Praktikum 5 A
Bekerja Dengan Bash Shell

PERCOBAAN

1. Profile

- File .bash_profile dijalankan pada home direktori pemakai yang login. File.bash_profile adalah hidden file, sehingga untuk melihatnya gunakan opsi a pada instruksi ls.

```
3123600005$ ls -a
.              .bash_logout  .config    Downloads   .gnupg  Pictures  .ssh
..             .bashrc       Desktop    .face       .local  .profile  Templates
.bash_history  .cache        Documents  .face.icon  Music   Public    Videos
3123600005$ more .profile
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
```

Analisa:

- Perintah ls -a digunakan untuk menampilkan semua file dan direktori yang ada dalam direktori saat ini, termasuk file yang diawali dengan titik (.) yang biasanya disembunyikan.
- Disini saya menggunakan file .profile karena tidak ada file bash_profile. Perintah more .profile digunakan untuk menampilkan isi dari file .profile secara halaman demi halaman dalam terminal. File .profile adalah skrip shell yang dijalankan pada saat login ke shell di sistem berbasis Unix dan Linux, dan umumnya berisi variabel lingkungan dan pengaturan lain yang dibutuhkan oleh pengguna.

- File .bash_logout akan diekseksi sesaat sebelum logout, berfungsi sebagai house clearing jobs, artinya membersihkan semuanya, misalnya menghapus temporary file atau job lainnya. Melihat file .bash_logout dengan instruksi

```
3123600005$ cat .bash_logout
# ~/.bash_logout: executed by bash(1) when login shell exits.

# when leaving the console clear the screen to increase privacy

if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
3123600005$
```

Analisa:
Perintah cat .bash_logout digunakan untuk menampilkan isi dari file .bash_logout di terminal. File .bash_logout adalah skrip yang dijalankan ketika sebuah sesi login shell Bash berakhir atau ketika pengguna melakukan logout

2. Menggunakan Feature History Bash

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

```
3123600005$ clear
```

Analisa:
Tombol panah atas pada keyboard akan menampilkan perintah yang telah digunakan sebelumnya.

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

```
3123600005$ cd
3123600005$ ls -l /etc
total 1048
-rw-r--r--  1 root root    3040 May 25  2023 adduser.conf
drwxr-xr-x  3 root root    4096 Apr  8 21:25 alsa
drwxr-xr-x  2 root root    4096 Apr  8 21:32 alternatives
-rw-r--r--  1 root root     401 Jan 11  2023 anacrontab
drwxr-xr-x  4 root root    4096 Apr  8 21:29 apache2
-rw-r--r--  1 root root     433 Aug 23  2020 apg.conf
drwxr-xr-x  2 root root    4096 Apr  8 21:19 apparmor
drwxr-xr-x  8 root root    4096 Apr  8 21:31 apparmor.d
-rw-r--r--  1 root root     833 Feb 11  2023 appstream.conf
drwxr-xr-x  9 root root    4096 Apr  8 21:31 apt
drwxr-xr-x  3 root root    4096 Apr  8 21:31 avahi
-rw-r--r--  1 root root    1994 Apr 24  2023 bash.bashrc
-rw-r--r--  1 root root      45 Jan 25  2020 bash_completion
-rw-r--r--  1 root root     367 Sep 23  2022 bindresvport.blacklist
drwxr-xr-x  2 root root    4096 Jan 27 04:48 binfmt.d
drwxr-xr-x  2 root root    4096 Apr  8 21:30 bluetooth
-rw-r--r--  1 root root    7374 Sep 19  2022 bogofilter.cf
drwxr-xr-x  3 root root    4096 Apr  8 21:24 ca-certificates
-rw-r--r--  1 root root    5989 Apr  8 21:30 ca-certificates.conf
drwxr-s---  2 root dip     4096 Apr  8 21:30 chatscripts
drwxr-xr-x  3 root root    4096 Apr  8 21:25 chromium
```

```
3123600005$ ls -l
total 32
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Desktop
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Documents
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Downloads
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Music
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Pictures
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Public
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Templates
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Videos
3123600005$ whoami
ivanrhmt
3123600005$ who
ivanrhmt tty2         2024-04-08 21:41 (tty2)
3123600005$
```

Analisa:

- Perintah cd digunakan untuk mengubah direktori kerja ke direktori home pengguna
- Perintah ls -l /etc digunakan untuk menampilkan isi dari direktori /etc dengan format panjang (long listing format). Direktori /etc berisi file konfigurasi sistem yang penting dan seringkali spesifik untuk sistem tersebut.
- Perintah ls -l digunakan untuk menampilkan isi dari direktori saat ini (atau direktori yang ditentukan) dalam format panjang atau "long listing format".
- Perintah whoami digunakan untuk menampilkan username dari pengguna yang saat ini terautentikasi atau login di terminal.
- Perintah who digunakan untuk menampilkan informasi tentang pengguna yang saat ini log in ke sistem. Perintah ini memberikan gambaran umum tentang pengguna yang aktif pada mesin tersebut, termasuk informasi seperti nama pengguna, terminal yang digunakan, waktu login, dan lainnya.

➢ Untuk memeriksa apakah perintah ini ditambahkan pada history, dapatmenggunakan perintah history untuk melihat semua perintah yang pernah dimasukkan.

```
3123600005$ history
    1  PS1='3123600005$ '
    2  clear
    3  cd
    4  pwd
    5  clear
    6  mkdir Tes
    7  cd
    8  ls
    9  rm Tes
   10  rmdir Tes
   11  ls
   12  clear
   13  PS1='3123600005$ '
   14  clear
   15  ls -a
   16  more .profile
   17  clear
   18  more .bashrc
   19  clear
   20  cat .bash_logout
   21  clear
   22  cd
   23  ls -l /etc
```

Analisa:

Perintah history digunakan untuk menampilkan daftar perintah yang telah dijalankan oleh pengguna di terminal. Setiap perintah yang dieksekusi di terminal dicatat dalam file riwayat (history file), dan perintah history membantu dalam melihat, mengelola, atau mengulang perintah-perintah tersebut.

- 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.

```
3123600005$ !26
who
ivanrhmt tty2         2024-04-08 21:41 (tty2)
3123600005$
```

Analisa:

Perintah !26 adalah contoh dari apa yang disebut "event designator." Event designator ini memungkinkan pengguna untuk mengakses dan menjalankan kembali perintah dari riwayat perintah berdasarkan nomor urutnya. Disini, perintah nomor urut 26 merupakan perintah who, maka terminal akan menampilkan informasi tentang pengguna yang saat ini log in ke sistem.

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

```
3123600005$ !?etc?
ls -l /etc
total 1048
-rw-r--r--  1 root root    3040 May 25  2023 adduser.conf
drwxr-xr-x  3 root root    4096 Apr  8 21:25 alsa
drwxr-xr-x  2 root root    4096 Apr  8 21:32 alternatives
-rw-r--r--  1 root root     401 Jan 11  2023 anacrontab
drwxr-xr-x  4 root root    4096 Apr  8 21:29 apache2
-rw-r--r--  1 root root     433 Aug 23  2020 apg.conf
drwxr-xr-x  2 root root    4096 Apr  8 21:19 apparmor
drwxr-xr-x  8 root root    4096 Apr  8 21:31 apparmor.d
-rw-r--r--  1 root root     833 Feb 11  2023 appstream.conf
drwxr-xr-x  9 root root    4096 Apr  8 21:31 apt
drwxr-xr-x  3 root root    4096 Apr  8 21:31 avahi
-rw-r--r--  1 root root    1994 Apr 24  2023 bash.bashrc
-rw-r--r--  1 root root      45 Jan 25  2020 bash_completion
-rw-r--r--  1 root root     367 Sep 23  2022 bindresvport.blacklist
drwxr-xr-x  2 root root    4096 Jan 27 04:48 binfmt.d
drwxr-xr-x  2 root root    4096 Apr  8 21:30 bluetooth
-rw-r--r--  1 root root    7374 Sep 19  2022 bogofilter.cf
drwxr-xr-x  3 root root    4096 Apr  8 21:24 ca-certificates
-rw-r--r--  1 root root    5989 Apr  8 21:30 ca-certificates.conf
drwxr-s---  2 root dip     4096 Apr  8 21:30 chatscripts
drwxr-xr-x  3 root root    4096 Apr  8 21:25 chromium
```

Analisa:
Perintah !?etc? adalah contoh lain dari "event designator" yang memanfaatkan pencarian berdasarkan konten perintah. Fungsi dari !?etc? adalah untuk menjalankan kembali perintah terakhir yang mengandung kata "etc" di dalamnya. Ini berguna untuk mengulang perintah sebelumnya yang spesifik tanpa perlu mengingat seluruh perintah atau nomornya.

