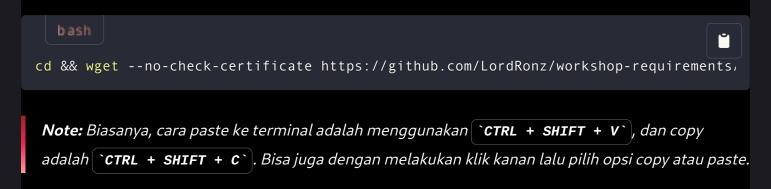
# Modul Workshop Network and Security Lab B201

# **Prerequisite**

Agar lebih mudah untuk mengikuti, diharapkan peserta sudah memiliki sistem maupun VM yang terinstall linux. Linux yang digunakan boleh dari Debian, Ubuntu, Arch, maupun turunannya.

Kemudian, buka terminal atau command prompt dan jalankan perintah berikut:



# A. Pengenalan Linux

Linux/UNIX merupakan salah satu contoh dari open source yang sering digunakan oleh instansi atau masyarakat umum. Pada tahun 1969, Ken Thompson dan Dennis Ritchie (developer bahasa C), para peneliti di AT&T Bell Laboratory Amerika, membuat sistem operasi UNIX, yaitu cikal bakal dari Linux. UNIX mendapatkan perhatian besar karena merupakan sistem operasi pertama yang dibuat bukan oleh hardware maker. Selain itu, karena seluruh source code yang pernah dibuat menggunakan bahasa C, sehingga mempermudah pemindahannya ke berbagai platform. (Priambodo, 2014).

Dalam waktu singkat UNIX berkembang secara pesat dan terpecah dalam dua aliran. yaitu UNIX yang dikembangkan oleh Universitas Berkeley dan yang dikembangkan oleh AT&T. Dari sini lahirlah proyek POSIX (Portable Operating System Interface for UNIX) yang dimotori oleh IEEE (The Institute of Electrical and Electronics Engineers) yang bertujuan untuk menetapkan spesifikasi standar UNIX. (Priambodo, 2014). Tentu saja untuk memberikan perspektif lebih silahkan ke **wikipedia**.

# **B. Linux System Administration**

Linux System administration merupakan perintah-perintah dalam mengoperasikan sistem operasi Linux. Mulai dari instalasi, pemeliharaan, hingga tool-tool untuk mendeteksi gejala-gejala yang mungkin akan menimbulkan masalah. Berikut ini adalah beberapa dari command-command linux yang penting untuk diketahui.

### > man

### reference >

Command `man` merupakan command untuk menampilkan manual page dari system. Setiap argumen page biasanya adalah nama dari program, fungsi (biasanya bahasa C) atau utilitas. Kemudian manual page untuk program tersebut akan ditampilkan. Sebuah section apabila diberikan maka akan membuat man hanya menampilkan bagian section tersebut.

Daftar dibawah merupakan section number dari manual dan diikuti dengan tipe halaman yang ada di dalamnya.

