

MODUL LABORATORIUM SISTEM OPERASI

FAKULTAS SAINS DAN TEKNOLOGI
PROGRAM STUDI INFORMATIKA



Penyusun Modul
Ir. Sumarno
Hamzah Setiawan , S.Kom
Roni Pambudi, S.Kom



DAFTAR ISI

Pokok Bahasan 4 Proses Thread dan Deadlock.....	5
Pendahuluan.....	5
Penyajian (Tutorial)	5
Lembar Kerja dan Tugas.....	6
Kunci Lembar Tugas	8
Referensi	9
Pokok Bahasan 5 Manajemen File dan Disk.....	10
Pendahuluan.....	10
Penyajian (Tutorial)	10
Lembar Kerja dan Tugas.....	11
Kunci Lembar Tugas	14
Referensi	14
Pokok Bahasan 6 Sistem Operasi Tersebar	15
Pendahuluan.....	15
Penyajian (Tutorial)	15
Lembar Kerja dan Tugas.....	15
Kunci Lembar Tugas	16
Referensi	16
Lampiran.....	18
Riwayat Hidup Penulis	18

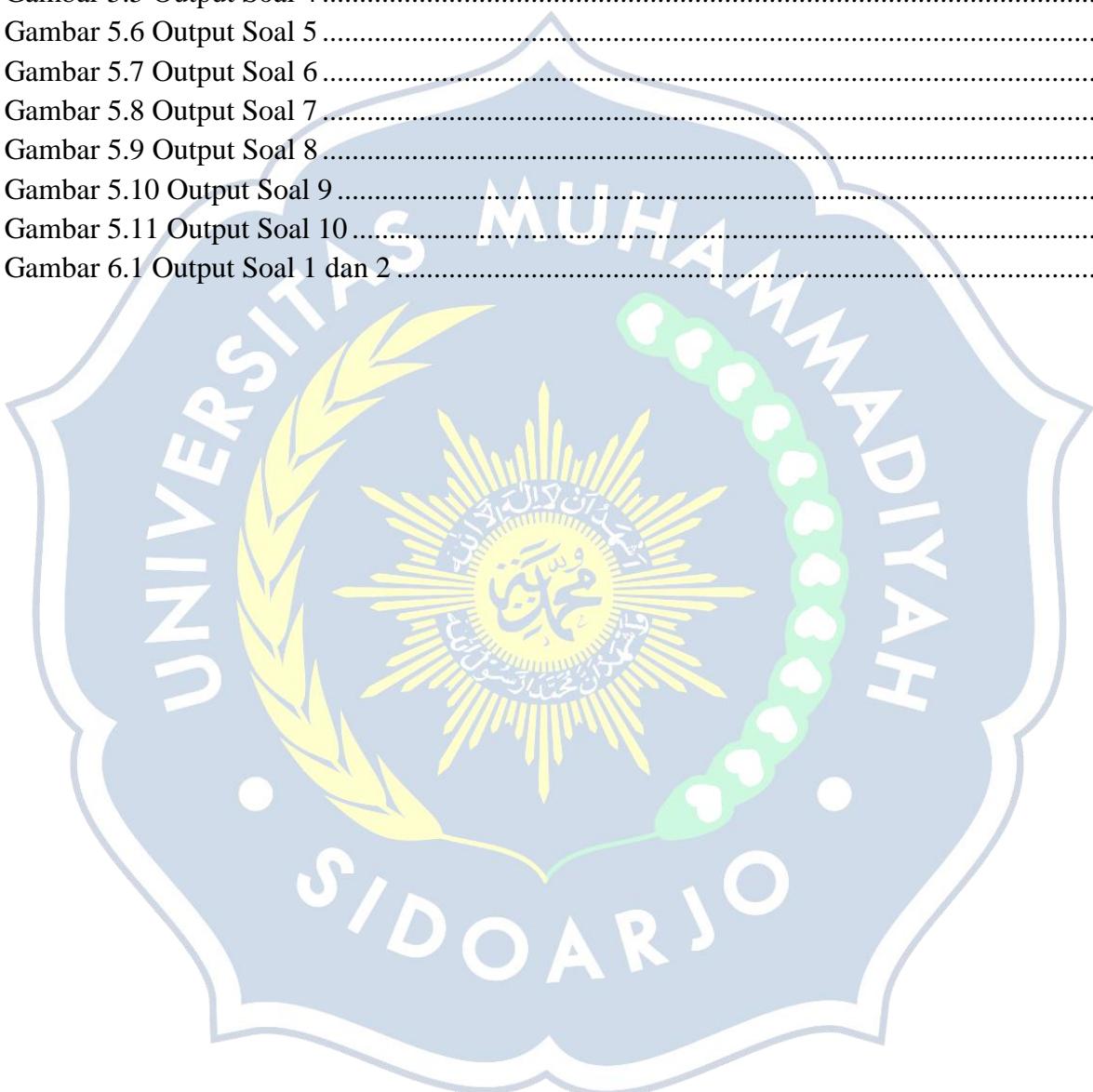
DAFTAR TABEL

Tabel 5 Deskripsi Nomor Sinyal	5
Tabel 6 Direktori Standar Linux.....	10



DAFTAR GAMBAR

Gambar 5.1 Struktur File Linux	10
Gambar 5.2 Output Soal 1	15
Gambar 5.3 Output Soal 2	15
Gambar 5.4 Output Soal 3	16
Gambar 5.5 Output Soal 4	16
Gambar 5.6 Output Soal 5	16
Gambar 5.7 Output Soal 6	17
Gambar 5.8 Output Soal 7	17
Gambar 5.9 Output Soal 8	17
Gambar 5.10 Output Soal 9	18
Gambar 5.11 Output Soal 10	18
Gambar 6.1 Output Soal 1 dan 2	21



POKOK BAHASAN 4

Proses Thread dan Deadlock

TUJUAN

1. Mengenal konsep proses di linux.
2. Mengerti konsep sinyal dan bagaimana cara mengelola sinyal tersebut.

DASAR TEORI

KONSEP PROSES DI LINUX

Setiap kali intruksi diberikan pada linux shell, maka kernel akan menciptakan sebuah proses id. Proses ini disebut juga dengan terminology Unix: sebagai sebuah Job. Proses id (PID) di mulai dari 0, yaitu proses “INK”, kemudian diikuti oleh proses berikutnya (terdaftar pada /etc/inittab).

