**JOBSHEET 2**

**DASAR SISTEM OPERASI**

**CAPAIAN PEMBELAJARAN**

1. Mengelola beragam device atau piranti I/O yang berbeda karakteristiknya dan dapat digunakan dengan baik oleh proses-proses diantaranya:

· Penjadwalan peranti I/O

· Buffering

· Reservasi device

· Penanganan kesalahan pengakses peranti I/O

1. Melakukan perintah Dasar SO:

· Login

**TEORI DAN PRAKTIKUM**

**TEORI**

1. **PENJADWALAN PERANTI I/O**

Penjadwalan perangkat I/O adalah proses yang penting dalam mengelola beragam perangkat atau piranti I/O dengan karakteristik yang berbeda. Ini melibatkan pengaturan urutan akses ke perangkat I/O agar dapat dimanfaatkan secara efisien oleh proses-proses yang berjalan dalam sistem komputer. Dengan melakukan penjadwalan perangkat I/O dengan baik, sistem dapat mengoptimalkan penggunaan sumber daya dan meningkatkan kinerja keseluruhan. **Penjadwalan FCFS (First-Come, First-Served)**: Perangkat I/O dilayani berdasarkan urutan kedatangan permintaan. Metode ini sederhana tetapi tidak selalu efisien karena tidak mempertimbangkan prioritas atau karakteristik khusus perangkat. **Penjadwalan Prioritas**: Setiap permintaan I/O diberi prioritas tertentu, dan perangkat dilayani sesuai dengan prioritasnya. Prioritas dapat ditentukan berdasarkan berbagai faktor, seperti kepentingan proses atau jenis data yang ditangani. **Penjadwalan SJF (Shortest Job First)**: Perangkat I/O dilayani berdasarkan waktu yang diperlukan untuk menyelesaikan permintaan tersebut. Permintaan dengan waktu pengerjaan yang lebih pendek akan dilayani terlebih dahulu. **Penjadwalan Round Robin**: Setiap permintaan I/O diberi waktu tertentu untuk dilayani, dan jika tidak selesai dalam waktu tersebut, permintaan tersebut dipindahkan ke antrian belakang. Metode ini memastikan bahwa setiap permintaan I/O mendapat porsi waktu yang adil. **Penjadwalan Real-Time**: Digunakan dalam sistem yang membutuhkan respons cepat terhadap perangkat I/O. Permintaan I/O yang memiliki batas waktu tertentu untuk dilayani dengan prioritas lebih tinggi. **Penjadwalan Prioritas Dinamis**: Prioritas diberikan secara dinamis berdasarkan karakteristik atau kondisi saat ini dari perangkat I/O atau proses yang meminta layanan I/O. Implementasi penjadwalan perangkat I/O yang efektif membutuhkan pemahaman yang mendalam tentang karakteristik perangkat keras dan kebutuhan aplikasi yang berjalan di atasnya.

1. **BUFFERING**

Buffering adalah teknik yang umum digunakan dalam sistem komputer untuk mengatasi perbedaan kecepatan antara proses-produksi data dan proses-konsumsi data. Ini terjadi ketika dua entitas dalam sistem memiliki kecepatan kerja yang berbeda, seperti ketika data diproduksi lebih cepat daripada yang dapat dikonsumsi atau sebaliknya. Buffering membantu dalam menyamakan laju kedua proses tersebut dengan menyimpan sementara data dalam suatu buffer (penyimpanan sementara) sehingga data dapat diambil atau dikirimkan dengan kecepatan yang sesuai. Ada beberapa jenis buffering, termasuk: **Buffer Input**: Digunakan ketika data masuk ke sistem dengan kecepatan yang lebih cepat daripada yang dapat ditangani oleh proses pengolahan. Data masuk disimpan dalam buffer input sehingga proses pengolahan dapat mengambil data dari buffer sesuai dengan kebutuhannya. **Buffer Output**: Digunakan ketika data diproduksi dengan kecepatan yang lebih cepat daripada yang dapat ditransmisikan ke tujuan akhir. Data yang telah diproses disimpan dalam buffer output sehingga dapat dikirimkan dengan kecepatan yang sesuai ke perangkat atau sistem lain. **Buffer Input-Output (I/O)**: Digunakan dalam komunikasi antara perangkat I/O dan prosesor. Data yang diterima dari perangkat I/O atau akan dikirim ke perangkat I/O disimpan dalam buffer I/O untuk mengakomodasi perbedaan kecepatan antara perangkat I/O dan prosesor. Manfaat utama dari buffering termasuk mengoptimalkan penggunaan sumber daya sistem, mengurangi latensi, dan menghindari kehilangan data akibat ketidakcocokan kecepatan antara proses-produksi dan proses-konsumsi. Buffering merupakan komponen penting dalam desain sistem komputer modern untuk memastikan kinerja yang efisien dan handal.

