

**LAPORAN TUGAS AKHIR SEMESTER
PEMROGRAMAN BERBASIS SHELL
SISTEM OPERASI**



Disusun oleh:

Nama : Rangga Dwi Saputra

NIM : 2341720248

Kelas : TI 1B

Dosen Pembimbing :

1.Erfan Rohadi, ST., M.Eng., Ph.D.

2.Mohammad Faried Rahmat, S.ST., M.Tr.T

**D4 Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang**

2024

Deskripsi Proyek

```
PEMROGRAMAN SHELL
=====

-----
Masukkan operasi yang ingin dijalankan :
1. Melihat isi Direktory
2. Melihat Proses Parent dan Child
3. ID Proses
4. Manajemen Memori
5. Kalkulator
-----
Masukkan pilihan: █
```

Tugas akhir Mata Kuliah Sistem Operasi ini didasarkan dari Pemrograman Shell yang telah diajarkan pada pertemuan sebelumnya. Yang mana kami menjalankan program berbasis shell tersebut menggunakan linux. Program di implementasikan dalam Shell Script, sebuah program penterjemah perintah sebagai user interface (penghubung) agar user dapat mengakses layanan dari kernel (program inti sistem operasi).

Shell script dapat dijalankan melalui terminal linux agar bisa mengakses system operasinya. Dalam tugas ini saya menggunakan wsl terminal kali linux, yang mana dapat dijalankan meskipun saya menggunakan Windows 11. Untuk kode editornya sendiri saya menggunakan Visual Studio Code.

Tugas kami disini adalah membuat daftar menu yang bisa dijalankan dalam program shell minimal 2 fitur. Dari tugas ini saya memutuskan untuk mengimplementasikan praktikum-praktikum sebelumnya sejumlah lima fitur, yang terdiri dari 1 fitur dasar berupa melihat isi direktory, 3 fitur unggulan berupa melihat proses parent dan child, id proses dan manajemen memory, dan 1 fitur tambahan yaitu fungsi kalkulator.

Detail Penjelasan Program

1. Melihat Isi Direktori

Dalam fitur ini pengguna dapat melihat isi directory tersebut hanya dengan menggunakan command ``ls``. Program akan menampilkan isi directory tersebut. Berikut implemantasian output dari fitur tersebut ;

```
-----  
Masukkan pilihan: 1  
Documentation  rangka.sh  README.md
```

Dapat dilihat dari directory ini terdapat directory Bernama Documentation, dan 2 jenis file yakni rangka.sh dan README.md

2. Melihat Proses Parent dan Child

Dalam fitur ini pengguna dapat melihat proses parent (induk) dan juga anaknya dari system computer yang sedang dijalankan. Program tersebut dijalankan berdaarkan susuan kode sebagai berikut,

```
echo "Proses parent dan child:"  
ps -ef --forest
```

Perintah ini digunakan untuk menampilkan daftar proses yang sedang berjalan di sistem dalam format tree (pohon), sehingga memudahkan untuk melihat hubungan antara proses parent dan child.

- `ps`: Command untuk menampilkan snapshot dari proses yang sedang berjalan.
- `-e`: Menampilkan semua proses di sistem.
- `-f`: Menggunakan format panjang yang menampilkan informasi lebih lengkap termasuk UID (user ID), PID (process ID), PPID (parent process ID), dan sebagainya.
- `--forest`: Menampilkan proses dalam bentuk pohon untuk menunjukkan hierarki proses. Ini membuatnya lebih mudah untuk melihat proses mana yang dimulai oleh proses lain.