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

```
   31  ls -l /etc
   32  history
3123600005$
```

Analisa:
Di dalam history, yang ditampilkan bukanlah perintah !?etc?, tetapi perintah ls -l /etc

- Apabila string tidka ditemukan pada perintha history maka akan terdapat pesan error.

```
3123600005$ !?wombat99?
bash: !?wombat99?: event not found
3123600005$
```

Analisa:
Akan menampilkan pesan tidak ditemukan karena user tidak pernah melakukan perintah dengan string wombat99.

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

```
3123600005$ !who
who
ivanrhmt tty2         2024-04-08 21:41 (tty2)
3123600005$ !whoa
whoami
ivanrhmt
3123600005$
```

Analisa:
Perintah !who akan menjalankan kembali perintah terakhir yang dimulai dengan kata "who" seperti perintah who dan whoami.

- Anda bisa menggantikant string pada perintah history, terutama pada perintah yang panjang. Misalnya ketik cat /bin/bash | strings | grep shell | less dan tekan Enter. Maka akan menampilkan semua string pada file /bin/bash yang berisi kata ”shell”. Untuk keluar tekan q. Jika ingin menampilkan kata ”alias”, maka Anda tidak perlu mengetik perintah yang panjang lagi, tetapi cukup ketik ^shell^alias^ dan tekan Enter maka akan menggantikan kata ”shell” dengan ”alias”.

```
parse_shellopts
shell_glob_filename
set_shellopts
reset_shell_options
restricted_shell
subshell_envp
expand_words_shellexp
shell_initialized
shell_version_string
find_shell_builtin
shell_eof_token
subshell_level
shell_compatibility_level
set_login_shell
shell_is_restricted
shell_tty
initialize_shell_options
subshell_top_level
find_shell_variable
optimize_shell_function
initialize_shell_builtins
initialize_shell_variables
this_shell_function
static_shell_builtins
subshell_argv
shell_name
debugging_login_shell
execute_shell_function
shellstart
:
```

```
alias_expand_word
legal_alias_name
remove_alias
it_aliases
parser_save_alias
progcomp_alias
find_alias
alias_expand
unalias_doc
delete_all_aliases
parser_expanding_alias
unalias_builtin
initialize_aliases
get_alias_value
alias_expand_all
parser_restore_alias
expand_aliases
add_alias
unalias
unalias [-a] name [name ...]
expand_aliases
progcomp_alias
               includes aliases, builtins, and functions, if and only if
      -P       force a PATH search for each NAME, even if it is an alias,
      -t       output a single word which is one of `alias', `keyword',
               `function', `builtin', `file' or `', if NAME is an alias,
    A useful alias to use with this is r='fc -s', so that typing `r cc'
Remove each NAME from the list of defined aliases.
      -a       remove all alias definitions
:
```

Analisa:

- Perintah cat /bin/bash | strings | grep shell | less merupakan serangkaian pipelined commands yang digunakan untuk menganalisis dan menyaring isi dari file biner /bin/bash, yang merupakan shell Bash. Pertama, cat /bin/bash digunakan untuk menampilkan isi biner dari Bash. Kemudian, strings mengekstrak string yang dapat dibaca dari data biner tersebut. Setelah itu, grep shell menyaring keluaran untuk hanya menunjukkan baris yang mengandung kata "shell". Akhirnya, less memungkinkan pengguna untuk melihat hasilnya secara interaktif dalam bentuk yang dapat di-scroll, memudahkan pengguna untuk menjelajahi dan menganalisis string terkait dengan shell secara rinci. Ini adalah teknik yang berguna untuk debugging atau analisis keamanan, memberikan wawasan tentang komponen dan fungsi yang terkait dengan Bash.
- Perintah ^shell^alias^ digunakan untuk mengganti kata pertama yang terdeteksi yaitu "shell" dengan kata "alias" dalam perintah terakhir yang dijalankan, dan kemudian menjalankannya kembali secara otomatis.

3. Mengubah Feature History Bash

➢ Bash shell akan menyimpan perintah history meskipun telah log out dan log in kembali. File .bash_history menyimpan file history yang terdapat pada home directory.

```
3123600005$ cd
3123600005$
```

Analisa:

Perintah cd digunakan untuk mengubah direktori kerja ke direktori home pengguna

- Lihat beberapa baris pada file .bash_history dengan ketik tail .bash_history dan tekan Enter. File ini bukan file yang up to date.

```
3123600005$ tail .bash_history
cd
pwd
clear
mkdir Tes
cd
ls
rm Tes
rmdir Tes
ls
clear
3123600005$
```

Analisa:
Perintah tail .bash_history digunakan untuk menampilkan bagian akhir dari file .bash_history. File .bash_history menyimpan riwayat perintah yang telah dijalankan oleh pengguna dalam shell Bash. Fungsi dasar dari perintah tail adalah untuk menampilkan beberapa baris terakhir dari file teks; secara default, ia menampilkan 10 baris terakhir.

- Ketik history dan tekan Enter. Maka akan terlihat baris terakhir adalah perintah history dan baris sebelumnya adalah tail .bash_history. Perintah history bersifat up to date, karena disimpan pada memory sistem.

```
   49  tail .bash_history
   50  history
3123600005$
```

Analisa:
Perintah history bersifat up to date, karena disimpan pada memory sistem

- Ketik perintah berikut $ echo ‘Ini perintah saya’

```
3123600005$ echo 'Ini perintah saya'
Ini perintah saya
3123600005$
```

Analisa:
Perintah echo ‘Ini perintah saya’ digunakan untuk menampilkan teks ‘Ini perintah saya’ ke terminal.

- Log out dan log in kembali sebagai user yang sama. Ketik history dan tekan Enter. Maka perintah echo ’Ini perintah saya’ akan berada pada baris terakhir. Lihat file .bash_history, maka perintah tsb akan terdapat pada file .bash_history.

```
   52  echo 'Ini perintah saya'
   53  history
ivanrhmt@ivanrhmt:~$ tail .bash_history
clear
tail .bash_history
PS1='3123600005$ '
clear
history
clear
tail .bash_history
history
clear
echo 'Ini perintah saya'
ivanrhmt@ivanrhmt:~$
```

Analisa:

- Perintah history menampilkan perintah echo ‘Ini perintah saya’ di baris sebelum terakhir di atas perintah history karena perintah history bersifat up to date.
- Perintah tail .bash_history menampilkan perintah echo ‘Ini perintah saya’ di baris terakhir karena .bash_history mencatat perintah-perintah setelah mereka dijalankan dan biasanya disimpan saat shell ditutup atau setelah banyak perintah dijalankan.

➢ Ketik history|less untuk melihat perintah history terakhir pada screen. Tekan spacebar untuk melihat file lebih banyak. Untuk keluar tekan q

```
   47  history
   48  clear
   49  tail .bash_history
   50  history
   51  clear
   52  echo 'Ini perintah saya'
   53  history
   54  tail .bash_history
   55  PS1='3123600005$ '
   56  clear
   57  history|less
(END)
```

Analisa:

Perintah history|less memungkinkan pengguna untuk melihat dan menavigasi riwayat perintah yang telah dijalankan dengan lebih efisien dan interaktif. Perintah history menghasilkan daftar perintah yang telah digunakan, lengkap dengan nomor urut, dan dengan menggunakan less, output dari history ditampilkan satu halaman pada satu waktu. Ini memudahkan pengguna untuk menggulir ke atas atau ke bawah melalui daftar perintah tersebut, mencari perintah tertentu, atau hanya mengkaji kembali perintah yang telah dijalankan.

