasi strategi promosi dan juga kerjasama yang dimiliki oleh universitas, maka khusus untuk WR1 (tidak termasuk sekretaris) diberikan hak akses untuk membaca file yang berada pada direktori kerja bagian promosi dan kerjasama.

```
[root@serveransible Keuangan]# chmod 770 RKA_unit
[root@serveransible Keuangan]#
```

Gambar 106

Penjelasan: RKA unit dapat diakses oleh semua unit.

```
[root@serveransible Universitas_Beyond_Imagination]# setfacl -m u:Scarlett:r WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Gambar 107

Penjelasan: WR1 dapat membaca file didirektori WR3

16. Kondisi berikutnya adalah, khusus untuk bagian kerja SDM sekretaris dari WR2 tidak diberikan hak akses sama sekali.

```
[root@serveransible WR2]# setfacl -m g:WR1:rwx SDM
[root@serveransible WR2]# setfacl -m g:WR3:rwx SDM
[root@serveransible WR2]# setfacl -m u:Evans:rwx SDM
[root@serveransible WR2]#
```



Gambar 108

17. Khusus untuk unit keuangan bagian kerja Bursar Mahasiswa, Sam dan sekretaris WR2 tidak mendapatkan hak akses sama sekali pada direktori tersebut.

```
[root@serveransibile WR2]# chown Witwicky Keuangan
[root@serveransibile WR2]# setfacl -m u:Scarlett:rwx Keuangan
[root@serveransibile WR2]# setfacl -m g:WR3:rwx Keuangan
[root@serveransibile WR2]# chmod 770 Keuangan
[root@serveransibile WR2]#
```

Gambar 109

Penjelasan: semua unit dapat mengakses kecuali sam dan Johhanson

18. Dikarenakan masing-masing WR harus membutuhkan akses cepat untuk beberapa informasi tertentu maka dibutuhkan hard link atau soft link yang ada pada direktori kerja pribadi merujuk ke beberapa file berikut. Untuk WR1 dibutuhkan link ke file program_bina_karakter.txt pada direktori kerja pembinaan karakter, WR2 dibutuhkan link ke file RKA_unit_kemahasiswaan.txt pada direktori kerja RKA Unit, dan pada WR3 dibutuhkan link ke file kerjasama_dalam_dan_luar_negri.txt yang berada pada direktori kerja Kerjasama.

```
[root@serveransibile Kemahasiswaan]# cd Bina_karakter
[root@serveransibile Bina_karakter]# nano program_bina_karakter.txt
[root@serveransibile Bina_karakter]#
```

Gambar 110

Penjelasan: pembuatan program_bina_karakter.txt dalam direktori Bina_karakter.

```
[root@serveransibile WR2]# cd Keuangan
[root@serveransibile Keuangan]# cd RKA_unit
[root@serveransibile RKA_unit]# nano RKA_unit_kemahasiswaan.txt
[root@serveransibile RKA_unit]#
```

Gambar 111

Penjelasan: pembuatan program RKA_unit_kemahasiswaan.txt

```
[root@serveransibile ~]# cd Universitas_Beyond_Imagination
[root@serveransibile Universitas_Beyond_Imagination]# cd WR3
[root@serveransibile WR3]# cd Kerjasama
[root@serveransibile Kerjasama]# cd Kerjasama
[root@serveransibile Kerjasama]# ls
[root@serveransibile Kerjasama]# nano kerjasama_dalam_dan_luar_negeri.txt
[root@serveransibile Kerjasama]#
```

Gambar 112

Penjelasan: proses pembuatan file kerjasama_dalam_dan_luar_negeri.txt pada direktori kerjasama.

```
[root@serveransibile Universitas_Beyond_Imagination]# ln -s /Universitas_Beyond_Imagination/WR1/Kemahasiswaan/Bina_karakter/program_bina_karakter.txt /WR1
[root@serveransibile Universitas_Beyond_Imagination]#
```

Gambar 113

```
[Imagination]# ln -s /Universitas_Beyond_Imagination/WR1/Kemahasiswaan/Bina_karakter/program_bina_karakter.txt /WR1  
[Imagination]#
```

Gambar 114

Penjelasan: command diatas merupakan soft link ke file program_bina_karakter.txt

```
[root@serveransible ~]# ln -s /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA_unit_kemahasiswaan.txt /WR2  
[root@serveransible ~]# ls -l /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA_unit_kemahasiswaan.txt /WR2  
ls: cannot access /Universitas Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA unit kemahasiswaan.txt: No such file or directory  
lrwxrwxrwx 1 root root 80 Sep 23 14:12 /WR2 -> /Universitas Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA unit kemahasiswaan.txt  
[root@serveransible ~]#
```

Gambar 115

Penjelasan: command diatas merupakan soft link ke file program_RKA_unit_kemahasiswaan.txt

```
[root@serveransible ~]# ls -l /Universitas_Beyond_Imagination/WR3/Kerjasama/Kerjasama/kerjasama_dalam_dan_luar_negeri.txt /WR3  
ls: cannot access /Universitas Beyond_Imagination/WR3/Kerjasama/Kerjasama/kerjasama dalam dan luar negeri.txt: No such file or directory  
lrwxrwxrwx 1 root root 91 Sep 23 14:25 /WR3 -> /Universitas Beyond_Imagination/WR3/Kerjasama/Kerjasama/kerjasama dalam dan luar negeri.txt  
[root@serveransible ~]#
```

Gambar 116

Penjelasan: command diatas merupakan soft link ke file kerjasama_dalam_dan_luar_negeri.txt

BAB 5. FILESYSTEM AND LOGICAL VOLUME MANAGER

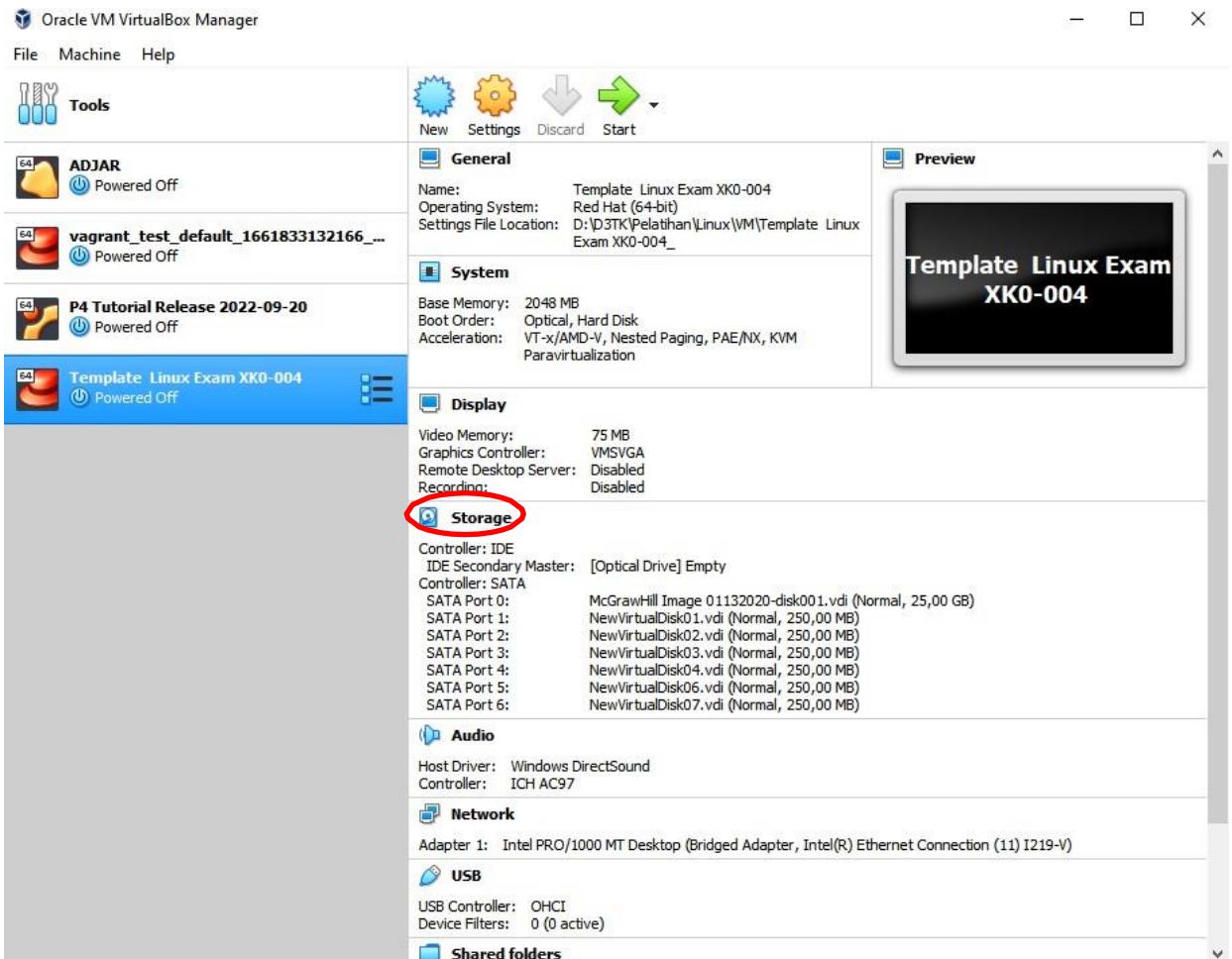
A. Logical Volume Configuration

Selesaikanlah praktikum ini dengan menggunakan mesin virtual yang telah diinstal pada komputer masing-masing. Uji coba pada praktikum ini menggunakan VirtualBox versi 6.0.4, segala konfigurasi terkait mesin virtual disesuaikan dengan jenis dan versi mesin virtual yang diinstal.

Konfigurasi pada mesin virtual

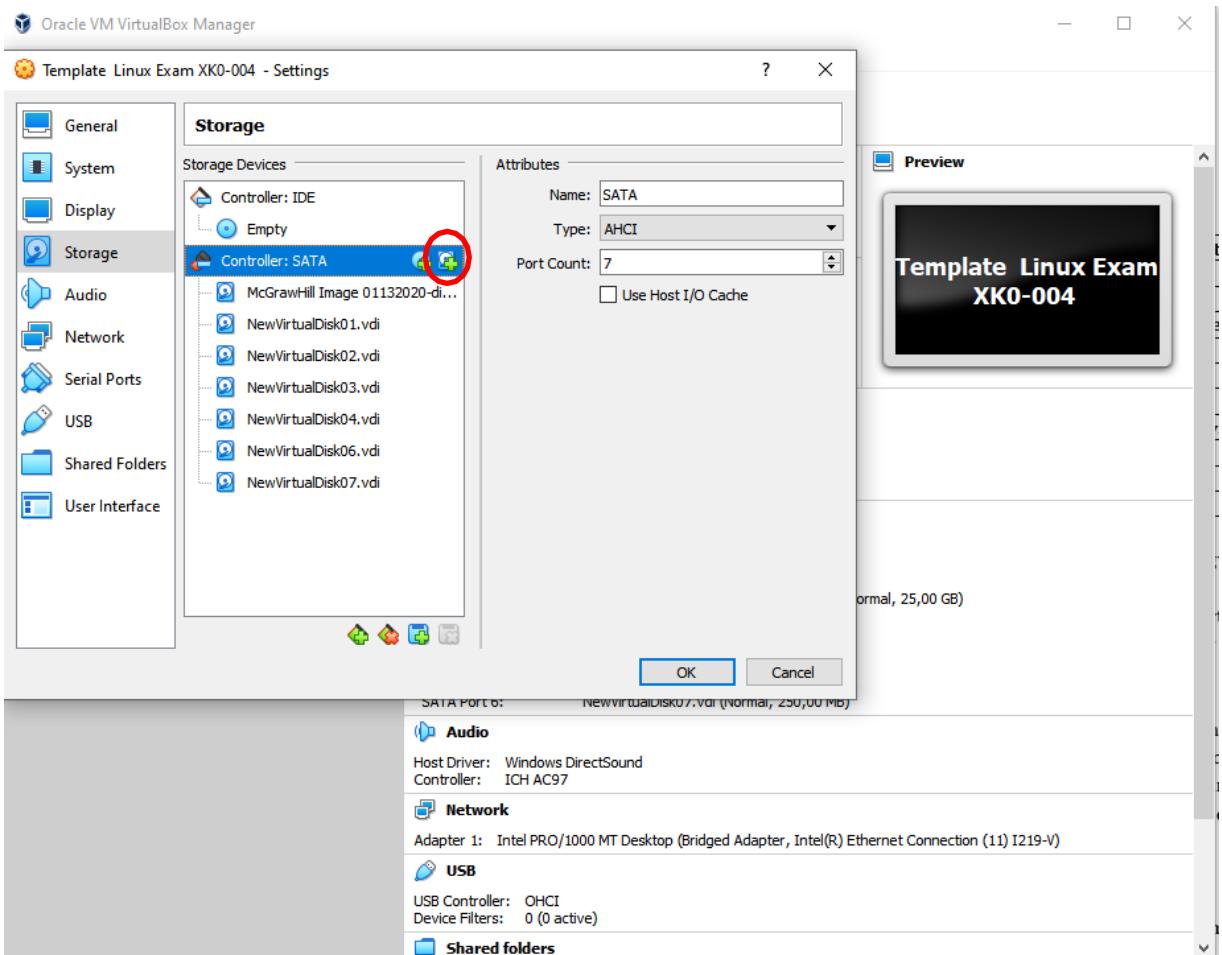
Sebelum menjalankan langkah-langkah berikut pastikan bahwa sistem operasi yang berjalan pada mesin virtual sudah dimatikan.

1. Klik menu Storage untuk menambahkan 6 storage.



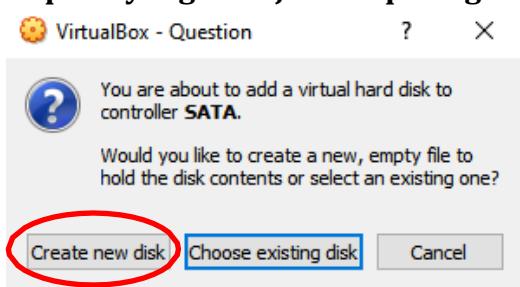
Gambar 117

2. Tambahkan 6 virtual disk dengan menekan ikon petak sebelah kanan dengan tanda plus hijau, seperti yang ditunjukkan pada gambar berikut.



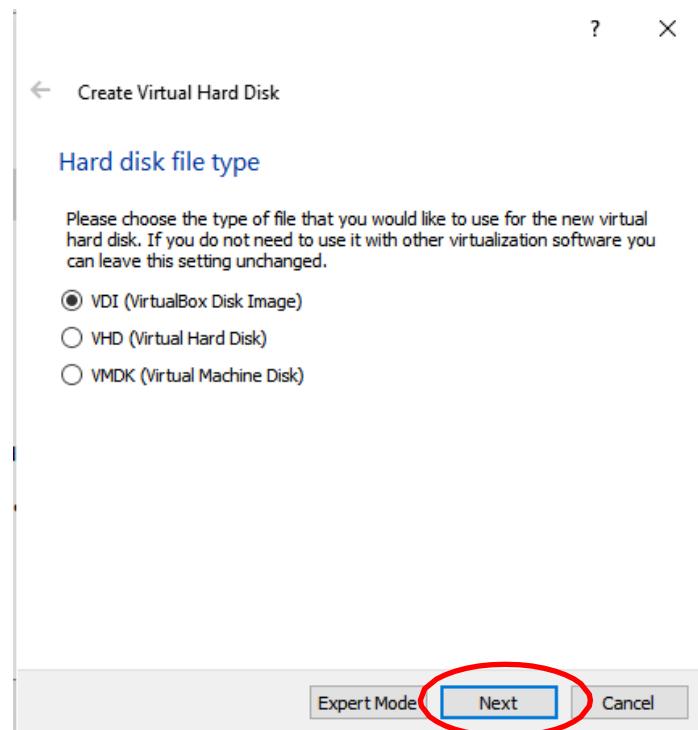
Gambar 118

seperti yang ditunjukkan pada gambar berikut.



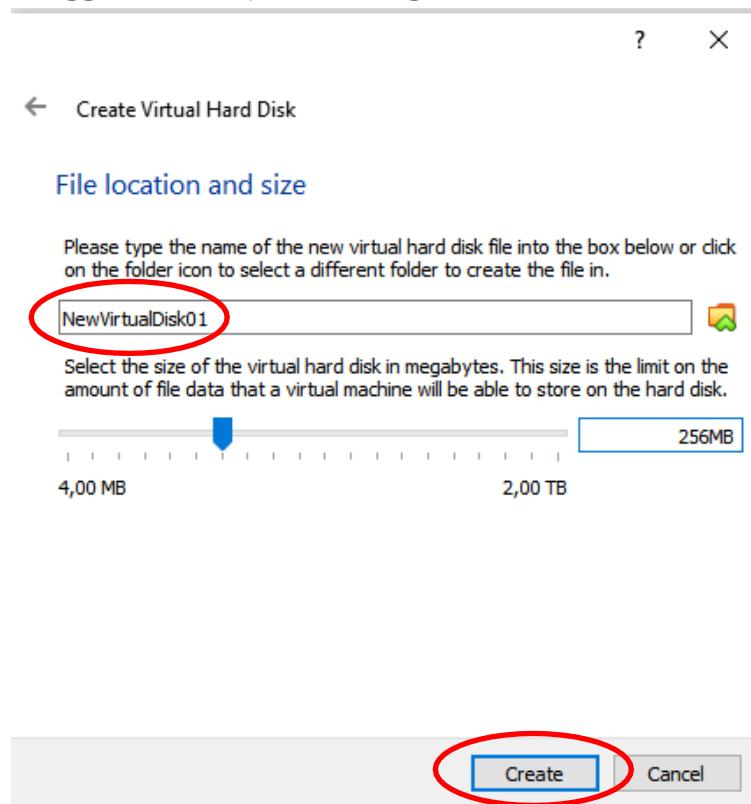
Gambar 119

4. Window berikutnya adalah menentukan tipe file untuk virtual hard disk yang baru. Pilihlah VDI (VirtualBox Disk Image), seperti yang ditunjukkan pada gambar berikut, kemudian pada Window selanjutnya pilih Dynamically Allocated.



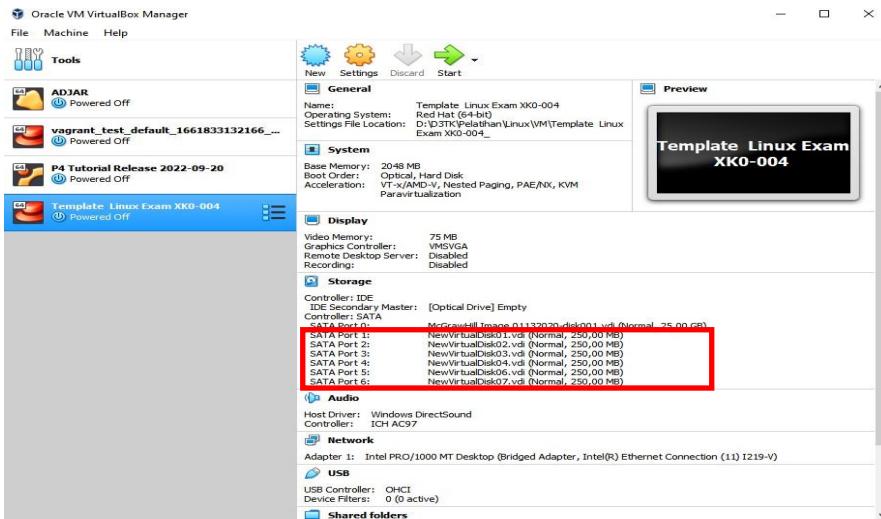
Gambar 120

5. Ubahlah nama pada hard drive. Misalnya, diberi nama NewVirtualDisk01 sampai virtual disk keenam. Buatlah ukuran untuk setiap virtual disk 256 MB, selanjutnya klik tombol Create. Pastikan penamaan yang unik untuk setiap virtual disk yang baru sehingga tidak terjadi naming conflict.



Gambar 121

6. Ulangi langkah-langkah sebelumnya hingga tercipta 6 disk yang baru dengan ukuran setiap disk 256MB. Klik OK pada pengaturan Storage untuk menyelesaikan pembuatan hard drive baru. Ilustrasi berikut menunjukkan hasil akhir, yang memperlihat 6 disk yang baru telah ditambahkan pada mesin virtual. Selanjutnya nyalakan mesin virtual.



Gambar 122

Setelah keenam virtual disk yang baru telah ditambahkan pada mesin virtual, selanjutnya jalankan langkah-langkah berikut.

- 1. Setelah sistem operasi berjalan, buka aplikasi terminal.**
- 2. Konfigurasi LVM dapat dieksekusi menggunakan akun root dengan perintah su -**
- 3. Untuk mengetahui hard drive yang telah ditambahkan pada sistem operasi jalankan perintah fdisk -l. Misalnya, pada gambar berikut keenam hard drive yang baru ditunjukkan pada Disk /dev/sdb - Disk /dev/sdg.**

```
Disk /dev/sde: 268 MB, 268435456 bytes, 524288 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdf: 268 MB, 268435456 bytes, 524288 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdd: 268 MB, 268435456 bytes, 524288 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdg: 268 MB, 268435456 bytes, 524288 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-root: 18.2 GB, 18249416704 bytes, 35643392 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/centos-swap: 2147 MB, 2147483648 bytes, 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

[root@localhost josa]#
```

Gambar 123

4. Pilihlah 2 hard drive yang akan dikonversi menjadi physical volume sebagai penampung LVM, dengan menjalankan perintah pvcreate. Misalnya, seperti yang ditunjukkan pada gambar berikut:

```
# pvcreate /dev/sdb ; pvcreate /dev/sdc ; pvdisplay
Physical volume "/dev/sdc" successfully created.
--- Physical volume ---
PV Name           /dev/sda2
VG Name            centos
PV Size           <19.00 GiB / not usable 3.00 MiB
Allocatable        yes (but full)
PE Size            4.00 MiB
Total PE          4863
Free PE           0
Allocated PE      4863
PV UUID           HmfR1w-MUcB-kxDI-DIKL-Iw2K-K0t6-K0ri8N

"/dev/sdb" is a new physical volume of "256.00 MiB"
--- NEW Physical volume ---
PV Name           /dev/sdb
VG Name            centos
PV Size           256.00 MiB
Allocatable        NO
PE Size            0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           Yk0NdM-443j-pgYl-Zk5y-sprk-WdZK-CvBcRy