Berikut hasil program yang ditunjukkan ;

```

-----
Masukkan pilihan: 2
Proses parent dan child:
UID      PID  PPID  C  STIME TTY          TIME CMD
root      1    0    0  10:17 hvc0      00:00:00 /init
root      6    1    0  10:17 hvc0      00:00:00 plan9 --control-socket 5 --log-level 4 --server-fd 6 --pipe-fd 8 --log-truncate
root      9    1    0  10:17 ?          00:00:00 /init
root     10    9    0  10:17 ?          00:00:00 \_ /init
dinn     11   10    0  10:17 pts/0      00:00:00 \_ -bash
dinn    175   11    0  11:33 pts/0      00:00:00 \_ /bin/bash ./rangga.sh
dinn    181  175  99  11:35 pts/0      00:00:00 \_ ps -ef --forest
Apakah ingin mengulangi operasi lagi? (y/n) : █

```

3. ID Proses

Memungkinkan pengguna untuk melihat proses id yang sedang dijalankan dalam system computer. Program ini dijalankan dari kode program sebagai berikut:

```

child_process() { # mendefinisikan loop yang mencetak PID (Process ID)
# dari proses yang sedang berjalan dan tidur selama 2 detik setiap iterasi
    for i in {1..5}; do
        echo "Ini adalah process $$"
        sleep 2
    done
}
# Menjalankan fungsi di latar belakang
child_process &
parent_pid=$! # menyimpan PID dari proses terakhir yang
diluncurkan di latar belakang menggunakan operator '&'

# Ensure the parent process waits for the child to complete
wait $parent_pid

```

Program akan dijalankan sebanyak 5 kali untuk menampilkan id prosesnya. Berikut hasil keluarannya ;

```

-----
Masukkan pilihan: 3
Ini adalah process 175
Ini adalah process 175
Ini adalah process 175
Ini adalah process 175
Ini adalah process 175
Apakah ingin mengulangi op

```

4. Manajemen Memory

Manajemen memori adalah proses mengendalikan dan mengkoordinasikan memori utama komputer. Ini memastikan bahwa blok ruang memori dikelola dan dialokasikan dengan benar sehingga sistem operasi (OS), aplikasi , dan proses lain yang berjalan memiliki memori yang diperlukan untuk menjalankan operasinya. Untuk mengetahui dan melihat memorinya yang dijalankan, yaitu melalui kode program sebagai berikut;

```
sudo dmesg | more
```

- dmesg (display message or driver message) adalah perintah yang digunakan untuk menampilkan buffer log pesan kernel. Pesan-pesan ini mencakup informasi tentang hardware dan driver pada sistem, serta berbagai pesan sistem lainnya. Ini sangat berguna untuk debug dan memecahkan masalah sistem, terutama yang terkait dengan hardware dan driver.
- more adalah pager untuk menampilkan teks satu layar pada satu waktu. Ini berguna saat Anda memiliki output panjang yang tidak muat di satu layar, sehingga Anda bisa membaca output tersebut sedikit demi sedikit dengan menekan spasi untuk melanjutkan ke bagian berikutnya.

Output akan berhenti setelah satu layar penuh teks. Anda dapat menekan:

- Space untuk melihat layar berikutnya.
- Enter untuk melihat baris berikutnya.
- q untuk keluar dari more.

