**LAPORAN PRAKTIKUM SISTEM OPERASI**

****

**Praktikum : 5 (Lima)**

**Nama : Reja Revaldy F**

**Nim : C030322067**

**Mata Kuliah : Sistem Operasi**

**Dosen Pengampu : Fuad Sholihin, S.T., M.Kom**

**PROGRAM STUDI D3 TEKNIK INFORMATIKA**

**JURUSAN TEKNIK ELEKTRO**

**POLITEKNIK NEGERI BANJARMASIN**

**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI**

PRAKTIKUM 5

Bash Shell

### TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan shell dan sebutkan shell yang ada di sistem operasi Linux.

Shell adalah program yang menyediakan antarmuka pengguna untuk berinteraksi dengan sistem operasi. Shell bertindak sebagai penghubung antara pengguna dan sistem operasi, memungkinkan pengguna untuk menjalankan perintah-perintah sistem dan program-program lainnya.

1. Apa kegunaan wildcard, kunci Tab dan history ?

* Wildcard: Wildcard digunakan untuk mencocokkan pola atau kumpulan file atau direktori dalam perintah shell. Karakter wildcard yang umum digunakan adalah "" (mencocokkan nol atau lebih karakter), "?" (mencocokkan satu karakter), dan "[]" (mencocokkan karakter dalam set tertentu). Misalnya, ".txt" akan mencocokkan semua file dengan ekstensi .txt dalam direktori saat ini.
* Kunci Tab: Kunci Tab digunakan untuk melengkapi otomatis perintah, nama file, atau direktori saat mengetik di shell. Ketika Anda mulai mengetik sesuatu dan menekan Tab, shell akan mencoba untuk melengkapi atau menampilkan pilihan yang tersedia berdasarkan konteks.
* History: History menyimpan daftar perintah yang telah dieksekusi sebelumnya dalam shell. Dengan menggunakan fitur history, Anda dapat melihat, mengedit, dan menjalankan kembali perintah-perintah sebelumnya tanpa perlu mengetik ulang. Biasanya, tombol panah atas dan bawah pada keyboard digunakan untuk mengakses perintah sebelumnya dalam riwayat.

1. Apa yang dimaksud fitur tilde (~), parameter $, kurung kurawal { }, perintah ` ` atau $( ), $[ ], tanda \, tanda petik satu (’) dan petik dua (”)

* Fitur tilde (): Tilde () digunakan sebagai pintasan untuk direktori home pengguna. Misalnya, "~/Documents" akan merujuk pada direktori "Documents" di direktori home pengguna.
* Parameter $: Parameter $ digunakan untuk mengakses nilai dari variabel lingkungan (environment variables) atau parameter-posisional dalam shell. Misalnya, $HOME akan mengembalikan nilai dari variabel lingkungan HOME yang berisi direktori home pengguna.
* Kurung kurawal { }: Kurung kurawal digunakan untuk membuat ekspansi string atau brace expansion dalam shell. Misalnya, echo {1..3} akan menghasilkan "1 2 3".
* Perintah atau $( ): Perintah backticks ( ) atau perintah dalam tanda kurung $( ) digunakan untuk mengeksekusi perintah shell di dalam perintah shell lainnya. Misalnya, output=$(ls) atau output=ls akan menjalankan perintah "ls" dan menyimpan outputnya ke dalam variabel "output".
* $[ ]: Notasi $[ ] digunakan untuk evaluasi aritmatika dalam shell. Misalnya, $[2 + 2] akan menghasilkan nilai 4.
* Tanda : Tanda backslash () digunakan untuk menghindari interpretasi karakter khusus atau escape karakter. Misalnya, untuk mencetak karakter petik satu (') secara literal, Anda dapat menggunakan tanda petik satu dengan backslash (''').
* Tanda petik satu (’): Tanda petik satu (' ') digunakan untuk menandai string literal di dalam shell, di mana variabel atau ekspresi dalam tanda petik satu tidak akan diinterpretasikan atau dievaluasi. Misalnya, echo 'Hello $NAME' akan mencetak "Hello $NAME" sebagai output, bukan nilai dari variabel NAME.
* Tanda petik dua ("): Tanda petik dua (" ") digunakan untuk membuat string yang dapat diinterpretasikan oleh shell, sehingga variabel atau ekspresi di dalam tanda petik dua akan dievaluasi. Misalnya, echo "Hello $NAME" akan mencetak "Hello" diikuti oleh nilai dari variabel NAME.

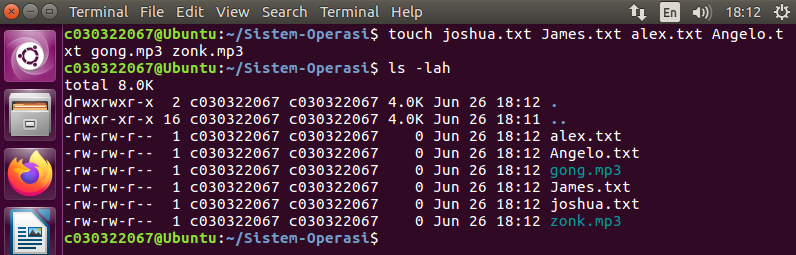
### PERCOBAAN

1. Login sebagai user.
2. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
3. Selesaikan soal-soal latihan

### Percobaan 1 : Wildcard

1. Membuat beberapa file

$ touch joshua.txt James.txt alex.txt Angelo.txt gong.mp3 zonk.mp3

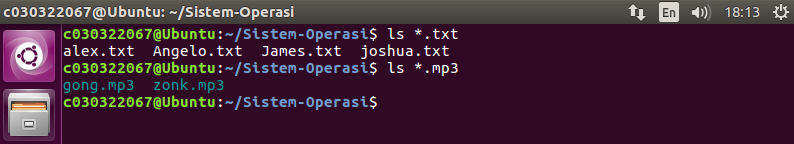


Analisa : Perintah di atas menggunakan touch untuk membuat beberapa file sekaligus. File-file yang dibuat adalah joshua.txt, James.txt, alex.txt, Angelo.txt, gong.mp3, dan zonk.mp3.