"/dev/sdc" is a new physical volume of "256.00 MiB"
--- NEW Physical volume ---
PV Name           /dev/sdc
VG Name            centos
PV Size           256.00 MiB
Allocatable        NO
PE Size            0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           tEdhDP-8dK6-j1Pd-XbmH-f8xw-9g4p-JKbxo5

[root@localhost josal]# _
```

Gambar 124

5. Jika physical volume sudah terbentuk, buatlah volume group dengan nama home, sepertiperintah berikut:

```
# vgccreate home /dev/sdb /dev/sdc ; vgdisplay
```

```
--- NEW Physical volume ---
PV Name           /dev/sdc
VG Name            centos
PV Size           256.00 MiB
Allocatable        NO
PE Size            0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           tEdhDP-8dK6-j1Pd-XbmH-f8xw-9g4p-JKbxo5

[root@localhost josal]# pvcreate home /dev/sdb /dev/sdc ; vgdisplay
Device home not found (or ignored by filtering).
Physical volume "/dev/sdb" successfully created.
Physical volume "/dev/sdc" successfully created.
--- Volume group ---
VG Name           centos
System ID         lvm2
Format            lvm2
Metadata Areas    1
Metadata Sequence No 3
VG Access         read/write
VG Status         resizable
MAX LV            0
Cur LV            2
Open LV           2
Max PV            0
Cur PV            1
Act PV             1
VG Size           <19.00 GiB
PE Size            4.00 MiB
Total PE          4863
Alloc PE / Size   4863 / <19.00 GiB
Free  PE / Size   0 / 0
VG UUID           4cRviT-Zs5Z-GfPe-tTkq-FWEd-ahIt-Qzow0a

[root@localhost josal]# _
```

Gambar 125

6. Alokasikan 300 MB Logical Volume yang disebut dengan `home_directory` dari volume group yang sudah dibuat, seperti perintah berikut:

```
# lvcreate -L 300M -n home_directory ; lvdisplay
```

```
[root@localhost josal]# lvdisplay
--- Logical volume ---
LV Path          /dev/centos/swap
LV Name          swap
VG Name          centos
LV UUID          cNU8mm-kb4q-hCEm-KSpx-GdM8-Lw8A-9tdKUp
LV Write Access  read/write
LV Creation host, time localhost, 2022-08-30 08:53:07 +0700
LV Status        available
# open           2
LV Size          2.00 GiB
Current LE       512
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 8192
Block device    253:1

--- Logical volume ---
LV Path          /dev/centos/root
LV Name          root
VG Name          centos
LV UUID          WeTgSK-lFEy-Q7yB-RTzg-xGN1-bmWo-GIullt
LV Write Access  read/write
LV Creation host, time localhost, 2022-08-30 08:53:08 +0700
LV Status        available
# open           1
LV Size          <17.00 GiB
Current LE       4351
Segments         1
Allocation       inherit
Read ahead sectors auto
- currently set to 8192
Block device    253:0

[root@localhost josal]# _
```

Gambar 126

7. Buatlah filesystem kemudian mount ke direktori yang baru, seperti perintah berikut:

```
# mkfs.ext4 /dev/home/home_directory ; mkdir home_directory
# mount /dev/home/home_directory /home_directory/
# lvdisplay
```

```
[root@localhost ~]# mkfs.ext4 /dev/home/home_directory ; mkdir home_directory
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=8)
Fragment size=1024 (log=8)
Stride=0 blocks, Stripe width=0 blocks
76912 inodes, 307200 blocks
15360 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=33947648
38 block groups
8192 blocks per group, 8192 fragments per group
2024 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[root@localhost ~]# _
```

Gambar 127

```
[root@localhost ~]# mount /dev/mapper/home-home_directory /home
mount: /dev/mapper/home-home_directory is already mounted or /home busy
      /dev/mapper/home-home_directory is already mounted on /home
[root@localhost ~]# _
```

Gambar 128

```
-- Logical volume --
LV Path          /dev/mapper/home-home_directory
LV Name          home_directory
VG Name          home
LV UUID          68Esqu-1SIf-wcXb-zQAs-UUAs-spfU-Lhid5x
LV Write Access  read/write
LV Creation host, time localhost.localdomain, 2022-10-14 16:20:58 +0700
LV Status        available
# open           1
LV Size          300.00 MiB
Current LE       75
Segments         2
Allocation       inherit
Read ahead sectors auto
- currently set to 8192
Block device    253:2

[root@localhost ~]# _
```

Gambar 129

8. Resize logical volume `home_directory` menjadi 400MB kemudian tampilkan hasilnya dengan perintah `df -h`.

```
# lvresize -r -L 400M /dev/mapper/home-home_directory
# df -h
```

```
[root@localhost ~]# lvresize -r -L 400M /dev/mapper/home-home_directory
  Size of logical volume home/home_directory changed from 300.00 MiB (75 extents) to 400.00 MiB (100 extents).
Logical volume home/home_directory successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/home-home_directory is mounted on /home: on-line resizing required
old_desc_blocks = 3, new_desc_blocks = 4
The filesystem on /dev/mapper/home-home_directory is now 409600 blocks long.

[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/centos-root 17G   1.5G  16G  9% /
/devtmpfs        901M     0  901M  0% /dev
tmpfs            912M     0  912M  0% /dev/shm
tmpfs            912M  8.7M  903M  1% /run
tmpfs            912M     0  912M  0% /sys/fs/cgroup
/dev/sda1        1014M  149M  866M  15% /boot
tmpfs            183M     0  183M  0% /run/user/1000
/dev/mapper/home-home_directory 388M  2.3M  355M  1% /home
[root@localhost ~]#
```

Gambar 130

9. Jika kapasitas masih dibutuhkan, tambahkan 50MB, seperti perintah berikut:

```
# lvresize -r -L +50M /dev/mapper/home-home_directory
# df -h
```

```
[root@localhost ~]# lvresize -r -L +50M /dev/mapper/home-home_directory
  Rounding size to boundary between physical extents: 52.00 MiB.
  Size of logical volume home/home_directory changed from 400.00 MiB (100 extents) to 452.00 MiB (113 extents).
Logical volume home/home_directory successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Filesystem at /dev/mapper/home-home_directory is mounted on /home: on-line resizing required
old_desc_blocks = 4, new_desc_blocks = 5
The filesystem on /dev/mapper/home-home_directory is now 462848 blocks long.

[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/centos-root 17G   1.5G  16G  9% /
/devtmpfs        901M     0  901M  0% /dev
tmpfs            912M     0  912M  0% /dev/shm
tmpfs            912M  8.7M  903M  1% /run
tmpfs            912M     0  912M  0% /sys/fs/cgroup
/dev/sda1        1014M  149M  866M  15% /boot
tmpfs            183M     0  183M  0% /run/user/1000
/dev/mapper/home-home_directory 430M  2.3M  403M  1% /home
[root@localhost ~]# _
```

Gambar 131

10. Tambahkan logical volume `home_directory` pada file `/etc/fstab`, sehingga logical volume dapat di-mount ketika sistem diaktifkan. Pada line yang sama terapkan kuota untuk user dan group, sehingga penggunaan storage dapat dibatasi, seperti perintah berikut:

```
/dev/home/home_directory      /home_directory  ext4  defaults,usrquota,grpquota  1 2
```

Gambar 132

11. Selanjutnya, untuk mengaktifkan pengaturan kuota yang telah didefinisikan pada file

`/etc/fstab`, maka jalankan perintah berikut:

```
quotacheck -cug /home_directory
```

```
[root@localhost ~]# quotacheck -cug /home_directory  
[root@localhost ~]# █
```

Gambar 133

Jika kuota berhasil diaktifkan, maka akan terbentuk dua file yaitu `aquota.user` dan `aquota.group`.

```
# cd /home_directory/  
# ls -l  
total 30  
-rw----- 1 root root 7168 Sep 29 06:51 aquota.group  
-rw----- 1 root root 7168 Sep 29 06:51 aquota.user  
[root@localhost ~]# cd /home_directory/  
[root@localhost home_directory]# ls -l  
total 26  
-rw----- 1 root root 6144 Oct 13 21:18 aquota.group  
-rw----- 1 root root 6144 Oct 13 21:18 aquota.user  
drwx----- 2 root root 12288 Oct 13 21:05 lost+found  
[root@localhost home directory]# █
```

Gambar 134

12. Selanjutnya, jika user ditambahkan pada sistem, maka home direktori akan diarahkan ke logical volume `/home_directory`. Hal ini mungkin dilakukan dengan mengubah file

`/etc/default/useradd`. Ubahlah variable `HOME` mengarah ke direktori `home` yang baru yaitu `/home_directory`. Uji cobalah apakah ketika user yang baru ditambahkan maka direktori `home` yang terbentuk ada pada `/home_directory`.

GNU nano 2.3.1

File: /etc/default/useradd

```
# useradd defaults file  
GROUP=100  
HOME=/home_directory  
INACTIVE=-1  
EXPIRE=  
SHELL=/bin/bash  
SKEL=/etc/skel  
CREATE_MAIL_SPool=yes
```

Gambar 135

13. Kuota dapat diterapkan pada single user ataupun group dengan menggunakan perintah edquota *username/group*. Buatlah sebuah user dengan username test1, seharusnya direktori home yang tercipta berada pada /home_directory, seperti berikut:

```
# pwd  
/home_directory/test1
```

Gambar 136

14. Kemudian terapkan kuota untuk user test1 dengan menjalankan perintah edquota *username* misalnya edquota test1. Ketika perintah ini dijalankan kuota untuk user test1 dapat dikonfigurasi, tambahkan 55KB pada bagian hard, seperti gambar berikut:

```
1 Disk quotas for user test1 (uid 1003):  
2   Filesystem          blocks      soft      hard      inodes      soft      hard  
3   /dev/sda6              0          0          0          0          0          0          0  
4   /dev/mapper/home-home_directory    0          0      55000          0          0          0          0
```

Gambar 137

15. Lakukanlah uji coba untuk memastikan bahwa user hanya dapat menggunakan direktori home sesuai dengan kuota yang telah ditetapkan dengan menambahkan file pada direktori home user test1. Selanjutnya, untuk memeriksa kuota yang telah digunakan user, jalankan perintah seperti pada gambar berikut:

```
# quota test1  
Disk quotas for user test1 (uid 1003):  
  Filesystem  blocks  quota  limit  grace  files  quota  limit  grace  
/dev/mapper/home-home_directory        55001*    0    55000            1      0      0
```

Gambar 138

16. Untuk mendapatkan laporan penggunaan kuota untuk seluruh user dapat dilakukan dengan menjalankan perintah repquota -a, seperti gambar berikut:

```
# repquota -a  
*** Report for user quotas on device /dev/sda6  
Block grace time: 7days; Inode grace time: 7days  
                                Block limits                      File limits  
User           used     soft     hard grace     used     soft     hard grace  
-----  
root          --       28       0       0             5       0       0       0  
  
*** Report for user quotas on device /dev/mapper/home-home_directory  
Block grace time: 7days; Inode grace time: 7days  
                                Block limits                      File limits  
User           used     soft     hard grace     used     soft     hard grace  
-----  
root          --       13       0       0             2       0       0       0  
test1         +-     55001      0    55000            1      0      0
```

Gambar 139

17. Jika kapasitas pada sebuah volume group perlu ditambah, maka dapat dilakukan dengan menjalankan perintah vgextend. Sebelum menambah kapasitas dari hard drive lainnya, harus dipastikan bahwa hard drive tersebut belum dialokasikan menjadi filesystem untuk program lain, dengan perintah fdisk -l. Misalnya, logical volume home yang dialokasikan untuk direktori home kapasitasnya perlu ditambah dan terdapat hard drive

```
# vgextend home /dev/sdg
[root@localhost ~]# vgextend home /dev/sdg
Physical volume "/dev/sdg" successfully created.
Volume group "home" successfully extended
[root@localhost ~]# █
```

Gambar 140

yang free pada /dev/sdg, maka perintah yang akan dijalankan seperti berikut:

Periksalah apakah kapasitas untuk logical volume home sudah bertambah.

18. Selanjutnya, hard drive (physical volume) dapat dipisahkan dari volume group, namun harus dipastikan terlebih dahulu bahwa physical volume tersebut tidak sedang digunakan oleh logical volume dengan menjalankan perintah pvdisplay. Misalnya, hapuslah physical volume /dev/sdg yang telah dialokasikan pada volume group home, dengan menjalankan perintah berikut:

```
# vgreduce home /dev/sdg
```

Jalankanlah kembali instruksi pada nomor 17, kemudian bagilah volume group dengan membentuk volume group yang baru. Namun sebelumnya dipastikan bahwa logical volume tidak dapat dibagi pada volume group yang berbeda. Setiap logical volume yang sudah terbentuk harus berada pada physical volume yang telah digabungkan pada sebuah volume group apakah volume group yang baru atau volume group yang lama. Misalnya, membagi volume group yang baru dengan volume

```
# vgsplit home new_home /dev/sdg
```

group yang telah ada, seperti perintah berikut:

BAB 6. FILESYSTEM AND LOGICAL VOLUME MANAGER – PART II

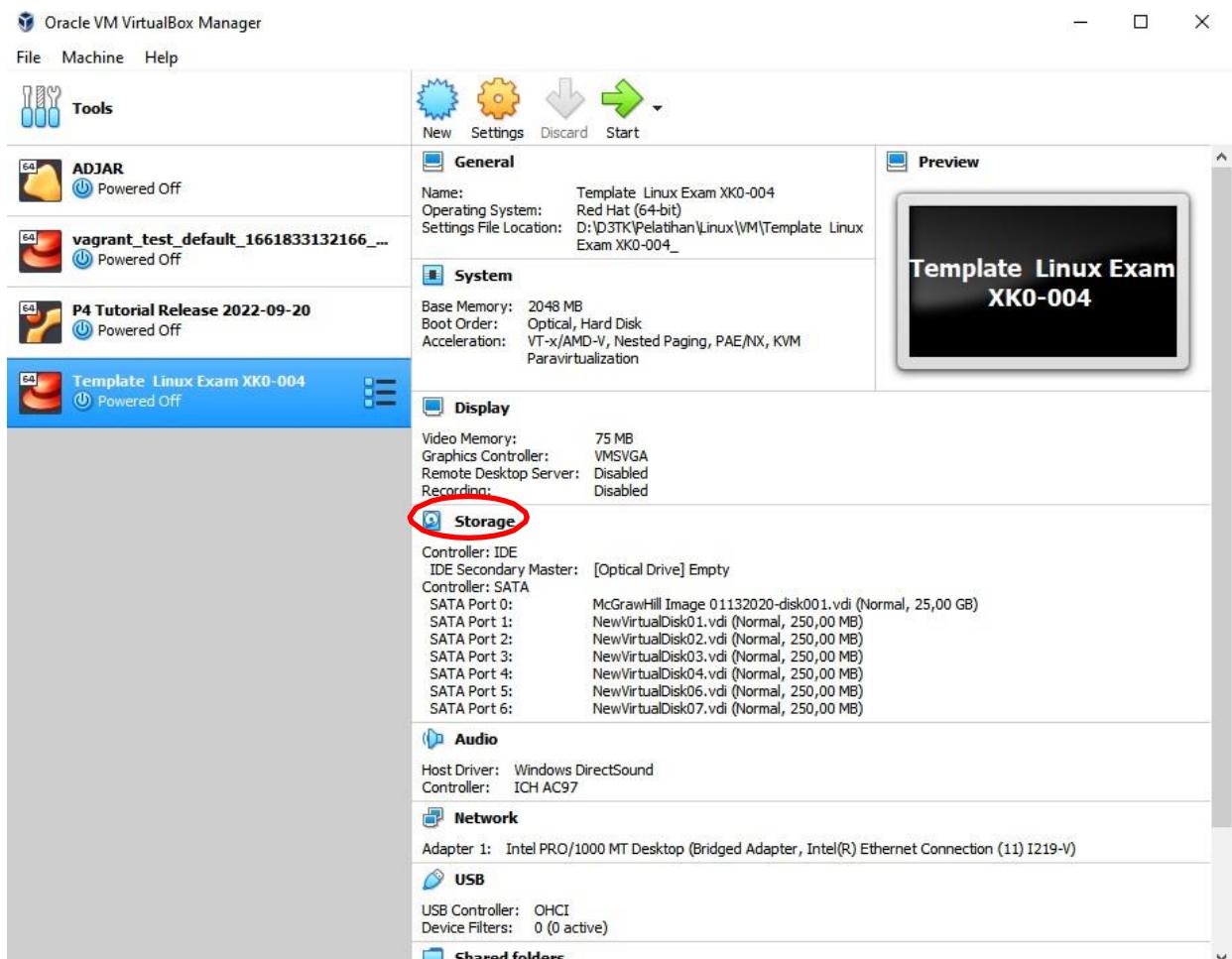
A. Logical Volume Configuration

Selesaikanlah praktikum ini dengan menggunakan mesin virtual yang telah diinstal pada komputer masing-masing. Uji coba pada praktikum ini menggunakan VirtualBox versi 6.0.4, segala konfigurasi terkait mesin virtual disesuaikan dengan jenis dan versi mesin virtual yang diinstal.

Konfigurasi pada mesin virtual

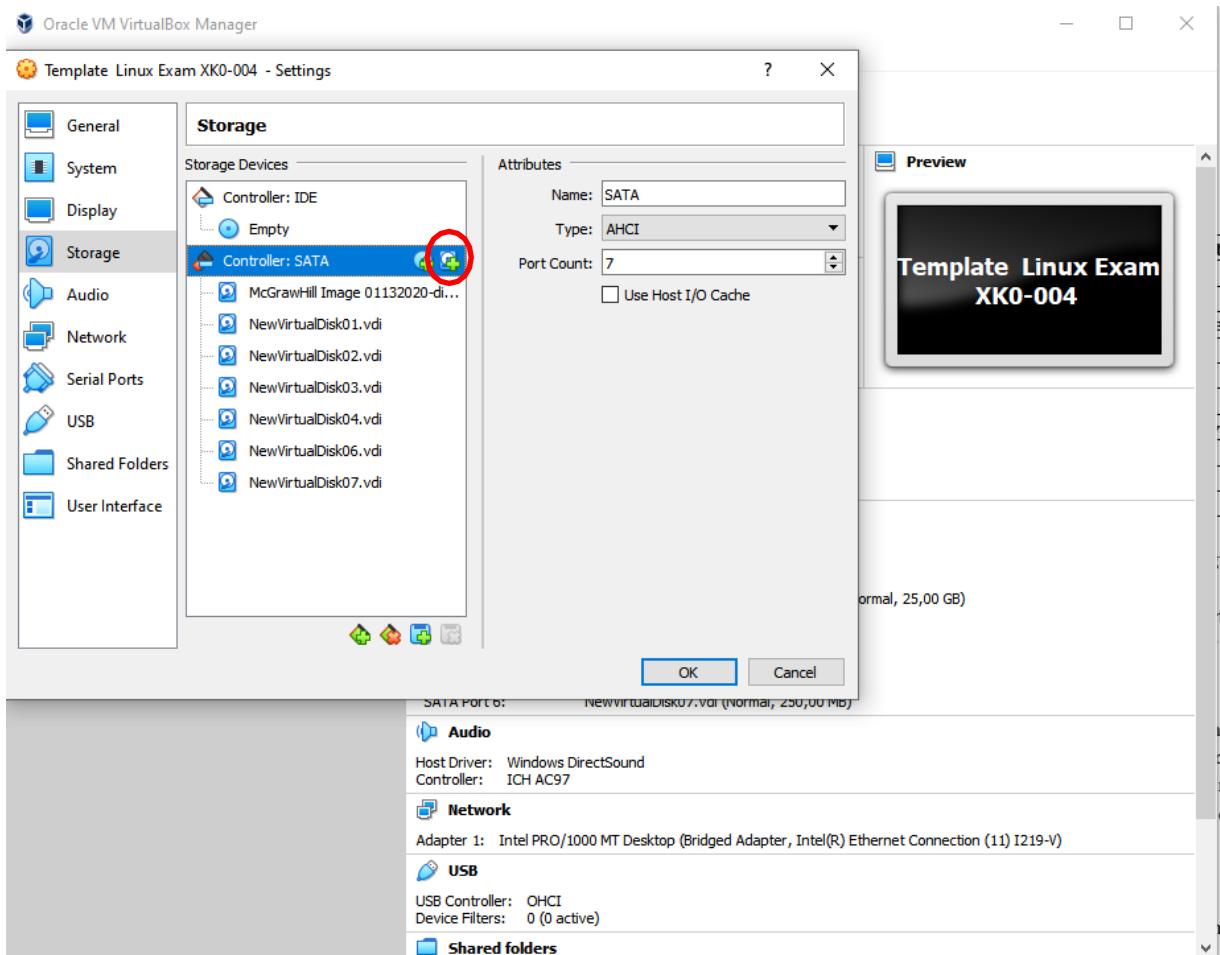
Sebelum menjalankan langkah-langkah berikut pastikan bahwa sistem operasi yang berjalan pada mesin virtual sudah dimatikan.

1. Klik menu Storage untuk menambahkan 6 storage.



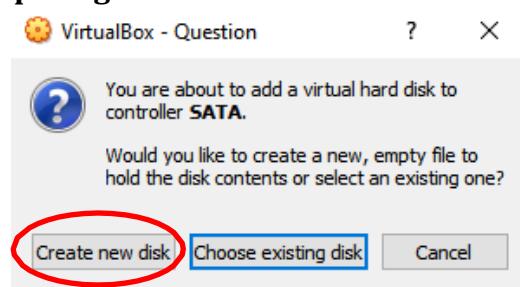
Gambar 141

2. Tambahkan 6 virtual disk dengan menekan ikon petak sebelah kanan dengan tanda plus hijau, seperti yang ditunjukkan pada gambar berikut.



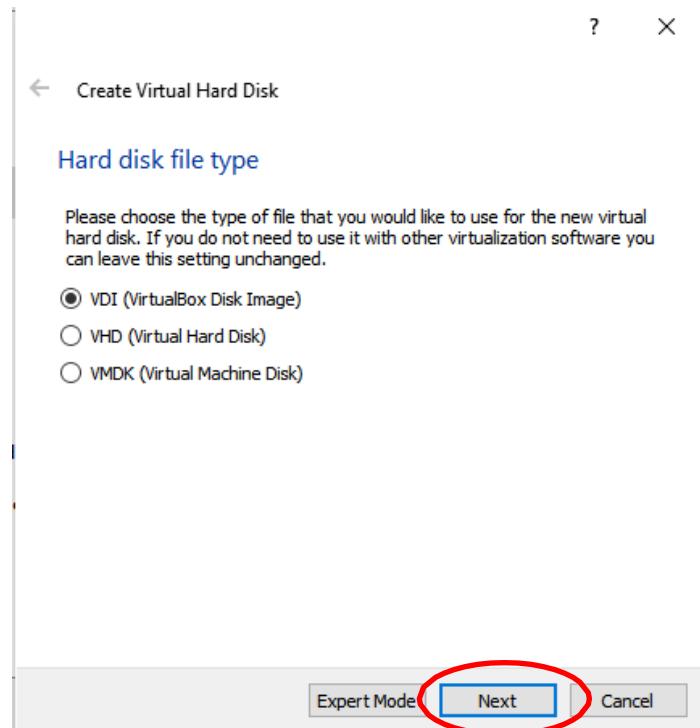
Gambar 142

seperti yang ditunjukkan pada gambar berikut.



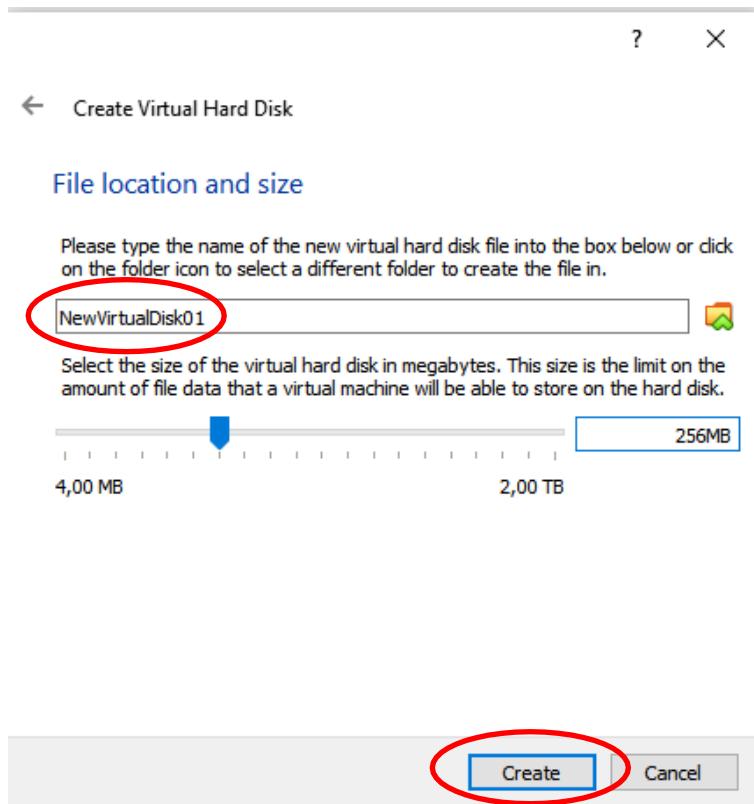
Gambar 143

4. Window berikutnya adalah menentukan tipe file untuk virtual hard disk yang baru. Pilihlah VDI (VirtualBox Disk Image), seperti yang ditunjukkan pada gambar berikut, kemudian pada Window selanjutnya pilih Dynamically Allocated.



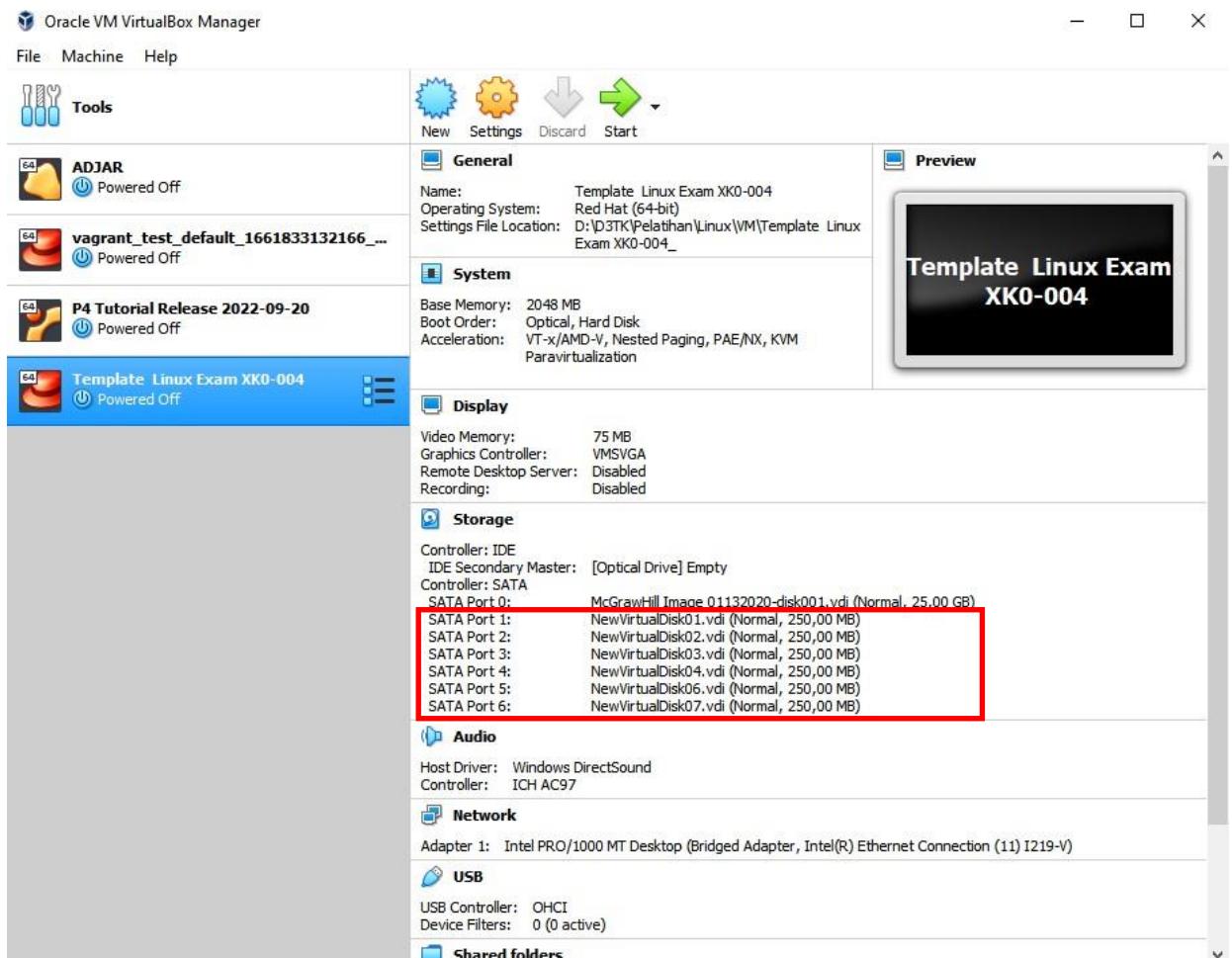
Gambar 144

5. Ubahlah nama pada hard drive. Misalnya, diberi nama NewVirtualDisk01 sampai virtual disk keenam. Buatlah ukuran untuk setiap virtual disk 256 MB, selanjutnya klik tombol Create. Pastikan penamaan yang unik untuk setiap virtual disk yang baru sehingga tidak terjadi naming conflict.



Gambar 145

6. Ulangi langkah-langkah sebelumnya hingga tercipta 6 disk yang baru dengan ukuran setiap disk 256MB. Klik OK pada pengaturan Storage untuk menyelesaikan pembuatan hard drive baru. Ilustrasi berikut menunjukkan hasil akhir, yang memperlihat 6 disk yang baru telah ditambahkan pada mesin virtual. Selanjutnya nyalakan mesin virtual.



Gambar 146

B. Logical Volume Configuration

Setelah keenam virtual disk yang baru telah ditambahkan pada mesin virtual, selanjutnya jalankan langkah-langkah berikut.

- 1. Setelah sistem operasi berjalan, buka aplikasi terminal.**
- 2. Konfigurasi LVM dapat dieksekusi menggunakan akun root dengan perintah su -**
- 3. Untuk mengetahui hard drive yang telah ditambahkan pada sistem operasi jalankan perintah fdisk -l. Misalnya, pada gambar berikut keenam hard drive yang baru ditunjukkan pada Disk /dev/sdb – Disk /dev/sdg.**

The screenshot shows a terminal window titled "Template Linux Exam XK0-004 [Running] - Oracle VM VirtualBox". The window contains the output of the "fdisk -l" command. It lists six drives: /dev/sda7, /dev/sdb, /dev/sdc, /dev/sdd, /dev/sde, /dev/sdf, and /dev/sdg. Each drive entry provides details about its size (262 MB), byte count (262144000), sectors (512000), and sector size (512 bytes). The terminal prompt at the bottom indicates the user is root (@localhost:~).