Beberapa tipe proses:

- **Foreground**

Proses yang didapatkan oleh pemakai langsung pada terminal (interaktif dialog).

- **Batch**

Proses yg dikumpulkan dan dijalankan secara sekuensial (satu persatu) Proses Batch tidak di asosiasikan (berinteraksi) dengan terminal.

- **Daemon**

Proses yang menunggu permintaan (request) dari proses lainnya dan menjalankan tugas sesuai dengan permintaan tersebut. Bila tidak ada request, maka program ini akan berada dalam kondisi “idle” dan tidak menggunakan waktu hitung CPU. Umumnya nama proses daemon di UNIX berakhiran d, misalnya inetd, named, popd dll.

SINYAL

Proses dapat mengirim dan menerima sinyal dan ke posisi lainnya. Proses mengirim sinyal melalui intruksi “kill” dengan format

```
$ Kill [ -nomor sinyal ] [ nomor PID ]
```

Nomor sinyal : 1 s/d maksimum nomor sinyal yang di definisikan system Standar nomor sinyal yang ter penting adalah:

Tabel Deskripsi Nomor Sinyal

No. Sinyal	Nama	Deskripsi
1	SIGHUP	Hangup, sinyal dikirim bila proses terputus, misalnya melalui putusnya hubungan modem
2	SIGINT	Sinyal Interup, melalui ^c
3	SIGQUIT	Sinyal Quit melalui ^\
9	SIGKILL	Sinyal Kill, menghentikan proses
15	SIGTERM	Sinyal terminasi software

MENGIRIM SINYAL

Mengirim sinyal adalah satu alat komunikasi antar proses, yaitu memberitahukan proses yang sedang berjalan bahwa ada sesuatu yang harus dikendalikan. Berdasarkan sinyal yang dikirim ini maka proses dapat berinteraksi dan administrator/programmer dapat menentukan reaksi tersebut. Mengirim sinyal menggunakan intruksi.

```
$ Kill [-nomor sinyal] [nomor PID]
```

Sebelum mengirim sinyal PID proses yang akan dikirim harus diketahui terlebih dahulu.