1. Melihat daftar file yang mempunyai extension sama

$ ls \*.txt

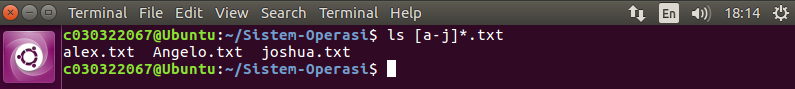
$ ls \*.mp3



Analisa : Perintah di atas menggunakan ls untuk menampilkan daftar file dalam direktori yang memenuhi pola tertentu. Pada perintah pertama (ls \*.txt), hanya file-file dengan ekstensi .txt yang akan ditampilkan. Pada perintah kedua (ls \*.mp3), hanya file-file dengan ekstensi .mp3 yang akan ditampilkan.

1. Melihat daftar file dengan jangkauan tertentu

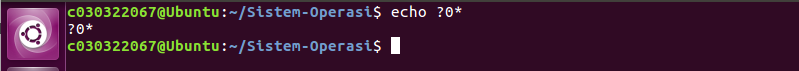
$ ls [a-j]\*.txt



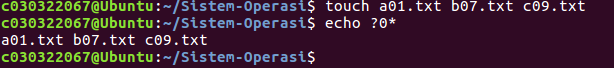
Analisa : Perintah di atas menggunakan ls untuk menampilkan daftar file dalam direktori yang dimulai dengan huruf a hingga j dan memiliki ekstensi .txt. Misalnya, file-file seperti apple.txt, banana.txt, dan jelly.txt akan ditampilkan.

1. Mencari nama file dengan single karakter tertentu

$ echo ?0\*

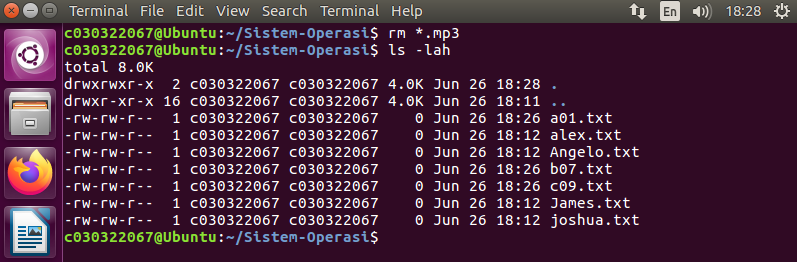


Analisa : Perintah di atas menggunakan echo untuk mencetak daftar file yang memiliki karakter pertama sebagai karakter tunggal (single character) dan diikuti dengan angka 0, serta memiliki karakter lain setelahnya. Misalnya, file-file seperti a01.txt, b07.txt, dan c09.txt akan dicetak. (Karena di percobaan tidak ada file tersebut maka saya buat file tersebut di bawahnya)



1. Menghapus file yang mempunyai extension .mp3

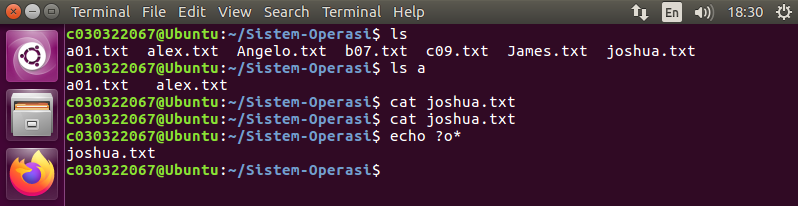
$ rm \*.mp3



Analisa : Perintah di atas menggunakan rm untuk menghapus semua file yang memiliki ekstensi .mp3 dalam direktori. File-file dengan ekstensi .mp3 akan dihapus. Harap berhati-hati saat menggunakan perintah rm karena perintah ini akan menghapus file secara permanen. Pastikan untuk memeriksa kembali file-file yang akan dihapus sebelum menjalankan perintah ini.

### Percobaan 2 : Kunci Tab

1. Gunakan kunci Tab dan lihat hasilnya



Analisa : Tab disini berguna seperti autocomplete.

### Percobaan 3 : Perintah History

1. Bash shell menyimpan ”history” perintah yang digunakan sebelumnya. Andadapat mengaksis history dalam beberapa cara. Cara paling mudah adalah menggunakan Panah Atas. Maka perintah sebelumnya akan ditampilkan.

Analisa : dengan menggunakan panah atas kita bisa mengakses command yang pernah kita jalankan sebelumnya.

1. Berikutnya, berikan Bash shell beberapa perintah untuk diingat. Masukkan perintah berikut dan tekan Enter pada setiap baris.