```
Template Linux Exam XK0-004 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal

File Edit View Search Terminal Help
/dev/sda7      34512896  38707199  2097152  8e  Linux LVM

Disk /dev/sdb: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdc: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdd: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sde: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdf: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdg: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

root@localhost:~
```

Gambar 147

- 4. Pilihlah 2 hard drive yang akan dikonversi menjadi physical volume**
- 5. sebagai penampung LVM, dengan menjalankan perintah pvcreate. Misalnya, seperti yang ditunjukkan pada gambar berikut:**

```
# pvcreate /dev/sdb ; pvcreate /dev/sdc ; pvdisplay
```

```
[root@localhost ~]# pvcreate /dev/sde ; pvcreate /dev/sdf ; pvdisplay
Physical volume "/dev/sde" successfully created.
Physical volume "/dev/sdf" successfully created.
"/dev/sdf" is a new physical volume of "262.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sdf
VG Name
PV Size         262.00 MiB
Allocatable     NO
PE Size          0
Total PE        0
Free PE         0
Allocated PE    0
PV UUID         9Kec1C-m6aP-oa2j-L10V-3WmA-ZQ2z-yhWkeq

"/dev/sde" is a new physical volume of "262.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sde
VG Name
PV Size         262.00 MiB
Allocatable     NO
```

Gambar 149

C. Jika physical volume sudah terbentuk, buatlah volume group dengan nama home, sepertiperintah berikut:

```
# vgcreate home /dev/sdb /dev/sdc ; vgdisplay
[root@localhost ~]# vgcreate home /dev/sde /dev/sdf ; vgdisplay
Volume group "home" successfully created
--- Volume group ---
VG Name          home
System ID
Format          lvm2
Metadata Areas   2
Metadata Sequence No 1
VG Access       read/write
VG Status        resizable
MAX LV          0
Cur LV          0
Open LV          0
Max PV          0
Cur PV          2
Act PV          2
VG Size         520.00 MiB
PE Size          4.00 MiB
Total PE        130
Alloc PE / Size 0 / 0
Free PE / Size 130 / 520.00 MiB
```

Gambar 150

1. Alokasikan 300 MB Logical Volume yang disebut dengan `home_directory` dari volume group yang sudah dibuat, seperti perintah berikut:

```
# lvcreate -L 300M -n home_directory ; lvdisplay  
# mkfs.ext4 /dev/home/home_directory ; mkdir home_directory  
# mount /dev/home/home_directory /home_directory/  
# lvdisplay  
[root@localhost ~]# lvcreate -L 300M -n home_directory home ; lvdisplay  
Logical volume "home_directory" created.  
--- Logical volume ---  
LV Path          /dev/home/home_directory  
LV Name          home_directory  
VG Name          home  
LV UUID          11TqBa-ulUI-UzXZ-7StQ-8RNW-ViOB-irW47S  
LV Write Access  read/write  
LV Creation host, time localhost.localdomain, 2022-10-13 20:34:26 -0700  
LV Status        available  
# open           0  
LV Size          300.00 MiB  
Current LE       75  
Segments         2  
Allocation       inherit  
Read ahead sectors auto  
- currently set to 8192  
Block device     253:0  
[root@localhost ~]# █
```

Gambar 151

Buatlah filesystem kemudian mount ke direktori yang baru, seperti perintah berikut:

```
[root@localhost ~]# mkfs.ext4 /dev/home/home_directory ; mkdir home_directory  
mke2fs 1.42.9 (28-Dec-2013)  
Filesystem label=  
OS type: Linux  
Block size=1024 (log=0)  
Fragment size=1024 (log=0)  
Stride=0 blocks, Stripe width=0 blocks  
76912 inodes, 307200 blocks  
15360 blocks (5.00%) reserved for the super user  
First data block=1  
Maximum filesystem blocks=33947648  
38 block groups  
8192 blocks per group, 8192 fragments per group  
2024 inodes per group
```

Gambar 152

```
[root@localhost ~]# mkfs.ext4 /dev/home/home_directory ; mkdir home_directory  
mke2fs 1.42.9 (28-Dec-2013)  
/dev/home/home_directory is apparently in use by the system; will not make a filesystem  
here!  
mkdir: cannot create directory 'home_directory': File exists  
[root@localhost ~]# mount /dev/home/home_directory /home_directory/  
mount: /dev/mapper/home-home_directory is already mounted or /home_directory busy  
[root@localhost ~]# lvdisplay  
--- Logical volume ---  
LV Path          /dev/home/home_directory  
LV Name          home_directory  
VG Name          home  
LV UUID          11TqBa-ulUI-UzXZ-7StQ-8RNW-ViOB-irW47S  
LV Write Access  read/write  
LV Creation host, time localhost.localdomain, 2022-10-13 20:34:26 -0700  
LV Status        available  
# open           1  
LV Size          300.00 MiB  
Current LE       75  
Segments         2  
Allocation       inherit  
Read ahead sectors auto  
- currently set to 8192  
Block device     253:0  
[root@localhost ~]# █
```

Gambar 153

2. Resize logical volume home_directory menjadi 400MB kemudian tampilkan hasilnya dengan perintah df -h.

```
# lvresize -r -L 400M /dev/home/home_directory
[root@localhost ~]# lvresize -r -L 400M /dev/home/home_directory
fsck from util-linux 2.23.2
/dev/mapper/home-home_directory: clean, 11/76912 files, 19977/307200 blocks
  Size of logical volume home/home_directory changed from 300.00 MiB (75 extents)
) to 400.00 MiB (100 extents).
  Logical volume home/home_directory successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/mapper/home-home_directory to 409600 (1k) blocks
.
The filesystem on /dev/mapper/home-home_directory is now 409600 blocks long.

[root@localhost ~]#
```

Gambar 154

```
# df -h
```

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        895M    0  895M   0% /dev
tmpfs          910M    0  910M   0% /dev/shm
tmpfs          910M   11M  900M   2% /run
tmpfs          910M    0  910M   0% /sys/fs/cgroup
/dev/sda3       18G   6.0G  12G  34% /
/dev/sdal      297M  287M  9.9M  97% /boot
tmpfs          182M   24K  182M   1% /run/user/0
[root@localhost ~]#
```

Gambar 155

3. Jika kapasitas masih dibutuhkan, tambahkan 50MB, seperti perintah berikut:

```
# lvresize -r -L +50M /dev/home/home_directory
[root@localhost ~]# lvresize -r -L +50M /dev/home/home_directory
Rounding size to boundary between physical extents: 52.00 MiB.
fsck from util-linux 2.23.2
/dev/mapper/home-home_directory: clean, 11/101200 files, 23297/409600 blocks
  Size of logical volume home/home_directory changed from 400.00 MiB (100 extent
s) to 452.00 MiB (113 extents).
  Logical volume home/home_directory successfully resized.
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/mapper/home-home_directory to 462848 (1k) blocks
.
The filesystem on /dev/mapper/home-home_directory is now 462848 blocks long.

[root@localhost ~]#
```

Gambar 156

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        895M    0  895M   0% /dev
tmpfs          910M    0  910M   0% /dev/shm
tmpfs          910M   11M  900M   2% /run
tmpfs          910M    0  910M   0% /sys/fs/cgroup
/dev/sda3       18G   6.0G  12G  34% /
/dev/sdal      297M  287M  9.9M  97% /boot
tmpfs          182M   24K  182M   1% /run/user/0
[root@localhost ~]#
```

Gambar 157

```
# df -h
```

4. Tambahkan logical volume `home_directory` pada file `/etc/fstab`, sehingga logical volume dapat di-mount ketika sistem diaktifkan. Pada line yang sama terapkan kuota untuk user dan group, sehingga penggunaan storage dapat dibatasi, seperti perintah berikut:

```
/dev/home/home_directory          /home_directory ext4 defaults,usrquota,grpquota      1 2
```

```
#  
# /etc/fstab  
# Created by anaconda on Mon Feb 21 02:42:29 2022  
#  
# 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  
#  
UUID=dc658591-d809-4491-0e7f6e0a0c03 /           xfs  defaults        0 0  
UUID=e6b78aec-715b-4b2a-9e11-7447e0b05d33 /boot    xfs  defaults        0 0  
UUID=c0335641-6e38-4f24-affc-bd3ac2b6119b swap     swap  defaults        0 0  
/dev/home/home_directory          /home_directory ext4 defaults,usrquota,grpquota      1 2
```

Gambar 158

5. Selanjutnya, untuk mengaktifkan pengaturan kuota yang telah didefinisikan pada file

/etc/fstab, maka jalankan perintah berikut:

```
quotacheck -cug /home_directory
```

```
[root@localhost ~]# quotacheck -cug /home_directory  
[root@localhost ~]# █
```

Gambar 159

Jika kuota berhasil diaktifkan, maka akan terbentuk dua file yaitu `aquota.user` dan `aquota.group`.

```
# cd /home_directory/  
# ls -l  
total 30  
-rw----- 1 root root 7168 Sep 29 06:51 aquota.group  
-rw----- 1 root root 7168 Sep 29 06:51 aquota.user  
[root@localhost ~]# cd /home_directory/  
[root@localhost home_directory]# ls -l  
total 26  
-rw----- 1 root root 6144 Oct 13 21:18 aquota.group  
-rw----- 1 root root 6144 Oct 13 21:18 aquota.user  
drwx----- 2 root root 12288 Oct 13 21:05 lost+found  
[root@localhost home directory]# █
```

Gambar 160

6. Selanjutnya, jika user ditambahkan pada sistem, maka home direktori akan diarahkan ke logical volume `/home_directory`. Hal ini mungkin dilakukan dengan mengubah file

`/etc/default/useradd`. Ubahlah variable `HOME` mengarah ke direktori home yang baru yaitu `/home_directory`. Uji cobalah apakah ketika user yang baru ditambahkan

```
GNU nano 2.3.1                                         File: /etc/default/useradd  
  
# useradd defaults file  
GROUP=100  
HOME=/home_directory  
INACTIVE=-1  
EXPIRE=  
SHELL=/bin/bash  
SKEL=/etc/skel  
CREATE_MAIL_SPOOL=yes
```

Gambar 161

maka direktori home yang terbentuk ada pada /home_directory.

7. Kuota dapat diterapkan pada single user ataupun group dengan menggunakan perintah edquota username/group. Buatlah sebuah user dengan username test1, seharusnya direktori home yang tercipta berada pada /home_directory, seperti berikut:

```
# pwd  
/home_directory/test1
```

```
[root@localhost home_directory]# ls  
aquota.group aquota.user lost+found test1  
[root@localhost home_directory]#
```

Gambar 162

8. Kemudian terapkan kuota untuk user test1 dengan menjalankan perintah edquota *username* misalnya edquota test1. Ketika perintah ini dijalankan kuota untuk user test1 dapat dikonfigurasi, tambahkan 55KB pada bagian hard, seperti gambar berikut:

```
1 Disk quotas for user test1 (uid 1003):  
2   Filesystem          blocks      soft      hard      inodes      soft      hard  
3   /dev/sda6              0          0          0          0          0          0          0  
4   /dev/mapper/home-home_directory    0          0      55000          0          0          0          0  
Disk quotas for user test1 (uid 1016):  
Filesystem          blocks      soft      hard      inodes      soft      hard  
/dev/mapper/home-home_directory    0          0      55000          0          0          0          0
```

Gambar 163

9. Lakukanlah uji coba untuk memastikan bahwa user hanya dapat menggunakan direktori home sesuai dengan kuota yang telah ditetapkan dengan menambahkan file pada direktori home user test1. Selanjutnya, untuk memeriksa kuota yang telah digunakan user, jalankan perintah seperti pada gambar berikut:

```
# quota test1  
Disk quotas for user test1 (uid 1003):  
  Filesystem  blocks  quota  limit  grace  files  quota  limit  grace  
/dev/mapper/home-home_directory    55001*    0    55000            1        0        0  
[root@localhost ~]# quota test1  
Disk quotas for user test1 (uid 1016): no limited resources used  
[root@localhost ~]#
```

Gambar 164

10. Untuk mendapatkan laporan penggunaan kuota untuk seluruh user dapat dilakukan dengan menjalankan perintah repquota -a, seperti

```
# repquota -a
*** Report for user quotas on device /dev/sda6
Block grace time: 7days; Inode grace time: 7days
          Block limits           File limits
User      used   soft   hard grace   used   soft   hard grace
-----
root     --     28     0     0           5     0     0     0

*** Report for user quotas on device /dev/mapper/home-home_directory
Block grace time: 7days; Inode grace time: 7days
          Block limits           File limits
User      used   soft   hard grace   used   soft   hard grace
-----
root     --     13     0     0           2     0     0     0
test1    +-   55001     0   55000       1     0     0     0
```

Gambar 165

```
*** Report for user quotas on device /dev/mapper/home-home_directory
Block grace time: 7days; Inode grace time: 7days
          Block limits           File limits
User      used   soft   hard grace   used   soft   hard grace
-----
root     --     13     0     0           2     0     0     0

[root@localhost ~]#
```

Gambar 166

12. Jika kapasitas pada sebuah volume group perlu ditambah, maka dapat dilakukan dengan menjalankan perintah vgextend. Sebelum menambah kapasitas dari hard drive lainnya, harus dipastikan bahwa hard drive tersebut belum dialokasikan menjadi filesystem untuk program lain, dengan perintah fdisk -l. Misalnya, logical volume home yang dialokasikan untuk direktori home kapasitasnya perlu ditambah dan terdapat hard drive

```
# vgextend home /dev/sdg
[root@localhost ~]# vgextend home /dev/sdg
Physical volume "/dev/sdg" successfully created.
Volume group "home" successfully extended
[root@localhost ~]# █
```

Gambar 167

yang free pada /dev/sdg, maka perintah yang akan dijalankan seperti berikut:

Periksalah apakah kapasitas untuk logical volume home sudah bertambah.

13. Selanjutnya, hard drive (physical volume) dapat dipisahkan dari volume group, namun harus dipastikan terlebih dahulu bahwa physical volume tersebut tidak sedang digunakan oleh logical volume dengan menjalankan perintah `pvdisplay`. Misalnya, hapuslah physical volume `/dev/sdg` yang telah dialokasikan pada volume group `home`, dengan menjalankan perintah berikut:

```
# vgreduce home /dev/sdg
[root@localhost ~]# vgreduce home /dev/sdg
Removed "/dev/sdg" from volume group "home"
[root@localhost ~]# pvdisplay
--- Physical volume ---
PV Name           /dev/sde
VG Name           home
PV Size          262.00 MiB / not usable 2.00 MiB
Allocatable       yes (but full)
PE Size          4.00 MiB
Total PE         65
Free PE          0
Allocated PE     65
PV UUID          IgnZZv-r33d-Hgkz-F334-ip5X-dG2R-72VdSw

--- Physical volume ---
PV Name           /dev/sdf
VG Name           home
PV Size          262.00 MiB / not usable 2.00 MiB
Allocatable       yes
PE Size          4.00 MiB
Total PE         65
Free PE          17
Allocated PE     48
PV UUID          9Kec1C-m6aP-oa2j-L10V-3WmA-ZQ2z-yhWkeq

"/dev/sdg" is a new physical volume of "262.00 MiB"
```

Gambar 168

Jalankanlah kembali instruksi pada nomor 17, kemudian bagilah volume group dengan membentuk volume group yang baru. Namun sebelumnya dipastikan bahwa logical volume tidak dapat dibagi pada volume group yang berbeda. Setiap logical volume yang sudah terbentuk harus berada pada physical volume yang telah digabungkan pada sebuah volume group apakah volume group yang baru atau volume group yang lama. Misalnya, membagi volume group yang baru dengan volume group yang telah ada, seperti

```
# vgsplit home new_home /dev/sdg
```

perintah berikut:

```
[root@localhost ~]# vgextend home /dev/sdg
Volume group "home" successfully extended
[root@localhost ~]# vgsplit home new_home /dev/sdg
New volume group "new_home" successfully split from "home"
[root@localhost ~]# █
```

Gambar 169

BAB 7. SAMBA DOMAIN CONTROLLER

A. Instalasi Samba pada CentOS

Langkah-langkah untuk melakukan instalasi Samba pada CentOS:

1. Instal samba-client dengan mengeksekusi perintah berikut

```
yum install samba samba-client
```

Gambar 170

2. Setelah proses instalasi selesai, jalankan servis samba dan jalankan perintah yang akan menjalankan servis secara otomatis pada saat system boot. Eksekusi perintah berikut:

```
# systemctl start smb.service  
# systemctl start nmb.service  
# systemctl enable smb.service  
# systemctl enable nmb.service
```

Gambar 171

Servis smbd menyediakan layanan berbagi file (sharing file) dan layanan cetak (printing) yang aktif pada port TCP 139 dan 445. Selanjutnya, servis nmbd menyediakan layanan NETBIOS melalui layanan penamaan IP terhadap klien yang aktif pada port UDP 137.