LEMBAR KERJA DAN TUGAS

LANGKAH-LANGKAH

1. Masuk ke system operasi Linux.
2. Login menggunakan user dan password Anda.
3. Gunakan intruksi ststus proses : ps
4. Gunakan intruksi untuk mengelola sinyal: kill, trap, nohup.

PERCOBAAN

Percobaan 1 : Status Proses

1. Instruksi ps (proses status) digunakan untuk melihat kondisi proses yang ada. PID adalah Nomor Identitas Proses, tty adalah nama terminal dimana proses tersebut aktif, STAT berisi S (sleeping) dan R (Running), COMMAND merupakan instruksi yang digunakan.

\$ ps

2. Untuk melihat factor / elemen lainnya, gunakan option -u (user). %CPU adalah presentasi CPU time yang digunakan oleh proses tersebut. %MEM adalah presentasi sistem memori yang digunakan proses, SIZE adalah jumlah memori yang digunakan, RSS (Real System Storage) adalah jumlah memori yang digunakan, START adalah “kapan proses tersebut diaktifkan.

\$ ps -u

3. Mencari proses yang spesifik pemakai. Proses di atas hanya terbatas pada proses milik pemakai, dimana pemakai tersebut melakukan login.

\$ ps -u [nama user]

4. Mencari proses lainnya gunakan option a (aR) dan au (all user)

\$ ps -a

Percobaan 2 : Sinyal

1. Membuat shell script dengan nama loop.sh.

\$ nano loop.sh

```
#!/bin/bash
# Sebuah shell script : loop.sh
while :
do
    echo ".\c"
    sleep 10
done
```

2. Eksekusi file loop.sh sebagai background.

\$ chmod +x loop.sh
\$./loop.sh &

3. Melihat proses ID dari `loop.sh`.

```
$ ps
```

4. Menghentikan proses. Nomor 15 (SIGTERM) merupakan default.

```
$ kill -15 [ nomor PID ] atau
```

```
$ kill [ nomor PID ]
```

5. Menghentikan proses secara mutlak.

```
$ kill -9 [ nomor PID ]
```

Percobaan 3 : Mengelola Sinyal

1. Membuat file `prog.sh`.

```
$ nano prog.sh
```

```
#!/bin/sh
echo "Program berjalan .."
while :
do
    echo "X"
    sleep 20
done
```

2. Jalankan program tersebut karena program melakukan looping, maka stop dengan mengirim sinyal interrupt (^C).

```
$ chmod +x prog.sh
```

```
$ ./prog.sh
```

3. Jalankan program tersebut sebagai background. Catat nomor PID proses, tekan <Enter> untuk ke foreground dan periksa melalui intruksi `ps`.

```
$ ./prog.sh &
```

```
$ ps
```

4. Kirimkan sinyal terminasi sebagai berikut.

```
$ kill [nomor PID]
```

5. Ubahlah program `prog.sh` dengan instruksi trap untuk menangkap sinyal yang dikirim.

```
$ nano prog.sh
```

```
#!/bin/sh
trap "" 1 2 3 15
echo "Program berjalan .."
while :
do
    echo "X"
    sleep 20
done
```

6. Jalankan program tersebut sebagai background. Coba lakukan kill dengan nomor PID proses tersebut.

```
$ ./prog.sh &
```

```
$ kill [nomor PID] atau
```

```
$ kill -1 [nomor PID] atau
$ kill -2 [nomor PID] atau
$ kill -15 [nomor PID]
```

7. Perintah kill di atas tidak akan menghentikan proses karena dihalangi dengan perintah trap. Cobalah menggunakan nomor sinyal 9.

```
$ kill -9 [nomor PID]
```

Percobaan 4 : No Hangup

1. Adakalanya sebuah proses memerlukan waktu yang cukup lama misalnya proses sortir, sehingga perlu dilakukan sebagai proses background. Namun bila proses masih berlangsung dan kita melakukan logout, maka otomatis proses akan ikut berhenti, yang artinya proses sortir harus diulang kembali. Simulasi dari proses sort.

```
$ nano myjob.sh
```

```
#!/bin/bash
while :
do
    find / -print > berkas
    sort berkas -o hasil
    echo "Proses selesai pada `date`" >>
proses.log
done
```

