

# Shell for Power User

@sopier

## Contents

<b>1</b>	<b>Bash Shell</b>	<b>2</b>
1.1	Beberapa shortcutkey berguna: . . . . .	2
1.2	Letak File / Folder yang sudah di hapus . . . . .	2
<b>2</b>	<b>GNU Screen</b>	<b>2</b>
2.1	Tips dan Trik Navigasi pada GNU screen . . . . .	3
2.2	Berpindah screen Deattach / Reattach . . . . .	3
<b>3</b>	<b>IPython</b>	<b>3</b>
<b>4</b>	<b>VIM</b>	<b>3</b>
4.1	Navigasi Dasar . . . . .	3
4.2	Perintah Dasar . . . . .	4
4.3	Menyimpan Session Buffer . . . . .	4
4.4	Menjalankan perintah shell pada VIM . . . . .	4

# 1 Bash Shell

## 1.1 Beberapa shortcut berguna:

1. Berpindah secara cepat dari satu aplikasi (misal vim atau python interpreter) ke shell, 'Ctrl+z', Untuk kembali, 'fg' atau bisa dengan melihat dulu antrian job dengan perintah 'jobs', lalu 'fg 1' atau 'fg 2' sesuai dengan output dari perintah 'jobs'.

## 1.2 Letak File / Folder yang sudah di hapus

Pernahkah Anda ingin melakukan *undo* terhadap file atau folder yang sudah Anda hapus sebelumnya? Di mana letaknya? Bagi Anda pengguna Ubuntu, letak file tersebut adalah di

```
~/.local/share/Trash/files
```

# 2 GNU Screen

Buka aplikasi terminal favorit Anda, pastikan Anda sudah meng-*install* paket GNU screen dengan cara (Ubuntu):

```
$ sudo apt-get install screen
```

Untuk memudahkan navigasi, buatlah file `.screenrc` pada direktori home Anda.

```
$ touch .screenrc
```

```
$ vim .screenrc
```

Isikan baris konfigurasi berikut:

```
# An alternative hardstatus to display a bar at the bottom
# listing the windownames and highlighting the current windowname
# in blue. (This is only enabled if there is no hardstatus setting
# for your terminal)

hardstatus on
hardstatus alwayslastline
hardstatus string "%{.bW}%-w%{.rW}%n %t%{-}%+w %= %{..G} %H %{..Y} %m/%d %Ca "
```

Fungsi dari script konfigurasi di atas adalah untuk menampilkan semacam *status bar* pada layar monitor Anda. Isi dari *status bar* tersebut adalah daftar window yang sudah Anda buat, disertai dengan namanya, misal `0 bash`, `1 bash` dan seterusnya. Selain itu, konfigurasi di atas akan menampilkan **user** dan penunjuk waktu (hari dan jam).

## 2.1 Tips dan Trik Navigasi pada GNU screen

```
^ a c  => create new terminal window
^ a k  => kill current terminal window
^ a a  => quick move between terminal window
^ S    => split screen horisontally
^ a |  => split screen vertically
^ a tab => move between splitted screen
^ d    => detach a terminal window
^ r    => reattach a detached a terminal window
^ a A  => gives a new window name
```

## 2.2 Berpindah screen Deattach / Reattach

Kasus: Anda sekarang berada pada mode GUI (Ctrl + Alt + F7), dan sedang membuka aplikasi terminal, yang didalamnya Anda sedang melakukan *editing* sebuah file dan melakukan koneksi via *ssh* ke *server*.

Selanjutnya Anda ingin melanjutkan pekerjaan Anda tersebut ke modus CLI (Ctrl + Alt + F1), bagaimana melakukan *porting* terminal Anda tersebut?

Berikut caranya, pastikan Anda berada pada mode CLI (Ctrl + Alt + F1):

```
$ screen -list
$ screen -D
$ screen -R
```

*Voila*, sekarang Anda berhasil melakukan *porting* terminal Anda dan siap melanjutkan pekerjaan Anda. :)

## 3 IPython

## 4 VIM

### 4.1 Navigasi Dasar

```
j, k, h, l  => turun, naik, kiri, kanan
gg          => menuju dokumen paling atas
G          => menuju dokumen paling bawah
^d          => turun satu layar
^u          => naik satu layar
i           => mode insert aktif
I           => mode insert aktif, kursor di awal baris
a           => mode insert aktif, kursor di kanan kursor sekarang
A           => mode insert aktif, kursor di akhir baris
o           => mode insert aktif, kursor di bawah baris sekarang
O           => mode insert aktif, kursor di atas baris sekarang
y           => copy hasil sorot dari visual mode
```

yy	=>	copy baris
p	=>	paste baris dibawah kursor
P	=>	paste baris diatas kursor sekarang
v	=>	aktifkan visual mode (sorot)
V	=>	aktifkan visual model (sorot satu baris)
Esc	=>	keluar dari mode insert

## 4.2 Navigasi Lanjutan

dw	=>	delete one word after
db	=>	delete one word before
d5w	=>	delete 5 word forward
d5b	=>	delete 5 word before

## 4.3 Perintah Dasar

:w	=>	simpan dokumen
:q	=>	keluar vim
:q!	=>	keluar tanpa simpan
:wq	=>	keluar dan simpan
:r	=>	pipng hasil output shell command ke document
:mksession	=>	lihat subbagian Session Buffer

## 4.4 Menyimpan Session Buffer

Suatu saat Anda sedang melakukan *development* aplikasi web menggunakan Python Django. Anda memanfaatkan fitur **buffer** yang ada pada VIM, sehingga Anda cukup butuh satu terminal untuk *editing* semua file-file Anda.

VIM mempunyai fitur *make session* (**mksession**) yang berguna untuk menyimpan sesi pekerjaan yang sedang Anda lakukan. Berikut cara menyimpan sesi pekerjaan Anda dalam vim:

```
:mksession ~/today.ses
```

Dengan fitur ini Anda tidak perlu lagi membuka *buffer* file satu persatu, tapi cukup menggunakan perintah berikut ketika memulai vim:

```
$ vim -S ~/today.ses
```

## 4.5 Menjalankan perintah shell pada VIM

Kasus: Ketika sedang bekerja dengan LaTeX, setiap kali ingin melakukan *compile* dari file **tex** ke dalam PDF, Anda melakukan perintah **pdflatex namafile.text** pada terminal Anda.

Cara konvensional: Anda membuka 2 tab terminal, di mana terminal yang satu khusus untuk melakukan *compiling*.

Cara efisien: Anda tidak perlu membuka tab terminal baru, cukup dari vim Anda dapat menjalankan perintah shell dengan menambahkan tanda ! di depan perintah tersebut.

Contoh untuk melakukan *compiling* file tex

```
:!pdflatex nama_file.tex
```