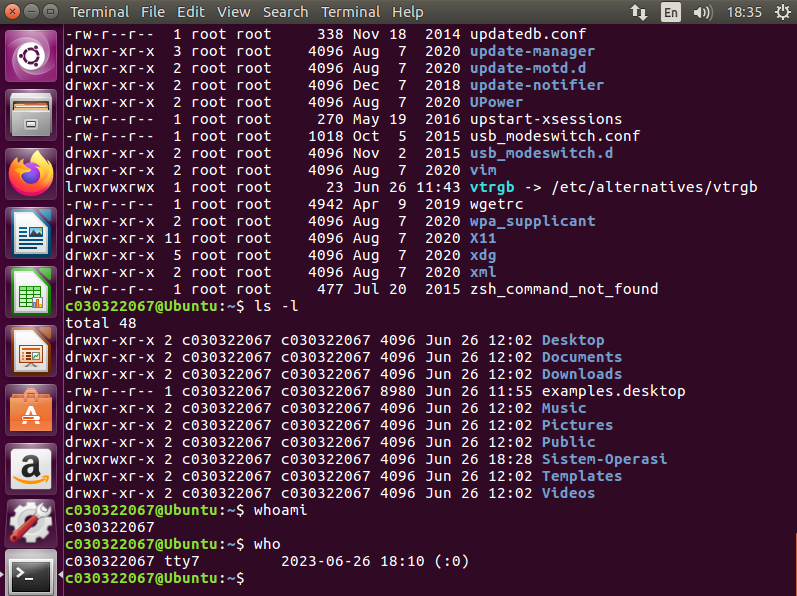
$ cd

$ ls –l /etc

$ ls –l

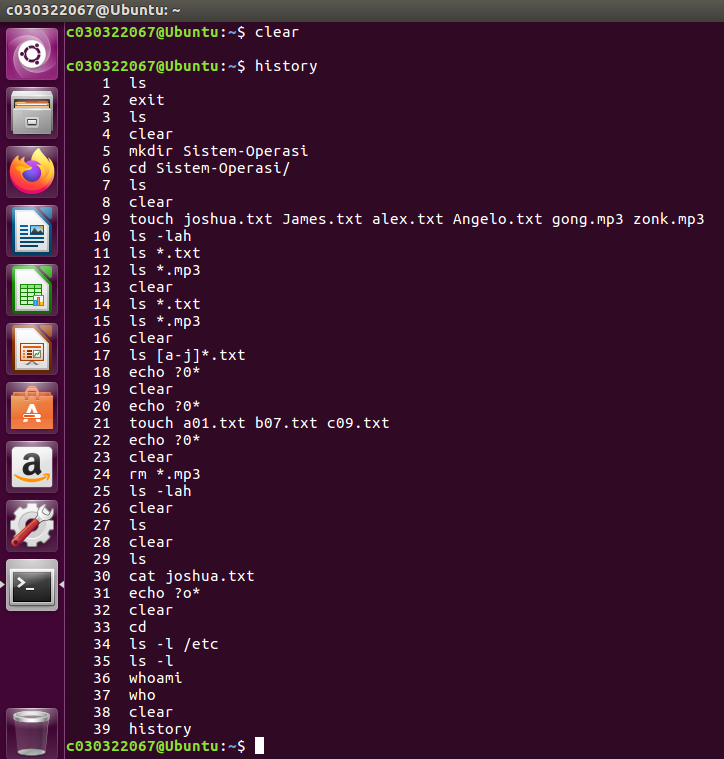
$ whoami

$ who



Analisa : TIdak ada hal yang spesial disini seperti perintah perintah yang pernah dikerjakan di praktikum sebelumnya.

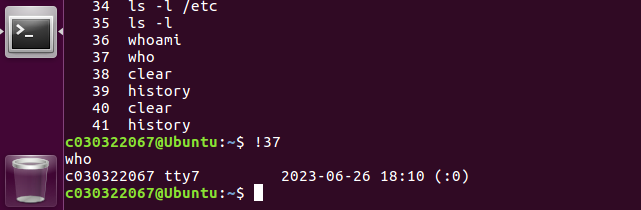
1. Untuk memeriksa apakah perintah ini ditambahkan pada history, dapat menggunakan perintah history untuk melihat semua perintah yang pernah dimasukkan.



Analisa : Perintah history berguna untuk memeriksa riwayat perintah dan dapat digunakan untuk merujuk kembali ke perintah sebelumnya, mengulang perintah tertentu, atau mencari perintah tertentu yang telah dimasukkan sebelumnya

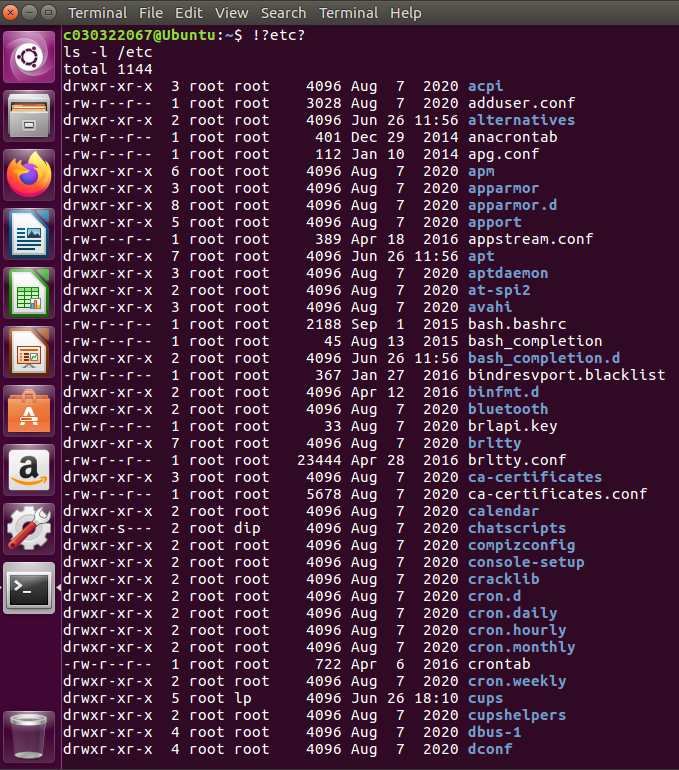
1. Anda dapat memilih perintah sebelumnya dengan menggunakan Panah Atas,tetapi hal ini tidak efisien untuk perintah yang semakin bertambah banyak. Cara yang mudah menggunkaan nomor pada perintah history atau mencarinya. Untuk memilih dan mengeksekusi perintah dengan nomor, masukkan kunci ! diikuti nomor perintah.

