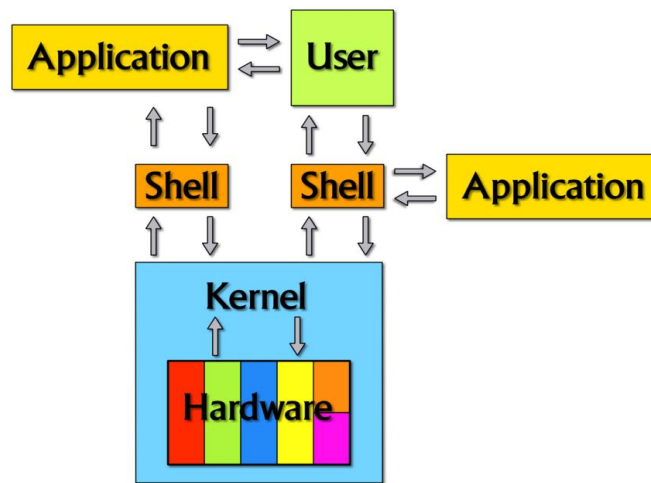


TEORI



Gbr. Shell berhubungan langsung dengan kernel sistem operasi

1. Shell

Shell adalah program (penterjemah perintah) yang menjembatani user dengan sistem operasi. Dalam hal ini *kernel* (inti sistem operasi), umumnya shell menyediakan *prompt* sebagai *user interface*, tempat dimana user mengetikkan perintah-perintah yang diinginkan baik berupa perintah internal shell (*internal command*), ataupun perintah eksekusi suatu file program (*eksternal command*), selain itu shell memungkinkan user menyusun sekumpulan perintah pada sebuah atau beberapa file untuk dieksekusi sebagai program.

Dennis Ritchie dan Ken Thompson (Creator UNIX) berpikirtentang media apa yang harus dibuat agar orang dapat ‘mencoba’ UNIX mereka. Kemudian dibuatlah *shell* dengan nama *Bourne Shell (sh)* yang dibuat oleh SR Bourne. Semenjak itu bermunculan juga jenis shell yang lain yaitu : **C shell** (dibuat oleh Bill Joy), **Korn shell** (dibuat David G. Korn) dan milik GNU sendiri yaitu *Bourne Again Shell (bash)* sebuah shell versi **gratis**.

Secara default dalam Linux menggunakan *bash shell*. Untuk melihat tipe shell dari user ketikkan sintaknya yaitu : `$grep nama_user/etc/passwd`

```
[yudaganteng@rumahstudio yuda]$ grep yuda /etc/passwd
yuda:x:500:500::/home/yuda:/bin/bash
```

Shell	Jenis
/bin/csh atau /usr/bin/csh	C shell
/bin/sh	Bourne shell (bash)
/bin/rsh	Restricted Bourne Shell
/bin/jsh	Bourne shell termasuk jobs control
/bin/ksh	Korn shell
/bin/rksh	Restricted Korn shell

Gbr. Jenis Shell

Dari contoh diatas username yuda menggunakan shell jenis bash. Shell secara garis besar lebih powerfull dibandingkan dengan window manager . Yang perlu diingat karena linux ditulis dengan bahasa C dan bahasa C bersifat case sensitif yang artinya berlaku **pembedakan** antara huruf kecil dan besar, maka perintah di shell ketik dalam huruf kecil jika diketik dalam huruf besar shell tidak akan mengenali.

Manfaat yang dapat diperoleh dengan adanya **shell** adalah:

1. Menangani sebuah prosedur yang terdiri dari sejumlah perintah.
2. Membangun utilitas yang sangat bermanfaat bagi banyak pemakai.
3. Menyederhanakan sebuah perintah yang sering dipergunakan.

2. Struktur Prompt di shell

[username@hostname pwd]\$

contoh :

```
[yudaganteng@rumahstudio /home]$
```

Rumus diatas merupakan struktur prompt shell dimana username ialah **nama user**, hostname ialah **nama host** (default-nya localhost), sedangkan pwd ialah direktory sekarang (letak kursor berada), tanda \$ berarti sebagai user saja, untuk *root* menggunakan tanda #.