2. Jalankan proses tersebut sebagai proses background.

```
$ chmod +x myjob.sh
$ ./myjob.sh &
```

3. Kemudian logout dan login kembali. Periksa sampai dimana job bekerja.

```
$ ps
```

4. Gunakan nohup (NoHangup) agar job tetap berjalan meskipun pemakai logout. Catatan : fungsi ini tidak berjalan di non system V (Linux).

```
$ ./myjob.sh &
$ nohup myjob.sh
```

5. Kemudian logout dan login kembali. Periksa apakah job masih bekerja.

```
$ ps
```

TUGAS

1. Login menggunakan user Anda dan lihat status proses, perhatikan kolom keluaran dari perintah ps –au sebagai berikut:
 - a. Sebutkan nama-nama proses yang bukan root.
 - b. Tulis PID dan COMMAND dari proses yang paling banyak mengganggu CPU time.
 - c. Sebutkan buyut proses dan PID dari proses tersebut.
 - d. Sebutkan beberapa proses daemon.
 - e. Pada prompt login lakukan hal-hal sebagai berikut:

```
$ ssh
$ who
```

```
$ bash
$ ls
$ sh
$ ps
```

2. Modifikasi program prog.sh sebagai berikut:

```
$ nano prog.sh
```

```
#!/bin/bash
trap "echo Hello Goodbye ; exit 0" 1 2 3
15
while :
do
    find / -print > berkas
    sort berkas -o hasil
    echo "Proses selesai pada `date`" >>
proses.log
    sleep 60
done
```

Jalankan program tersebut sebagai background. Coba lakukan kill dengan nomor sinyal 1, 2, 3 dan 15 pada nomor PID proses tersebut. Apakah proses berhenti atau tetap berjalan? Nomor sinyal berapa yang digunakan untuk menghentikan proses di atas?

3. Modifikasi program myjob.sh. buatlah trap sedemikian rupa sehingga bila proses tersebut dihentikan (kill), otomatis file berkas akan terhapus.

```
$ nano myjob.sh
```

```
#!/bin/bash
trap #lakukan perubahan di baris ini
while :
do
    find / -print > berkas
    sort berkas -o hasil
    echo "Proses selesai pada `date`" >> proses.log
    sleep 60
done
```

```
$ kill -15 [nomor PID]
$ ls -l
```

REFERENSI

- [Stallings2001] William Stallings. Hak Cipta © 2001. Operating Systems. Prentice Hall.
- [Tanenbaum1992] Andrew S. Tanenbaum. Hak Cipta © 1992. Modern Operating Systems. Prentice-Hall Inc..
- [Silberschatz2000] Avi Silberschatz, Peter Galvin, dan Grag Gagne. Hak Cipta © 2000. Operating Systems CONCEPTS

POKOK BAHASAN 5

Manajemen File dan Disk

PENDAHULUAN

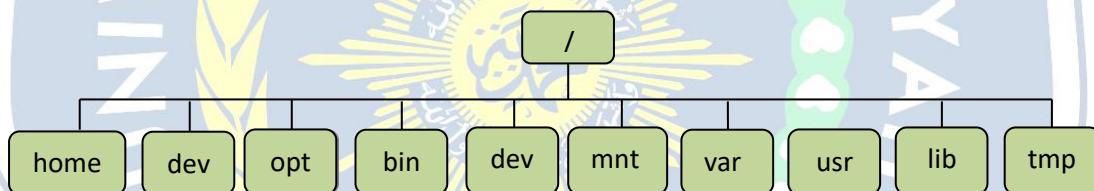
Pada pokok bahasan ini akan dibahas mengenai perintah-perintah dasar manajemen file dan perintah-perintah manajemen disk. Sehingga setelah mempelajari bab ini diharapkan mahasiswa mampu:

- a) Mengenal manajemen file di linux
- b) Mengenal direktori dan file di linux
- c) Mengenal ijin akses direktori dan file
- d) Mengenal konsep owner dan group
- e) Mengenal manajemen disk di linux

PENYAJIAN (TUTORIAL)

Manajemen File

File merupakan kumpulan data atau informasi yang saling berhubungan. Sistem operasi Linux mempunyai manajemen file berbentuk seperti piramida. Membentuk tingkatan struktur file seperti pohon silsilah keluarga. Struktur ini memudahkan pengguna linux untuk mengelola berbagai file.