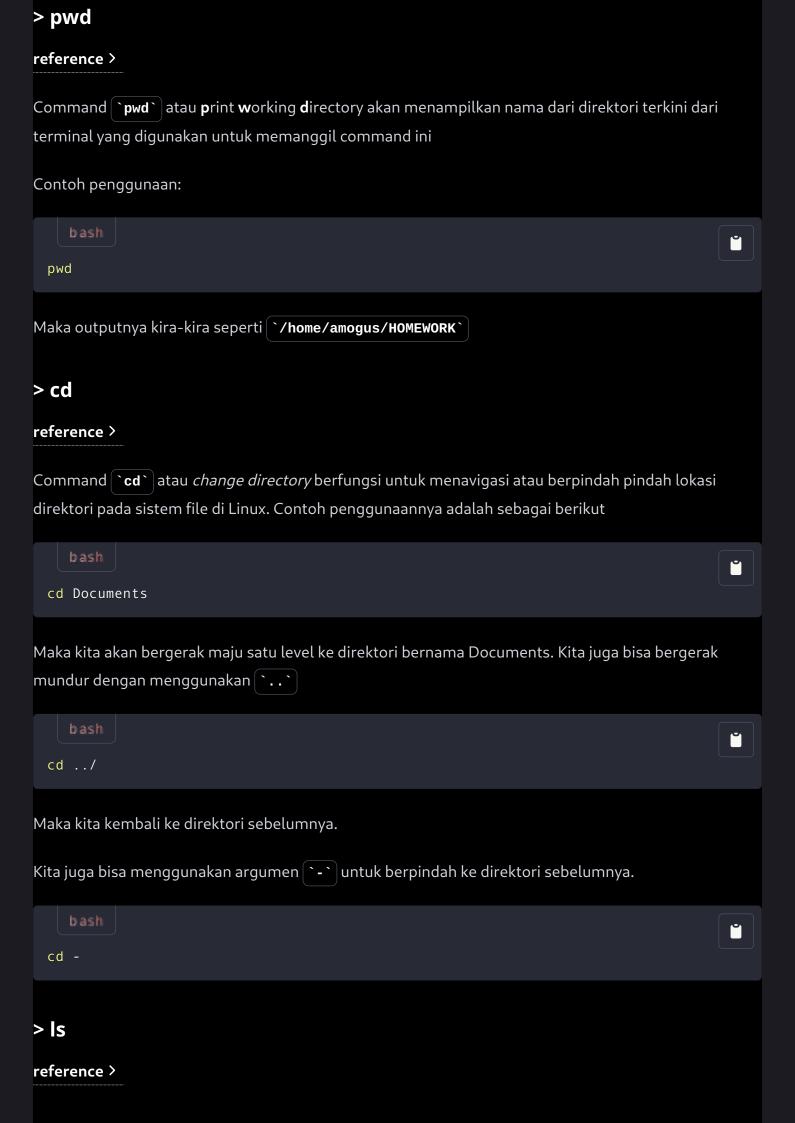
Section Number	Туре		
1	Executable programs or shell commands		
2	System calls (functions provided by the kernel)		
3	Library calls (functions within program libraries)		
4	Special files (usually found in /dev)		
5	File formats and conventions, e.g. /etc/passwd		
6	Games		
7	Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)		
8	System administration commands (usually only for root)		
9	Kernel routines [Non standard]		

Berikut ini adalah contoh penggunaannya.