3. Langkah Pemrograman Shell

Apa saja sih tool yang di butuhkan untuk membuat *script shell*?

Sederhana saja kok, hanya butuh text editor kesayangan yang anda miliki seperti (vi/ vim/ nano). Kemudian bagaimana ya cara atau langkah-langkah membuat shell script? Untuk membuat script shell langkah-langkah nya yaitu:

Pertamax, anda harus membuat sebuah file dengan text editor seperti *vi*.

Contoh: \$vi file | setelah masuk kedalam file tulis script yang ingin dibuat. Setelah script selesai dibuat, lalu simpan dengan menekan tombol “esc” dilanjutkan dengan :wq atau :x

Kedua, setelah menulis atau membuat script shell rubahlah atribut ijin akses file atau permission dari file script yang telah dibuat menjadi execute (x), caranya sebagai berikut:

Contoh :

```
$ chmod +x your-script-name  
$ chmod +x file, atau  
$ chmod 755 your-script-name
```

Ketiga, selanjutnya mengeksekusi shell script, dengan cara sebagai berikut :

```
$ ./your-script-name  
$ ./file
```

Selesai sudah proses membuat script shell, sangat mungkin anda dapat menjadi programmer SHELL, yen tertarik !

Contoh Script Penjumlahan :

```
#!/bin/bash
clear;
echo "program penjumlahan";
echo -n "masukkan angka pertama: ";
read angka1

echo -n "masukkan angka kedua: ";
read angka2

n=$((angka1+angka2))
echo "hasil penjumlahan antara $angka1 dan $angka2 adalah $n"
```

4. Contoh Program Shell

Yaitu menyusun atau mengelompokkan beberapa perintah *shell* (*internal* ataupun *eksternal* command) menjadi kumpulan perintah yang melakukan tugas tertentu sesuai tujuan penulisnya.

Kelebihan shell di LINUX dibanding sistem operasi lain adalah bahwa *shell* di LINUX memungkinkan kita untuk menyusun serangkaian perintah seperti halnya bahasa pemrograman (*interpreter language*), melakukan proses I/O, menyeleksi kondisi, looping, membuat fungsi, dsb. adalah proses - proses yang umumnya dilakukan oleh suatu bahasa pemrograman, jadi dengan shell di linux kita dapat membuat program seperti halnya bahasa pemrograman. Untuk pemrograman shell pemakai unix atau linux menyebutnya sebagai *script shell*. Sekarang coba kita membuat script shell yang menampilkan informasi berikut:

1. Waktu system
2. Info tentang anda
3. jumlah pemakai yang sedang login di system

contoh scriptnya :

```
#!/bin/bash
#myinfo

#membersihkan tampilan layar
clear

#menampilkan informasi
echo -n "Waktu system   :"; date
echo -n "Anda          :"; whoami
echo -n "Banyak pemakai :"; who | wc -l
```

sebelum dijalankan jangan lupa untuk merubah permission file *myinfo* sehingga dapat dieksekusi oleh anda

```
[fajar@linux$]chmod 755 myinfo
[fajar@linux$]./myinfo
Waktu system   : Sat Nov 25  22:57:15 BORT 2001
Anda          : fajar
Banyak pemakai : 2
```

tentunya layout diatas akan disesuaikan dengan system yang anda gunakan statement echo dengan opsi -n akan membuat posisi kursor untuk tidak berpindah ke baris baru karena secara default statement echo akan mengakhiri proses pencetakan ke standar output dengan karakter baris baru (newline).