$ !<Nomor Perintah> Contoh : !781



Analisa : ! diikuti oleh nomor perintah memungkinkan Anda untuk dengan cepat mengulang atau mengeksekusi kembali perintah tertentu tanpa harus mencarinya secara manual dalam history.

1. Anda dapat mencari perintah dengan menyertakan perintah yang diinginkan.Misalnya !?etc?! akan menjalankan perintah ls –l /etc yang sebelumnya digunakan.



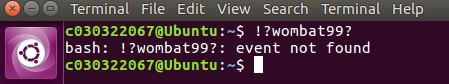
Analisa : Penggunaan !?<kata-kunci>?! memungkinkan Anda untuk dengan cepat mencari dan mengeksekusi kembali perintah yang mengandung kata-kunci tertentu dalam history perintah. Ini adalah cara yang efisien untuk menemukan dan menjalankan kembali perintah yang relevan dalam riwayat, tanpa perlu mencari secara manual.

1. Kemudian gunakan perintah history, maka akan terlihat perintah ls –l /etc yang kedua dan bukan !?etc?



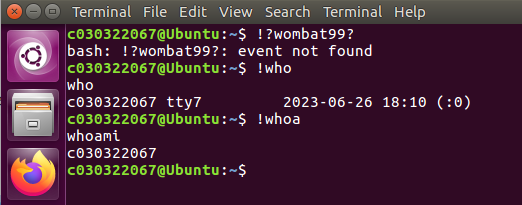
1. Apabila string tidka ditemukan pada perintah history maka akan terdapat pesan error.

$ !?wombat99?



Analisa : Pada perintah tersebut, !?wombat99? digunakan untuk mencari perintah yang mengandung string "wombat99" dalam riwayat perintah sebelumnya. Jika string "wombat99" tidak ditemukan dalam riwayat perintah, maka akan muncul pesan error yang mengindikasikan bahwa perintah tersebut tidak dapat ditemukan.

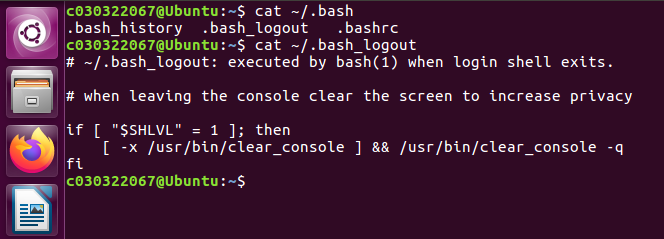
1. Jika diketikkan !who maka yang dijalankan adalah perintah who. Tetapi bila Anda ketikkan !whoa maka yang dijalankan adalah perintah whoami.



Analisa : Pada perintah !who, itu akan mengeksekusi perintah who dari riwayat perintah sebelumnya. Ini berarti perintah who akan dijalankan. Namun, pada perintah !whoa, itu akan mengeksekusi perintah yang mengandung string "whoa" dari riwayat perintah sebelumnya. Jika ada perintah sebelumnya yang mengandung "whoa", maka perintah tersebut akan dijalankan. Jadi, jika dalam riwayat perintah terdapat perintah seperti whoami, maka perintah whoami akan dijalankan jika menggunakan !whoa.

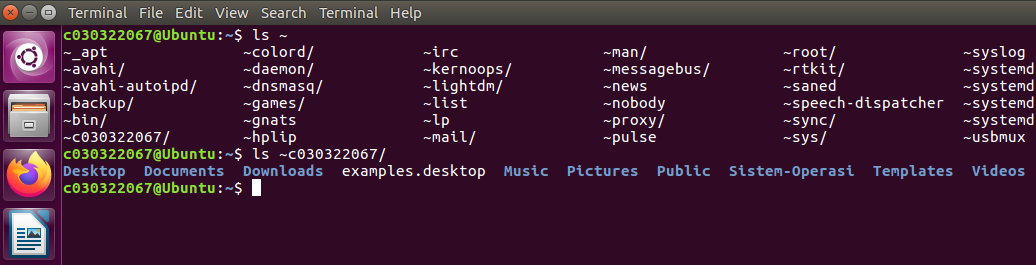
### Percobaan 4 : Lebih Lanjut Tentang Perintah Baris

1. Penggunaan tanda ~ yang merefensi ke home directory



Analisa : Pada perintah ini, ~ digunakan sebagai tanda referensi ke home directory pengguna. Dalam konteks ini, ~ akan digantikan dengan path lengkap ke home directory pengguna yang sedang aktif. Misalnya, jika home directory pengguna adalah /home/user, maka perintah cat ~/.bash\_profile akan menggantinya dengan /home/user/.bash\_profile dan kemudian menjalankan perintah cat pada file .bash\_profile yang ada di home directory tersebut.

1. Penggunaan tanda ~ yang merefensi ke home directory user lain



Analisa : Dengan menggunakan ~userlain sebagai tanda referensi, kita dapat merujuk ke file atau direktori dalam home directory pengguna lain tanpa harus menuliskan path lengkap secara manual. Namun, perlu diperhatikan bahwa akses ke home directory pengguna lain harus diizinkan dan hak akses yang tepat diberikan untuk melihat dan membaca file atau direktori yang dimaksud.