1. **RESERVASI**

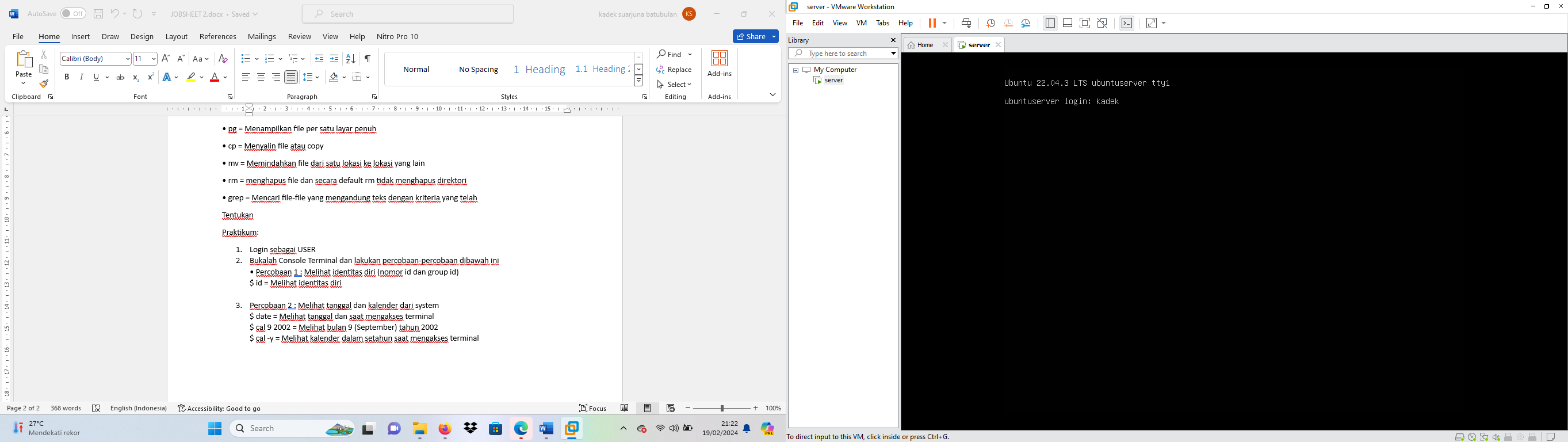
Reservasi perangkat (device reservation) adalah sebuah konsep dalam sistem komputer di mana sebuah perangkat keras atau sumber daya sistem didedikasikan atau dialokasikan secara eksklusif untuk penggunaan tertentu atau proses tertentu untuk jangka waktu tertentu. Ini berarti bahwa perangkat tersebut tidak dapat digunakan oleh proses atau aplikasi lain selama periode waktu yang telah ditentukan. Konsep ini sering digunakan dalam lingkungan di mana sumber daya terbatas harus dibagi secara adil di antara proses-proses yang bersaing. **Reservasi CPU**: Sebuah proses atau aplikasi mungkin memerlukan akses eksklusif ke CPU untuk menyelesaikan tugas kritisnya dengan cepat dan tanpa gangguan. Dalam kasus ini, CPU dapat "direservasi" untuk proses tersebut sehingga tidak digunakan oleh proses-proses lain. **Reservasi Memori**: Sebuah proses yang membutuhkan jumlah memori yang besar untuk menjalankan operasi khusus, seperti pengolahan data besar atau pemrosesan paralel, mungkin melakukan reservasi memori untuk memastikan bahwa memori tersebut tersedia dan tidak digunakan oleh proses lain. **Reservasi Perangkat I/O**: Ketika aplikasi memiliki kebutuhan yang sangat tinggi terhadap perangkat I/O tertentu, seperti hard disk atau jaringan, mereka dapat melakukan reservasi perangkat untuk memastikan akses eksklusif dan penggunaan bandwidth maksimum. **Reservasi Jaringan**: Dalam lingkungan jaringan, aplikasi atau layanan mungkin memerlukan bandwidth jaringan tertentu untuk memastikan kinerja yang konsisten. Dengan melakukan reservasi, bandwidth jaringan dapat dialokasikan secara eksklusif untuk aplikasi atau layanan tersebut. Reservasi perangkat dapat membantu dalam menghindari konflik sumber daya, memprioritaskan akses ke sumber daya yang penting, dan meningkatkan kinerja keseluruhan sistem. Namun, penggunaan reservasi harus dikelola dengan hati-hati untuk memastikan bahwa sumber daya tidak disia-siakan dan untuk mencegah situasi di mana proses-proses lain mengalami kesulitan dalam mendapatkan akses ke sumber daya yang mereka perlukan.