Anda boleh mencoba tanpa menggunakan opsi -n, dan lihat perbedaannya. opsi lain yang dapat digunakan adalah -e (enable), memungkinkan penggunaan backslash karakter atau karakter sekuen seperti pada bahasa C atau perl, misalkan :

```
echo -e "\abunyikan bell"
```

jika dijalankan akan mengeluarkan bunyi bell, informasi opsi pada statement echo dan backslash karakter selengkapnya dapat dilihat via man di prompt shell.

```
[fajar@linux$]man echo
```

5. Membuat Program

Kita mencoba membuat program dengan struktur kontrol dengan CASE. Untuk menjalankan program atau mengcompile dengan perintah chmod +x (nama_file). Jika tidak ada yg error, maka eksekusinya yaitu bash (nama_file). Berikut listingnya :

```
#!/bin/bash

clear
echo "*****";
echo " Menu Makanan Murah Kampus Siang Hari ";
echo "*****";
echo "      1. Bakso      ";
echo "      2. Gado-gado  ";
echo "      3. Siomay     ";
echo "      4. EsTea      ";
echo "      5. Exit       ";
echo "-----";
read -p "Masukkan pilihan agan [1-5]: " pil;
echo "-----";
case $pil in
1)
    echo "agan memilih makan Bakso";
    echo -n "Mau berapa mangkuk gan?? = ";
    read jum
    let bayar=jum*3000;
;;
2)
    echo -n "Gado-gadonya Mau makan berapa porsi gan?? = ";
    read jum
    let bayar=jum*4000;
;;
3)
    echo -n "Banyak siomay mau berapa gan?? = ";
    read jum
    let bayar=jum*500
;;
4)
    echo -n "Mau berapa botol nie gan?? = ";
    read jum
    let bayar=jum*1000
;;
*)
    echo "Diet ne agan2 ga mau makan?? :) Terima kasih Gan"
    echo "-----"
    exit
;;
esac

echo "Berarti agan harus bayar = Rp. $bayar"
echo "*Trima kasih sudah membeli* :)"
echo "-----"
```

a. Logika Program

Pada listing diatas kita ketahui bahwa kasus diatas memakai struktur kontrol karena memakai perintah **Case**. Biar Mudah! Dalam ubuntu kita menulis script shell menggunakan teks editor yaitu Vi/gedit untuk menulis statemen perintahnya. Lalu dikompile di terminal.

Caranya yaitu simpan nama file yang sudah dibuat pada teks editor, simpan dimana saja, lalu compile dengan perintah `chmod +x nama_file`, jika program running maka ketik `bash nama_file`.

Berikut penjabarannya :

- **#!/bin/bash** Statement ini merupakan nama shell yang dibuat pada program ini, seperti yg kita ketahui Shell itu bertugas untuk menerjemah perintah yang berisikan sejumlah perintah dimana dapat dijadikan sebagai satu prosedur dan dapat disimpan pada suatu file.
- **Clear** disini berarti membersihkan layar, jika biasanya di windows kita memakai perintah `cls` yaitu clear screen.
- **Echo** digunakan untuk mencetak dan menampilkan yang ada di dalam string, perintah ini seperti `printf` pada bahasa C.
- **Echo "Menu Makanan Murah Kampus Siang Hari";**
Statement ini berarti menampilkan atau mencetak Menu Makanan Murah Kampus Siang Hari, setiap diakhir perintah atau statement harus diakhiri dengan tanda titik koma (;).
- **Read -p "Masukkan Pilihan agan [1-5] : " pil;**
Statement read digunakan untuk membaca data dari papan ketik (keyboard) dan meletakkannya pada variabel, -p merupakan ungkapan untuk menggunakan pilihan. Jadi statement ini merupakan inputan dari user.
- **Case \$pil in Pil**
merupakan nama variabel yang saya pilih, Semua perintah Linux menghasilkan nilai keluaran yang disebut sebagai nilai exit atau \$. Ini untuk menentukan pilihan yang akan dijalankan.
 - **1) Echo "agan memilih makan bakso";**
ini merupakan tampilan pilihan pertama. -n merupakan ungkapan dari operator string. Read jum merupakan membaca jumlah yang telah diinput oleh user, jum disini merupakan variabel bebas. `Let bayar=jum*3000` statement `let` digunakan sebagai pernyataan untuk kondisi yang menggunakan ekspresi numerik dan operasi aritmatika. Dalam hal ini jumlah dari yg user input akan dikalikan dengan 3000. Sedangkan antara pernyataan yang satu dengan pernyataan yg lain dipisahkan dg tanda titik koma dua kali (; ;).
 - **2) pilihan kedua** yaitu pilihan untuk gado-gado,
statement ini juga sama seperti pilihan pertama, tapi harga dari jumlah inputan user dikalikan dengan 4000.
 - **3) pilihan ketiga** pilihan siomay
untuk hasil operasi aritmatika pilihan user dikalikan dengan 500. Jadi setiap siomay yang diinput user hasilnya akan dikali 500.
 - **4) pilihan keempat** pilihan untuk estea user menginput mau berapa banyak botol lalu perintah `let` akan menjumlahkannya hasil aritmatikanya.
 - **5) pilihan kelima** pilihan untuk exit dimana akan mencetak "Diet ne agan2 ga mau makan ?? J trima kasih gan. Berarti pilihan sudah keluar atau exit.
- **Tanda "*"** menyatakan bagian yang dieksekusi bila tidak ada nilai yang cocok dengan nilai di dalam variabel. Dalam hal ini jika user menginput pilihan yang tidak ada di dalam daftar pilihan maka akan dieksekusi tidak cocok, maka akan mencetak "Pilihan ga ada tuh Gan !! monggo dicoba lagi. **Jika**