```
[root@localhost snadapdap]# systemctl start smb.service  
[root@localhost snadapdap]# systemctl start nmb.service  
[root@localhost snadapdap]# systemctl enable smb.service  
Created symlink from /etc/systemd/system/multi-user.target.wants/smb.service to /usr/lib/systemd/system/smb.service.  
[root@localhost snadapdap]# systemctl enable nmb.service  
Created symlink from /etc/systemd/system/multi-user.target.wants/nmb.service to /usr/lib/systemd/system/nmb.service.  
[root@localhost snadapdap]# █
```

Gambar 172

Setelah Samba berhasil maka seluruh port yang penting harus terbuka, dengan menjalankan perintah berikut

```
firewall-cmd --permanent --zone=public --add-service=samba  
firewall-cmd --zone=public --add-service=samba
```

```
/usr/bin/systemctl start samba
[root@localhost snadapdap]# firewall-cmd --permanent --zone=public --add-service=samba
success
[root@localhost snadapdap]# firewall-cmd --zone=public --add-service=samba
success
[root@localhost snadapdap]#
```

Gambar 173

B. Konfigurasi Samba

Studi kasus yang digunakan pada praktikum ini adalah modul Pengelolaan User dan Group pada Linux. Pada modul ini diberikan satu contoh penerapan Samba untuk divisi Finance, dengan langkah-langkah sebagai berikut:

1. Tambahkan group Finance dengan menggunakan perintah groupadd.

```
# groupadd finance
```

```
[root@localhost snadapdap]# groupadd finance
[root@localhost snadapdap]# cat /etc/group
root:x:0:
vduemotl:x:1000:
finance:x:1004:
[root@localhost snadapdap]#
```

Gambar 174

2. Buatlah folder data yang kemudian setelahnya didalam folder /data, sama seperti pada modul Pengelolaan User dan Group. Langkah ini bisa dilewatkan jika folder telah dipersiapkan pada praktikum sebelumnya.

```
# mkdir data
```

```
[root@localhost snadapdap]# mkdir data
[root@localhost snadapdap]# cd data
[root@localhost data]#
```

Gambar 175

3. Tambahkan folder finance dibawah folder data dengan path folder /data/finance.

```
# mkdir finance
```

```
[root@localhost data]# mkdir finance
[root@localhost data]# cd finance
[root@localhost finance]# pwd
/home/snadapdap/data/finance
[root@localhost finance]# █
```

Gambar 176

4. Ketiga perintah berikut dijalankan untuk mengubah mode akses folder /data/finance menjadi 770 secara rekursif. Kemudian ubah konteks SELinux security menjadi samba_share_t pada folder /data/finance. Selanjutnya, ubah hak kepemilikan folder /data/finance dengan root sebagai pemilik folder dan juga group finance.

```
chmod -R 770 /data/finance/
chcon -t samba_share_t /data/finance
chown -R root:finance /data/finance
```

```
[root@localhost finance]# pwd
/home/snadapdap/data/finance
[root@localhost finance]# chmod -R 770 /home/snadapdap/data/finance/
[root@localhost finance]# chcon -t samba_share_t /home/snadapdap/data/finance
[root@localhost finance]# chown -R root:finance /home/snadapdap/data/finance
[root@localhost finance]# █
```

Gambar 177

5. Berikutnya, dari user yang sudah diciptakan pada modul praktikum sebelumnya, contohnya bylardo dari bagian Finance, maka user bylardo harus juga ditetapkan sebagai user samba. Smbpasswd adalah perintah yang digunakan untuk menciptakan user samba seperti perintah berikut.

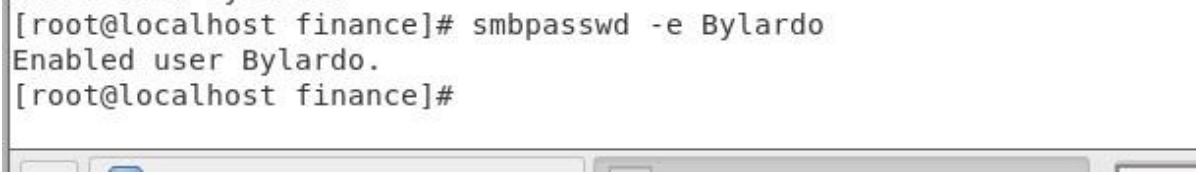
```
# smbpasswd -a bylardo
New SMB password:
Retype new SMB password:
Added user bylardo.
```

```
[root@localhost ~]# cd finance
[root@localhost finance]# smbpasswd -a Bylardo
New SMB password:
Retype new SMB password:
Added user Bylardo.
[root@localhost finance]# █
```

Gambar 178

6. Setelah akun bylardo ditetapkan menjadi user samba, selanjutnya user bylardo diaktifkan dengan menggunakan perintah smbpasswd dengan argumen -e.

```
# smbpasswd -e bylardo  
Enabled user bylardo.
```



```
[root@localhost finance]# smbpasswd -e Bylardo  
Enabled user Bylardo.  
[root@localhost finance]#
```

Gambar 179

7. Setelah itu, lakukan konfigurasi pada file /etc/samba/smb.conf. Sebelumnya, sangat dihindari untuk memodifikasi file konfigurasi yang original. Oleh karena itu sebelum menambahkan konfigurasi maka buatkan file cadangan.

Duplikat file smb.conf menjadi smb.conft.ori

```
[root@localhost finance]# sudo cp /etc/samba/smb.conf /etc/samba/smb.conft.ori
```

Gambar 180

Tambahkan konfigurasi folder berbagi pada file konfigurasi /etc/samba/smb.conf seperti konfigurasi berikut.

- [Finances] ↳ nama folder berbagi yang akan terlihat
- Path ↳ hierarki folder yang akan dibagi melalui servis samba, dalam hal ini adalah /data/finance.
- Browseable ↳ folder berbagi dapat diakses dan dieksplor
- Writeable ↳ isi folder dapat ditulis dengan menambahkan folder atau file.
- Read only ↳ folder tidak dibatasi hanya dengan mode baca
- Valid user ↳ user valid yang dapat mengakses folder berbagi.

```
[Finances]  
path = /data/finance  
browseable = yes  
writeable = yes  
read only = no  
valid users = @finance
```

Gambar 181

Tambahkan potongan teks diatas setelah dibuka nano etc/samba/smb.conf seperti berikut:

```
snadapdap@localhost:/home/snadapdap/data/finance
File Edit View Search Terminal Help
GNU nano 2.3.1           File: /etc/samba/smb.conf          Modified

path = /var/lib/samba/drivers
write list = @printadmin root
force group = @printadmin
create mask = 0664
directory mask = 0775

[Finances]
path = /home/snadapdap/data/finance
browseable = yes
writeable = yes
read only = no
valid users = @finance
```

Gambar 182

8. Jika ada perubahan yang diterapkan pada file konfigurasi samba, maka diakhiri dengan mengaktifkan kembali servis dari samba dan netbios, seperti perintah berikut.

```
# systemctl restart smb.service
# systemctl restart nmb.service
```

```
[root@localhost finance]# systemctl restart smb.service
[root@localhost finance]# systemctl restart nmb.service
[root@localhost finance]#
```

Gambar 183

C. Akses samba dari Windows

Setelah Samba dikonfigurasi maka dilakukan pengujian akses terhadap folder berbagi yang dalam hal ini adalah folder /data/finance. Langkah-langkah berikut untuk mengakses folder berbagi melalui sistem operasi Windows.

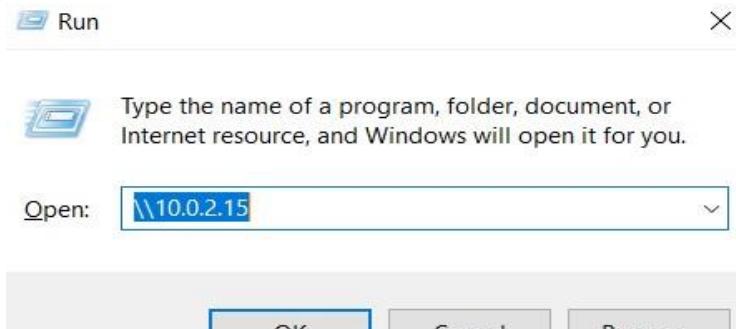
1. Akseslah folder berbagi melalui IP Address server samba yang telah dikonfigurasi, sebagai contoh seperti gambar berikut.

Contoh:

```
[root@localhost finance]# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
  inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::fa4c:216:cb7b:9686 prefixlen 64 scopeid 0x20<link>
      ether 08:00:27:7e:4c:df txqueuelen 1000 (Ethernet)
      RX packets 187 bytes 220398 (215.2 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 178 bytes 22446 (21.9 KiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
  inet 127.0.0.1 netmask 255.0.0.0
  inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 114 bytes 13008 (12.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 114 bytes 13008 (12.7 KiB)
```

Gambar 184



Gambar 185

2. Jika server samba dapat diakses maka muncul tampilan berikut dengan cara memberikan username bylardo dan password bylardo.



Gambar 186

3. Jika akun bylardo berhasil login, maka folder Finance akan muncul pada folder berbagai sesuai dengan yang telah dikonfigurasi pada konfigurasi samba.

4. Selanjutnya, dilakukan uji coba dengan membuat folder yang baru.

12 > Finances

Name	Date modified	Type	Size
test	30/11/2022 20:03	File folder	

Gambar 187

BAB 8. PENGELOLAAN OBJEK PADA ACTIVE DIRECTORY MENGGUNAKAN POWERSHELL

Pengelolaan Objek Active Directory dengan Menggunakan PowerShell

Dengan memperkenalkan bahasa skrip PowerShell, Microsoft memberikan lebih banyak kontrol terhadap sistem dan operasi Windows. PowerShell juga memungkinkan enginer untuk menggunakan fitur tambahan yang tidak dapat digunakan melalui GUI. Terdapat module Active Directory PowerShell, yang dapat digunakan untuk mengelola Active Directory Domain services, Active Directory Lightweight Directory Services AD LDS), dan objek. Objek pada Active Directory masih tetap dapat dikelola menggunakan Command Prompt, tetapi PowerShell menyediakan pengelolaan komponen dan operasi pada Active Directory dengan cara yang lebih advance. Setiap Server yang menjalankan AD DS atau AD LDS akan memiliki modul Active Directory PowerShell secara default.

A.1 Menciptakan, Memodifikasi dan Menghapus Objek di Active Directory

Menciptakan, memodifikasi dan menghapus objek adalah tugas pengelolaan AD yang dilakukan secara umum. Operasi ini dapat dijalankan dengan menggunakan GUI, namun jika sudah menyangkut operasi yang lebih kompleks dan juga waktu eksekusi operasi, maka penggunaan skrip pada PowerShell dapat menjadi pilihan. Oleh karena itu, rekomendasi terbaik adalah untuk menggunakan beberapa cara (GUI/Skrip) tergantung dari tugas administrasi yang dijalankan.

A.1.1 Menciptakan Objek pada Active Directory

Setiap objek memiliki sekumpulan atribut yang berbeda-beda. Ketika Anda menciptakan objek, maka nilai untuk setiap atribut tersebut harus ditetapkan. Beberapa dari atribut ini bersifat wajib, dan beberapa diantaranya tidak. Anda harus menyesuaikan atribut apa saja yang akan digunakan dengan mengacu kepada preferensi organisasi/perusahaan yang Anda kelola.

Menciptakan Objek User

Untuk menciptakan objek User pada Active Directory, dapat menggunakan cmdlet NewADUser di PowerShell. Anda dapat melihat sintaks penuh dari perintah tersebut beserta dengan tipe data yang diterima.

`Get-Command New-ADUser -Syntax`

```

PS C:\Users\Administrator> Get-Command New-ADUser -Syntax

New-ADUser [-Name] <string> [-WhatIf] [-Confirm] [-AccountExpirationDate <datetime>] [-AccountNotDelegated <bool>] [-AccountPassword <securestring>] [-AllowReversiblePasswo
rdEncryption <bool>] [-AuthenticationPolicy <ADAuthenticationPolicy>] [-AuthenticationPolicySilo <ADAuthenticationPolicySilo>] [-AuthType <ADAuthType>] [-CannotChangePasswo
rd <bool>] [-Certificates <X509Certificate[]>] [-ChangePasswordAtLogon <bool>] [-City <string>] [-Company <string>] [-CompoundIdentitySupported <bool>] [-Country <string>]
[-Credential <pscredential>] [-Department <string>] [-Description <string>] [-DisplayN
ame <string>] [-Division <string>] [-EmailAddress <string>] [-EmployeeID <string>] [-E
mployeeNumber <string>] [-Enabled <bool>] [-Fax <string>] [-GivenName <string>] [-Home
Directory <string>] [-HomeDrive <string>] [-HomePage <string>] [-HomePhone <string>] [-
Initials <string>] [-Instance <ADUser>] [-KerberosEncryptionType <ADKerberosEncryptio
nType>] [-LogonWorkstations <string>] [-Manager <ADUser>] [-MobilePhone <string>] [-Of
fice <string>] [-OfficePhone <string>] [-Organization <string>] [-OtherAttributes <has
htable>] [-OtherName <string>] [-PassThru] [-PasswordNeverExpires <bool>] [-PasswordNo
tRequired <bool>] [-Path <string>] [-POBox <string>] [-PostalCode <string>] [-Principa
lAllowedToDelegateToAccount <ADPrincipal[]>] [-ProfilePath <string>] [-SamAccountName
<string>] [-ScriptPath <string>] [-Server <string>] [-ServicePrincipalNames <string[]>
] [-SmartcardLogonRequired <bool>] [-State <string>] [-StreetAddress <string>] [-Sur
name <string>] [-Title <string>] [-TrustedForDelegation <bool>] [-Type <string>] [-User
PrincipalName <string>] [<CommonParameters>]

PS C:\Users\Administrator>

```

Gambar 188

Untuk menciptakan user yang baru dengan menggunakan PowerShell, parameter minimum yang harus digunakan adalah **-Name**, yang kemudian dapat diikuti dengan parameter lainnya. Berikut adalah contoh untuk membuat sebuah akun user:

New-ADUser -Name "Toni Spark" -GivenName "Toni" -Surname "Spark" -SamAccountName "tspark" -UserPrincipalName tspark@hogwarts.edu -Path "OU=IT,DC=hogwarts,DC=edu" -AccountPassword (Read-Host -AsSecureString "Type Password for User") -Enabled \$true

Perintah ini memiliki parameter sebagai berikut

- Name** Menentukan nama penuh
- GivenName** Menentukan nama depan
- Surname** Menentukan nama belakang
- SamAccountName** Menentukan nama pengguna (username)
- UserPrincipalName** Menentukan UPN untuk akun pengguna
- Path** Menentukan path OU. Jika path tidak ditentukan maka objek akan diletakkan pada default container.

g. **-AccountPassword** ↗ Menentukan password pengguna sesuai dengan masukan dari pengguna, yang kemudian password akan dikonversi oleh sistem sesuai dengan tipe data yang relevan.

h. **-Enabled** ↗ Menentukan apakah status akun dinonaktifkan atau diaktifkan.

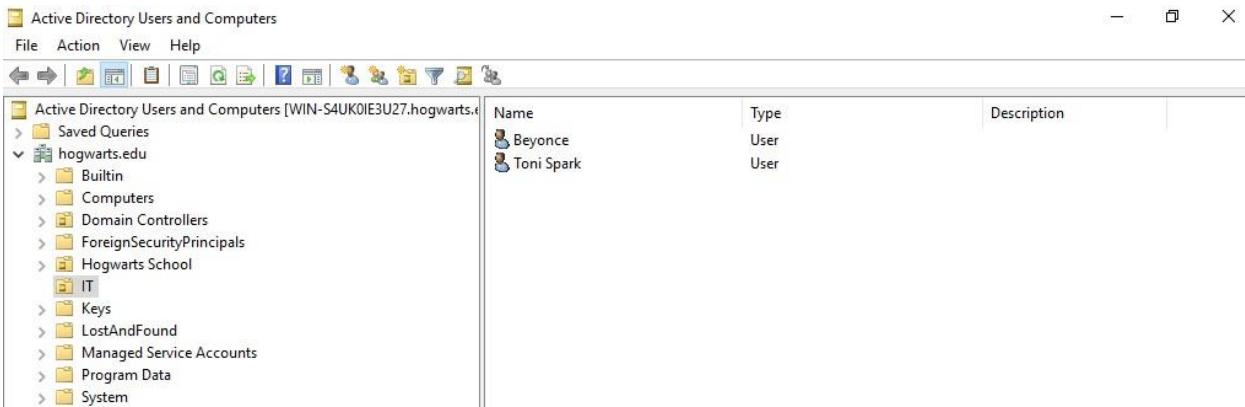
Hasil eksekusi perintah diatas ditampilkan pada gambar dibawah.

The screenshot shows a Windows Server 2016 environment within Oracle VM VirtualBox. A PowerShell window titled 'Administrator: Windows PowerShell ISE' is open. The command PS C:\Users\Administrator> New-ADUser -Name "Toni Spark" -GivenName "toni" -Surname "Spark" Read-Host -AsSecureString was entered, but it failed because 'Read-Host -AsSecureString' is not recognized as a cmdlet. The command PS C:\Users\Administrator> New-ADUser -Name "Toni Spark" -GivenName "toni" -Surname "Spark" was then successfully run, creating a user object with the following properties:

Property	Value
DistinguishedName	CN=Toni Spark,OU=IT,DC=hogwarts,DC=edu
Enabled	True
GivenName	toni
Name	Toni Spark
ObjectClass	user
ObjectGUID	e8205b61-ac59-4919-964c-89a99648d07a
SamAccountName	tspark
SID	S-1-5-21-3926235987-3091820138-3660749175-1115
Surname	Spark
UserPrincipalName	tspark@hogwarts.edu

Gambar 189

Hasil dari eksekusi perintah diatas dapat dilihat pada Active Directory User and Computer. Pada OU IT telah berhasil diciptakan sebuah user dengan nama Toni Spark, seperti yang ditampilkan pada gambar dibawah.



Gambar 190

1. Menambahkan user dengan menggunakan skrip

Selain mengeksekusi beberapa perintah untuk menciptakan beberapa objek user, maka operasi ini dapat dilakukan dengan menggunakan file comma-separated values (CSV) yang berisi data atribut dan menggunakan file ini untuk menciptakan akun dalam sekali jalan. Untuk demonstrasi, buatlah sebuah file .csv yang berisi data dibawah ini.

The screenshot shows a Microsoft Notepad window titled 'ADUser1 - Notepad'. The file contains the following CSV data:

```

Name,GivenName,Surname,SamAccountName,UserPrincipalName,Path
Coba1User1,Coba1,User1,cuser1@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba2User2,Coba2,User2,cuser2@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba3User3,Coba3,User3,cuser3@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba4User4,Coba4,User4,cuser4@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba5User5,Coba5,User5,cuser5@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba6User6,Coba6,User6,cuser6@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba7User7,Coba7,User7,cuser7@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba8User8,Coba8,User8,cuser8@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba9User9,Coba9,User9,cuser9@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"
Coba10User10,Coba10,User10,cuser10@hogwarts.edu, "OU-IT,DC=hogwarts,DC=edu"

```

Gambar 191

Baris pertama pada file .csv merupakan atribut, sedangkan baris-baris berikutnya adalah nilai dari atribut. Nilai dari atribut akan digunakan dengan menggunakan skrip berikut.