➢ Untuk melihat berapa banyak perintah history yang ada pada file ketik berikut dan output yang keluar serupa di bawah ini

```
3123600005$ wc -l .bash_history
52 .bash_history
3123600005$
```

Analisa:
Perintah wc –l .bash_history digunakan untuk menghitung jumlah baris dalam file .bash_history. File .bash_history menyimpan riwayat perintah yang telah dijalankan oleh pengguna dalam shell Bash. Disini jumlah baris dalam file .bash_history saya adalah 52.

- Output menunjukkan bahwa 1000 perintah history disimpan pada file history. Untuk melihat jangkauan (limit) perintah history digunakan variabel HISTSIZE. Untuk melihat jangkauan history ketik sebagai berikut

```
3123600005$ set|grep HISTSIZE
HISTSIZE=1000
3123600005$
```

Analisa:
Perintah set|grep HISTSIZE digunakan untuk mencari dan menampilkan nilai dari variabel lingkungan HISTSIZE. Variabel HISTSIZE menentukan jumlah maksimum perintah yang Bash simpan dalam riwayatnya selama sebuah sesi. Dengan menjalankan set | grep HISTSIZE, perintah ini menampilkan entri yang berhubungan dengan HISTSIZE dari semua variabel yang diset oleh shell, memberikan informasi tentang batasan penyimpanan riwayat saat ini.

- Bila ingin memperbesar jangkauan file history, maka ubahlah variabel HISTSIZE pada skrip startup yang disebut .bashrc pada home directory

```
3123600005$ echo 'HISTSIZE=5000' >> .bashrc
3123600005$
```

Analisa:
Perintah echo ‘HISTSIZE=5000’ >> .bashrc digunakan untuk menambahkan pengaturan ke file konfigurasi .bashrc, yang secara otomatis menetapkan variabel lingkungan HISTSIZE ke 5000 setiap kali sebuah terminal Bash diinisialisasi. Penggunaan operator >> memastikan bahwa baris ini ditambahkan ke bagian bawah file .bashrc tanpa menghapus isi yang ada sebelumnya, sehingga mengubah kapasitas riwayat Bash agar dapat menyimpan hingga 5000 perintah.

- Log out dan log in kembali sebagai user yang sama. Lihat perubahan variabel HISTSIZE.

```
3123600005$ set|grep HISTSIZE
HISTSIZE=5000
3123600005$
```

Analisa:
Perintah set|grep HISTSIZE digunakan untuk menampilkan nilai dari variabel HISTSIZE yang awalnya 1000 menjadi 5000.

- Ketikkan perintah history beberapa kali, maka perintah ini akan disimpan pada BASH history meskipun yang diketikkan perintahnya sama.

```
   65  clear
   66  set|grep HISTSIZE
   67  history
3123600005$
```

Analisa:
Disini Ketika saya menjalankan perintah history beberapa kali, perintahnya tidak tersimpan pada BASH history. Mungkin karena variabel HISTCONTROL sudah bernilai 'ignoredups' dari awal Ketika linux di install.

- Anda dapat melakukan konfigurasi BASH agar tidak menambah perintah ke history jika perintah yang diketikkan sama dengan sebelumnya. Hal ini dilakukan dengan menambahkan variabel HISTCONTROL dan diberikan nilai ignoredups pada file .bashrc

```
3123600005$ echo 'HISTCONTROL=ignoredups' >> .bashrc
3123600005$
```

Analisa:
Perintah echo 'HISTCONTROL=ignoredups' >> .bashrc digunakan untuk menghindari penyimpanan perintah yang sama berurutan.

4. Mengubah Prompt Shell

- Prompt Bash shell dikonfigurasi dengan men-setting nilai variabel PS1. Selain menampilkan string statik sebagai prompt, Anda dapat menampilkan menjadi dinamis. Contohnya, apabila ingin menunjukkan current directory atau current time. Ketik PS1='\t:' dan tekan Enter untuk menampilkan waktu sistem dalam format 24 jam sebagai prompt Bash. Format dalam HH:MM:SS

```
3123600005$ PS1='\t:'
12:41:41:
```

Analisa:
Perintah PS1='\t:' digunakan untuk mengkonfigurasi tampilan prompt pada terminal. Variabel PS1 memungkinkan pengguna menyesuaikan bagaimana prompt ditampilkan, dan dalam kasus ini, perintah tersebut mengatur prompt untuk secara eksklusif menampilkan waktu saat ini dengan format 24 jam (HH:MM:SS), diikuti oleh tanda titik dua (:).

- Untuk menampilkan format 12 jam dengan indikator am dan pm ketik sebagai berikut :

```
12:41:41:PS1='\@:'
12:46 PM:
```

Analisa:
Perintah PS1='\@:' digunakan untuk mengatur prompt terminal agar menampilkan waktu saat ini dalam format 12 jam, lengkap dengan indikator AM atau PM, diikuti oleh tanda titik dua (:).

- Kebanyakan orang menginginkan prompt Bash menampilkan current working directory. Direktory dapat ditampilkan dalam bentuk keseluruhan path atau hanya nama direktory. Karakter \w menampilkan hanya nama direktory. Jika current directory adalah home directory, maka tampil prompt ~:

```
12:49 PM:PS1='\w: '
~:
```

Analisa:
Perintah PS1='\w:' digunakan untuk mengatur prompt terminal agar menampilkan direktori kerja saat ini, diikuti oleh tanda titik dua (:).

- Ketik cd /usr/sbin untuk melihat prompt /usr/sbin:

```
~: cd /usr/sbin
/usr/sbin:
```

Analisa:
Ketika melakukan perintah cd /usr/sbin, promt terminal akan menampilkan /usr/sbin karena /usr/sbin merupakan direktori kerja saat ini.

- Ketik PS1='\W:' untuk melihat prompt sbin:

```
/usr/sbin: PS1='\W: '
sbin:
```

Analisa:
Perintah PS1='\W:' digunakan untuk mengatur prompt terminal agar menampilkan hanya nama direktori kerja saat ini, bukan path lengkapnya, diikuti oleh tanda titik dua (:).

- Ada beberapa prompt BASH lain yang dapat diubah, yaitu PS2, PS3 dan PS4. Prompt PS2 digunakan sebagai prompt sekunder. Untuk melihat bagaimana penggunaannya, ketik echo 'Hello (tanpa diakhiri penutup quote) dan tekan Enter. Simbol lebih besar dari (>) akan muncul. Hal ini memberitahukan bahwa BASH menunggu Anda menyelesaikan perintah. Ketik penutup quote (') dan tekan Enter. Perintah ini akan menyelesaikan prompt PS2, kata "Hello, " muncul diikuti dengan prompt PS1 pada baris baru

```
sbin: echo 'Hello
> '
Hello

sbin:
```

Analisa:
Saat mengetik echo 'Hello dan menekan Enter, Bash belum menganggap perintah tersebut sebagai perintah yang lengkap atau sintaksisnya benar. Sebagai respons, Bash tidak akan langsung menjalankan perintah, melainkan akan menampilkan

simbol > di baris berikutnya. Simbol > merupakan promt sekunder yang menunjukkan bahwa Bash sedang menunggu user untuk melengkapi perintah yang telah dimulai.

- Anda dapat mengubah prompt PS2 seperti mengubah prompt PS1. Ketik perintah berikut :

```
3123600005$ PS2='Selesai memasukkan perintah Anda: '
3123600005$
```

Analisa:
Perintah PS2=’Selesai memasukkan perintah Anda: ’ mengubah prompt sekunder dari prompt defaultnya (>) menjadi teks “Selesai memasukkan perintah Anda: ”.