1. **PENANGANAN KESALAHAN PENGAKSESAN PERANGKAT I/O**

Penanganan kesalahan pengaksesan perangkat I/O adalah bagian penting dari pengembangan sistem yang dapat diandalkan. Kesalahan dalam pengaksesan perangkat I/O dapat terjadi karena berbagai alasan, seperti kegagalan perangkat keras, kehilangan koneksi, atau masalah perangkat lunak. **Deteksi Kesalahan**: Sistem perlu dilengkapi dengan mekanisme deteksi kesalahan yang dapat mengenali adanya masalah saat pengaksesan perangkat I/O. Ini dapat mencakup penggunaan kode kesalahan, status perangkat, atau mekanisme lainnya untuk mengindikasikan kegagalan. **Pemulihan Otomatis**: Dalam beberapa kasus, sistem dapat dirancang untuk melakukan pemulihan otomatis saat terjadi kesalahan. Ini bisa berupa mencoba kembali operasi, mengalihkan ke perangkat cadangan, atau melakukan tindakan pemulihan lainnya tanpa campur tangan pengguna. **Notifikasi Kesalahan**: Penting untuk memberi tahu pengguna atau proses terkait tentang adanya kesalahan dalam pengaksesan perangkat I/O. Notifikasi ini dapat berupa pesan kesalahan yang jelas atau log yang mencatat kejadian kesalahan untuk pemeriksaan lebih lanjut. **Pengembalian Kode Kesalahan**: Ketika terjadi kesalahan, penting untuk memberikan informasi yang berguna kepada pengguna atau proses terkait. Ini bisa berupa pengembalian kode kesalahan yang spesifik yang dapat digunakan untuk menentukan tindakan selanjutnya. **Tindakan Pemulihan Manual**: Dalam beberapa kasus, tindakan pemulihan mungkin memerlukan intervensi manual. Ini bisa termasuk memeriksa koneksi perangkat keras, mengganti perangkat yang rusak, atau melakukan operasi pemulihan lainnya yang tidak dapat dilakukan secara otomatis. **Logging dan Pelaporan**: Kesalahan pengaksesan perangkat I/O harus dilacak dalam log sistem untuk pemeriksaan dan analisis lebih lanjut. Ini membantu dalam menentukan pola kesalahan yang mungkin terjadi secara berkala dan mengidentifikasi masalah sistem yang mendasarinya. **Pengaturan Kebijakan**: Sistem juga dapat dirancang dengan pengaturan kebijakan yang memungkinkan administrator untuk mengkonfigurasi cara penanganan kesalahan, termasuk langkah-langkah pemulihan yang diinginkan. **Pengujian dan Pemeliharaan**: Penting untuk melakukan pengujian reguler dan pemeliharaan perangkat I/O untuk mengidentifikasi potensi masalah sebelum mereka menjadi kegagalan yang signifikan. Hal ini membantu mencegah kesalahan pengaksesan perangkat I/O yang tidak diinginkan. Dengan menerapkan langkah-langkah ini, sistem dapat dikembangkan dengan kemampuan untuk menangani kesalahan pengaksesan perangkat I/O dengan efektif, meningkatkan keandalan dan kinerja keseluruhan.