Sistem file pada linux dimulai dari root yang ditandai dengan symbol “ / ”. Kemudian direktori dan subdirektori. Kita dapat membuat file mulai dari root kebawah. Pada umumnya terdapat dua macam file yang berbeda. Pertama merupakan jenis file biasa yang berguna untuk menyimpan data. Sebagian besar file merupakan jenis file biasa yang berguna untuk menyimpan berbagai jenis data. Sedangkan jenis file yang kedua ialah file direktori yang berguna untuk menyimpan dan menghimpun berbagai file-file.

Tabel Direktori Standar Linux:

Direktori	Deskripsi
/etc	Berisi file <i>administrative</i> dan file <i>executable</i> yang berguna untuk administrasi sistem.
/dev	Berisi file khusus yang merepresentasikan peralatan <i>hardware</i>
/bin	Berisi program standart Linux
/usr/sbin /usr/bin	Berisi utilitas Linux

/usr/lib	Berisi program <i>library</i> yang diperlukan untuk kompilasi program.
/tmp	Berisi file sementara, yang pada saat <i>bootstrap</i> akan dihapus.
/boot	Berisi file yang sangat penting untuk proses <i>bootstrap</i> . <i>Kernel vmlinuz</i> disimpan di direktori ini.
/proc	Berisi informasi tentang <i>kernel</i> Linux, proses dan <i>virtual system file</i> .
/var	Direktori variabel, artinya tempat penyimpanan LOG (catatan hasil <i>output</i> program), file ini dapat membengkak dan perlu dimonitor perkembangannya.
/home	Berisi direktori untuk pemakai Linux (pada SCO diletakkan pada /usr).
/mnt	Direktori untuk <i>mounting system file</i> .
/root	Home directory untuk <i>superuser</i> (root).
/usr/bin/X11	<i>Symbolic link</i> ke /usr/X11/bin, program untuk X-Window.
/usr/src	<i>Source code</i> untuk Linux
/opt	Option, direktori ini biasanya berisi aplikasi tambahan (“add-on”) seperti Netscape Navigator, kde, gnome, applix dll.

Manajemen Disk

Manajemen disk merupakan adalah sarana penyimpanan yang berfungsi untuk menyimpan data, file dan informasi secara permanen dalam suatu sistem computer. Media disk berada satu tingkat dibawah memori utama dalam hierarki memori. Media disk berbeda sekali dengan memori utama computer karena tidak mempunyai hubungan langsung dengan prosesor computer. Linux mensupport berbagai file system diantaranya adalah:

1. Ext2
2. Ext3
3. Ext4

LEMBAR KERJA DAN TUGAS

Percobaan 1: Direktori

1. Melihat direktori aktif

```
$ pwd
$ ls
```
2. Melihat Direktori HOME

```
$ echo $HOME
```
3. Membuat direktori dan subdirektori

```
$ cd /home/[nama user]
$ mkdir folderkul
$ mkdir folderku2
$ ls
$ mkdir A B C A/D A/E A/F B/G
$ ls
```

4. Menghapus direktori dan subdirektori


```
$ rmdir folderkul
$ rmdir B/G B
$ ls
```
5. Navigasi direktori


```
$ cd A
$ pwd
$ cd E
$ pwd
$ cd ..
$ pwd
$ cd ..
```

Percobaan 2: File

1. Membuat file


```
$ cat > fileku.txt
```

*Belajar membuat file
Semoga bermanfaat
Untuk yang belajar
Akhiri dengan menekan
[Ctrl-D]*

```
$ nano filenano.txt
```

*Belajar menggunakan nano
Nano adalah editor di linux
Nano mudah digunakan
Nano cocok untuk pemula
Semoga bermanfaat*

```
$ ls
```