- Kemudian ketik echo ’Hello (tanpa diakhiri penutup quote) dan tekan Enter. Pada baris berikutnya akan muncul Selesai memasukkan perintah Anda:. Kemudian ketikkan penutup quote (’) dan tekan Enter. Jika perintah selesai, maka kata Hello akan muncul diikuti prompt PS1 pada baris baru.

```
3123600005$ echo 'Hello
Selesai memasukkan perintah Anda: '
Hello

3123600005$
```

Analisa:
Promt sekunder yang awalnya simbol > berubah menjadi teks “Selesai memasukkan perintah Anda: ”.

- Prompt BASH dapat ditampilkan berwar na dengan melakukan setting colorsetting string . Sebagai contoh, prompt BASH di-set dengan \w\$, akan menampilkan current working directory yang diikuti $ (atau # jika anda login sebagai root). Untuk setting warna menjadi biru ketikkan berikut :

```
3123600005$ PS1='\033[0;34m\w\$ \033[0;37m'
/usr/sbin$
```

Analisa:
Perintah PS1=’\033[0;34m\w\$ \033[0;37m’ digunakan untuk mengatur prompt terminal dengan menambahkan pewarnaan untuk meningkatkan kejelasan dan estetika. Kode \033[0;34m mengatur warna teks menjadi biru, yang akan diterapkan pada penampilan direktori kerja saat ini, yang diindikasikan oleh \w. Simbol \w menggantikan direktori kerja saat ini dalam prompt. Tanda dolar ($) yang mengikuti \w menunjukkan akhir dari direktori dan sering digunakan sebagai pemisah sebelum pengguna mulai mengetik perintah. Kode \033[0;37m setelah $ mengatur warna kembali ke abu-abu terang untuk teks yang akan diketik pengguna setelah prompt

- Untuk mendapatkan prompt warna merah ketikkan berikut :

```
/usr/sbin$ PS1='\033[0;31m\w\$ \033[0;37m'
/usr/sbin$
```

Analisa:
Sama seperti perintah di atas, bedanya adalah kode \033[0;31m mengatur warna teks menjadi merah. 30=hitam, 31=merah, 32=hijau, 34=biru, 35=ungu, 36=cyan, 37=putih.

- Bila menginginkan beberapa warna, ketikkan perintah berikut

```
/usr/sbin$ PS1='\033[0;31m\w\033[0;32m\$ \033[0;37m'
/usr/sbin$
```

Analisa:
Kode \033[0;32m\$ mengubah simbol $ menjadi warna hijau.

- Anda bisa menampilkan atribut visual seperti lebih terang, berkedip dan warna kebalikannya. Untuk menampilkan prompt yang lebih terang, atribut control diganti 1, seperti perintah berikut :

```
          PS1='\033[1;34m\w\033[1;32m\$ \033[0;37m'
/usr/sbin$
```

Analisa:
Atribut control 1 digunakan untuk membuat teks terang dan tebal.

- Untuk menampilkan prompt dengan warna berkebalikan, atribut control diganti 7, seperti perintah berikut :

```
/usr/sbin$ PS1='\033[7;34m\w\033[7;32m\$ \033[0;37m'
/usr/sbin$
```

Analisa:
Atribut control 7 digunakan untuk mengubah warna latar promt.

- Untuk menampilkan prompt berkedip, atribut control diganti 5, seperti perintah berikut :

```
/usr/sbin$ PS1='\033[5;34m\w\033[5;32m\$ \033[0;37m'
/usr/sbin$
```

Analisa:
Atribur control 5 digunakan untuk mengubah promt menjadi berkedip.

5. Menambahkan otomatisasi ke Prompt Shell

- Pastikan Anda berada di home directory

```
3123600005$ cd ~
3123600005$
```

Analisa:

Perintah cd ~ digunakan untuk mengubah direktori kerja saat ini ke direktori home pengguna.

- Buatlah skrip sederhana untuk mengurut daftar file. Anda dapat menggunakan teks editor, tetapi karena hanya satu baris, gunakan perintah echo untuk membuat file.

```
3123600005$ echo 'sort ~/list > ~/r13; mv ~/r13 ~/list' > ~/sorter
3123600005$
```

Analisa:
Perintah echo 'sort ~/list > ~/r13; mv ~/r13 ~/list' > ~/sorter digunakan untuk membuat file skrip sederhana yang bernama sorter di direktori home pengguna. Isi dari skrip ini merupakan dua perintah yang dijalankan secara berurutan: pertama, perintah sort ~/list > ~/r13 yang mengurutkan isi dari file list yang berada di direktori home pengguna, kemudian menyimpan hasilnya dalam file sementara r13. Kedua, perintah mv ~/r13 ~/list yang memindahkan isi dari file r13 kembali ke file list, menggantikan isi aslinya dengan versi yang telah diurutkan.

- Buatlah file skrip diatas menjadi file executable

```
3123600005$ chmod +x sorter
3123600005$
```

Analisa:
Perintah chmod +x sorter digunakan untuk memberikan izin eksekusi pada file yang bernama sorter.

- Jalankan program sorter diatas setiap shell Bash menampilkan prompt PS1. Untuk melakukannya, buatlah variable PROMPT_COMMAND Dimana nilainya adalah nama dari program sorter.

```
3123600005$ PROMPT_COMMAND=~/sorter
3123600005$
```

Analisa:
Perintah PROMPT_COMMAND=~/sorter digunakan untuk memberi instruksi kepada shell untuk menjalankan skrip sorter yang berada di direktori home pengguna setiap kali prompt siap untuk menerima perintah baru.

- Ketikkan echo 'John Smith:13001'>>list dan tekan Enter. Jika file list tidak ada, akan dibuat secara otomatis, tetapi jika sudah ada, string 'John Smith:13001' akah ditambahkan

```
3123600005$ echo 'John Smith:13001'>>list
3123600005$
```

Analisa:
Perintah echo 'John Smith:13001'>>list digunakan untuk menambahkan teks John Smith:13001 ke akhir file yang bernama list.

- Ketik cat list dan tekan Enter. Maka Anda akan melihat isi file list. Pada saat ini, file mungkin mempunyai hanya satu baris sehingga tidak dapat dilihat apakah file sudah terurut.

```
3123600005$ cat list
John Smith:13001
3123600005$
```

Analisa:
Perintah cat list digunakan untuk menampilkan isi dari file yang bernama list langsung ke terminal atau standar output.

- Masukkan bebe rapa perintah serupa dengan point 5 tetapi dengan nama dan nomor yang berbeda. Kemudian ketik cat list dan tekan Enter.

```
3123600005$ echo 'Anita:13002'>>list
3123600005$ echo 'Ivan:13003'>>list
3123600005$ echo 'Lisa:13004'>>list
3123600005$ echo 'Marchel:13005'>>list
3123600005$ cat list
Anita:13002
Ivan:13003
John Smith:13001
Lisa:13004
Marchel:13005
3123600005$
```

Analisa:
Perintah cat list akan menampilkan isi dari list berdasarkan huruf abjad.

- Apabila Anda tidak menginginkan Shell Bash menampilkan file terurut sepanjang waktu, Anda tidak perlu menambahkan variable PROMPT_COMMAND=~/sorter pada file konfigurasi seperti .bashrc. Bila Anda ingin BASH berhenti menjalankan program sorter, maka ketikkan variable PROMPT_COMMAND= dan tekan Enter atau log out dan login Kembali

```
3123600005$ PROMPT_COMMAND=
3123600005$
```

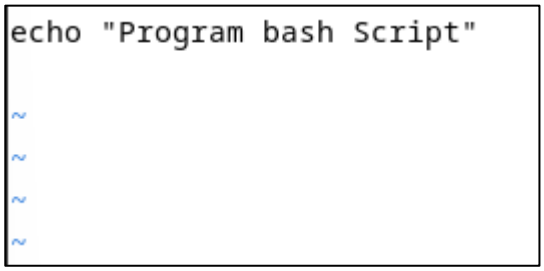
Analisa:
Dengan menjalankan perintah PROMPT_COMMAND=, maka tidak akan ada perintah tambahan yang dijalankan sebelum prompt ditampilkan, yang mengembalikan perilaku prompt ke kondisi default tanpa modifikasi atau eksekusi tambahan sebelum prompt muncul.