1. Penggunaan parameter/variabel $

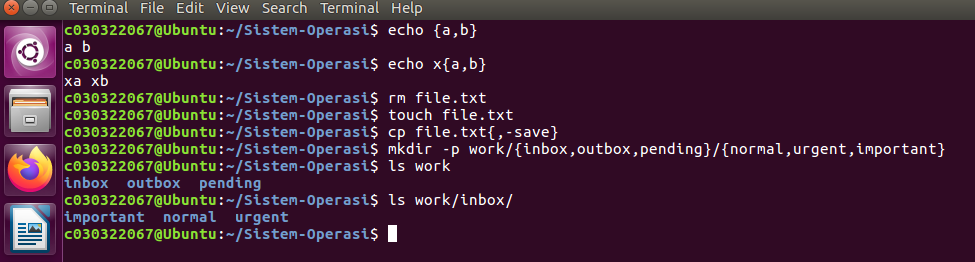
$ cd $HOME/public\_html

$ cd /home/teacher/public\_html



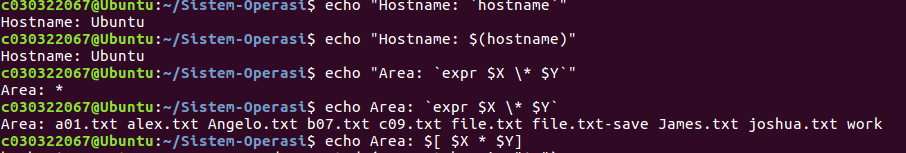
Analisa : Penggunaan parameter/variabel $ memungkinkan pengguna untuk menggantikan nilai atau path yang tepat dengan menggunakan variabel yang telah ditentukan sebelumnya. Ini memudahkan penggunaan perintah dengan referensi ke direktori atau file yang sering digunakan tanpa harus mengetikkan path lengkap secara manual setiap kali.

1. Penggunaan tanda kurung kurawal { }



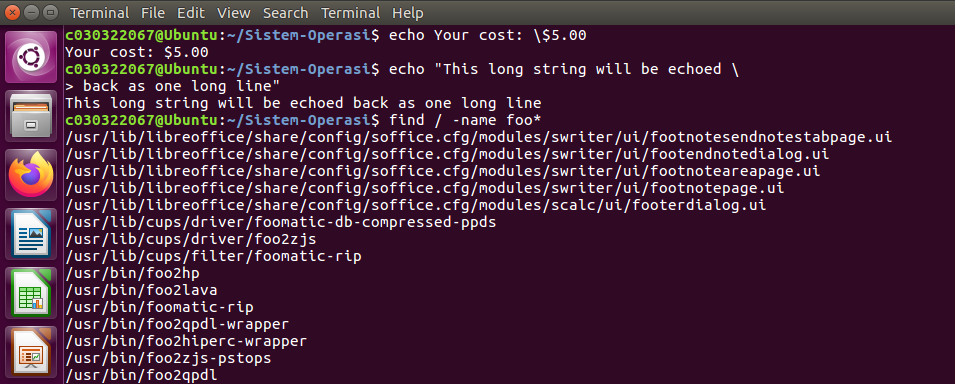
Analisa : Penggunaan tanda kurung kurawal { } memudahkan dalam pembuatan pola atau ekspansi string yang terkait. Ini berguna dalam pembuatan file/direktori dengan pola yang berulang atau ketika ingin melakukan operasi pada sejumlah file/direktori yang serupa.

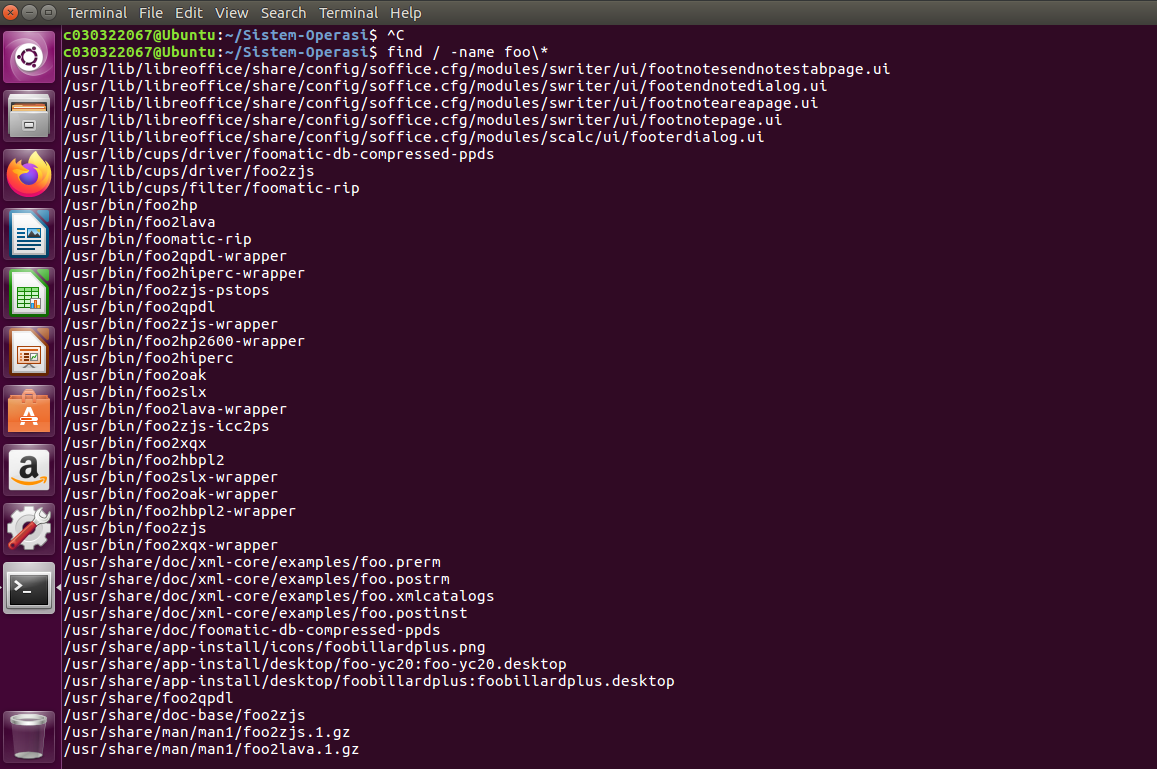
1. Perintah output ` ` atau $( ) dan aritmatika -$[ ]