Command dibawah akan menampilkan manual dari fungsi (`fork`) dalam bahasa C





Command `ls` atau list ini berfungsi untuk menampilkan informasi dari file-file yang berada pada direktori tertentu (secara default adalah direktori terkini). Untuk penggunaan lengkapnya bisa dilihat di reference atau dengan command `man ls`. Command dibawah akan menampilkan file-file yang tidak *hidden* pada `pwd`. bash Ů ls Kira-kira outputnya seperti ini ) ls committlint.config.js next-sitemap.js src node modules tailwind.config.js cypress cypress.json package.json tsconfig.json postcss.config.js LICENSE yarn-error.log yarn.lock next.config.mjs public next-env.d.ts README.md Apabila kita menggunakan argumen seperti (`-1`) maka outputnya akan lebih deskriptif, dan bila menggunakan (`-A`) maka outputnya akan menyertakan file yang berawalan dengan titik (.) alias hidden. Argumen (`-h`) akan mengubah tampilan ukuran file menjadi human readable. Biasanya saya menggunakan kombinasi 3 argumen tersebut: bash ľ ls -lAh Outputnya kira-kira seperti ini

```
> ls -lAh
total 612K
             1 lordronz lordronz
                                  450 Jan 14 20:37 committlint.config.js
-rw-r--r--
             7 lordronz lordronz 4.0K Jan 16 00:00 cypress
drwxr-xr-x
             1 lordronz lordronz
                                    3 Jan 16 00:00 cypress.json
-rw-r--r--
                                  147 Jan 14 20:37 .editorconfig
             1 lordronz lordronz
-rw-r--r--
             1 lordronz lordronz
                                  41 Jan 14 20:37 .env.example
-rw-r--r--
             1 lordronz lordronz
                                   23 Jan 14 20:37 .eslintignore
-rw-r--r--
             1 lordronz lordronz
                                  708 Jan 14 20:37 .eslintrc.json
-rw-r--r--
             8 lordronz lordronz 4.0K Jan 16 11:34 .git
drwxr-xr-x
-rw-r--r--
             1 lordronz lordronz 4.7K Jan 14 20:37 .gitattributes
             3 lordronz lordronz 4.0K Jan 14 20:37 .github
drwxr-xr-x
             1 lordronz lordronz
                                  499 Jan 16 00:00 .gitignore
-rw-r--r--
             3 lordronz lordronz 4.0K Jan 14 20:49 .husky
drwxr-xr-x
             1 lordronz lordronz 1.1K Jan 15 18:44 LICENSE
-rw-r--r--
             1 lordronz lordronz
                                  131 Jan 14 20:37
                                                   .lintstagedrc.json
-rw-r--r--
             5 lordronz lordronz 4.0K Jan 16 11:33
drwxr-xr-x
                                  825 Jan 16 11:32 next.config.mjs
             1 lordronz lordronz
-rw-r--r--
             1 lordronz lordronz
                                  201 Jan 14 20:37 next-env.d.ts
-rw-r--r--
             1 lordronz lordronz
                                  353 Jan 15 18:44 next-sitemap.js
-rw-r--r--
drwxr-xr-x 532 lordronz lordronz
                                  20K Jan 16 11:32 node modules
             1 lordronz lordronz 1.8K Jan 16 11:32 package.json
-rw-r--r--
             1 lordronz lordronz
                                   83 Jan 14 20:37 postcss.config.js
-rw-r--r--
             1 lordronz lordronz
                                  392 Jan 14 20:37 .prettierignore
-rw-r--r--
             1 lordronz lordronz
                                   85 Jan 14 20:37 .prettierrc.json
-rw-r--r--
             4 lordronz lordronz 4.0K Jan 16 10:31 public
drwxr-xr-x
```

# Linux file permission

### reference >

Pada section sebelumnya apabila kita melakukan command `ls -l` maka kita dapat melihat file permission atau file mode yang mana berguna untuk meregulasi level interaksi pada file atau direktori yang dapat dilakukan oleh system. Ini mirip seperti access control pada windows, namun lebih advanced. Inilah mengapa linux sangat secure (dengan catatan penggunanya menggunakan dengan benar, tentu saja).

Pada output dari **`ls**` sebelumnya, terdapat kolom drwxrwxrwx, penjelasannya adalah sebagai berikut.



Tiap-tiap dari permission `rwx` yang diatas dirincikan sebagai berikut:

	Karakter	Efek pada file	Efek pada direktori
Read permission	·-`	File tidak dapat dibuka	Isi direktori tidak dapat dilihat
Read permission	(`r`)	File dapat dibuka	Isi direktori dapat dilihat
Write permission	(,-,)	File tidak dapat dimodifikasi	lsi direktori tidak dapat dimodifikasi
Write permission	`w`	File dapat dimodifikasi	Isi direktori dapat dimodifikasi, seperti membuat file atau folder baru
Execute permission	`-`	File tidak dapat dieksekusi	Direktori tidak dapat diakses dengan command `cd`
Execute permission	`x`	File dapat dieksekusi	Direktori dapat diakses atau dibuka dengan command

### > chmod

### reference >

Command **`chmod`** berfungsi untuk mengubah permission dari sebuah file. Kita dapat menggunakan symbolic maupun numeric mode untuk mengganti permission.

Format dari syntax symbolic modenya adalah sebagai berikut.

```
chmod [OPTIONS] [ugoa...][-+=]perms...[,...] FILE...
```

Flag pertama yaitu `[ugoa]` menyatakan user dengan permission mana yang mesti diubah.

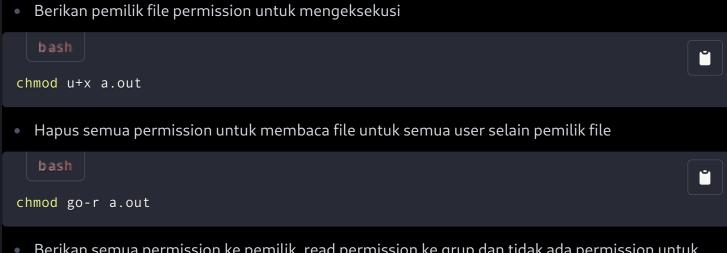
- **`u`** Pemilik file
- (`g`) Semua user yang ada di grup
- **`o`** Semua user lain yang ada di luar grup
- (`a`) Semua user, sama dengan (`ugo`)

Jika flag user ini tidak diberikan maka default adalah `a`.