nilainya sesuai, perintah yang terletak sesudah nilai `n` akan dijalankan hingga ditemukan tanda double semicolon (`::`) yang berarti proses akan diteruskan ke perintah setelah kata `esac`.

- Echo “Berarti `agan` harus bayar = Rp.`$bayar`” statement ini akan mencetak isi dari string tersebut dengan ketentuan `$bayar` berdasarkan hasil perhitungan jumlah aritmatika yang telah dihitung diatas. Dan statement terakhir menampilkan atau mencetak “Terima kasih sudah membeli”.

b. Output Program

```

jestry@jestry-imoet: ~/Menu
*****
Menu Makanan Murah Kampus Siang Hari
*****
1. Bakso
2. Gado-gado
3. Siomay
4. EsTea
5. Exit

-----
Masukkan pilihan agan [1-5]: 1
-----
agan memilih makan Bakso
Mau berapa mangkuk gan?? = 4
Berarti agan harus bayar = Rp. 12000
*Trima kasih sudah membeli* :)

jestry@jestry-imoet:~/Menu$ chmod +x pilihan
jestry@jestry-imoet:~/Menu$ bash pilihan

jestry@jestry-imoet: ~/Menu
*****
Menu Makanan Murah Kampus Siang Hari
*****
1. Bakso
2. Gado-gado
3. Siomay
4. EsTea
5. Exit

-----
Masukkan pilihan agan [1-5]: 2
-----
Gado-gadonya Mau makan berapa porsi gan?? = 2
Berarti agan harus bayar = Rp. 8000
*Trima kasih sudah membeli* :)

jestry@jestry-imoet:~/Menu$ chmod +x pilihan
jestry@jestry-imoet:~/Menu$ bash pilihan

jestry@jestry-imoet: ~/Menu
*****
Menu Makanan Murah Kampus Siang Hari
*****
1. Bakso
2. Gado-gado
3. Siomay
4. EsTea
5. Exit

-----
Masukkan pilihan agan [1-5]: 5
-----
Diet ne agan2 ga mau makan?? :) Terima kasih Gan

jestry@jestry-imoet:~/Menu$

```


LAB

Tujuan

- Mengetahui dan memahami Fungsi pemrograman Shell
- Mampu menggunakan Shell dalam membuat dan mengedit program
- Membuat Report Lab

Materi

1. Praktekkan perintah-perintah dan kasus sesuai teori diatas
2. Berikan juga dokumentasi penyelesaian dari study kasus yang diberikan Asisten
3. Simpan Screenshot dan keterangan dalam laporan lab.

Ketentuan Laporan

1. Laporan adalah kombinasi dari screenshot dan keterangan
2. Kirim via email ke : osd3ti2015@gmail.com
 - Subyek : **Kelas-NIM-NAMA-Shell**
 - Nama File : kelas-nim-nama-shell.pdf
 - Harus dalam bentuk .PDF
3. Batas Waktu : 1 Hari setelah Lab diselesaikan, jam 23:59