```
1 Import-Csv "C:\Users\Administrator\Desktop\AD\ADusers_1.csv" | ForEach-Object {
2     $upn = $_.SamAccountName + "hogwarts.edu"
3
4     New-ADUser -Name $_."Name"
5     -GivenName $_."GivenName"
6     -Surname $_."Surname"
7     -SamAccountName $_."SamAccountName"
8     -UserPrincipalName $upn
9     -Path $_."Path"
10    -AccountPassword (ConvertTo-SecureString "Pa$$wOrd_2018" -AsPlainText -Force) -Enabled $true
11 }
```

Gambar 192

Pada skrip diatas, perintah Import-Csv digunakan untuk mengimport file CSV yang telah dibuat. Pada skrip diset nilai variabel \$upn = \$_SamAccountName + "@hogwarts.edu", nilai variabel \$upn akan menjadi nilai untuk paramater -UserPrincipalName. Selanjutnya, pada skrip ditentukan password untuk seluruh akun menggunakan -accountPassword (ConvertTo-SecureString "Pa\$\$word_2018" -AsPlainText -force), seperti gambar dibawah.

```
Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
PS C:\Users\Administrator> Import-Csv "C:\Users\Administrator\Desktop\AD\ADUsers_1.csv"
$upn = $_.SamAccountName + ".hogwarts.edu"

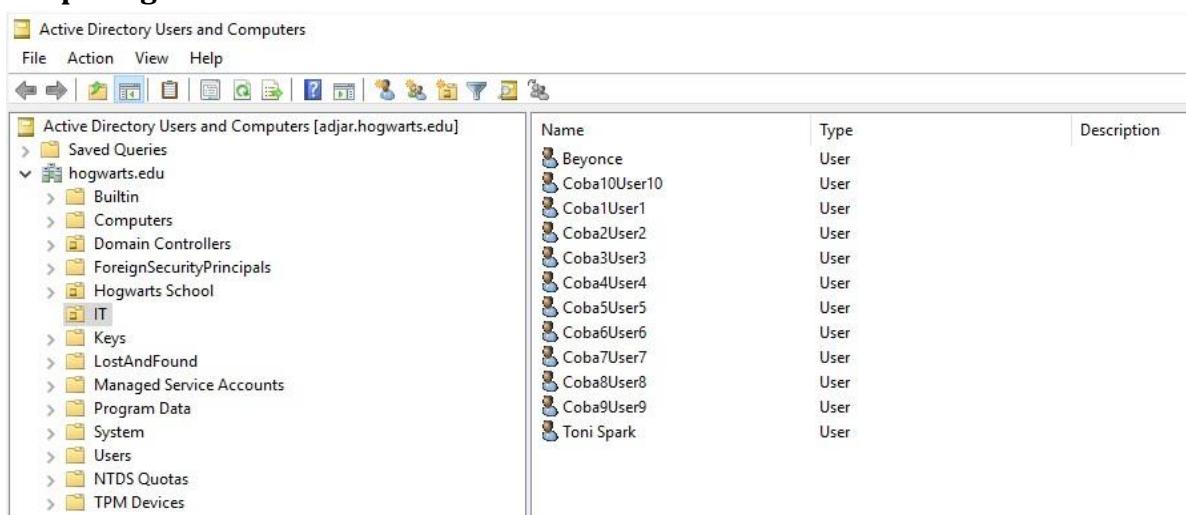
New-ADUser -Name $_."Name"
-GivenName $_."GivenName"
-Surname $_."Surname"
-SamAccountName $_."SamAccountName"
-UserPrincipalName $upn
-Path $_."Path"
-AccountPassword (ConvertTo-SecureString "Pa$$w0rd_2018" -AsPlainText)
}

PS C:\Users\Administrator>
```

```
Untitled1.ps1* X
1 Import-Csv "C:\Users\Administrator\Desktop\AD\ADUsers_1.csv"
2 $upn = $_.SamAccountName + ".hogwarts.edu"
3
4 New-ADUser -Name $_."Name"
5 -GivenName $_."GivenName"
6 -Surname $_."Surname"
7 -SamAccountName $_."SamAccountName"
8 -UserPrincipalName $upn
9 -Path $_."Path"
10 -AccountPassword (ConvertTo-SecureString "Pa$$w0rd_2018"
11
```

Gambar 193

Hasil eksekusi skrip tersebut dapat diperiksa pada Active Directory Users and Computers tepatnya pada OU IT, telah tercipta user Coba1User1 hingga Coba10User10, seperti pada gambar berikut.



Gambar 194

2. Modifikasi objek pada Active Directory

Ketika menciptakan objek-objek pada Active Directory, nilai untuk setiap atribut yang berbeda akan ditetapkan. Setelah objek diciptakan, akan ada situasi yang dibutuhkan untuk memodifikasi nilai dari atribut atau mengisi nilai dari atribut yang kosong. Perintah SetADUser dapat digunakan untuk mengubah dan menambahkan nilai atribut pada objek user AD yang sudah ada sebelumnya.

```
Set-ADUser Coba1User1 -OfficePhone "0923456790" -City "London"
```

Pada perintah diatas, nilai untuk atribut -OfficePhone dan City ditambahkan pada user Coba1User1.

Ada suatu waktu, ketika perubahan pada sebuah atribut diperlukan.

```
Set-ADUser Coba1User1 -OfficePhone "012221456790"
```

Pada perintah diatas, nilai atribut -OfficePhone digantikan dengan nilai yang baru.

Dari perintah diatas, modifikasi dilakukan pada user tertentu, tetapi hal ini bukan merupakan praktik yang tepat jika harus melakukannya pada jumlah user yang besar. Oleh karena itu, untuk perlu ada kombinasi antar perintah Set-ADUser dengan Get-ADUser. Dengan menggunakan kombinasi ini, maka objek akan dicari terlebih dahulu kemudian modifikasi dilakukan.

```
Get-AdUser -Filter { -SearchBase 'OU=IT,DC=hogwarts,DC=edu' } | Set-ADUser -City "London"
```

Dari perintah diatas, seluruh objek user yang berada pada lokasi OU=IT,DC=hogwarts,DC=edu akan dicari kemudian diubah nilai atribut City menjadi London.

Selanjutnya, seluruh user memiliki nilai atribut City adalah London dicari dan diubah nilainya menjadi Manchester.

```
Get-AdUser -Filter {City -like "London"} | Set-ADUser -City "Manchester"
```

Kombinasi query pencarian dengan modifikasi objek meringankan pekerjaan yang dilakukan secara manual, dan pekerjaan yang tidak mudah dilakukan jika harus menggunakan GUI.

3. Menghapus objek pada Active Directory

Menghapus objek pada AD sama mudahnya dengan modifikasi proses. Namun, hal yang lebih menantang adalah untuk menemukan objek yang harus dihapus. Untuk menghapus objek AD, dapat menggunakan perintah Remove-ADUser, seperti perintah berikut.

```
Get-Command Remove-ADUser -Syntax
```

Ketika menggunakan perintah Remove-ADUser harus melewatkkan parameter -Identity untuk akun tertentu, seperti perintah berikut.