**PRAKTIKUM:**

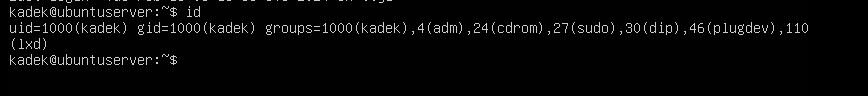
1. Login sebagai USER



1. Bukalah Console Terminal dan lakukan percobaan-percobaan dibawah ini

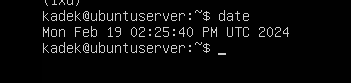
• Percobaan 1 : Melihat identitas diri (nomor id dan group id)

$ id = Melihat identitas diri



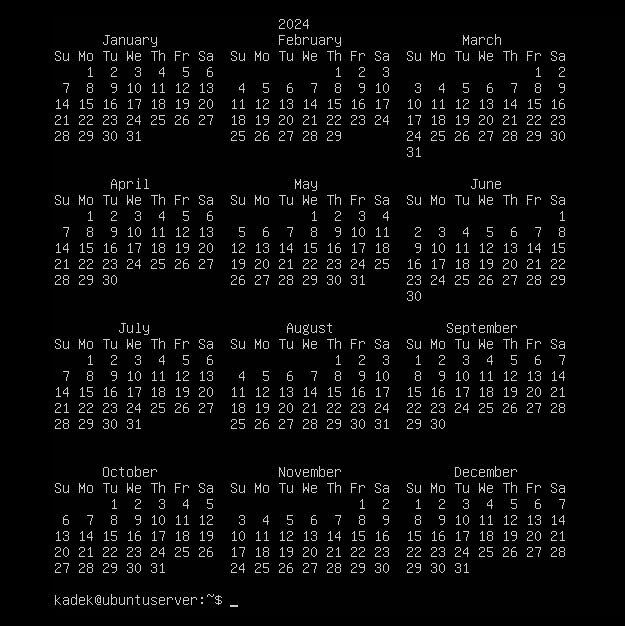
1. Percobaan 2 : Melihat tanggal dan kalender dari system

$ date = Melihat tanggal dan saat mengakses terminal



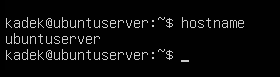
$ cal 20 2024 = Melihat bulan 20 (Februari) tahun 2024

$ cal -y = Melihat kalender dalam setahun saat mengakses terminal



1. Percobaan 3 : Melihat identitas mesin

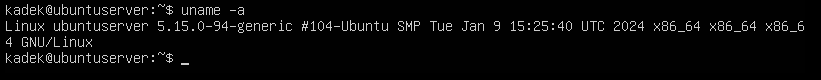
$ hostname = Melihat nama host dari computer



$ uname = Melihat sistem operasi



$ uname -a = Melihat informasi semua sistem operasi. Option adalah -a (all)



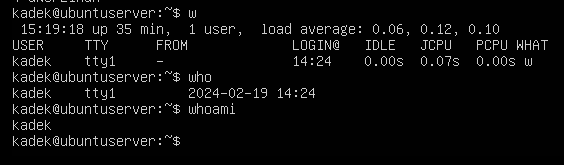
5. Percobaan 4 : Melihat siapa yang sedang aktif