Praktikum 5 B
Bekerja Dengan Bash Shell

PERCOBAAN

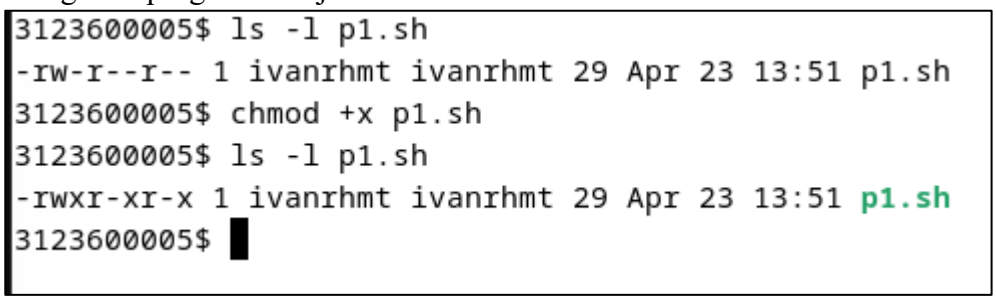
6. Membuat Bash-script dan menjalankannya

   - Membuat file p1.sh



Analisa:
Perintah vi p1.sh digunakan untuk membuka atau mengedit file skrip shell yang bernama p1.sh menggunakan editor teks vi. Editor vi adalah salah satu editor teks yang paling umum digunakan di lingkungan Unix dan Linux, terkenal dengan kemampuannya yang luas dan efisiensi saat mengedit teks.

   - Mengubah program menjadi executable



Analisa:
- Perintah ls -l p1.sh digunakan untuk menampilkan informasi rinci tentang file yang bernama p1.sh dalam format panjang (long listing format). Perintah ini memberikan detail lebih lengkap dibandingkan dengan perintah ls biasa.
- Perintah chmod +x p1.sh digunakan untuk menambahkan izin eksekusi (execute permission) pada file yang bernama p1.sh.

- Menjalankan script

```
3123600005$ bash p1.sh
Program bash Script
3123600005$ sh p1.sh
Program bash Script
3123600005$ . p1.sh
Program bash Script
3123600005$ ./p1.sh
Program bash Script
3123600005$
```

Analisa:
Perintah di atas merupakan beberapa cara untuk menjalankan script file p1.sh.

- Konvensi dalam pembuatan script shell dinyatakan sebagai #!/bin/bash. Tambahkan pada file p1.sh konvensi tersebut

```
#!/bin/bash
echo "Program bash Script"
```

Analisa:
Perintah #!/bin/bash disebut sebagai "shebang" atau "hashbang". Baris ini sangat penting karena memberitahu sistem operasi interpreter mana yang harus digunakan untuk menjalankan skrip tersebut. Ini memastikan skrip dijalankan dengan shell atau interpreter yang benar, terlepas dari shell atau interpreter yang saat ini digunakan oleh pengguna.

- Buatlah file p2.sh

```
#!/bin/bash
echo "Program 2 bash script"
```

Analisa:
Perintah di atas merupakan perintah untuk membuat file p2.sh yang berisi mirip seperti fil p1.sh.

- Menjalankan beberapa program shell dalam satu baris instruksi yang dipisahkan dengan tanda ;

```
3123600005$ cat p1.sh ; cat p2.sh
#!/bin/bash
echo "Program bash Script"
#!/bin/bash
echo "Program 2 bash script"
3123600005$ ./p1.sh ; ./p2.sh
Program bash Script
bash: ./p2.sh: Permission denied
3123600005$
```

Analisa:

- Perintah cat p1.sh ; cat p2.sh digunakan untuk menampilkan isi dari dua file skrip, p1.sh dan p2.sh, secara berturut-turut ke output standar, biasanya layar terminal. Perintah cat sendiri adalah singkatan dari "concatenate" yang berarti menyatukan, tetapi sering digunakan untuk menampilkan isi file. Tanda semicolon (;) yang digunakan di antara dua perintah cat berfungsi sebagai pemisah, memungkinkan kedua perintah dijalankan secara berurutan dalam satu baris; eksekusi cat p2.sh akan dimulai setelah cat p1.sh selesai tanpa memperhatikan hasil dari perintah pertama.
- Perintah ./p1.sh ; ./p2.sh digunakan untuk menjalankan dua skrip shell yang berada di direktori saat ini, pertama p1.sh dan kemudian p2.sh. Tanda semicolon (;) yang digunakan di antara kedua perintah bertindak sebagai pemisah yang memungkinkan kedua skrip dijalankan secara berurutan dalam satu baris perintah. Eksekusi dari ./p2.sh akan dimulai hanya setelah ./p1.sh telah selesai dijalankan, tanpa memandang apakah p1.sh berhasil atau gagal.

- Menjalankan script sebagai prosees background, sehingga prompt tidak menunggu program tersebut selesai. Untuk menjalankan proses background gunakan tanda & pada akhir instruksi.

```
3123600005$ ./p1.sh &
[1] 2225
3123600005$ Program bash Script
./p2.sh &
[2] 2226
[1]   Done                    ./p1.sh
3123600005$ Program 2 bash script

[2]+  Done                    ./p2.sh
3123600005$
```

Analisa:

Perintah di atas enginstruksikan sistem untuk menjalankan kedua skrip, p1.sh dan p2.sh, secara bersamaan di latar belakang. Simbol & setelah nama skrip memberi tahu shell untuk menjalankan proses tersebut sebagai job latar belakang, yang memungkinkan shell untuk segera kembali ke prompt dan siap menerima perintah lain tanpa menunggu skrip pertama selesai. Dengan demikian, kedua skrip tersebut dijalankan secara paralel, memungkinkan kinerja multitasking tanpa menghalangi interaksi pengguna dengan terminal.

Untuk menjalankan 2 program atau lebih dalam satu block, kemudian dieksekusi sebagai proses background, gunakan tanda (…)

```
3123600005$ (ls ; who) > hasil &
[1] 2228
3123600005$ cat hasil
Desktop
Documents
Downloads
hasil
list
Music
p1.sh
p2.sh
Pictures
Public
sorter
Templates
Videos
ivanrhmt tty2         2024-04-23 13:49 (tty2)
[1]+  Done                    ( ls --color=auto; who ) > hasil
3123600005$
```

Analisa:
Perintah tersebut digunakan untuk memasukkan isi dari perintah ls dan who secara berurutan ke dalam file yang bernama hasil, kemudian dieksekusi sebagai proses background.

7. Job Control

- Proses foreground

```
3123600005$ ps x
    PID TTY      STAT   TIME COMMAND
   1054 ?        Ss     0:00 /lib/systemd/systemd --user
   1055 ?        S      0:00 (sd-pam)
   1070 ?        S<sl   0:07 /usr/bin/pipewire
   1079 ?        S<sl   0:01 /usr/bin/wireplumber
   1080 ?        S<sl   0:01 /usr/bin/pipewire-pulse
   1081 ?        Ss     0:00 /usr/bin/dbus-daemon --session
   1082 ?        SLsl   0:00 /usr/bin/gnome-keyring-daemon -
   1099 tty2     Ssl+   0:00 /usr/libexec/gdm-wayland-sessio
   1104 ?        Ssl    0:00 /usr/libexec/gvfsd
   1106 tty2     Sl+    0:00 /usr/libexec/gnome-session-bina
   1179 ?        Sl     0:00 /usr/libexec/gvfsd-fuse /run/us
   1193 ?        Ssl    0:00 /usr/libexec/gcr-ssh-agent /run
   1194 ?        Ssl    0:00 /usr/libexec/gnome-session-ctl
   1196 ?        Ss     0:00 ssh-agent -D -a /run/user/1000/
```

Analisa:
Perintah ps x digunakan untuk menampilkan informasi tentang proses yang sedang berjalan, dengan spesifik memperlihatkan proses yang tidak terikat dengan terminal serta proses yang dimiliki oleh pengguna saat ini. Perintah ps sendiri

merupakan singkatan dari "process status" dan digunakan secara luas untuk memonitor aktivitas proses pada sistem berbasis Unix.