```
Remove-ADUser -Identity "beyonce"
```

Perintah diatas akan menghapus objek user AD yaitu beyonce. Sebelum, objek dihapus akan muncul konfirmasi.

Perintah untuk menghapus user juga dapat dikombinasikan dengan query pencarian untuk menemukan objek, seperti perintah berikut.

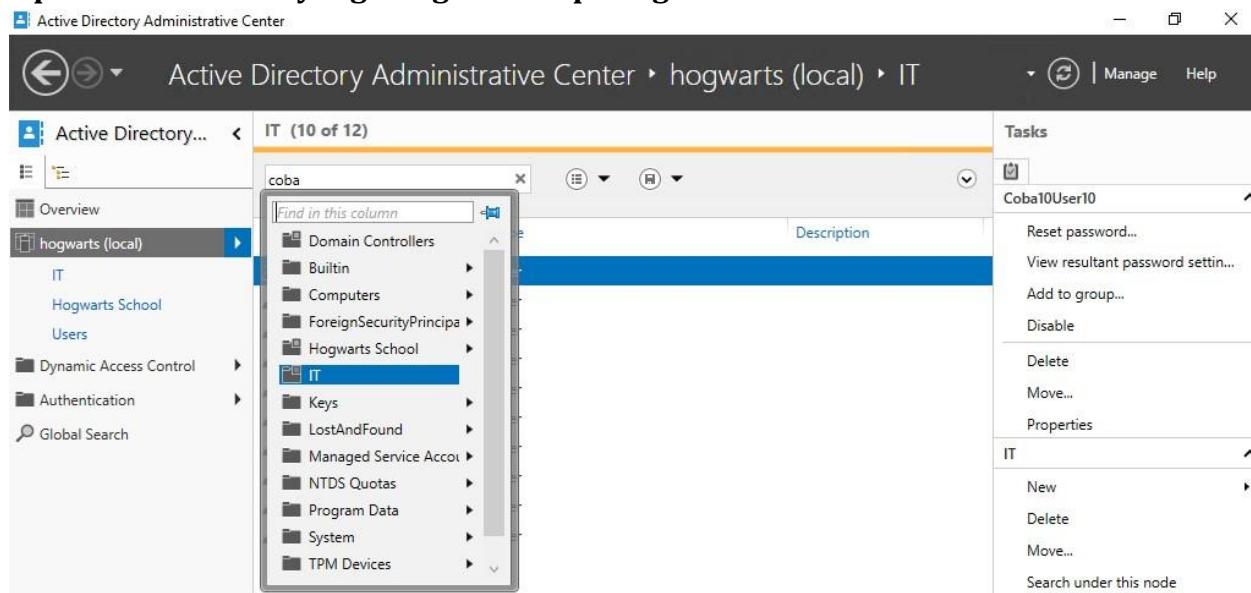
```
Get-ADUser -Filter {Name -like "Coba"} | Remove-ADUser
```

Pada perintah diatas, pengguna yang namanya diawali Coba pada seluruh direktori akan dihapus.

4. Mencari objek pada Active Directory

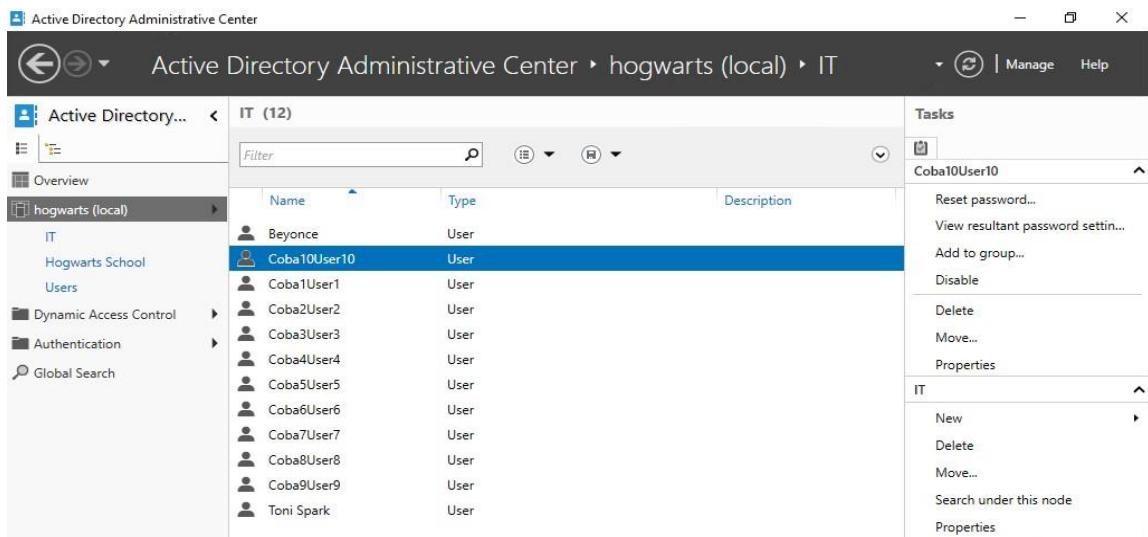
Active Directory Administrative Center (ADAC) telah meningkatkan kemampuan query dan filter objek pada AD, melalui sebuah GUI yang akan menarik hasil lebih cepat dibandingkan dengan PowerShell.

Pada ADAC, ketika klik kanan pada nama domain terdapat kotak filter untuk mendapatkan kontainer yang diinginkan seperti gambar dibawah.



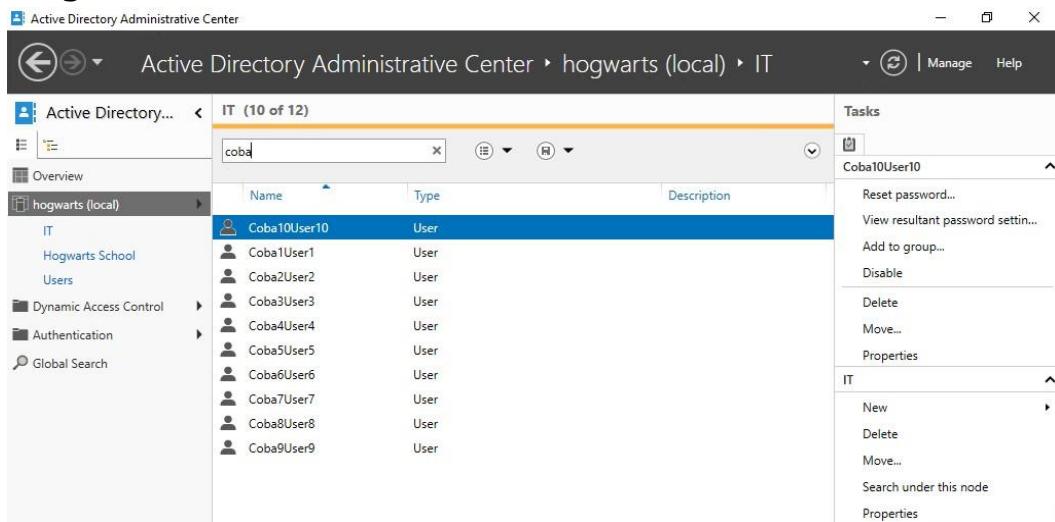
Gambar 195

Pada contoh ini akan dicari seluruh user yang berada pada OU IT, seperti pada gambar dibawah.



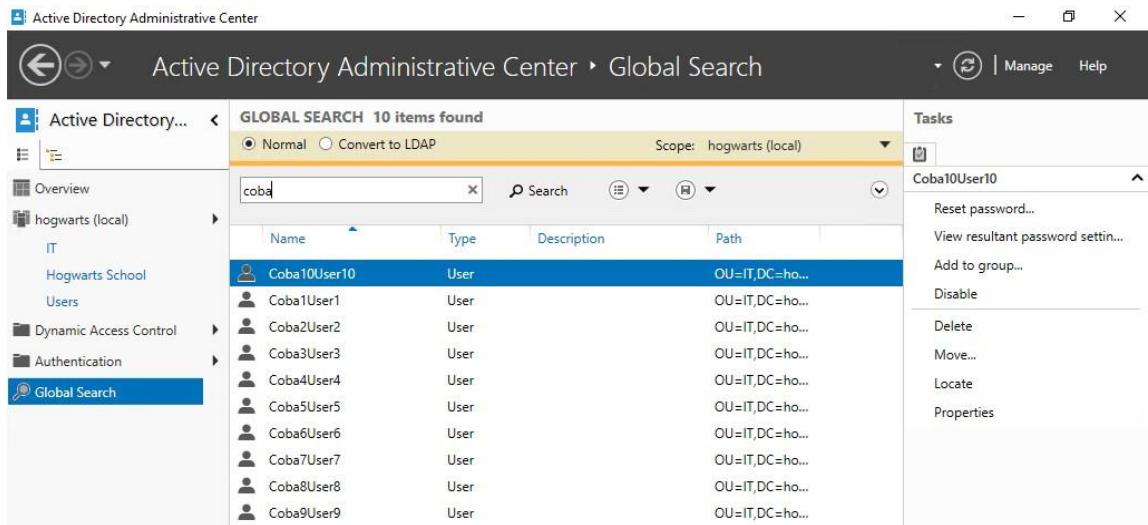
Gambar 196

Selanjutnya pada field pencarian ketikkan user yang diawali dengan nama Coba, sehingga akan dihasilkan seluruh user yang namanya diawali dengan Coba, seperti gambar dibawah.

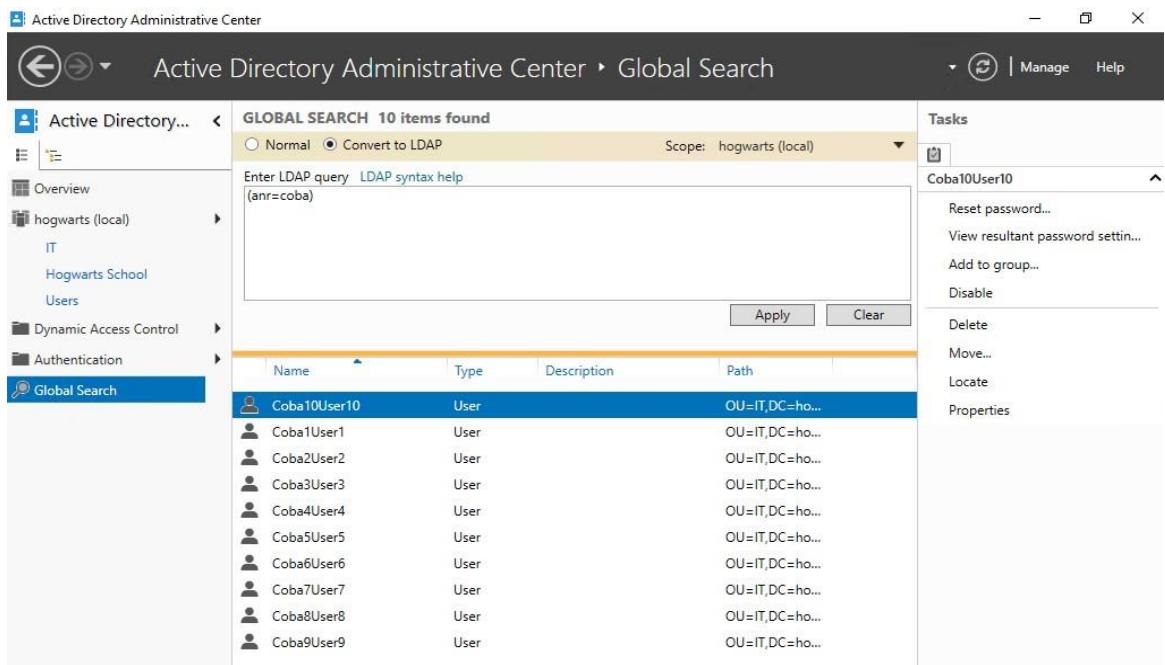


Gambar 197

Selain itu ADAC juga memiliki fitur yang disebut sebagai Global Search, yang dapat digunakan untuk mencari objek diseluruh direktori. Fitur ini memungkinkan pencarian berbasis teks atau querie berbasis LDAP, seperti gambar dibawah.

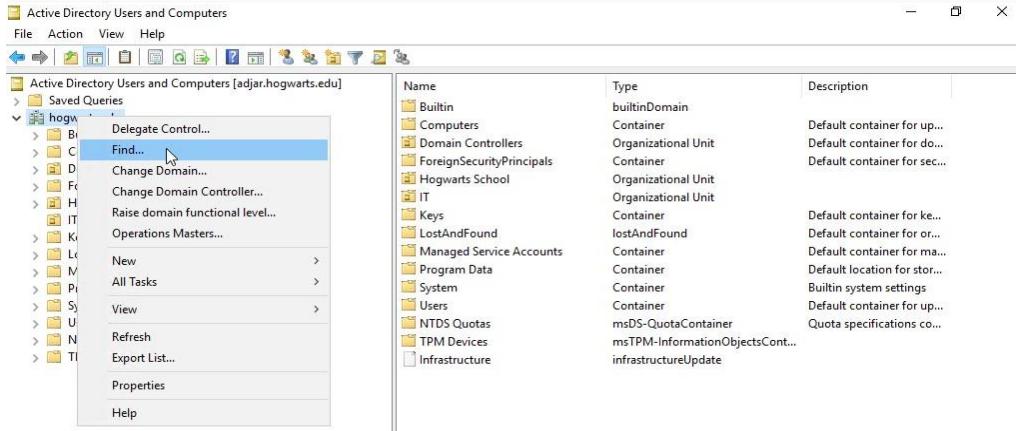


Gambar 198



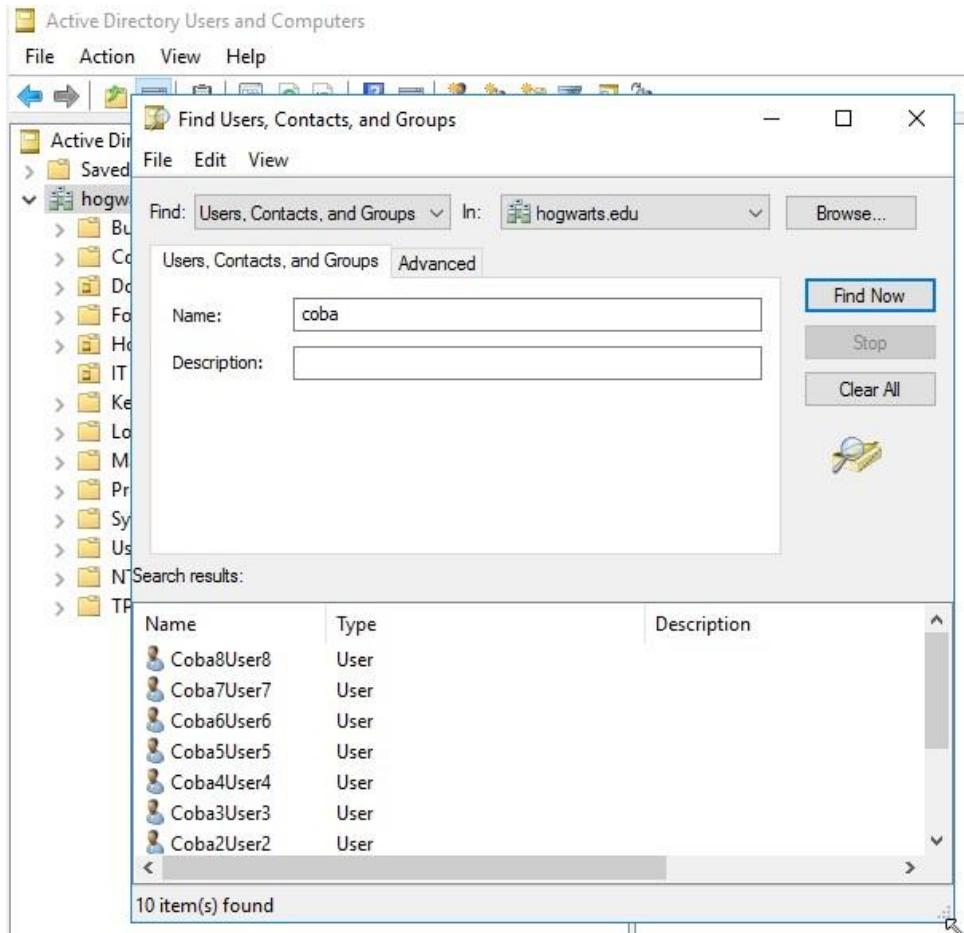
Gambar 199

Pencarian juga bisa dilakukan melalui Active Directory User and Computer ADUC. Fitur Find dapat ditemukan dengan klik kanan pada setiap kontainer, seperti pada gambar dibawah.



Gambar 200

Fitur Find juga memungkinkan untuk melakukan pencarian berbasis teks juga pencarian yang bersifat advance berdasarkan atribut dan nilainya, seperti pada gambar dibawah.



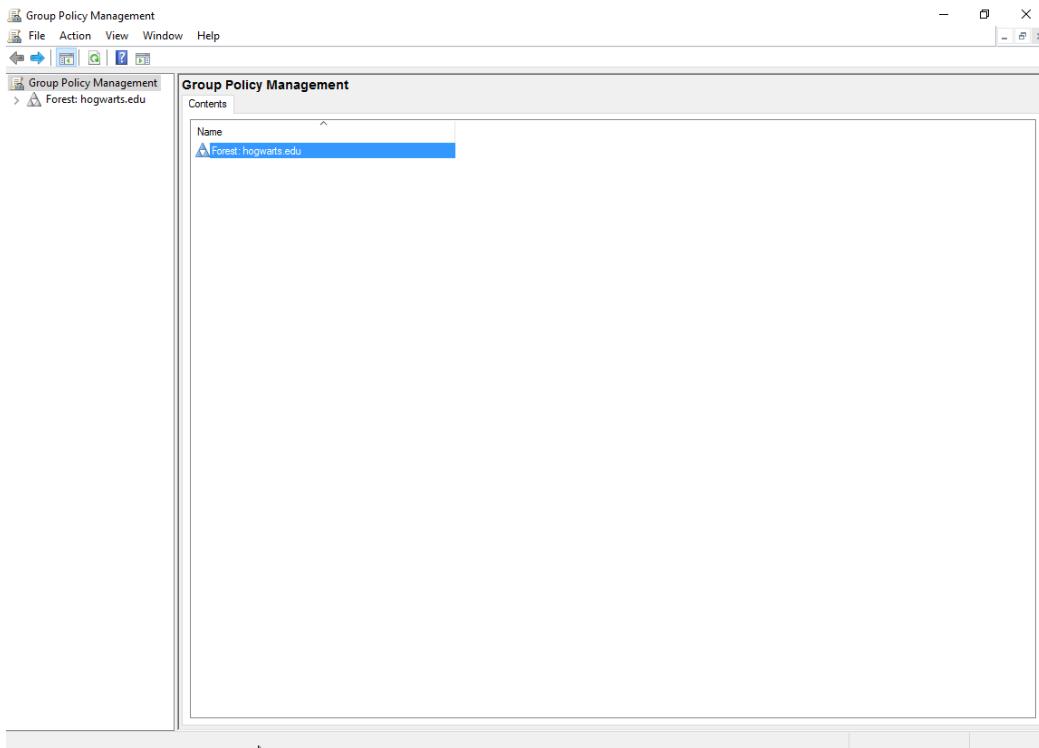
Gambar 201

BAB 9. MANAGING GROUP POLICY & ADVANCED AD MANAGEMENT WITH POWERSHELL

A. Pengelolaan Group Policy pada AD

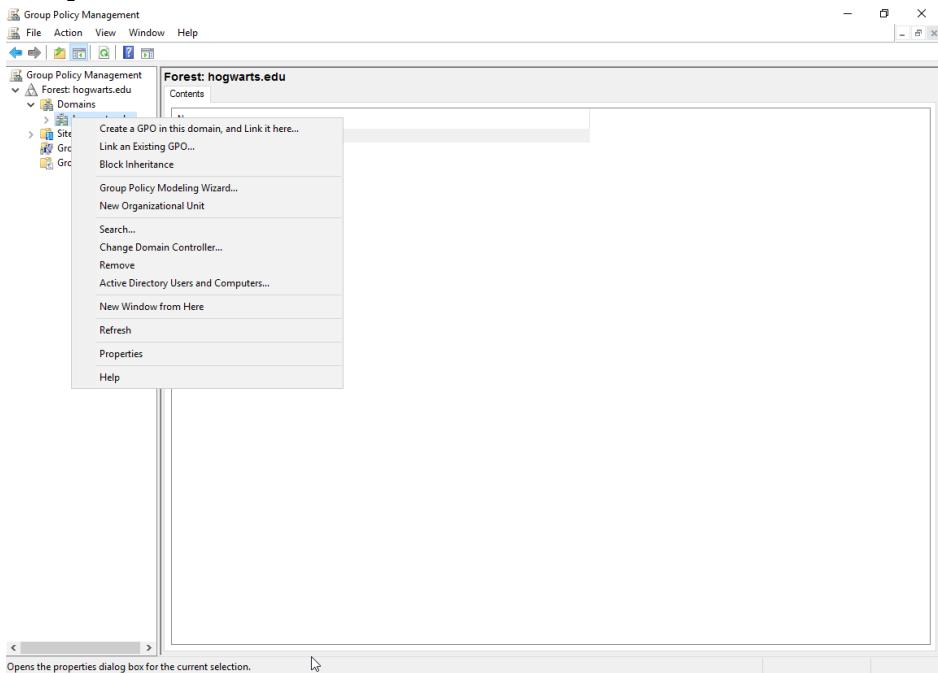
Untuk menciptakan Group Policy Object(GPO) pada sebuah AD dapat dilakukan melalui langkah-langkah berikut:

1. Buka fitur Group Policy Management melalui server manager. Melalui GP Management, dapat ditetapkan berbagai jenis policy kepada Organizational Unit (OU).



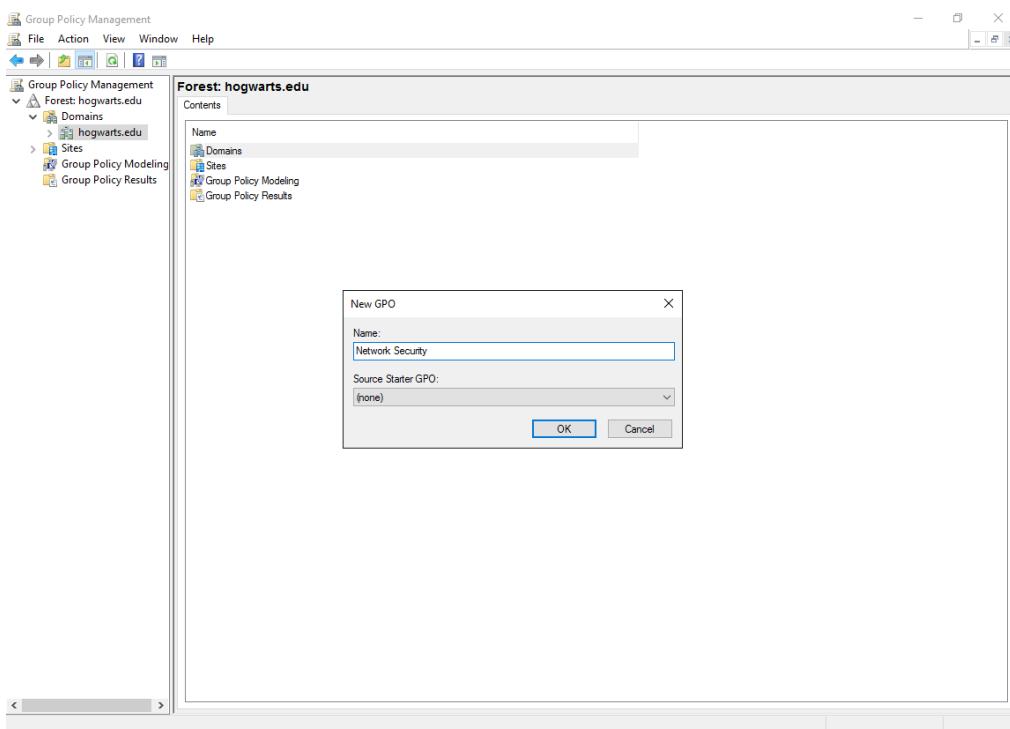
Gambar 202

2. Setelah fitur GP management terbuka, selanjutkan klik kanan pada nama domain dan klik pilihan “create GPO in this domain and link here”.



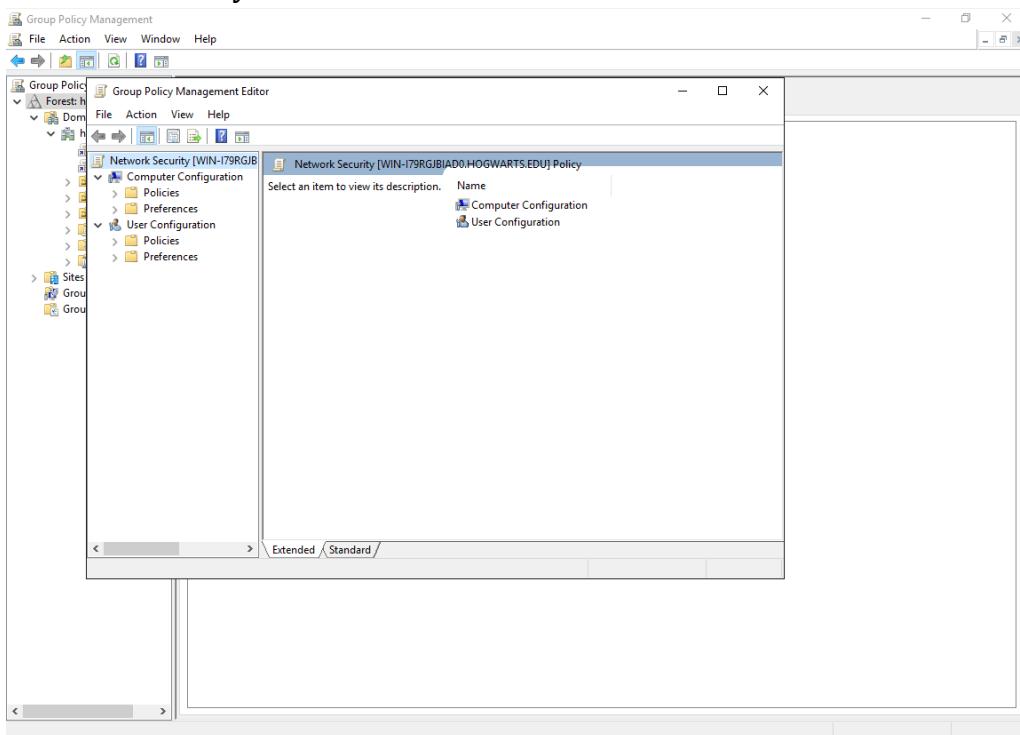
Gambar 203

3. Selanjutnya, berilah nama GPO. Contohnya adalah “Network Security”, kemudian klik OK.



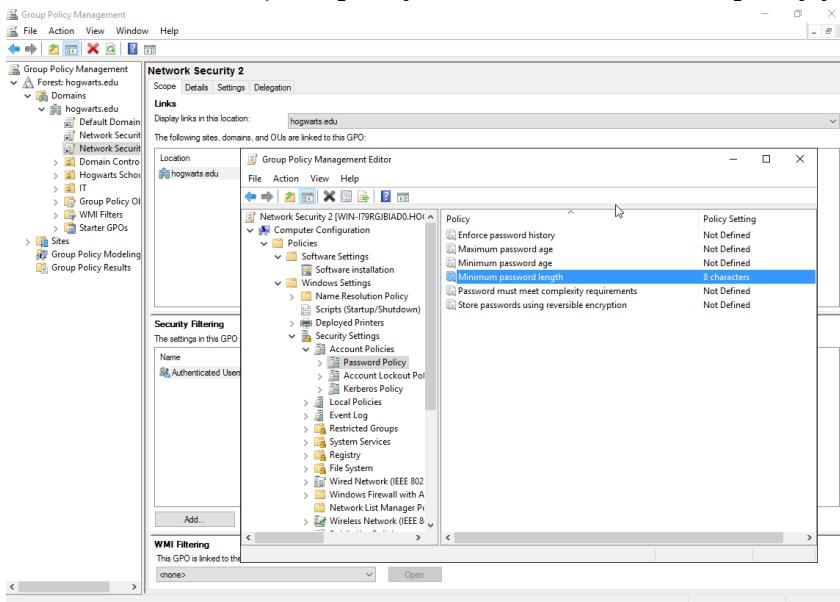
Gambar 204

4. Setelah itu akan terbentuk sebuah GPO yang baru. Kemudian, klik kanan pada GPO Network Security dan klik edit.



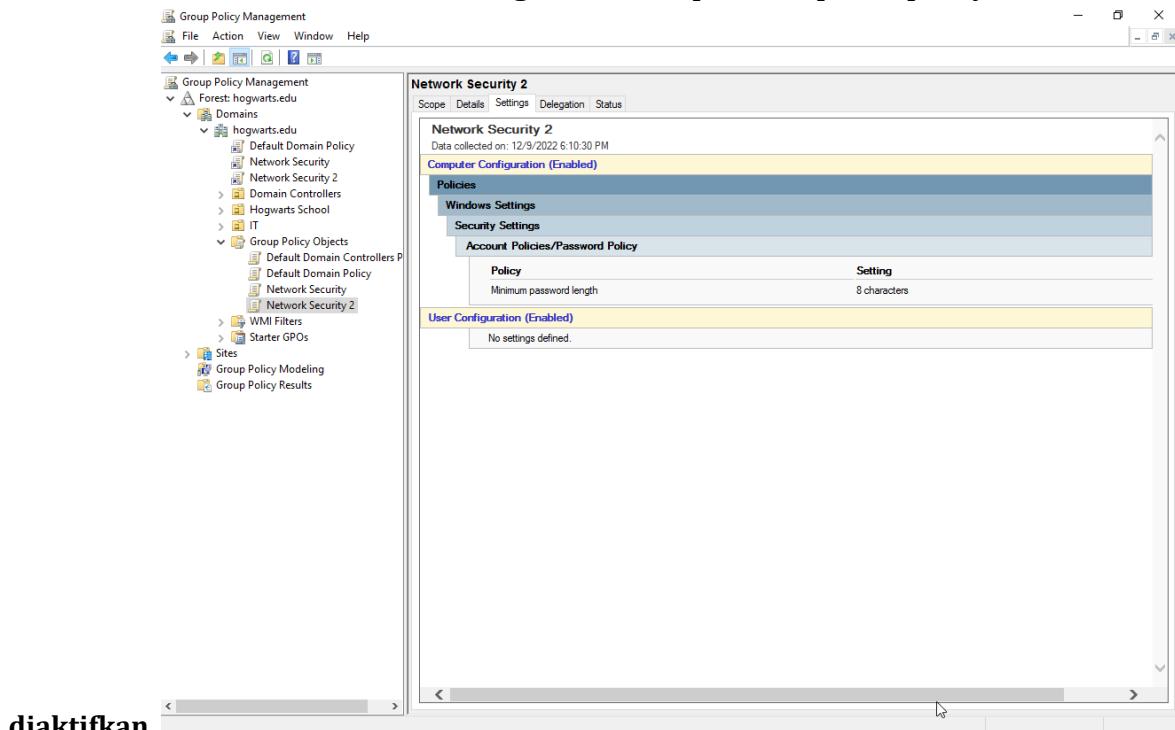
Gambar 205

5. Selanjutnya, GP akan ditentukan. Sebagai contoh sederhana, pada Computer Configuration akan ditetapkan GPO pada bagian Windows Settings kemudian pilih Security Settings. Pilihlah Account Policies, kemudian Klik Password Policies. Pada bagian layar sebelah kanan akan ditampilkan jenis-jenis policy yang dapat dimodifikasi. Pilihlah salah satu dari jenis policy tersebut dan tentukan policy yang Anda tetapkan.



Gambar 206

6. Untuk memverifikasi bahwa GPO telah berhasil ditetapkan maka klik GPO yang telah Anda modifikasi, lalu klik Settings kemudian periksa apakah policy telah



diaktifkan.

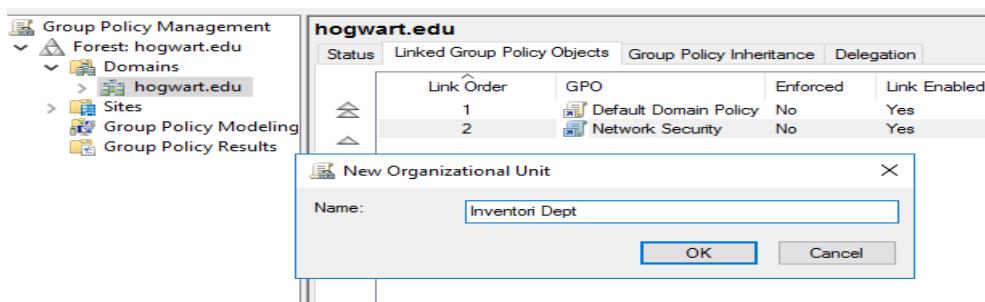
Gambar 207

B. Jenis-jenis Penerapan Group Policy (GP) pada OU

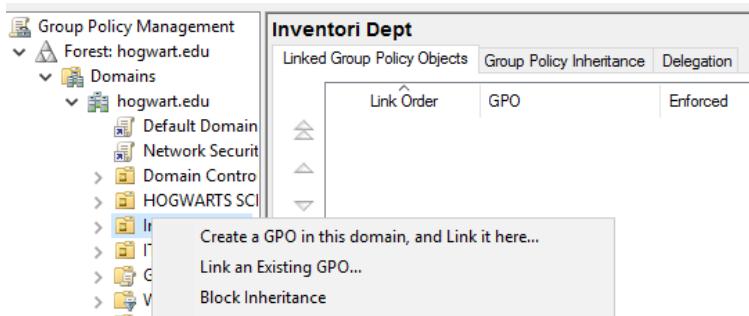
1. Block Inheritance GP

*catatan: Anda harus menambahkan OU baru pada DC hogwarts.edu dengan nama Inventori Dept.

Untuk memblok pewarisan group policy (inheritance group policy) kepada sebuah Organizational Unit (OU), maka klik kanan pada OU target dan pilih pilihan Block Inheritance.

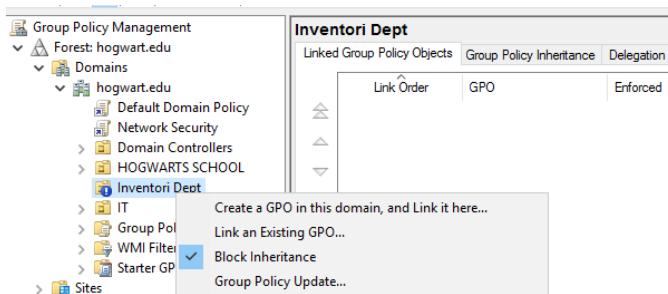


Gambar 208



Gambar 209

Hasilnya GP akan diblok pada OU target.

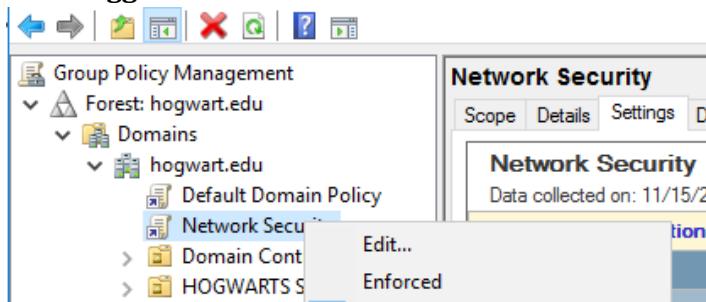


Gambar 210

2. Enforced GP

Enforced GP akan memberlakukan policy kepada semua OU yang berada dibawah Active Directory. Artinya, ketika GP ditegakkan (enforce) maka OU yang telah diberikan jenis Block Inheritance GP pun akan ditegakkan untuk mengambil alih (override) policy yang telah diberlakukan. Oleh karena itu, harus berhati-hati ketika memilih jenis penerapan GP ini terutama jika sudah ditentukan bahwa ada OU yang akan menerima GP yang berbeda. Untuk mengaktifkan Enforced GP, klik kanan GPO (contoh: Network Security) lalu pilih enforced.

*catatan: untuk membuktikan enforced GP, buatlah GP yang lain dengan nama Information Security, kemudian modifikasi dengan policy yang berbeda dari Network Security. Lalu aktifkan jenis Enforced GP pada Information security. Amati apakah policy pada OU yang telah diset pada nomor 1 berubah? Tunjukkan perubahan policy yang terjadi menggunakan cara A.6.

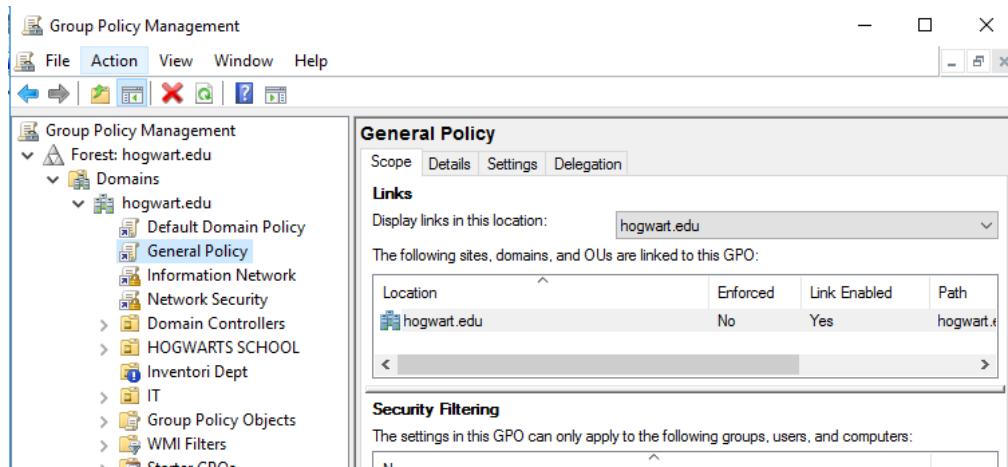


Gambar 211

3. Link Enabled GP

Linked Enabled GP akan memberlakukan bahwa policy akan ditautkan kepada OU, sehingga policy akan diterapkan kepada seluruh objek yang berada dibawah OU tersebut. Klik kanan pada GPO dan pilih Link enabled.

*catatan: buatlah GPO yang baru dengan nama General Policy, kemudian pada GPO ini aktifkan Link Enabled GP.



Gambar 212

Secara default GP akan mengambil waktu sekitar 90 menit untuk mengupdate perubahan policy pada klien. Jika perubahan perlu ditetapkan secara langsung, maka dapat dijalankan dengan perintah gpupdate /force yang melakukan pemeriksaan dan pembaharuan policy.

C. Pengelolaan User dan Group pada AD secara advance dengan PowerShell

*catatan: untuk bagian ini Anda akan menggunakan windows server 2016 yang sudah diberi nama superhero.edu, kemudian superhero.edu harus didaftarkan menjadi anggota dari (member of) domain hogwarts.edu.

A. Last log on time

Skrip berikut ditujukan untuk mendapatkan laporan waktu terakhir pengguna log on ke AD.