2. Menghapus file

```
$ rm fileku.txt
$ ls
```

3. Mengubah file

```
$ nano filenano.txt
```

*Selamat ...
 Telah diubah !!!*

```
$ ls
```

Percobaan 3: Ijin Akses

1. Melihat ijin akses


```
$ ls -l
```
2. Mengubah ijin akses


```
$ chmod 444 filenano.txt
```

```
$ ls -l

$ chmod 666 filenano.txt
$ ls -l

$ chmod 777 filenano.txt
$ ls -l
```

Percobaan 4: Owner

1. Melihat owner file
\$ ls -l
2. Mengubah owner file
\$ chown [nama user]:[nama user] filenano.txt
\$ ls -l

Percobaan 5: Group

1. Melihat group file
\$ ls -l
2. Mengubah group file
\$ chgrp root filenano.txt
\$ ls -l

Percobaan 6: Disk

1. Melihat daftar partisi
\$ fdisk -l
\$ blkid
2. Melihat kapasitas partisi
\$ df -h
3. Melihat tipe harddisk
\$ hdparm -I /dev/sda
4. Melihat partisi swap
\$ swapon -s

Tugas

1. Buatlah subdirektori *server*, *client*, *switch* dan *hub* sekaligus pada direktori *computer*.
2. Buatlah file laptop.txt yang berisi nama lengkap dan nim anda pada subdirektori server dan copy-kan file tersebut ke sub direktori client, switch dan hub.
3. Ubahlah ijin akses file laptop.txt pada sub direktori server sehingga *user*, *group* dan *others* hanya dapat melakukan *read*.
4. Ubahlah ijin akses file laptop.txt pada sub direktori client sehingga *user*, *group* dan *others* hanya dapat melakukan *execute*.

5. Ubahlah ijin akses file laptop.txt pada sub direktori switch sehingga *user*, *group* dan *others* hanya dapat melakukan *write*.
6. Ubahlah ijin akses file laptop.txt pada sub direktori hub sehingga *user*, *group* dan *others* dapat melakukan *read*, *write* dan *execute*.
7. Hapuslah direktori *hub*.
8. Ubahlah owner file laptop.txt pada subdirektori server menjadi owner root.
9. Ubahlah group file laptop.txt pada subdirektori server menjadi group root.
10. Hapuslah direktori *computer*.

REFERENSI

1. [Stallings2001] William Stallings. Hak Cipta © 2001. Operating Systems. Prentice Hall.
2. [Tanenbaum1992] Andrew S. Tanenbaum. Hak Cipta © 1992. Modern Operating Systems. Prentice-Hall Inc..
3. [Silberschatz2000] Avi Silberschatz, Peter Galvin, dan Grag Gagne. Hak Cipta © 2000. Operating Systems CONCEPTS

POKOK BAHASAN 6

Sistem Operasi Tersebar

PENDAHULUAN

Pada pokok bahasan ini akan dibahas mengenai Pengenalan Konsep Komunikasi Jaringan Pengenalan Sistem Operasi Tersebar . Sehingga setelah mempelajari bab ini diharapkan mahasiswa mampu:

- Siswa mengetahui apa yang dibutuhkan agar komputer-komputer dapat saling berkomunikasi
- Siswa mengetahui sistem operasi tersebar diwujudkan melalui komunikasi jaringan

PENYAJIAN (TUTORIAL)

DEFINISI SISTEM OPERASI TERSEBAR

Sistem operasi tersebar adalah sebuah sistem yang komponennya berada pada jaringan komputer. Komponen tersebut saling berinteraksi dan melakukan komunikasi/koordinasi hanya dengan pengiriman pesan. Sistem operasi terdistribusi adalah salah satu implementasi sistem operasi tersebar, dimana sekumpulan komputer dan prosesor yang terhubung dalam satu jaringan. Koleksi-koleksi dari objek-objek ini secara tertutup bekerja secara bersama-sama untuk melakukan suatu tugas atau pekerjaan tertentu. Tujuan utamanya adalah untuk memberikan hasil secara lebih, terutama dalam:

- \$ File system
- \$ Name space
- \$ Waktu pengolahan
- \$ Keamanan Akses ke seluruh resources,