- Proses background

```
3123600005$ ps x > hasil &
[1] 2278
3123600005$
```

Analisa:
Perintah ps x > hasil & digunakan untuk mengeksekusi beberapa fungsi sekaligus. Perintah ps x memberikan daftar lengkap proses yang dijalankan oleh pengguna saat ini, termasuk proses yang tidak terikat dengan terminal manapun. Output dari perintah ini kemudian dialihkan ke file yang bernama hasil melalui operator pengalihan output >. Ini berarti bahwa semua data tentang proses akan disimpan dalam file tersebut, bukan ditampilkan di layar terminal. Simbol & pada akhir perintah menempatkan seluruh proses ini di latar belakang, yang memungkinkan terminal untuk tetap digunakan untuk perintah lain sementara perintah ini berjalan secara independen.

- Setiap job mempunyai PID yang tunggal (unique). Untuk melihat jobs yang aktif

```
3123600005$ jobs
[1]+  Done                    ps x > hasil
3123600005$
```

Analisa:
Jobs yang aktif adalah ps x > hasil dengan status done.

- Buatlah file ploop.sh. File ini tidak akan pernah berhenti kecuali ditekan Ctrl-C

```
#!/bin/bash
while [ true ]
do
        sleep 10
        echo "Hallo"
done
~
```

Analisa:
Perintah vi plop.sh memulai editor teks vi untuk membuat atau mengedit file bernama ploop.sh. Isi skrip yang tertulis di dalamnya menetapkan Bash sebagai interpreter dengan menggunakan shebang #!/bin/bash. Skrip tersebut berisi loop tak terbatas (while [true]) yang terus berjalan karena kondisinya selalu benar. Di dalam loop, perintah sleep 10 digunakan untuk membuat jeda eksekusi selama 10 detik antar iterasi untuk menghindari penggunaan CPU yang berlebihan. Setelah setiap jeda, skrip mencetak kata "Hallo" ke terminal. Ini berarti bahwa selama skrip ini berjalan, akan terus-menerus mencetak "Hallo" setiap 10 detik.

- Buatlah file ploop.sh menjadi executable. Jalankan program, akan ditampilkan kata Hallo setiap 10 detik. Untuk keluar program, tekan Ctrl-C (^C)

```
3123600005$ chmod +x ploop.sh
3123600005$ ./ploop.sh
Hallo
Hallo
Hallo
^C
3123600005$
```

Analisa:
Program tersebut akan menampilkan string “Hallo” setiap sepuluh detik dan berhenti Ketika user menekan ctrl+c.

- Jalankan program sebagai background

```
3123600005$ ./ploop.sh &
[1] 2368
3123600005$ Hallo
Hallo
Hallo
Hallo
```

Analisa:
Perintah ./plop.sh & digunakan untuk menjalankan file ploop.sh sebagai proses background.

- Periksa job yang aktif

```
jobs
[1]+  Running                 ./ploop.sh &
3123600005$ Hallo
Hallo
```

Analisa:
Jobs yang aktif adalah ./plop.sh & dengan status running.

- Ubah job menjadi foreground, gunakan fg dan nomor job, atau fg dengan argumen berupa nama jobs yang sedang berjalan, atau bila tidak rancu dengan jobs yang lain, gunakan huruf depan nama jobs yang sedang berjalan.

```
3123600005$ fg %Hallo
1
./ploop.sh
Hallo
fg %ploop.sh
Hallo
```

Analisa:
Perintah fg %1 digunakan untuk mengontrol pekerjaan (jobs) yang berjalan di latar belakang (background) pada shell. Perintah ini secara spesifik digunakan untuk membawa pekerjaan yang ditandai dengan nomor pekerjaan 1 (ditunjukkan oleh %1) ke latar depan (foreground).

- Untuk mengembalikan jobs tersebut ke background, tekan Ctrl-Z (^Z), kemudian jalankan instruksi bg

```
Hallo
^Z
[1]+  Stopped                 ./ploop.sh
3123600005$ bg
[1]+ ./ploop.sh &
3123600005$ jobsHallo

[1]+  Running                 ./ploop.sh &
3123600005$
```

Analisa:
Perintah bg digunakan untuk mengatur proses yang berhenti atau berada dalam kondisi tertunda untuk melanjutkan eksekusi, tetapi di latar belakang.

8. Manipulasi stack untuk Direktori

- Instruksi dirs digunakan untuk melihat stack direktori, pada output hanya ditampilkan direktori home ~

```
3123600005$ dirs
~
3123600005$
```

Analisa:
Perintah dirs merupakan bagian dari sistem manajemen tumpukan direktori shell yang memungkinkan pengguna untuk menyimpan, melihat, dan menavigasi ke lokasi direktori yang berbeda dengan mudah. Ini adalah perintah yang sangat berguna ketika bekerja dengan banyak direktori secara simultan

- Membuat 3 buah direktori

```
3123600005$ mkdir marketing sales support
3123600005$
```

Analisa:
Perintah mkdir marketing sales support digunakan untuk membuat 3 direktori dengan nama marketing, sales, support.

- Masukkan direktori sales ke dalam stack dengan instruksi pushd. Maka terdapat 2 direktori dalam stack yaitu $HOME/sales dan $HOME. Kemudian lihat direktori actual

```
3123600005$ pushd sales
~/sales ~
3123600005$ pwd
/home/ivanrhmt/sales
3123600005$
```

Analisa:
Perintah pushd sales digunakan untuk mengelola navigasi direktori dengan lebih efisien. Ketika dijalankan, perintah ini mengubah direktori kerja saat ini ke direktori yang disebut sales, sekaligus menyimpan lokasi direktori sebelumnya ke dalam sebuah tumpukan yang dikelola oleh shell.

- Masuk ke direktori support

```
3123600005$ pushd /home/ivanrhmt/support
~/support ~/sales ~
3123600005$ pwd
/home/ivanrhmt/support
3123600005$
```

Analisa:
Perintah pushd /home/ivanrhmt/support berfungsi untuk mengelola navigasi antar direktori secara efisien dengan melakukan dua tindakan utama. Pertama, perintah ini mengubah direktori kerja saat ini ke /home/ivanrhmt/support, yang memungkinkan pengguna untuk langsung berpindah ke direktori tersebut tanpa harus mengetikkan perintah cd. Kedua, sebelum melakukan perpindahan direktori, pushd menyimpan lokasi direktori kerja sebelumnya ke dalam tumpukan direktori yang dikelola oleh shell.

- Lakukan kembali untuk direktori marketing

```
3123600005$ pushd ../marketing
~/marketing ~/support ~/sales ~
3123600005$ pwd
/home/ivanrhmt/marketing
3123600005$
```

Analisa:
Perintah pushd ../marketing berguna untuk berpindah ke direktori marketing yang berada satu tingkat di atas direktori kerja saat ini, karena ../ merepresentasikan direktori induk dari lokasi saat ini. Selain itu, pushd juga menyimpan lokasi direktori kerja sebelum perubahan ke dalam tumpukan direktori yang dikelola oleh shell.

- Bila pushd dilakukan tanpa argumen, maka stack akan mengambil direktori berikutn

```
3123600005$ pushd
~/support ~/marketing ~/sales ~
3123600005$ pushd
~/marketing ~/support ~/sales ~
3123600005$ pushd
~/support ~/marketing ~/sales ~
3123600005$
```

Analisa:
Perintah pushd digunakan untuk mengelola tumpukan direktori, memudahkan navigasi antar direktori tanpa kehilangan referensi ke lokasi sebelumnya. Fungsi utama dari pushd adalah mengganti direktori kerja saat ini ke direktori yang ditentukan dalam perintah, sambil secara simultan menyimpan direktori sebelumnya ke dalam sebuah tumpukan yang diingat oleh shell.