```
1 ## Skrip untuk memfilter user berdasarkan waktu logon terakhir ##
2
3 $htmlformat = "<style>BODY{background-color:LightBlue;}</style>"
4 Get-ADUser -Filter * -Properties "LastLogonDate" |
5 sort-object -Property LastLogonDate -Descending | Select-Object Name,LastLogonDate |
6 ConvertTo-Html -head $htmlformat -Body "<H2>Akun AD Last Login Date</H2>" | Out-File C:\lastlogon.html
7 Invoke-Expression C:\lastlogon.html
```

Gambar 213

Ketikkanlah skrip tersebut, jalankan. Untuk mendapatkan hasil dari skrip ini, maka Anda harus menyediakan beberapa user pada superhero.edu yang kemudian akan login ke domain hogwarts.edu.

```

Administrator: Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled2.ps1* X
1 $htmlformat = "<style>BODY{background-color:LightBlue;}</style>"
2 Get-ADUser -Filter * -Properties "LastLogonDate"
3 Sort-Object -Property LastLogonDate -Descending | Select-Object Name,LastLogonDate
4 ConvertTo-Html -head $htmlformat -Body "<H2>Akun AD Last Login Date</H2>" | Out-File C:\lastlogon.html
5 Invoke-Expression C:\lastlogon.html

PS C:\Users\Administrator> $htmlformat = "<style>BODY{background-color:LightBlue;}</style>"
Get-ADUser -Filter * -Properties "LastLogonDate"
Sort-Object -Property lastlogondate -Descending | Select-Object Name,LastLogonDate
ConvertTo-Html -head $htmlformat -Body "<H2>Akun AD Last Login Date</H2>" | Out-File C:\lastlogon.html
Invoke-Expression C:\lastlogon.html

PS C:\Users\Administrator>

```

Gambar 214

B. Failed Login Report

Skip berikut ditujukan untuk mendapatkan laporan login yang gagal ke AD.

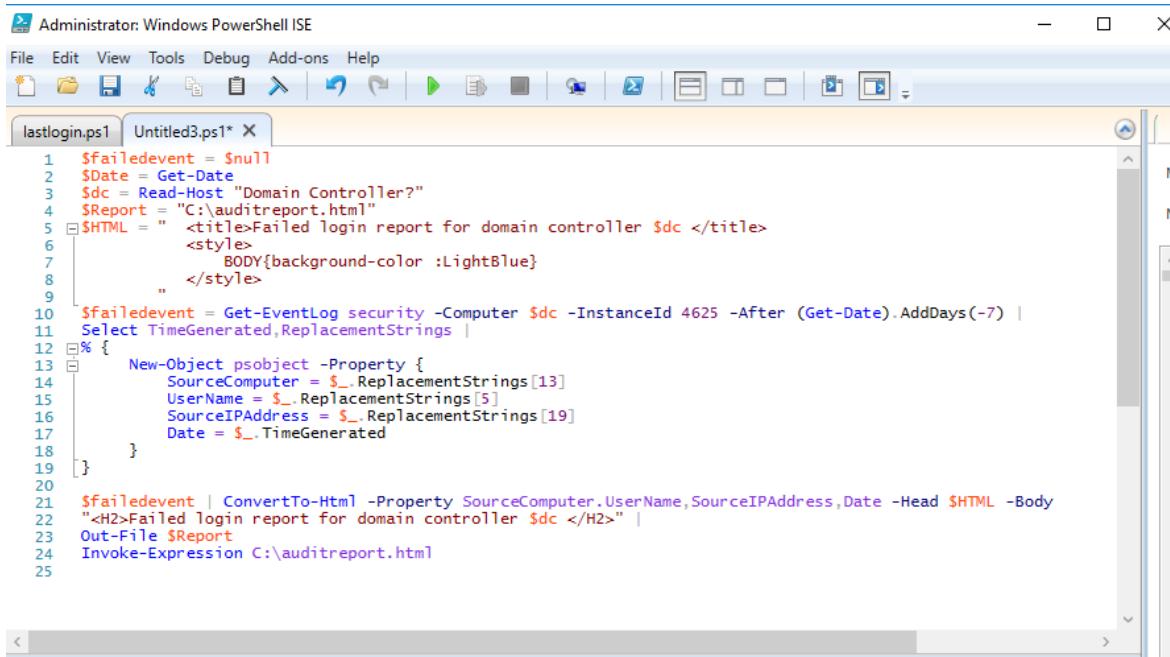
```

1 ## Laporan untuk DC Login Failures ##
2
3 $failedevent = $null
4 $Date = Get-Date
5 $dc = Read-Host "Domain Controller?"
6 $Report = "C:\auditreport.html"
7 $HTML = "    <title>Failed Login report for domain controller $dc </title>
8     <style>
9         BODY{background-color :LightBlue}
10    </style>
11
12 $failedevent = Get-EventLog security -Computer $dc -InstanceId 4625 -After ($Date).AddDays(-7) |
13 Select TimeGenerated,ReplacementStrings |
14 % {
15     New-Object psobject -Property {
16         SourceComputer = $_.ReplacementStrings[13]
17         UserName = $_.ReplacementStrings[5]
18         SourceIPAddress = $_.ReplacementStrings[19]
19         Date = $_.TimeGenerated
20     }
21 }
22
23 $failedevent | ConvertTo-Html -Property SourceComputer,UserName,SourceIPAddress,Date -Head $HTML -Body
24 "<H2> Failed login report for domain controller $dc</H2>" |
25 Out-File $Report
26 Invoke-Expression C:\auditreport.html

```

Gambar 215

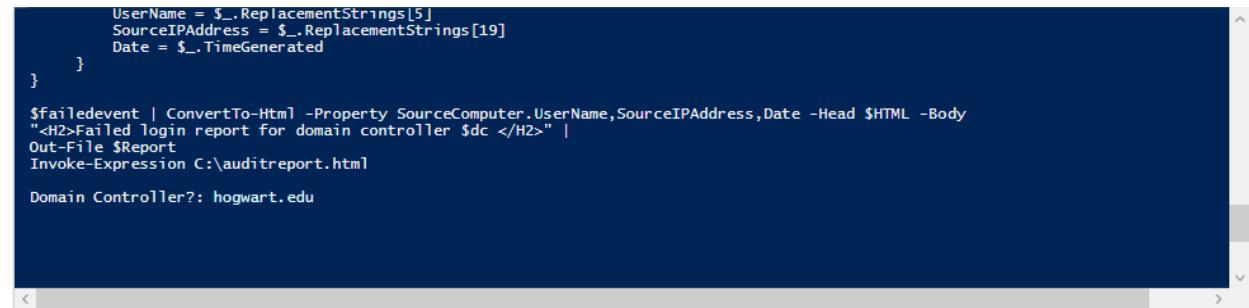
Ketikkanlah skrip tersebut, jalankan. Untuk mendapatkan hasil dari skrip ini, maka Anda harus menyediakan beberapa user pada superhero.edu yang kemudian akan gagal login ke domain hogwarts.edu.



The screenshot shows the Windows PowerShell ISE interface with the title bar "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar has icons for file operations like Open, Save, and Run. There are two tabs open: "lastlogin.ps1" (selected) and "Untitled3.ps1*". The code in "lastlogin.ps1" is as follows:

```
1 $Failedevent = $null
2 $Date = Get-Date
3 $dc = Read-Host "Domain Controller?"
4 $Report = "C:\auditreport.html"
5 $HTML = "<title>Failed login report for domain controller $dc </title>
6 <style>
7     BODY{background-color :LightBlue}
8 </style>
9 "
10 $Failedevent = Get-EventLog security -Computer $dc -InstanceId 4625 -After ($Date).AddDays(-7) |
11 Select TimeGenerated,ReplacementStrings |
12 % {
13     New-Object psobject -Property {
14         SourceComputer = $_.ReplacementStrings[13]
15         UserName = $_.ReplacementStrings[5]
16         SourceIPAddress = $_.ReplacementStrings[19]
17         Date = $_.TimeGenerated
18     }
19 }
20 $Failedevent | ConvertTo-Html -Property SourceComputer,UserName,SourceIPAddress,Date -Head $HTML -Body
21 "<H2>Failed login report for domain controller $dc </H2>" |
22 Out-File $Report
23 Invoke-Expression C:\auditreport.html
24
25
```

Gambar 216



The screenshot shows a terminal window with the following command and output:

```
UserName = $_.ReplacementStrings[5]
SourceIPAddress = $_.ReplacementStrings[19]
Date = $_.TimeGenerated
}

$Failedevent | ConvertTo-Html -Property SourceComputer,UserName,SourceIPAddress,Date -Head $HTML -Body
"<H2>Failed login report for domain controller $dc </H2>" |
Out-File $Report
Invoke-Expression C:\auditreport.html