Analisa : Penggunaan perintah output atau $( ) memungkinkan untuk menjalankan perintah di dalam perintah lain dan menggunakan outputnya sebagai bagian dari teks yang dicetak. Ini berguna ketika ingin memasukkan informasi dinamis, seperti nama host, ke dalam pesan atau tampilan. Penggunaan aritmatika -$[ ] atau $[ ] memungkinkan untuk melakukan perhitungan aritmatika langsung dalam perintah dan menggantikan bagian tersebut dengan hasil perhitungan. Ini mempermudah perhitungan dan penggunaan variabel dalam perintah. Namun, perlu diperhatikan bahwa format $[ ] sudah usang dan sebaiknya digantikan dengan $(()) untuk ekspresi aritmatika pada Bash yang lebih modern.

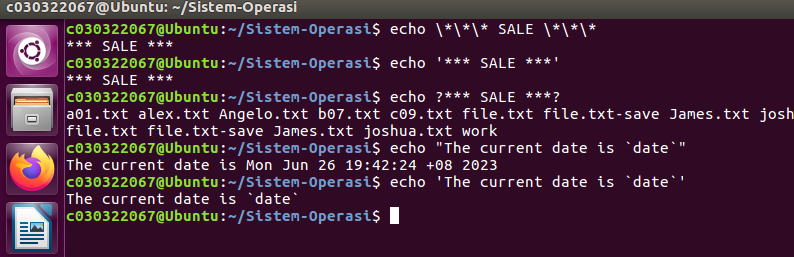
1. Parameter backslash (\)





Analisa : Penggunaan backslash () memberikan fungsi escape atau mengubah makna khusus pada karakter-karakter tertentu. Hal ini berguna saat ingin memasukkan karakter yang memiliki makna khusus dalam konteks tertentu, seperti karakter yang digunakan dalam ekspansi shell atau dalam eksekusi perintah.

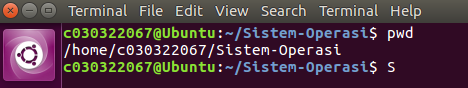
1. Membatasi ekspansi dengan tanda petik



Analisa : Penggunaan tanda petik tunggal (' ') dan tanda petik ganda (" ") dalam bash memungkinkan pembatasan ekspansi, di mana teks di dalamnya dapat dicetak secara harfiah atau dievaluasi sebagai perintah. Pemilihan tanda petik yang tepat tergantung pada kebutuhan dan pengaruh yang diinginkan terhadap ekspansi shell.

### Latihan

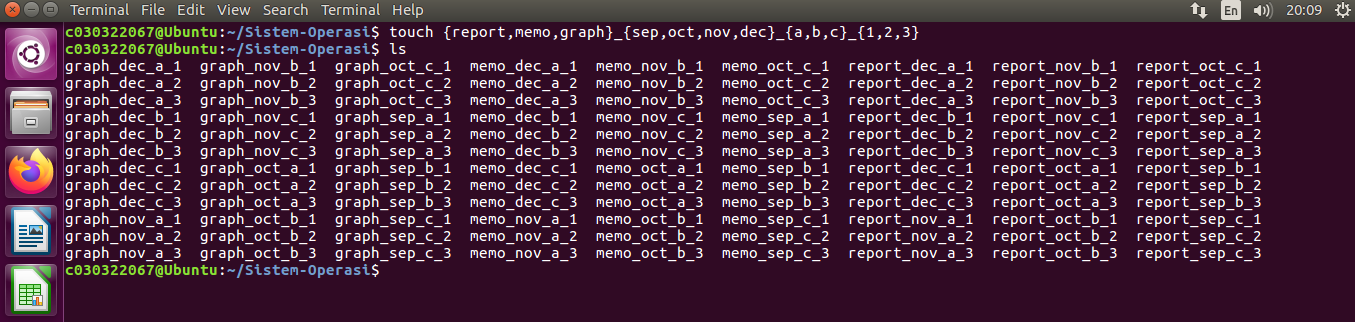
1. Login sebagai student di TTY 1. Lihatlah direktori yang dimasuki dengan perintah pwd.



Analisa : Login biasa dan jalankan perintah pwd.

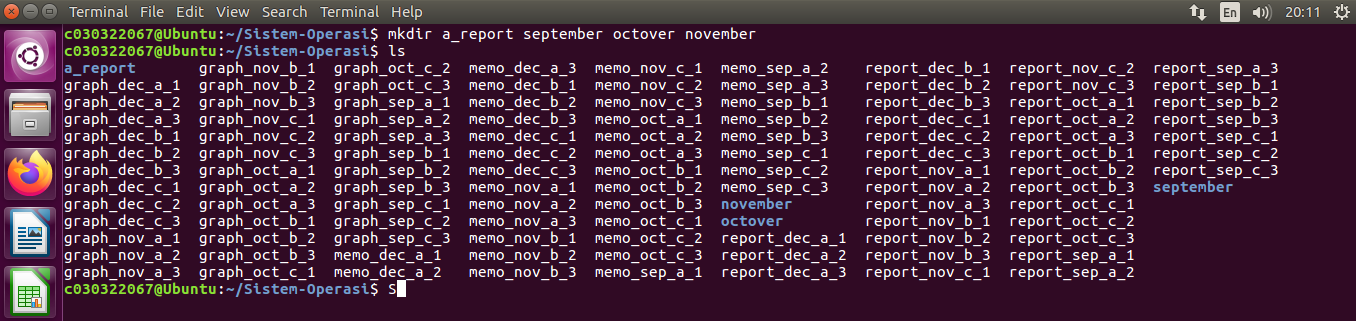
1. Buatlah beberapa file dengan perintah touch. Gunakan tanda kurung { } untuk membuat beberapa file dan lihat hasilnya dengan perintah ls.