- Untuk membuat direktori sales menjadi direktori paling atas (top stack), maka pushd dapat dilakukan dengan argumen +n, dimana n adalah nomor urut direktori tersebut

```
3123600005$ pushd +2
~/sales ~ ~/support ~/marketing
3123600005$ pwd
/home/ivanrhmt/sales
3123600005$
```

Analisa:
Perintah pushd +2 digunakan untuk memanipulasi dan mengatur ulang tumpukan direktori yang telah disimpan melalui perintah pushd sebelumnya. Fungsinya adalah untuk memutar tumpukan direktori sedemikian rupa sehingga direktori yang saat ini berada di posisi ketiga dari atas tumpukan (indeks 2, mengingat indeksasi dimulai dari 0) dipindahkan ke bagian atas tumpukan, sekaligus mengubah direktori kerja saat ini menjadi direktori tersebut.

- Untuk menghapus direktori dari stack, gunakan instruksi popd

```
3123600005$ popd
~ ~/marketing ~/support
3123600005$ popd +2
~ ~/marketing
3123600005$ dirs
~ ~/marketing
3123600005$
```

Analisa:

Perintah popd digunakan untuk mengelola tumpukan direktori, yang sering dibentuk menggunakan perintah pushd. Fungsi utama dari popd adalah menghapus direktori teratas dari tumpukan direktori yang telah disimpan dan sekaligus mengubah direktori kerja saat ini ke direktori yang baru dikeluarkan dari tumpukan tersebut. Ini memungkinkan pengguna untuk dengan cepat kembali ke direktori sebelumnya yang mereka kunjungi.

9. Alias

- Alias adalah mekanisme untuk memberi nama alias pada satu atau sekelompok instruksi. Untuk melihat alias yang sudah terdaftar pada system :

```
3123600005$ alias
alias ls='ls --color=auto'
3123600005$
```

Analisa:
Perintah alias digunakan untuk membuat singkatan atau nama baru untuk perintah yang lebih panjang, yang memungkinkan pengguna untuk mengetik perintah yang lebih pendek atau lebih mudah diingat sebagai pengganti perintah asli.

- Membuat beberapa alias

```
3123600005$ alias del='rm -i'
3123600005$ alias h='history'
3123600005$
```

Analisa:
Perintah alias del=’rm -i’ menciptakan alias bernama del yang menggantikan perintah rm -i. begitu juga dengan perintah alias h=’history.

- Gunakan instruksi hasil alias

```
3123600005$ ls
Desktop    Downloads  list       Music  p2.s
Documents  hasil      marketing  p1.sh  Pict
3123600005$ del hasil
rm: remove regular file 'hasil'? y
3123600005$ h | more
    1  PS1='3123600005$ '
    2  clear
    3  cd
    4  pwd
    5  clear
    6  mkdir Tes
    7  cd
```

Analisa:
- Perintah del hasil akan menjalankan perintah rm -i hasil

- Perintah h | more akan menjalankan perintah history | more

➢ Untuk menghapus alias gunakan instruksi unalias

```
3123600005$ unalias del
3123600005$ del files
bash: del: command not found
3123600005$
```

Analisa:
Perintah del files terdapat kesalahan karena alias del sudah dihapus dengan menggunakan instruksi unalias. Perintah unalias del digunakan untuk menghapus alias yang sebelumnya telah ditetapkan, dalam hal ini alias del. Penghapusan alias ini mengembalikan fungsionalitas asli dari nama perintah yang telah di-alias, dalam contoh ini del, yang berarti setelah perintah unalias del dijalankan, del tidak akan lagi berfungsi sebagai singkatan dari rm -i.

LATIHAN

1. Eksekusi seluruh profile yang ada :

a. Edit file profile /etc/profile dan tampilkan pesan sebagai berikut :

```
3123600005$ sudo nano /etc/profile
```

```
fi
echo 'Profile dari /etc/profile'

^G Help        ^O Write Out   ^W Whe
```

b. Asumsi nama anda student, maka edit semua profile yang ada yaitu :

```
3123600005$ nano /home/ivanrhmt/.bash_profile
3123600005$ nano /home/ivanrhmt/.bash_login
3123600005$ nano /home/ivanrhmt/.profile
3123600005$ nano /home/ivanrhmt/.bashrc
3123600005$
```

c. Ganti nama /home/student dengan nama anda sendiri. Pada setiap file tersebut, cantumkan instruksi echo, misalnya pada /home/ student/.bash_profile:

```
echo "Profile dari .bash_profile"
```

.bash_profile

d. Lakukan hal yang sama untuk file lainnya, sesuaikan tampilan dengan nama file yang bersangkutan.

```
echo "Profile dari .bash_login"
```

.bash_login

```
echo "Profile dari .profile"
```

.profile

```
echo "Profile dari .bashrc"
```

.bashrc

2. Jalankan instruksi subtitute user, kemudian keluar dengan perintah exit sebagai berikut :

```
3123600005$ su ivanrhmt
Profile dari .bashrc
ivanrhmt@ivanrhmt:/root$ exit
exit
3123600005$
```

kemudian gunakan opsi – sebagai berikut

```
3123600005$ su - ivanrhmt
Profile dari /etc/profile
Profile dari .bash_profile
ivanrhmt@ivanrhmt:~$ exit
logout
3123600005$
```

Jelaskan perbedaan kedua utilitas tersebut.:

- Ketika menjalankan su ivanrhmt, saya beralih ke pengguna "ivanrhmt" tetapi masih mempertahankan sebagian besar lingkungan shell dari pengguna sebelumnya (yang menjalankan perintah). Ini berarti variabel lingkungan seperti HOME, PATH, dan lain-lain bisa jadi tidak sepenuhnya diupdate untuk mencerminkan pengguna yang baru.
- Penggunaan su - ivanrhmt (dengan tanda minus atau sebagai su -l ivanrhmt yang berarti login) mengindikasikan bahwa Anda ingin beralih ke pengguna "ivanrhmt" dengan lingkungan yang sama seperti saat pengguna tersebut login secara normal melalui proses autentikasi (seperti melalui layar login atau melalui SSH).

3. Logout

    a. Edit file .bash_logout, tampilkan pesan dan tahan selama 5 detik, sebelum eksekusi logout

```
echo "Terima kasih atas sesi yang diberikan"
sleep 5
clear
```

    b. Edit file .bash_logout, tampilkan pesan dan tahan selama 4 detik, sebelum eksekusi logout

```
echo "Terima kasih atas sesi yang diberikan"
sleep 4
clear
```

4. History

    a. Ganti nilai HISTSIZE dari 1000 menjadi 20

```
3123600005$ HISTSIZE=20
3123600005$ h
  280  exit
  281  PS1='3123600005$ '
  282  clear
  283  nano /home/ivanrhmt/.bash_logout
  284  exit
  285  echo $0
  286  clear
  287  PS1='3123600005$ '
  288  nano ~/.bash_logout
  289  clear
  290  nano ~/.bash_logout
  291  exit
  292  PS1='3123600005$ '
  293  clear
  294  HISTSIZE=20
  295  h
  296  alias h='history'
  297  clear
  298  HISTSIZE=20
  299  h
3123600005$
```

b. Gunakan fasilitas history dengan mengedit instruksi baris ke 5 dari instruksi yang terakhir dilakukan.

```
3123600005$ !-5
alias h='history'
3123600005$
```

c. Ulangi instruksi yang terakhir. Gunakan juga ^P dan ^N untuk bernavigasi pada history buffer

```
3123600005$ alias h='history'
3123600005$ !!
alias h='history'
3123600005$ alias h='history'
3123600005$
```

d. Ulaingi instruksi pada history buffer nomor tertentu, misalnya nomor 150

```
3123600005$ !150
./p1.sh
Program bash Script
3123600005$
```

e. Ulangi instruksi dengan prefix “ls”

```
3123600005$ !ls
ls
Desktop    Downloads  marketing  p1.sh  Pictures  Public  sorter   Templates
Documents  list       Music      p2.sh  ploop.sh  sales   support  Videos
3123600005$ !?ls?
ls
Desktop    Downloads  marketing  p1.sh  Pictures  Public  sorter   Templates
Documents  list       Music      p2.sh  ploop.sh  sales   support  Videos
3123600005$
```

Jelaskan perbedaan instruksi diatas

- Perintah !ls Mengulangi perintah terakhir yang dimulai dengan string "ls". Contohnya Jika perintah terakhir yang Anda jalankan adalah ls -al, menggunakan !ls akan menjalankan kembali ls -al.
- Perintah !?ls? Mengulangi perintah terakhir yang mengandung string "ls" di mana saja dalam perintah tersebut. Contohnya Jika perintah terakhir yang Anda jalankan adalah echo files listed: ls -l, menggunakan !?ls? akan menjalankan kembali echo files listed: ls -l.