LANGKAH-LANGKAH

- Masuk ke sistem operasi Linux
- Login sebagai user.(user Anda)
- Masuk ke root dengan perintah **su**
- Instalasi paket software menggunakan **yum**

LEMBAR KERJA DAN TUGAS

Percobaan 1 : Yum script

- Install paket samba dari repository lokal umsida.

```
# cd /tmp/
# yum -y install samba
•Melihat files apa saja yang sudah di install
#ls /etc/samba
```

- Melihat file konfigurasi

```
#ls -l /etc/samba
```

- Melihat dokumen file

```
#rpm -qd samba
```

- Mengaktifkan software

```
# /etc/rc.d/init.d/smb start
```

- Merestart software

```
#/etc/rc.d/init.d/smb restart
```

- Mematikan software

```
#/etc/rc.d/init.d/smb stop
```

Percobaan 2 : Konfigurasi software

1. Untuk konfigurasi software yang sudah kita install tadi bisa menggunakan command **NANO**.

```
#nano /etc/samba/smb.conf
```

2. Mengisi file konfigurasi

[Global]

```
workgroup = workgroup
domain logons = No
domain master = No
security =
netbios name = ICTSERVER
map to guest = bad user
backend = tdb
```

[Share]

```
comment = Share
path = /home/share
read only = no
browseable = yes
public = yes
writeable = yes
```

3. Merestart software samba

```
#/etc/rc.d/init.d/smb restart
```

Percobaan 3 : Membuat folder

1. Membuat folder yang nantinya akan kita share

```
# mkdir /home/share
```

2. Memberi hak akses ke folder yang akan kita share

```
# chmod -R 777 /home/share
```

Percobaan 4 : Mencoba file share

- Mencoba browser file share
Places > Connect to server > masukkan ip komputer > connect
- Mengetahui ip komputer
ifconfig

TUGAS

1. Buatlah sebuah share folder dengan menggunakan password untuk bisa akses folder tersebut?
2. Ubah perijinan di smb.conf agar guest tidak bisa membuat folder di share folder

REFERENSI

1. [Stallings2001] William Stallings. Hak Cipta © 2001. Operating Systems. Prentice Hall.
2. [Tanenbaum1992] Andrew S. Tanenbaum. Hak Cipta © 1992. Modern Operating Systems. Prentice-Hall Inc..
3. [Silberschatz2000] Avi Silberschatz, Peter Galvin, dan Grag Gagne. Hak Cipta © 2000. Operating Systems CONCEPTS

LAMPIRAN

Riwayat Hidup Penulis



Ir. Sumarno, Lahir di surabaya tanggal 27 Mei 1961. Melanjutkan pendidikan ke S1 Universitas Muhammadiyah Surabaya dengan gelar Ir. Kemudian penulis menyelesaikan studi S2 nya di Magister Manajemen Universitas Muhammadiyah Malang. Kini menjabat Kepala BAU UMSIDA. Mata kuliah yang diampuh penulis yaitu Teori Bahasa Automata, Sistem Operasi , Jaringan Komputer dan Sistem Digital.



Hamzah Setiawan, lahir di Sidoarjo tanggal 30 April 1986. Setelah menyelesaikan Sekolah Dasar, SLTP, dan SLTA di Sidoarjo, melanjutkan Pendidikan ke Universitas Trunojoyo Madura. Meraih gelar sarjana (S.Kom) Fakultas Teknik Prodi Informatika tahun 2009. Saat ini penulis masih dalam proses study lanjut (S2) di Sekolah Tinggi Teknik surabaya (STTS) Prodi Teknologi Informasi. Kini menjabat Kasi Kemahasiswaan UMSIDA. Aktifitas keseharian berhidmat di Pimpinan Daerah Pemuda Muhammadiyah Sidoarjo dan Lembaga Pengembangan Cabang Ranting (LPCR) PWM Jatim. Mata Kuliah yang di ampuh penulis yaitu Algoritma dan Pemrograman, Aljabar Linier, Pengantar Teknologi Informasi, Metode Numerik, Sistem terdistribusi, Analisa desain dan Sistem Informasi, Pengembangan system berbasis Web



Roni Tambudi