Flag berikutnya adalah [-+=] menyatakan operasi yang akan menentukan permission apa yang akan dihapus, ditambah, atau diset.

- (`-`) Menghapus permission
- `+` Menambah permission
- (\*=\*) Mengubah permission menjadi permission yang diberikan. Jika tidak ada argumen permission yang diberikan, semua permission dari user yang siberikan pada flag sebelumnya akan dihapus.

Berikut beberapa contoh penggunaannya



 Berikan semua permission ke pemilik, read permission ke grup dan tidak ada permission untuk user lain.

chmod u=rwx,g=r,o= a.out

Untuk numeric mode, kita menggunakan angka yang merepresentasikan masing-masing permission.

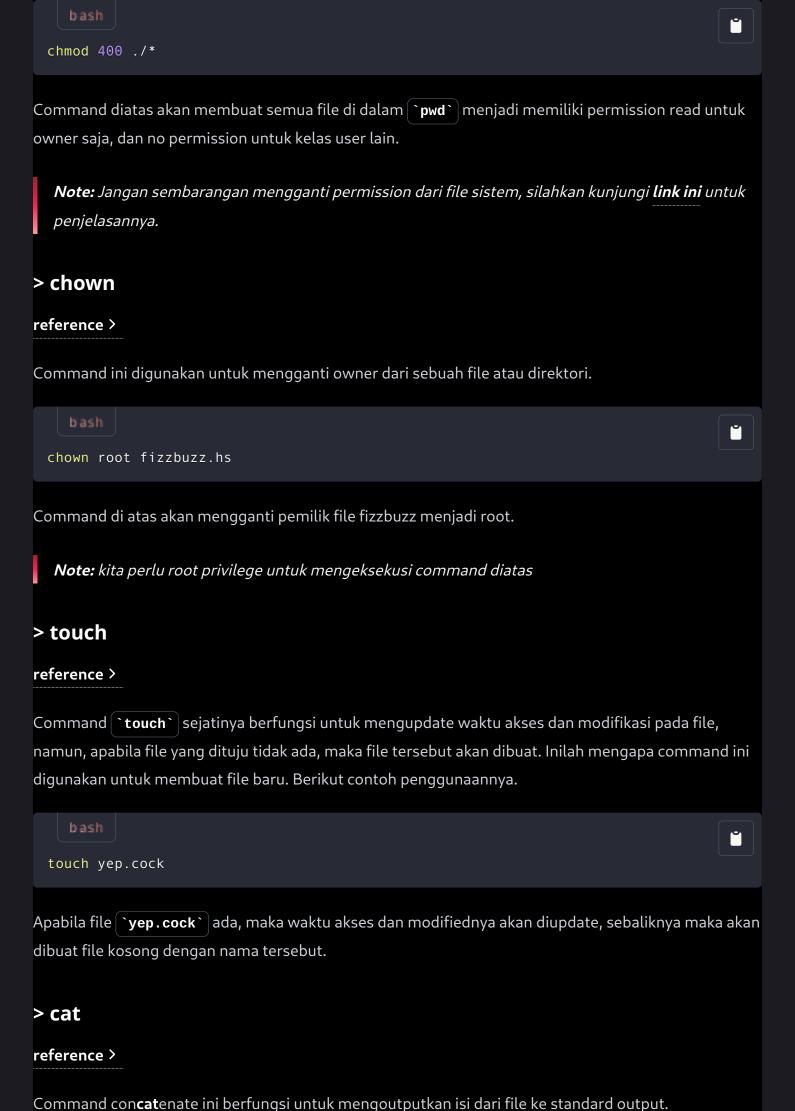
- (`r`)(read) = 4
- `w` (write) = 2
- `r` (execute) = 1
- (no permission) = 0

Angka dari permission yang dimiliki oleh tiap kelas user direpresentasikan dengan jumlah atau *sum* dari angka permission.