$ touch {report,memo,graph}\_{sep,oct,nov,dec}\_{a,b,c}\_{1,2,3}



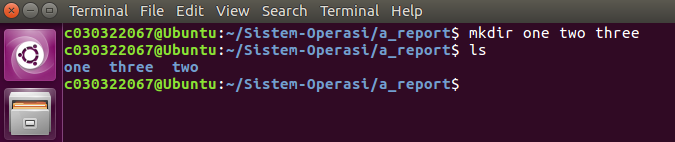
Analisa : akan menghasilkan beberapa file dengan pola nama yang telah ditentukan. Setiap bagian yang diapit oleh kurung kurawal {} akan menghasilkan variasi file yang sesuai.

1. Buatlah beberapa direktori a\_report september october november december dengan perintah mkdir. Gunakan ls untuk melihat hasilnya



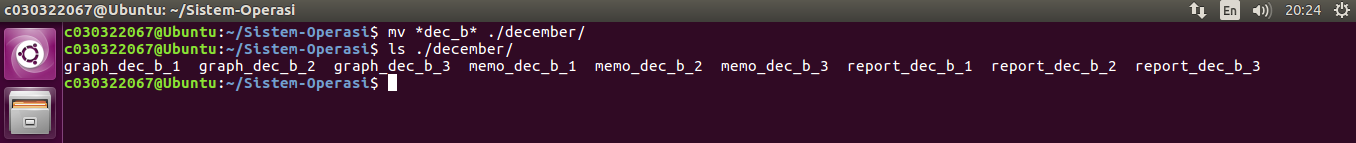
Analisa : Hanya membuat direktori biasa menggunakan command mkdir

1. Buatlah sub direktori di dalam a\_reports yang bernama one two three dan tampilkan hasilnya dengan perintah ls. Pastikan direktori aktif (current directory) berada di sub direktori a\_reports



Analisa : Hanya membuat direktori biasa menggunakan command mkdir didalam direktori a\_report

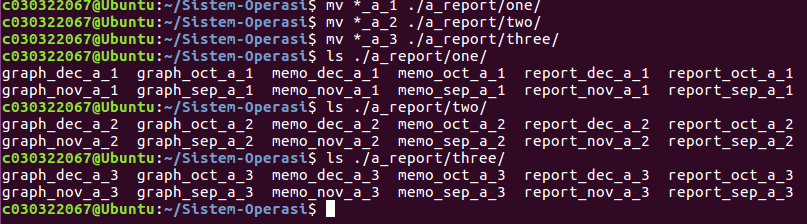
1. Pindahkan semua file laporan “b” pada home directory dan kelompokkan berdasarkan bulan (september, october, november dan december), misalnya graph\_dec\_b\_1 dipindah ke sub direktori december. Tuliskan langkah-langkah perintah yang digunakan. Gunakan perintah baris yang tepat sehingga perintah yang digunakan lebih sederhana (lebih sedikit).





Analisa : menggunakan \* dan mv untuk memindahkan file terkait ke direktori yang ditentukan.

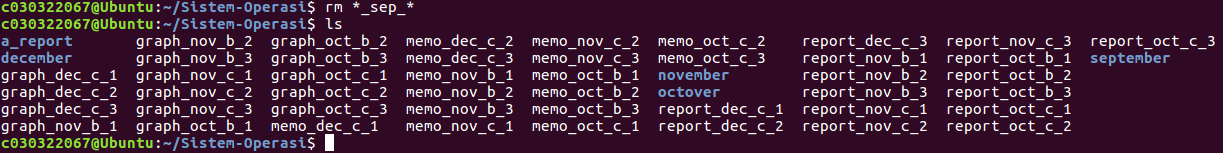
1. Kelompokkan file laporan “a” pada home directory berdasarkan nomor direktori (one, two dan three), misalnya graph\_dec\_a\_1 dipindah ke sub direktori a\_reports/one. Gunakan perintah baris yang tepat.



Analisa : menggunakan \* dan mv untuk memindahkan file terkait ke direktori yang ditentukan.

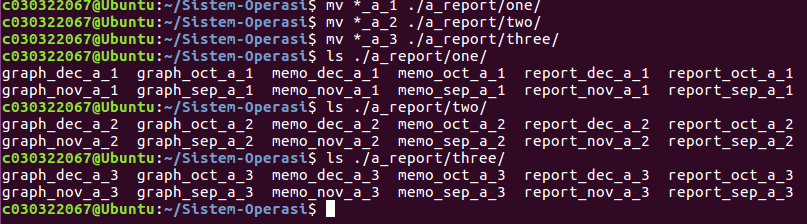
1. Hapus file bulan September (yang terdapat string “sep”). Gunakan perintah baris

yang tepat



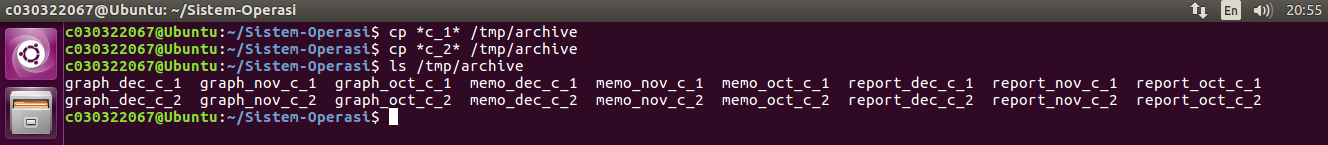
Analisa : menggunakan \* dan rm untuk menghapus file terkait yang telah ditentukan.