1. Mengetahui siapa saja yang aktif

$ w = Informasi user

$ who = Melihat nama user, tanggal, dan waktu yang sedang login saat ini

$ whoami = Melihat nama user yang sedang mengakses linux ini



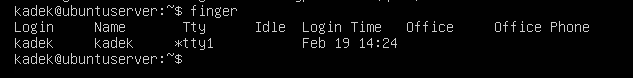
1. Mengubah informasi finger

$ chfn <user> = Mengubah informasi finger

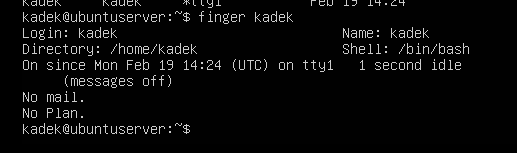


1. Melihat informasi finger

$ finger = Finger hampir sama dengan who, namun finger menyediakan informasi identitas user yang lebih lengkap dari who



$ finger <user> = Melihat informasi dari user yangspesifik yaitu “Damelia”



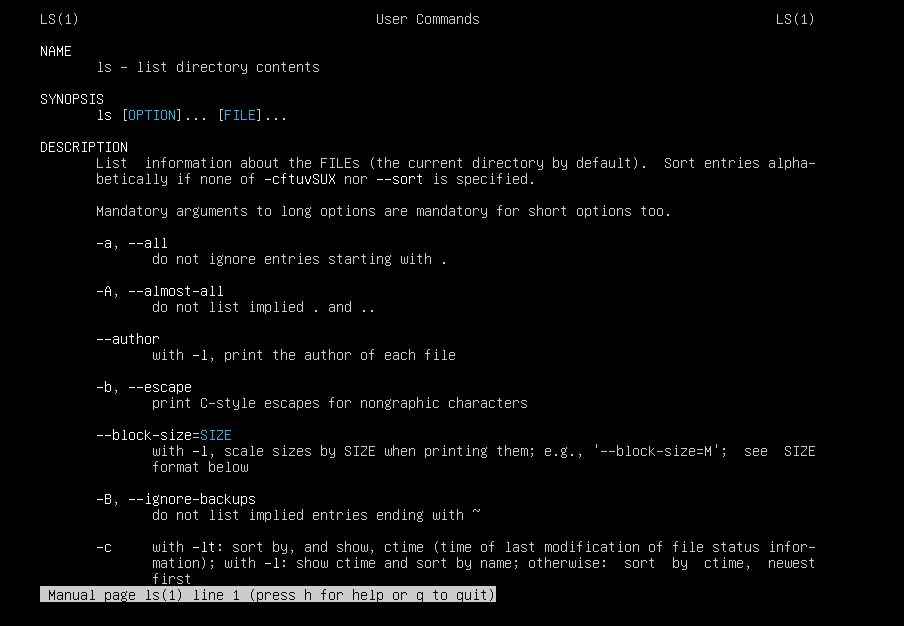
6. Percobaan 5 : Menggunakan manual

$ man = Singkatan dari manual yaitu untuk menampilkan informasi bantu untuk semua perintah LINUX. Perintah ini sangat bermanfaat bagi setiap pemakai LINUX karena dapat membantu mengingat kembali perintah-perintah LINUX

$ man ls = Menjelaskan informasi lengkap fungsi ls

$ man -k file = Menjelaskan informasi kibibytes

$ man 5 passwd = Menjelaskan informasi lengkap fungsi passwd(5)



1. Percobaan 6 : Menghapus layer

$ clear = membersihkan / menghapus layer terminal



1. Percobaan 7 : Mencari perintah yang deskripsinya mengandung kata kunci yang dicari

$ apropos date = Mencari perintah yang deskripsinya mengandung date

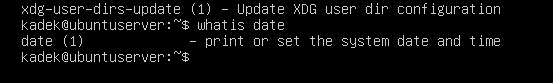
$ apropos mail = Mencari perintah yang deskripsinya mengandung mail

$ apropos telnet = Mencari perintah yang deskripsinya mengandung telnet



1. Percobaan 8 : Mencari perintah yang tepat sama dengan kunci yang dicari

$ whatis date = Mencari perintah yang sama dengan “date”