Domain Controller?: hogwart.edu
```

Gambar 217

BAB 10. AUDITING & MONITORING

Active Directory (AD) Auditing and Monitoring

A. Active Directory Audit

Sebagai jalan untuk mengidentifikasi ancaman keamanan pada sistem adalah melalui proses monitoring dan auditing secara berkelanjutan.

Sistem operasi Windows menyediakan sistem yang disebut advanced auditing yang mampu untuk mengidentifikasi isu keamanan pada sistem. Pada Windows secara default terdapat beberapa tipe aksi yang diaudit. Tipe audit ini ditangani oleh audit policy Windows.

Ada 10 kategori dari event yang dapat diaudit pada sistem Windows, sebagai berikut:

- System events
- Logon/logoff events
- Object access events
- Privileged use events
- Detailed tracking events
- Policy change events
- Account management events
- DS access events
- Account logon events
- Global object access auditing

Pada modul ini, hanya dua jenis event yang akan didemonstrasikan yaitu DS access events dan Object Access.

A.1 DS Access Events

Dengan mengaktifkan DS Access Events, maka event terhadap perubahan objek yang terjadi pada Active Directory Domain Services (AD DS) akan direkam. Hal ini bisa terjadi jika dan hanya jika System Access Control List (SACL) dikonfigurasi. DS Access events memiliki empat subkategori, sebagai berikut:

- Audit Directory Service Access
- Audit Directory Service Changes
- Audit Directory Service Replication
- Audit Detailed Directory Service Replication

Pada modul ini, hanya Audit Directory Service Changes yang akan didemonstrasikan. Audit Directory Service Changes memiliki empat kategori event yang berhubungan dengan perubahan pada AD DS, seperti:

- Create
- Delete
- Modify
- Move
- Undelete

Ketika sebuah objek diubah, maka nilai yang lama dan yang baru akan direkam. Event ini hanya akan direkam jika SAC dikonfigurasi. Setelah auditing diaktifkan maka, event berikut dapat ditemukan pada log keamanan sistem (security logs):

Table 4 Event 1 DS Access

Event ID	Event Message
5136	A directory service object was modified
5137	A directory service object was created
5138	A directory service object was undeleted
5139	A directory service object was moved
5141	A directory service object was deleted

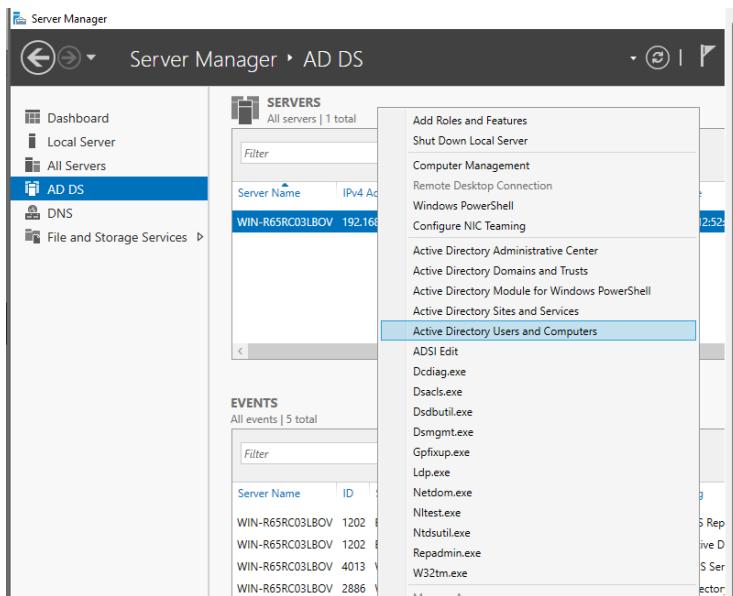
A.2 Object Access

Dengan mengaktifkan Object Access, maka akses pada file sistem dapat diaudit sehingga dapat diketahui user tertentu yang menciptakan, menghapus atau memodifikasi sebuah file atau direktori. Pada modul ini, akan didemonstrasikan bagaimana untuk mengkonfigurasi event auditing untuk file yang dibagikan melalui jaringan komputer (shared network folder). Setelah auditing dikonfigurasi, maka informasi dapat dilihat pada Event Viewer untuk menampilkan aksi yang dilakukan oleh user. B. Langkah-langkah demonstrasi praktikum

B.1 Mengaktifkan advanced security audit policies

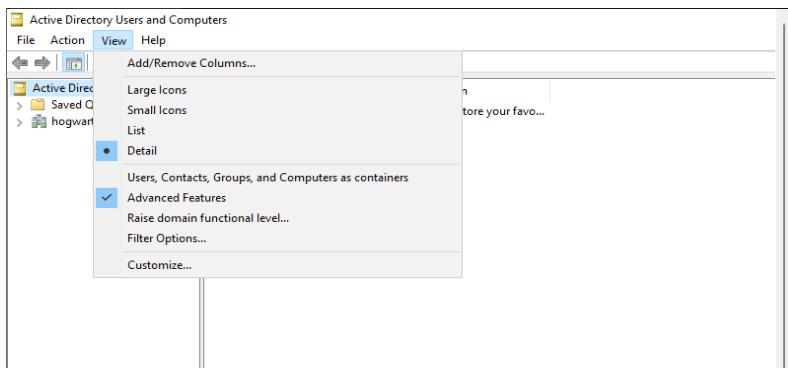
Ikuti langkah-langkah berikut untuk konfigurasi SACL. Setiap langkah berikan hasil screen shootnya.

1. Bukalah Active Directory Users and Computers



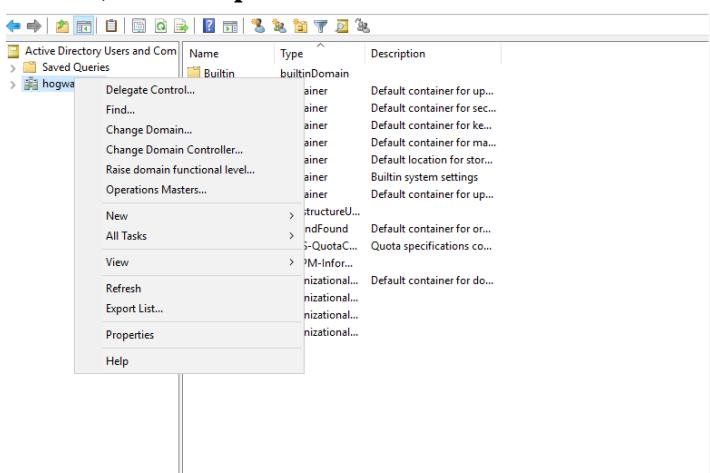
Gambar 218

2. Klik View Advanced Features



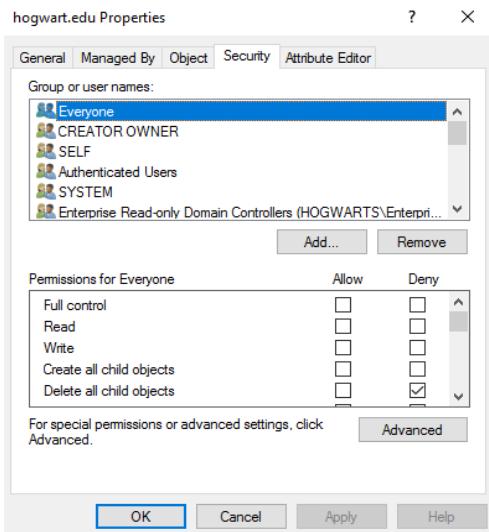
Gambar 219

3. Klik kanan pada OU atau objek yang auditingnya diaktifkan.
Kemudian, klik Properties. Dalam kasus ini silahkan pilih OU IT.



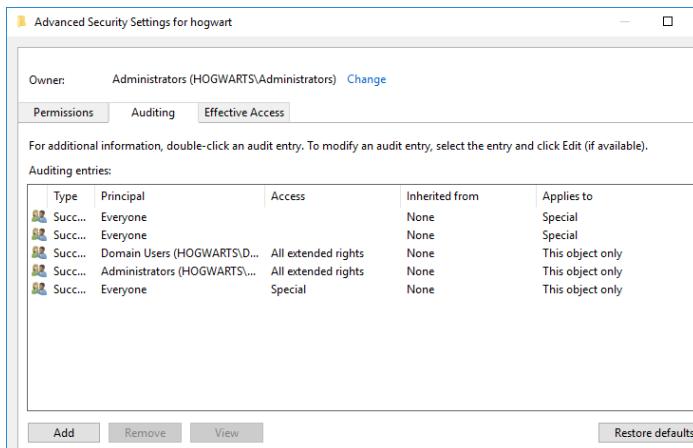
Gambar 220

4. Klik tab Security dan kemudian pilihan Advanced

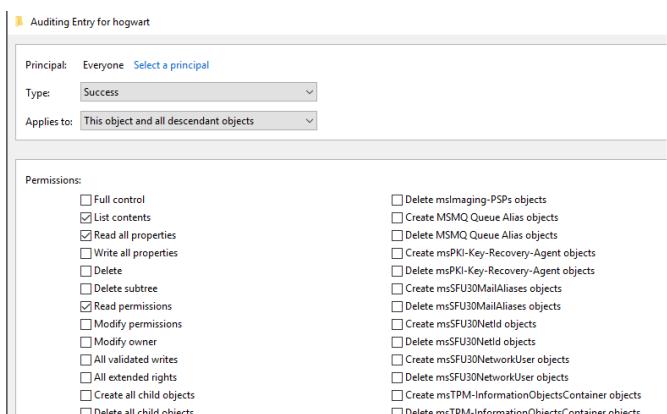


Gambar 221

5. Klik tab Auditing, kemudian klik tombol Add untuk menambahkan aturan keamanan yang baru ke SACL. Pada skenario ini, pada security principle pilih everyone yang artinya diterapkan untuk seluruh user yang berada dibawah OU IT.

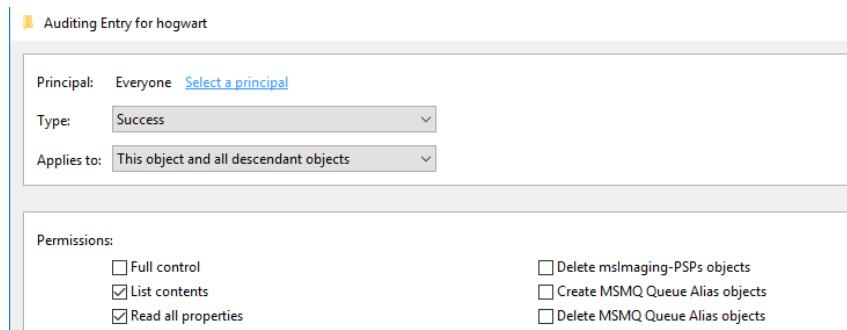


Gambar 222



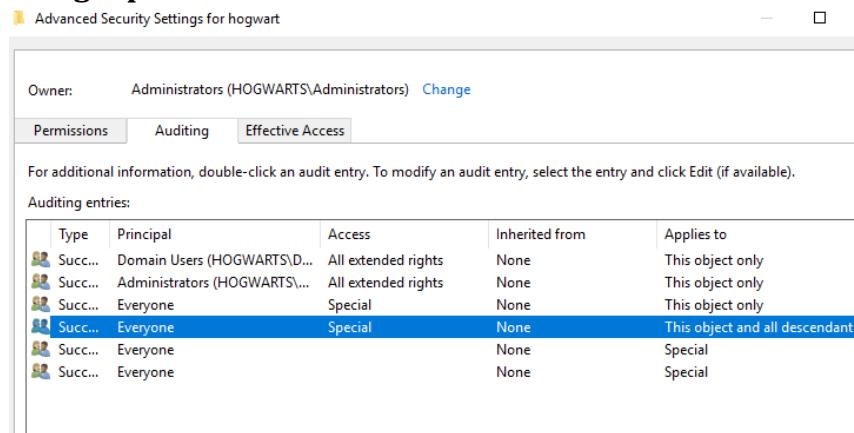
Gambar 223

6. Untuk pilihan type pilih tipe event Success, yang kemudian diterapkan ke This object and all descendant objects.



Gambar 224

7. Pilihlah permission yang akan ditetapkan. Dari permission secara default, pilih tambahan permission lainnya seperti menciptakan user dan menghapus user.

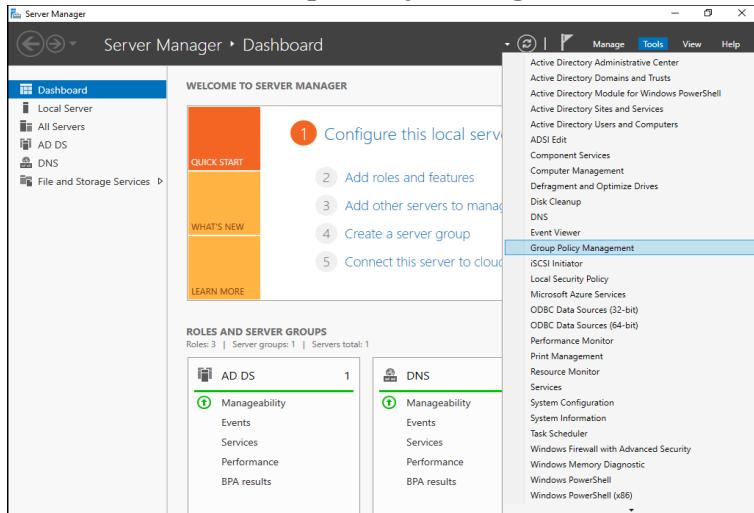


Gambar 225

B.2 Mengaktifkan audit policy DS Events

Setelah SACL diaktifkan, selanjutnya ikuti langkah-langkah berikut mengaktifkan advanced audit policy untuk DS, sebagai berikut:

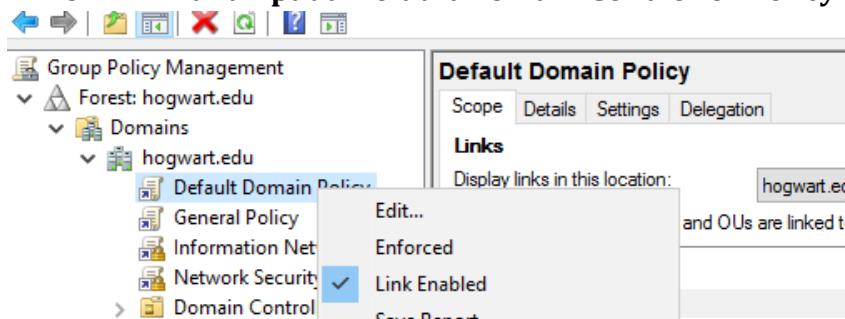
1. Bukalah Group Policy Management dari server manager.



Gambar 226

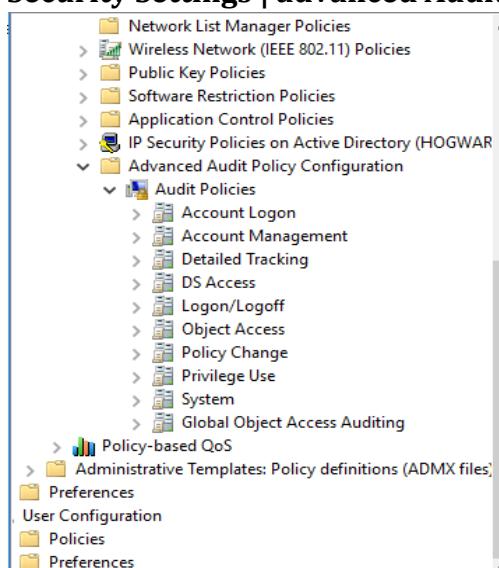
2. Selanjutnya ekspan Domain Controller OU.

3. Klik kanan pada Default Domain Controller Policy dan pilih Edit.



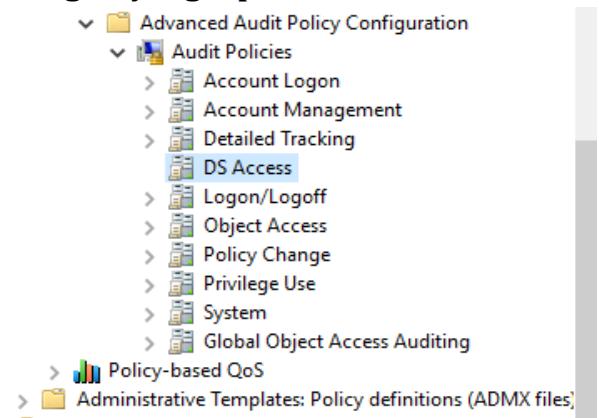
Gambar 227

4. Arahkan ke Computer Configuration | Policies | Windows Settings | Security Settings | advanced Audit Policy Configuration | audit Policies,



Gambar 228

5. Kemudian, akan ditemukan 10 kategori audit. Pada demonstrasi ini kategori yang dipilih adalah DS Access dan DS changes.

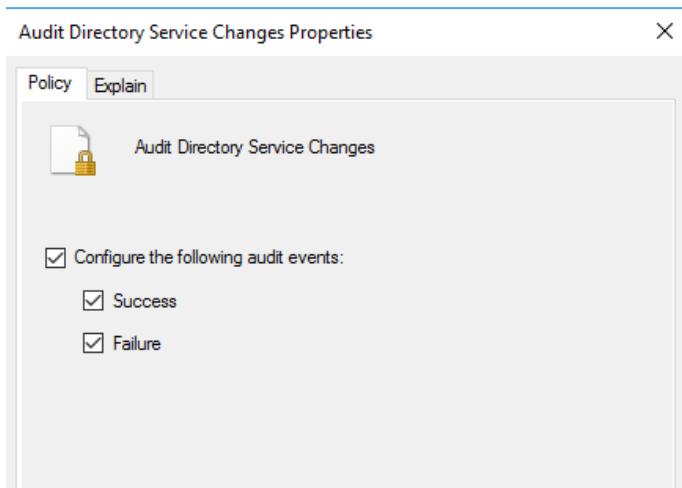


Gambar 229

6. Arahkan ke DS Access dan double-klik ke Subcategory. Untuk mengaktifkan auditing, pilihlah Configure the following audit events dan kemudian pilih event yang akan diaudit. Direkomendasikan memilih keduanya yaitu Success dan Failure.

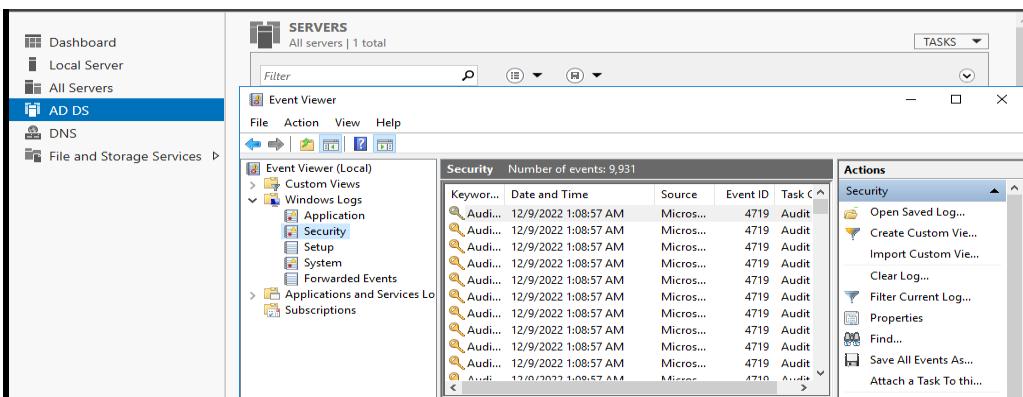
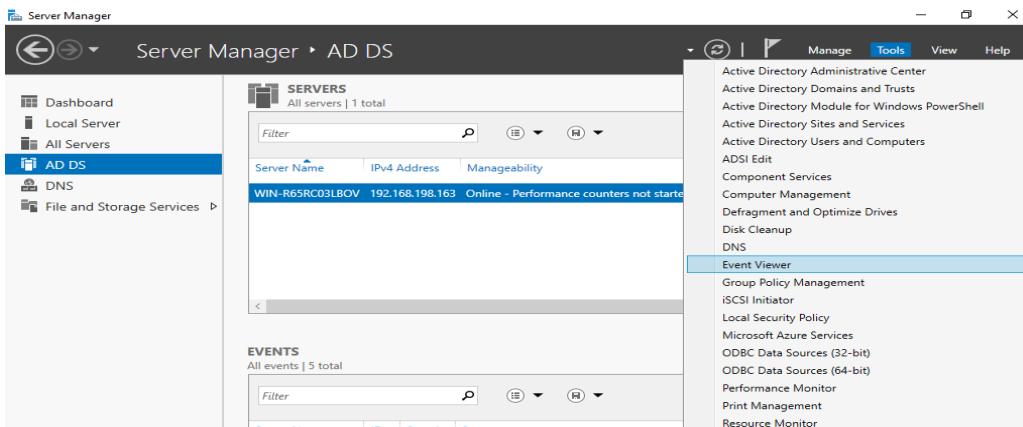
Subcategory	Audit Events
[101] Audit Detailed Directory Service Replication	Not Configured
[010] Audit Directory Service Access	Not Configured
[010] Audit Directory Service Changes	Not Configured
[101] Audit Directory Service Replication	Not Configured

Gambar 230



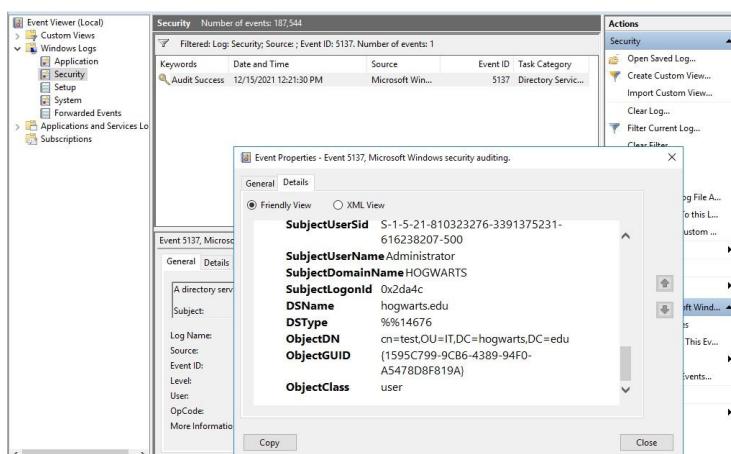
Gambar 231

7. Setelah group policy berhasil diterapkan, maka events yang berelasi dengan audit policy tersebut akan dituliskan di log. Anda dapat melihatnya di Event viewer, dengan cara melakukan filter log. Lakukan uji coba dengan cara menciptakan user baru, menghapus user, kemudian periksalah event pada Event Viewer. Anda dapat memfilter event ID yang akan diperiksa dengan mengacu pada event ID di Tabel 1.



Gambar 232

Hasilnya seperti berikut. Keterangannya adalah sebuah user dengan username test ditambahkan di OU IT oleh administrator.

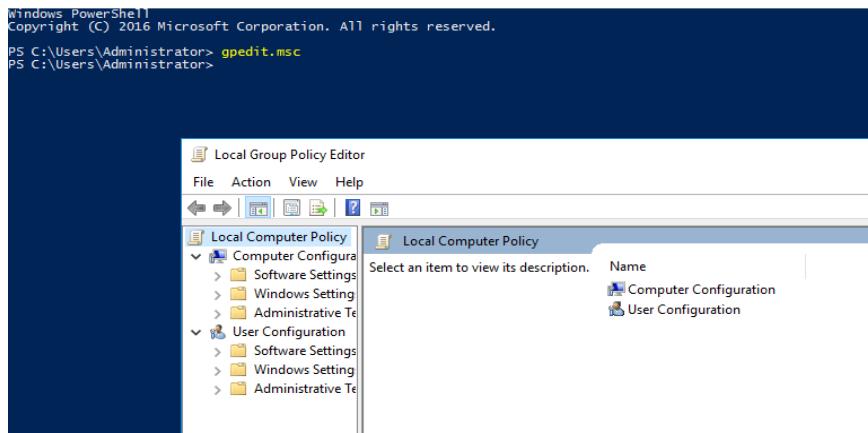


Gambar 233

B.3 Mengaktifkan audit policy Object Access

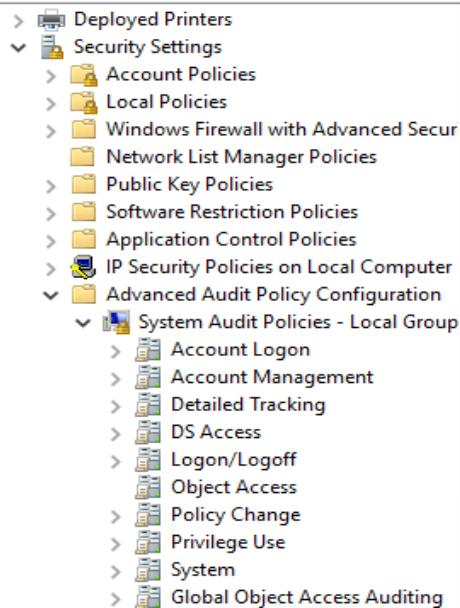
Pada bagian Object Access ini, akan diaktifkan auditing policy untuk akses file dan folder pada Windows Server 2016. Secara default file system object access audit policy tidak diaktifkan pada Windows server. Hal ini bisa dilakukan dengan cara mengkonfigurasi pengaturan audit menggunakan Group Policy. Pada praktikum ini akan mendemonstrasikan konfigurasi hanya pada satu komputer saja, sehingga bisa menggunakan Local Group Policy Editor. Untuk melakukan konfigurasi, ikuti langkah-langkah berikut:

1. Buka Local Group Policy Editor dengan menggunakan perintah gpedit.msc



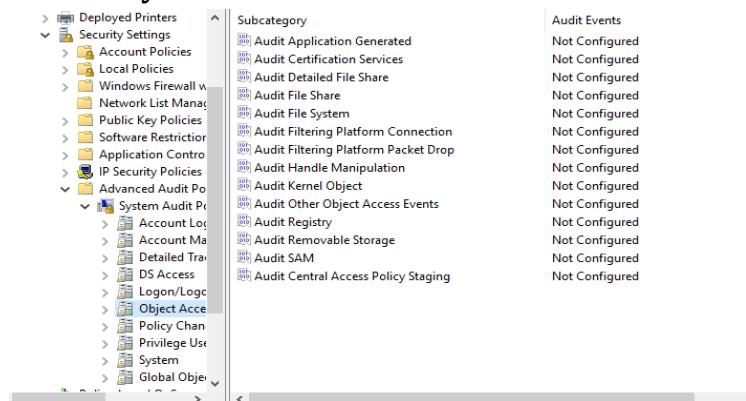
Gambar 234

2. Kemudian pilih Windows settings | Security Settings | Advanced Audit Policy Configuration | Object Access



Gambar 235

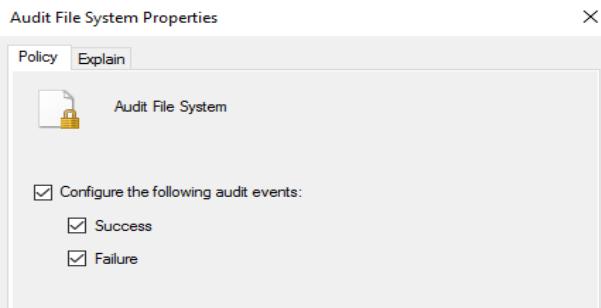
3. Buka Audit File system policy kemudian pilih kedua event yang akan direkam yaitu Success dan Failure.



Gambar 236

4.	Audit File System	Not Configured
	Audit Detailed File Share	Not Configured
	Audit File Share	Not Configured
	Audit Application Generated	Not Configured
	Audit Filtering Platform Connection	Not Configured
	Audit Filtering Platform Packet Drop	Not Configured
	Audit Handle Manipulation	Not Configured
	Audit Kernel Object	Not Configured
	Audit Other Object Access Events	Not Configured
	Audit Registry	Not Configured
	Audit Removable Storage	Not Configured
	Audit SAM	Not Configured
	Audit Central Access Policy Staging	Not Configured

Gambar 237

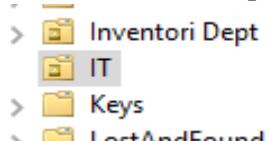


5.

Gambar 238

Selanjutnya lakukan konfigurasi pada folder yang akan dibagi lewat jaringan komputer, dengan langkah-langkah berikut:

1. Klik kanan pada folder yang akan dibagi, lalu pilih Properties.

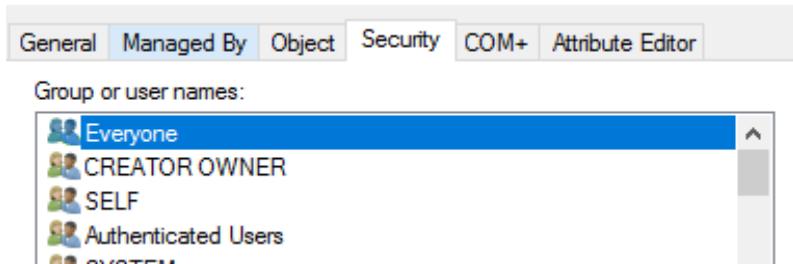


Gambar 239

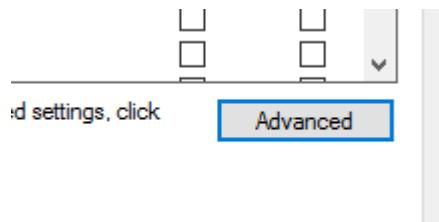


Gambar 240

3. Pilih tab Security, klik tombol Advanced lalu pilih tab Auditing.

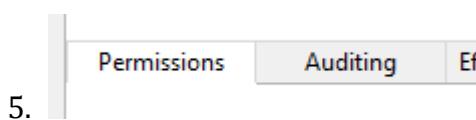


Gambar 241



4.

Gambar 242

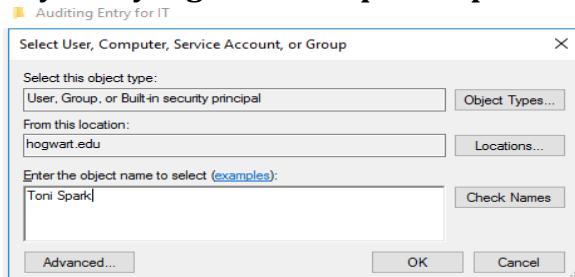


5.

Gambar 243

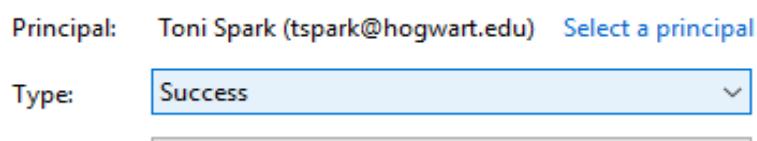
6. Klik tombol Add.

7. Pilihlah Principal dalam hal ini Anda bisa menggunakan user beyonce yang sudah diciptakan pada OU IT.



Gambar 244

8. Pilih type event yang akan direkam adalah Success 6. Kemudain applies to, This folder, subfolders, and files.



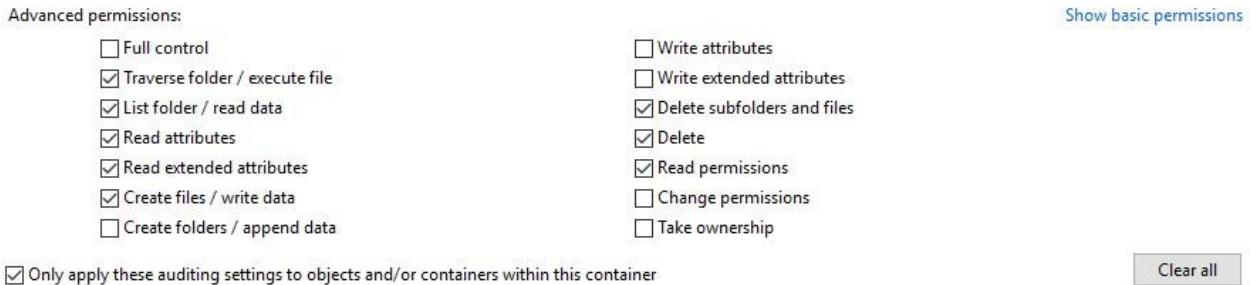
Gambar 245

9. Kemudian applies to, This folder, subfolders, and files.

Applies to: This object and all descendant objects

Gambar 246

7. Kemudian beri permission sesuai gambar berikut. Anda bisa melakukannya dengan klik Show advanced permission.

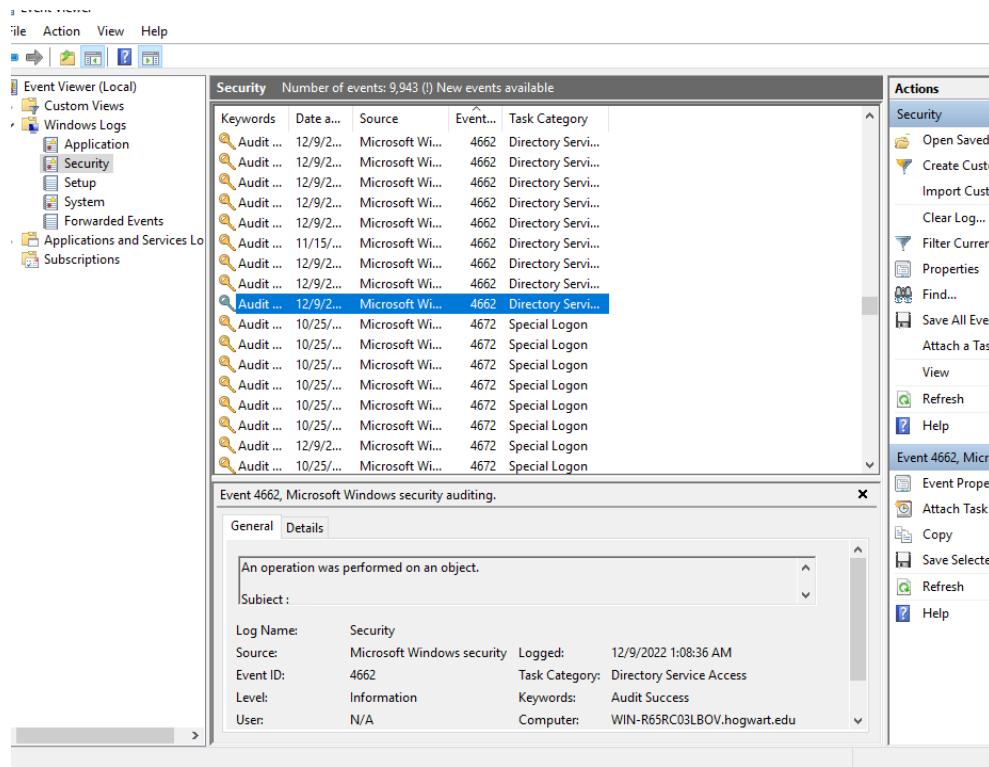


Gambar 247

8. Kemudian klik tombol Ok, Apply, dst. untuk menerapkan konfigurasi.

Untuk uji coba, Anda harus membuat sebuah folder yang dibagi (share) dari Windows Server, kemudian pada mesin yang berbeda harus dapat diakses oleh user yang telah Anda berikan audit policy dalam kasus ini adalah user beyonce. Pada folder yang anda bagi, silahkan buat beberapa file dan hapus file tersebut. Anda bisa mencoba untuk mencari event ID 4663 ketika sebuah file dihapus oleh user beyonce. Hasil dari eksekusi aksi ini dapat dilihat pada gambar berikut.

Gambar 248



Gambar 249