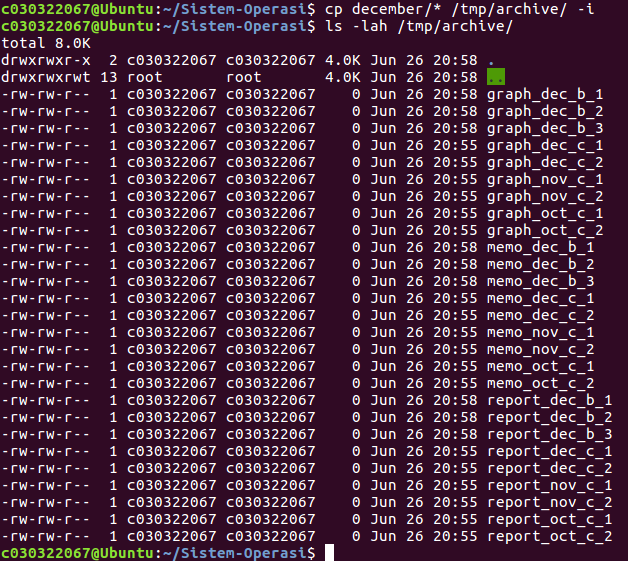
1. Pindahkan semua file laporan yang mengandung string ”a\_2” dan ”a\_3” ke sub direktori a\_reports/two dan a\_reports/three. Lakukan dari direktori yang aktif (current directory).



Analisa : menggunakan \* dan mv untuk memindahkan file terkait ke direktori yang ditentukan.

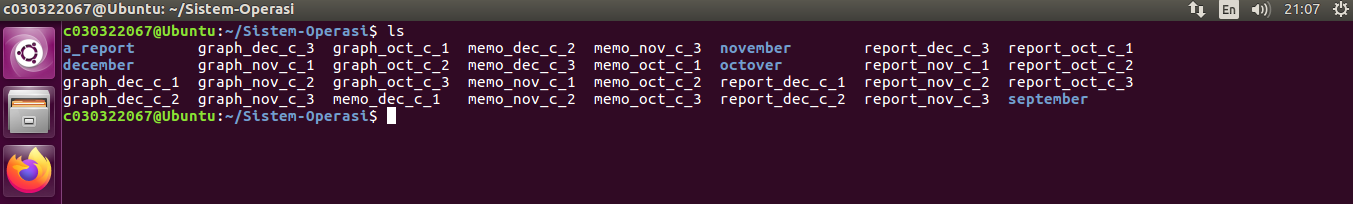
1. Kembali ke home directory dan lihatlah file yang tersisa. Apabila direktori ini berisi file laporan “c” berarti proses Anda sejauh ini sudah benar. Copy-kan laporan yang mengandung string “c\_1” dan “c\_2” ke sub direktori baru bernama archive di dalam sub direktori tmp (direktori standar Linux). Copy-kan juga laporan bulan Desember ke direktori /tmp/archive dengan pilihan –i pada perintah cp.





Analisa : Perintah di atas akan melakukan penyalinan file-file yang memiliki "c\_1" dan "c\_2" pada nama file-nya ke sub direktori "archive" di dalam direktori "tmp". Kemudian, file-file dalam direktori "december" akan dicopy ke direktori "/tmp/archive" dengan opsi -i yang akan meminta konfirmasi sebelum menimpa file yang sudah ada.

1. Tampilkan semua file laporan “c” pada home directory dengan menggunakan wildcard \*c\*. Kemudian hapus semua file laporan “c” tersebut. Pastikan tidak ada file yang dibuat pada latihan berada di home directory (sudah masuk ke sub direktori yang ada)





Analisa : Menghapus menggunakan rm \*c\*.

### Kesimpulan

Kesimpulan dari pengenalan Bash, penggunaan baris perintah dengan cepat, wildcard, kunci Tab, history, dan fitur tilde (~) sebagai referensi dari C shell adalah sebagai berikut:

1. Bash (Bourne Again Shell) adalah shell standar Linux yang menggabungkan fitur-fitur dari shell sebelumnya. Ini menyediakan antarmuka baris perintah yang kuat untuk berinteraksi dengan sistem operasi.
2. Penggunaan baris perintah dengan cepat memungkinkan pengguna untuk lebih efisien dalam berinteraksi dengan sistem. Fitur-fitur seperti wildcard, kunci Tab, dan history membantu mempercepat pengetikan perintah dan pemilihan file.
3. Wildcard adalah karakter khusus yang digunakan untuk memetakan atau mencocokkan beberapa nama file dengan pola tertentu. Contohnya, menggunakan \* untuk mencocokkan satu atau lebih karakter, ? untuk mencocokkan satu karakter, dan [a-z] untuk mencocokkan satu karakter dari rentang a-z.
4. Kunci Tab digunakan untuk melengkapi otomatis nama perintah atau nama file saat mengetik. Ini membantu menghemat waktu dan menghindari kesalahan penulisan.
5. History mencatat perintah-perintah yang telah dieksekusi sebelumnya, memungkinkan pengguna untuk mengulangi atau mengedit perintah yang sudah dimasukkan. Ini mempermudah pemakaian kembali perintah yang kompleks atau panjang.
6. Fitur tilde (~) adalah referensi dari C shell dan digunakan untuk merujuk ke home directory pengguna. Misalnya, ~/.bash\_profile merujuk ke file .bash\_profile di home directory.

Dengan memahami dan menggunakan fitur-fitur ini, pengguna dapat menjadi lebih produktif dan efisien dalam menggunakan shell Bash untuk mengelola sistem dan menjalankan perintah-perintah pada lingkungan baris perintah.