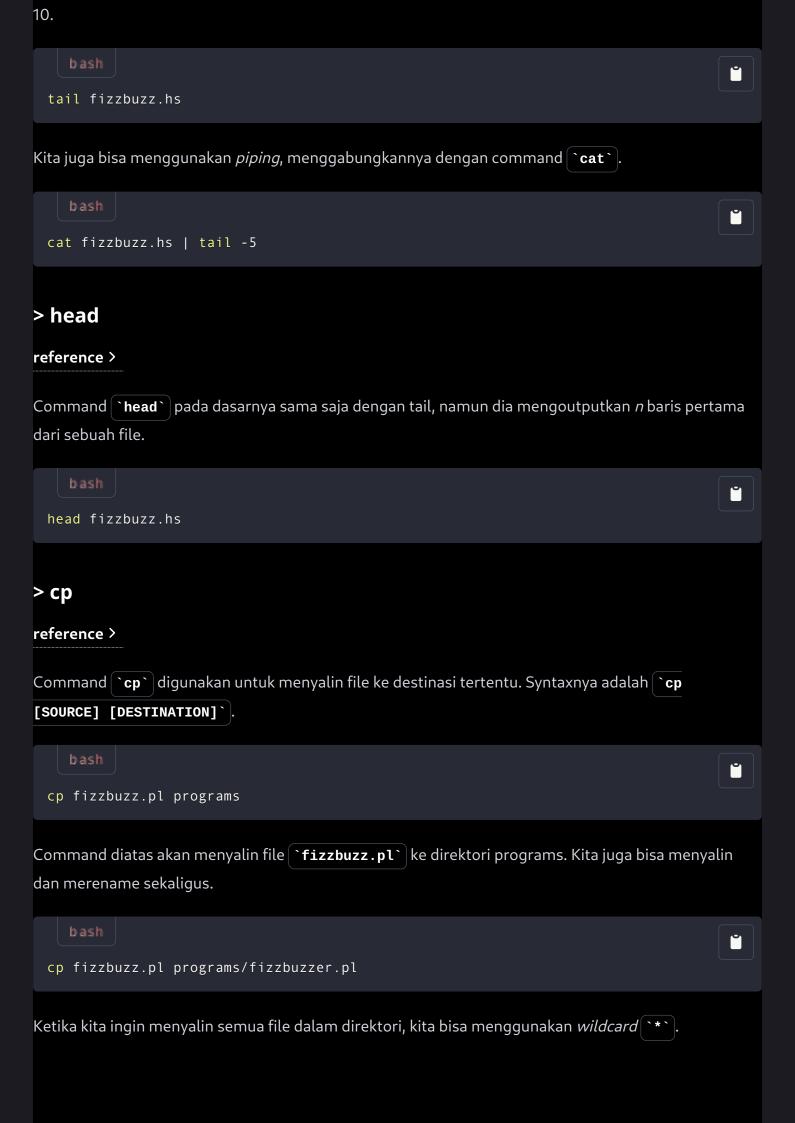
Misalnya command berikut

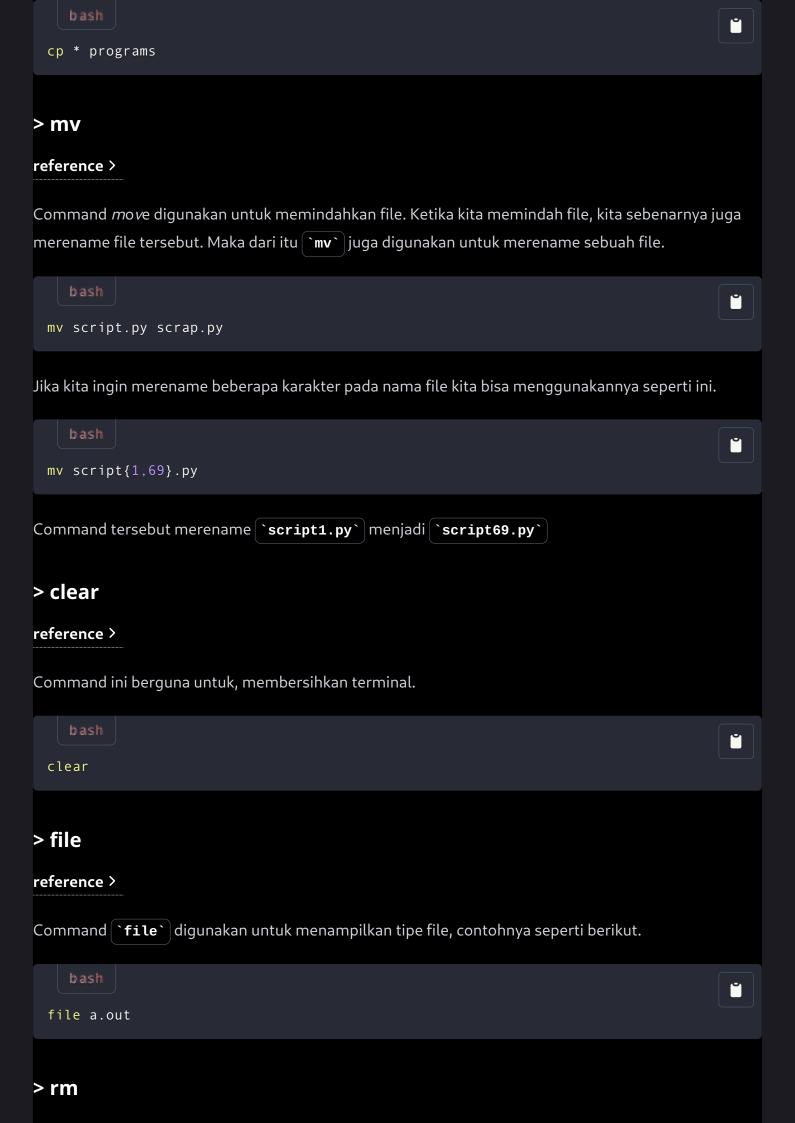
```
bash
chmod 744 a.out
```

Artinya, pemilik file memiliki permission read dan write karena 4 + 2 = 6. Lalu, grup dan user lain memiliki permission read saja.



Apabila argumen nama file tidak diberikan maka `cat` akan membaca dari standard input kemudian langsung mengoutputkannya bash ũ cat Command berikut akan mengoutputkan isi dari \`main.c` bash Ů cat main.c > mkdir reference > Command make directory ini berguna untuk... membuat direktori bash Ů mkdir ./Documents/projects Command diatas akan membuat direktori projects yang ada didalam folder Documents. > nano reference > Nano merupakan text editor default dari kebanyakan distro Linux. Saya menggunakan Arch Linux dan nano merupakan text editor defaultnya. > less reference > Command (`less`) digunakan untuk membaca file. Ada command yang mirip namun lebih primitif, `more` . Less is more than more. > tail reference > Command **`tail`** digunakan untuk mengoutputkan *n* baris terakhir dari sebuah file. Default *n* adalah





# Command `rm` digunakan untuk menghapus file maupun direktori. Command berikut akan menghapus file `main.py`. bash rm main.py Lalu untuk menghapus direktori beserta isinya kita menggunakan argumen `-r`. Berikut kita menghapus folder `poggers` beserta seluruh isinya. bash rm -r poggers

Namun kita perlu berhati-hati, karena command ini akan benar-benar menghapus file, tidak memindahkannya ke Recycle Bin seperti OS sebelah. Malahan kita bisa saja tidak sengaja menghapus file sistem.

**Note:** Perlu kehati-hatian dalam menggunakan command <code>`rm`</code>, karena user tidak akan ditanyakan konfirmasi, kecuali jika kita menggunakan flag <code>`-i`</code>. Flag seperti <code>`-rf`</code> akan menghapus file dan direktori secara rekursif tanpa ampun, sehingga akan destruktif jika targetnya adalah root <code>`/`</code>. Silahkan kunjungi **link ini** untuk informasi lebih lanjut.