1. Percobaan 9 : Manipulasi berkas (file) dan direktori

$ ls = Menampilkan current working directory

$ ls -l = Melihat semua file lengkap

$ ls -a = Menampilkan semua file atau direktori yang tersembunyi

$ ls -f = Menampilkan semua file atau direktori tanpa proses sorting

$ ls /usr = Menampilkan isi suatu direktori

$ ls / = Menampilkan isi direktori root

$ ls -F /etc = Menampilkan semua file atau direktori dengan menandai : tanda (/) untuk direktori, tanda asterik (\*) untuk file yang bersifat executable, tanda (@) untuk file symbolic link, tanda (=) untuk socket, tanda (%) untuk whiteout dan tanda (|) untuk FIFO

$ ls -1 /etc = Menampilkan file atau direktori secara lengkap yaitu terdiri dari nama file, ukuran, tanggal dimodifikasi, pemilik, group dan mode atau atributnya

$ ls -R /usr = Menampilkan semua file dan isi direktori. Argumen ini akan menyebabkan proses berjalan agak lama, apabila proses akan dihentikan dapat menggunakan ^c

1. Percobaan 10 : Melihat tipe file

$ file = Melihat tipe file

$ file \* = Melihat tipe semua file dan beberapa direktori

$ file /bin/ls = Melihat tipe file dalam /bin/ls

12. Percobaan 11 : Menyalin file

a. Meng-copy suatu file. Berikan opsi –I untuk pertanyaan interaktif bila file sudah ada

$ cp /etc/group f1 = Meng-copy /etc/group ke file f1

$ ls –l = Melihat file lengkap

$ cp –i f1 f2 = Meng-copy f1 ke f2

b. Meng-copy ke direktori

$ mkdir backup = Membuat direktori bermana backup

$ cp f1 f3 = Meng-copy f1 ke f3

$ cp f1 f2 f3 backup = Meng-copy f1, f2, f3 ke direktori backup

$ ls backup = Melihat isi direktori backup

$ cd backup = Masuk ke direktori backup

$ ls = Melihat file di dalam direktori backup yang sudah masuk sebelumnya di cd

13. Percobaan 12 : Melihat isi file

$ cat f1 = Menampilkan isi file f1

$ more f1 = Menampilkan file f1 per satu layar penuh

$ pg f1 = Menampilkan file f1 per satu layar penuh

14. Percobaan 13 : Mengubah nama file

$ mv f1 prog.txt = Merubah nama file f1 menjadi file prog.text

$ ls = melihat isi daftar file

$ mkdir mydir = Membuat direktori bermana mydir

$ mv f1 f2 f3 mydir = Memindahkan file f1, f2, f3 ke direktori mydir. Dikarenakan f1 sudah menjadi prog.text maka saya mengubah perintah menjadi $ mv f2 f3 prog.text mydir

15. Percobaan 14 : Menghapus file

$ rm f1 = Menghapus file f1

$ cp mydir/f1 f1 = Meng-copy mydir/f1 ke f1

$ cp mydir/f2 f2 = Meng-copy mydir/f1 ke f2

$ rm f1 = Menghapus file f1

$ rm –i f2 = Menghapus regular file f2 dengan konfirmasi

16. Percobaan 15 : Mencari kata atau kalimat dalam file grep akan mencari suatu variable dalam suatu baris tertentu, di dalam sembarang direktori pada semua file. Grep sangat berguna untuk menemukan kata tertentu dalam beberapa dokumen atau mencari adanya sebuah variable dalam sekelompok program.

$ grep root /etc/passwd = Mencari kata “root” pada /etc/passwd

$ grep “:0:” /etc/passwd = Mencari kata“:0:” pada /etc/passwd

$ grep student /etc/passwd = Mencari pada“student” pada /etc/passwd

**Tugas**

Lengkapi Jobsheet dengan menjalankan setiap perintahnya dan hasilnya di screenshot.