

PRAKTIKUM ALGORITMA DAN PEMROGRAMAN E1

PRAKTIKUM 2

COMMAND DI LINUX DAN KONDISIONAL

Daftar Materi

1. Operasi Command Linux
2. Kondisional

A. Operasi Command Linux di Server

Command Line interface (CLI) merupakan suatu mekanisme berinteraksi dengan sistem operasi komputer atau software dengan mengetikkan perintah untuk menjalankan suatu pekerjaan tertentu. Antar muka yang hanya berbasis teks ini sangat kontras dengan penggunaan klik mouse dalam lingkungan graphical user interface (GUI), atau menu pada text user interface (TUI) untuk memilih pilihan-pilihan. Metode memerintah komputer untuk menjalankan pekerjaan tertentu biasa disebut dengan “memasukkan” perintah: sistem menunggu pengguna untuk memutuskan kapan perintah teks yang dituliskan akan dikirim ke sistem dengan menekan tombol “Enter”.

Selanjutnya command line interpreter menerima, menganalisa, dan menjalankan perintah yang diperintahkan. Command line interpreter dapat berjalan dalam suatu terminal teks atau dalam jendela emulator terminal, misalnya klien shell secara remote contohnya PuTTY. Selama proses bekerja, pada umumnya perintah tersebut akan menghasilkan suatu keluaran (output) bagi pengguna dalam bentuk baris-baris teks pada CLI. Keluaran ini mestinya merupakan jawaban dari pertanyaan yang dikirim melalui perintah tadi, atau bisa juga merupakan informasi hasil akhir dari operasi yang berjalan

Langkah-Langkah Praktikum

1. Login server
`>ssh [NIM]@10.137.61.11`
2. Masukkan password dan ganti password
`[NIM]@10.137.61.11's password: [nim]@123`
3. Menampilkan isi file dan direktori
`$ ls`
4. Membuat direktori baru
`$ mkdir [nama_direktori]`
`$ mkdir praktikum1`
5. Berpindah direktori
`$ cd [nama_direktori]`
`$ cd praktikum1`
6. Membuat file bahasa c
`$ nano [nama_file].c`
`$ nano JarakGLBB.c`

7. Compile program dengan perintah

```
$ gcc [nama_file].c -o [nama_file_eksekusi]
$ gcc JarakGLBB.c -o JarakGLBB
```
8. Eksekusi program dengan perintah

```
$ ./[nama_file_eksekusi]
$ ./JarakGLBB
```
9. Keluar dari server

```
$ exit
```

Command Line Linux Lainnya:

1. Membuat direktori baru

```
$ mkdir
$ mkdir praktikum1
$ mkdir praktikum1
```
2. Membuat beberapa direktori sekaligus

```
$ mkdir -p dir1/dir2/dir3
$ mkdir -p latihan3/bab1/pilihan_ganda
$ mkdir -p latihan4/bab1/pilihan_ganda
```
3. Melihat isi seluruh direktori

```
$ ls -R
$ ls -R latihan3
```
4. Menghapus satu direktori

```
$ rmdir
$ rmdir latihan1
```
5. Menghapus direktori berikut isinya

```
$ rm -r
$ rm -r latihan3
```
6. Berpindah direktori

```
$ cd nama_direktori
$ cd latihan4
```
7. Berpindah langsung dalam beberapa tingkat direktori

```
$ cd direktori1/direktori2
$ cd latihan4/bab1/pilihan_ganda
```
8. Langsung kembali ke root direktori

```
$ cd /
```
9. Menyalin file

```
$ cp lokasi_file_sumber lokasi_file_tujuan
```
10. Menyalin sekaligus direktori berikut isinya

```
$ cp -r /folder_asal ./folder_tujuan
```
11. Mengganti nama file

```
$ mv file_awal file_baru
```
12. Menghapus file

```
$ rm nama_file
```

B. Kondisional (Analisa Kasus)

Kondisional (analisa kasus) adalah merupakan sebuah bentuk penguraian terhadap satu masalah menjadi beberapa sub-sub masalah yang saling lepas (asing). Penguraian satu masalah menjadi sub-sub masalah harus merepresentasi enumerasi semua kemungkinan kasus-kasusnya.

Konstruksi suatu analisis kasus dapat dimulai dari menentukan semua kondisi yang mungkin atau dimulai dari menentukan variasi aksi.

Mendefinisikan:

- **Kondisi**, ekspresi yang menghasilkan true atau false
- **Aksi**, hal yang dilaksanakan jika kondisi yang terkait dengan aksi telah terpenuhi

1. Satu Kasus

Satu Kasus merupakan bagian dari ekspresi kondisional yang hanya memiliki 1 kondisi, atau hanya dapat diuraikan menjadi 1 sub masalah.

Syarat: - Kondisi memiliki domain harga [true, false]

Notasi	Bahasa C
<pre><u>if</u> (<i>kondisi</i>) <u>then</u> <i>aksi</i></pre>	<pre>if (<i>kondisi</i>){ do_aksi; }</pre>

2. Dua Kasus

Dua Kasus merupakan bagian dari ekspresi kondisional yang hanya memiliki 2 kondisi, atau dapat diuraikan menjadi 2 sub masalah.

Syarat:

- Kondisi-1 dan Kondisi-2 memiliki domain harga [true, false]
- Kondisi-1 dan Kondisi-2 saling berlawanan

Notasi	Bahasa C
<pre><u>if</u> (<i>kondisi</i>) <u>then</u> <i>aksi-1</i> <u>else</u> { <u>not</u> <i>kondisi</i> } <i>aksi-2</i></pre>	<pre>if (<i>kondisi_1</i>){ do_aksi_1; } else{ do_aksi_2; }</pre>

3. Lebih dari Dua Kasus

Lebih dari Dua Kasus merupakan bagian dari ekspresi kondisional yang memiliki lebih dari 2 kondisi, atau dapat diuraikan menjadi lebih dari 2 sub masalah.

Syarat:

- Kondisi-1, ..., kondisi-n memiliki domain harga **[true, false]**
- Kondisi-1, ..., kondisi-n disjoint (tidak ada kasus yang sama tercakup pada dua buah kondisi)
- Kondisi mencakup semua kemungkinan

Pada Bahasa C untuk notasi depend on kita dapat menggunakan `switch` atau `if-else`. Nah masing-masing pernyataan digunakan sesuai dengan kebutuhannya masing-masing. Pernyataan `if-else` digunakan ketika menguji ekspresi berdasarkan rentang nilai/kondisi, atau ketika menggunakan operator logika (<, >, <= dan >=). Sedangkan pernyataan `switch` digunakan pada satu variable dengan nilai bervariasi, biasanya digunakan pada pilihan menu.

Notasi	Bahasa C
<p><u>depend on</u> (<i>nama-nama</i>)</p> <p><kondisi-1> : <aksi-1></p> <p><kondisi-2> : <aksi-2></p> <p><kondisi-3> : <aksi-3></p> <p>...</p> <p><kondisi-N> : <aksi-N></p>	<ul style="list-style-type: none">• Pernyataan switch<pre>switch (nama){ case kondisi_1: do_aksi_1; case kondisi_2: do_aksi_2; case kondisi_3: do_aksi_3; default: do_default; }</pre>• Pernyataan if-else<pre>if (kondisi_1){ do_aksi_1; } else if(kondisi_2){ do_aksi_2; } else if(kondisi_3){ do_aksi_3; } else{ do_aksi_4; }</pre>