### > sudo

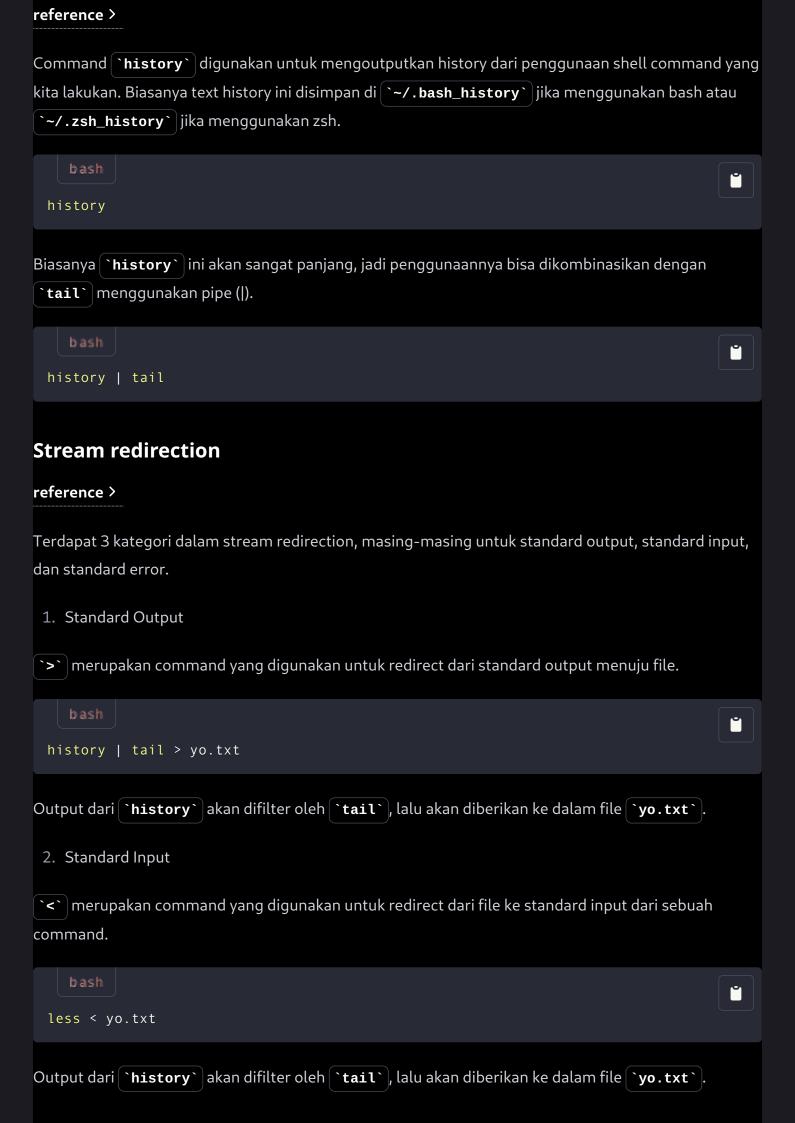
### reference >

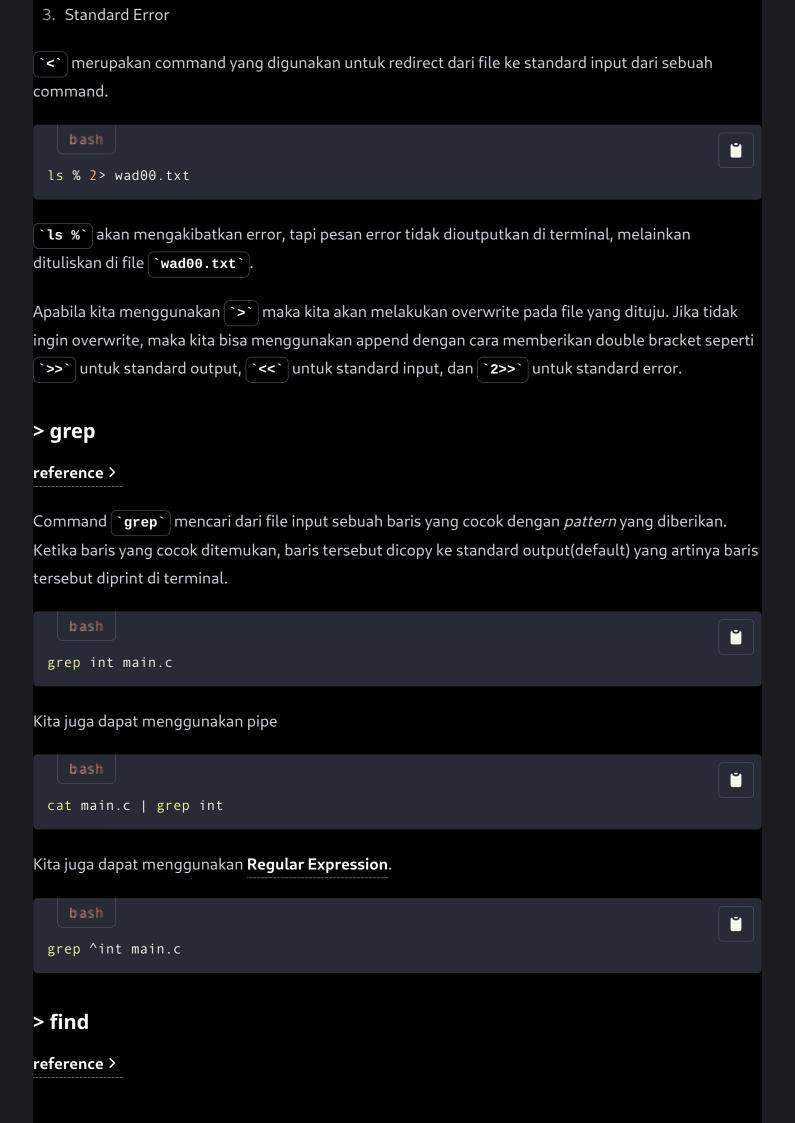
Command **s**uper**u**ser **do** digunakan untuk menjalankan command atau mengakses file yang dimiliki oleh *root user*. Jika kita memang sudah login dengan root user maka tidak perlu menggunakan command ini. Tapi sebaiknya kita jangan menggunakan root user untuk penggunaan sehari-hari karena alasan keamanan.

bash sudo cat /etc/shadow

Apabila kita mengeksekusi command diatas dengan menggunakan user non-root tanpa **sudo**, maka kita akan mendapat error *Permission Denied*.

# > history





Command [`find`] digunakan untuk mencari file yang ada di sistem dengan menggunakan berbagai parameter dan filter. Untuk mencari file dengan nama, kita menggunakan syntax berikut bash Ů find -name "main" Perlu diketahui perintah diatas bersifat case sensitive, sehingga mencari (`main` itu berbeda dengan `Main` |. Untuk mencari file tanpa memedulikan case, kita menggunakan opsi [`-iname` |. bash ľ find -iname "main" Apabila kita mau mencari file yang **tidak** mengikuti pattern tertentu, kita bisa melakukan invert pada pencarian dengan opsi (`-not`). bash ŭ find -not -iname "main" Atau kita bisa juga menggunakan tanda seru (!), dengan catatan diberikan backslash 📉 karena tanda seru merupakan karakter khusus. bash ũ find \! -iname "main" Selain menggunakan nama, kita juga bisa menggunakan tipe file, dengan parameter ( `-type` ) Berikut adalah beberapa descriptor yang dapat kita gunakan untuk menentukan tipe file: `**f**`: regular file `d`: directory `l`: symbolic link `c`: character devices `b` : block devices Kita dapat mencari semua file yang berakhiran | `.py` | dengan command berikut. bash Ŭ find -type f -name "\*.py"

Selain itu masih banyak lagi tipe filter yang dapat kita lakukan, silahkan mengunjungi referensi untuk selengkapnya.

# > which

### reference >

Command which akan mencetak path lengkap dari sebuah executable atau program yang akan dijalankan apabila argumen dari command ini diinputkan di shell prompt. Command ini akan mencari executable yang diberikan sebagai argumen di direktori yang terdapat pada environment variable *PATH. PATH* merupakan env yang memberi tahu shell di direktori mana kita harus mencari sebuah executable.