5. Prompt String (PS)

   a. Edit file .bash_profile, ganti prompt PS1 dengan '>'. Instruksi export diperlukan dengan parameter nama variable tersebut, agar perubahan variable PS1 dikenal oleh semua shell

```
echo "Profil
PS1='> '
export PS1
```

   Eksperimen hasil PS1 :

```
3123600005$ PS1="\! > "
302 > PS1="\d > "
Tue Apr 23 > PS1="\t > "
21:23:23 > PS1="Saya=\u > "
Saya=ivanrhmt > PS1="\w > "
~ > PS1="\h > "
ivanrhmt >
```

   b. Ubahlah warna shell prompt dengan warna biru dan berkedip.

```
ivanrhmt > PS1="\033[5;34m\3123600005$ "
3600005$
```

6. Bash script

   a. Buat 3 buah script p1.sh, p2.sh, p3.sh dengan isi masing-masing :

```
#!/bin/bash
echo "Program p1"
ls -l
~
```

   p1.sh

```
#!/bin/bash
echo "Program p2"
who
~
```

   p2.sh

```
#!/bin/bash
echo "Program p3"
ps x
~
```

   p3.sh

b. Jalankan script tersebut sebagai berikut dan perhatikan hasilnya :

$ ./p1.sh ; ./p3.sh ; ./p2.sh

```
3123600005$ ./p1.sh ; ./p3.sh ; ./p2.sh
Program p1
total 68
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Desktop
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Documents
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Downloads
-rw-r--r-- 1 ivanrhmt ivanrhmt   65 Apr 21 17:05 list
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 marketing
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Music
-rwxr-xr-x 1 ivanrhmt ivanrhmt   36 Apr 23 21:31 p1.sh
```

```
Program p3
    PID TTY      STAT   TIME COMMAND
   1049 ?        Ss     0:00 /lib/systemd/systemd --user
   1050 ?        S      0:00 (sd-pam)
   1066 ?        S<sl   0:02 /usr/bin/pipewire
   1073 ?        S<sl   0:00 /usr/bin/wireplumber
   1075 ?        S<sl   0:01 /usr/bin/pipewire-pulse
   1076 ?        SLsl   0:00 /usr/bin/gnome-keyring-daemon --foregrou
   1077 ?        Ss     0:00 /usr/bin/dbus-daemon --session --addres
   1092 tty2     Ssl+   0:00 /usr/libexec/gdm-wayland-session /usr/bi
   1099 tty2     Sl+    0:00 /usr/libexec/gnome-session-binary
   1102 ?        Ssl    0:00 /usr/libexec/gvfsd
   1166 ?        Sl     0:00 /usr/libexec/gvfsd-fuse /run/user/1000/g
   1188 ?        Ssl    0:00 /usr/libexec/gcr-ssh-agent /run/user/100
```

```
Program p2
ivanrhmt tty2         2024-04-23 20:49 (tty2)
3123600005$
```

$ ./p1.sh &

```
3123600005$ ./p1.sh &
[1] 2929
3123600005$ Program p1
total 68
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Desktop
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Documents
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Downloads
-rw-r--r-- 1 ivanrhmt ivanrhmt   65 Apr 21 17:05 list
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 marketing
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Music
-rwxr-xr-x 1 ivanrhmt ivanrhmt   36 Apr 23 21:31 p1.sh
-rwxr-xr-x 1 ivanrhmt ivanrhmt   34 Apr 23 21:32 p2.sh
-rwxr-xr-x 1 ivanrhmt ivanrhmt   35 Apr 23 21:34 p3.sh
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Pictures
-rwxr-xr-x 1 ivanrhmt ivanrhmt   59 Apr 23 14:50 ploop.sh
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Public
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 sales
-rwxr-xr-x 1 ivanrhmt ivanrhmt   37 Apr 21 16:56 sorter
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 support
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Templates
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Videos

[1]+  Done                    ./p1.sh
3123600005$
```

$ ./p1.sh $ ./p2.sh & ./p3.sh &

```
3123600005$ ./p1.sh $ ./p2.sh & ./p3.sh &
[1] 2935
[2] 2936
3123600005$ Program p3
Program p1
total 68
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Desktop
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Documents
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Downloads
-rw-r--r-- 1 ivanrhmt ivanrhmt   65 Apr 21 17:05 list
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 marketing
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Music
-rwxr-xr-x 1 ivanrhmt ivanrhmt   36 Apr 23 21:31 p1.sh
-rwxr-xr-x 1 ivanrhmt ivanrhmt   34 Apr 23 21:32 p2.sh
-rwxr-xr-x 1 ivanrhmt ivanrhmt   35 Apr 23 21:34 p3.sh
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Pictures
-rwxr-xr-x 1 ivanrhmt ivanrhmt   59 Apr 23 14:50 ploop.sh
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Public
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 sales
-rwxr-xr-x 1 ivanrhmt ivanrhmt   37 Apr 21 16:56 sorter
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 support
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Templates
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Videos
    PID TTY      STAT   TIME COMMAND
```

```
[1]-  Done                    ./p1.sh $ ./p2.sh
[2]+  Done                    ./p3.sh
3123600005$
```

$ (./p1.sh ; ./p3.sh) &

```
3123600005$ ( ./p1.sh ; ./p3.sh ) &
[1] 2940
3123600005$ Program p1
total 68
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Desktop
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Documents
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Downloads
-rw-r--r-- 1 ivanrhmt ivanrhmt   65 Apr 21 17:05 list
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 marketing
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Music
-rwxr-xr-x 1 ivanrhmt ivanrhmt   36 Apr 23 21:31 p1.sh
-rwxr-xr-x 1 ivanrhmt ivanrhmt   34 Apr 23 21:32 p2.sh
-rwxr-xr-x 1 ivanrhmt ivanrhmt   35 Apr 23 21:34 p3.sh
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Pictures
-rwxr-xr-x 1 ivanrhmt ivanrhmt   59 Apr 23 14:50 ploop.sh
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Public
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 sales
-rwxr-xr-x 1 ivanrhmt ivanrhmt   37 Apr 21 16:56 sorter
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr 23 15:24 support
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Templates
drwxr-xr-x 2 ivanrhmt ivanrhmt 4096 Apr  8 21:41 Videos
Program p3
    PID TTY      STAT   TIME COMMAND
   1049 ?        Ss     0:00 /lib/systemd/systemd --user
```

7. Jobs

   a. Buat shell-script yang melakukan loop dengan nama pwaktu.sh, setiap 10 detik, kemudian menyimpan tanggal dan jam pada file hasil.

```
#!/bin/bash
while [ true ]
do
        date >> hasil
        sleep 10
done
```

   b. Jalankan sebagai background; kemudian jalankan satu program (utilitas find) di background sebagai berikut :

```
jobs
find / -print > files 2>/dev/null &
jobs
```

c. Jadikan program ke 1 sebagai foreground, tekan ^Z dan kembalikan program tersebut ke background

```
fg %1
^Z
[1]+  Stopped                 ./pwaktu.sh
3123600005$ bg
[1]+ ./pwaktu.sh &
3123600005$
```

d. Stop program background dengan utilitas kill

```
   3016 pts/1    S      0:00 /bin/bash ./pwaktu.sh
   3053 ?        SNsl   0:00 /usr/libexec/tracker-extract-3
   3084 pts/1    S      0:00 sleep 10
   3087 pts/1    R+     0:00 ps x
3123600005$ kill 3016
3123600005$
```