Berikut hasil keluaran dari program tersebut;

```
Masukkan operasi yang ingin dijalankan :
1. Melihat Isi Direktori
2. Melihat proses parent dan proses child
3. Membuat dua proses sebanyak 5 kali
4. Melihat Manajemen Memori
5. Kalkulator
-----
Masukkan pilihan: 4
[sudo] password for dimi:
[ 0.000000] linux version 5.15.146.1-microsoft-standard-WSL2 (root@65c757a07562) (gcc (GCC) 11.2.0, GNU ld (GNU Binutils) 2.37) #1 SMP Thu Jan 11 04:09:03 UTC 2024
[ 0.000000] Command line: initrd=initrd.img WSL_ROOT_INIT=1 panic=1 nr_cpus=8 hv_utils.timesync_implicit=1 console=hvc0 debug=pty,legacy_count=0
[ 0.000000] KERNEL supported cpus:
[ 0.000000] Intel GenuineIntel
[ 0.000000] AMD AuthenticAMD
[ 0.000000] Centaur CentaurHauls
[ 0.000000] BIOS-provided physical RAM map:
[ 0.000000] BIOS-e820: [mem 0x0000000000000000-0x000000000009ffff] usable
[ 0.000000] BIOS-e820: [mem 0x00000000000a0000-0x00000000000bffff] reserved
[ 0.000000] BIOS-e820: [mem 0x0000000000100000-0x000000000001ffff] ACPI data
[ 0.000000] BIOS-e820: [mem 0x0000000000200000-0x0000000000b7ffff] usable
[ 0.000000] NX (Execute Disable) protection: active
[ 0.000000] DMI not present or invalid.
[ 0.000000] Hypervisor detected: Microsoft Hyper-V
[ 0.000000] Hyper-V: privilege flags low 0xc27f, high 0x3b8030, hints 0xc2c, misc 0xe0bed7b6
[ 0.000000] Hyper-V Host Build: 22621.10.0.3-0.3593
[ 0.000000] Hyper-V: Halted features: 0x0a0000
[ 0.000000] Hyper-V: LAPIC Timer Frequency: 0x1e8480
[ 0.000000] Hyper-V: Using hypercall for remote TLB flush
[ 0.000000] clocksource: hyperv_clocksource_tsc_page: mask: 0xffffffffffff max_cycles: 0x24e6a1710, max_idle_ns: 440795202120 ns
[ 0.000004] tsc: Detected 2499.333 MHz processor
[ 0.000010] e820: update [mem 0x00000000-0x000001fff] usable -> reserved
[ 0.000021] e820: remove [mem 0x000a0000-0x0000ffff] usable
[ 0.000026] last_pfn = 0xb7a00 max_arch_pfn = 0x000000000
[ 0.000030] smp/tbl: Configuration [0-7]: NO MC UC UC UC MB MP UC MT
[ 0.000121] Using GB pages for direct mapping
--More--
```

5. Kalkulator

Dalam fitur ini pengguna dapat menggunakan operator dasar seperti penjumlahan, pengurangan, perkalian, dan pembagian untuk menghitung angka angka. Namun, kekurangannya dalam kode program saya hanya masih bisa mengoperasikan 2 angka saja dan 1 operator, apabila ingin menjalankan yang lain maka harus mengulangi dari awal dengan inputan yang baru.

```
lanjut="y"
while [[ "$lanjut" == "y" ]]; do
    echo -e "Kalkulator"
    read -p "Masukkan angka pertama: " angka1
    read -p "Masukkan operator (+, -, /, \*): " operator
    read -p "Masukkan angka kedua: " angka2

    case $operator in
        +)
            hasil=$(echo "$angka1 + $angka2" | bc)
            echo "$angka1 + $angka2 = $hasil"
            ;;
        -)
            hasil=$(echo "$angka1 - $angka2" | bc)
            echo "$angka1 - $angka2 = $hasil"
            ;;
        \*)
            hasil=$(echo "$angka1 * $angka2" | bc)
            echo "$angka1 * $angka2 = $hasil"
            ;;
        /)
            hasil=$(echo "$angka1 / $angka2" | bc)
            echo "$angka1 / $angka2 = $hasil"
            ;;
        *)
            echo "Operator tidak valid"
            ;;
    esac

    read -p "Hitungan yang lain (y/n): " lanjut
done
```

Perlu diperhatikan bahwa untuk menjalankannya, pengguna harus terlebih dahulu menginstall `bc` untuk mendukung berjalannya operasi perhitungannya. Apabila belum

menginstallnya maka program tidak akan berjalan sebagaimana mestinya. Untuk menginstallnya dapat menggunakan *sudo apt install bc*

```
(dinn@LAPTOP-36IOUL75)-[/mnt/d/POLINEMA/Semester 2/SISTEM OPERASI/UAS-SISTEM]
$ sudo apt install bc
[sudo] password for dinn:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  0trace 7zip apt afflib-tools aircrack-ng amap amass amass-common android-fram
  android-libziparchive apache-users apache2 apache2-bin apache2-data apache2-ut
  libtcl3 libtcl3-dev libtcl3-dev 5.8.1-1 libtcl3-dev 5.8.1-1
```

Setelah bc terinstall, program dapat dijalankan seperti berikut;

```
5. Kalkulator
-----
Masukkan pilihan: 5
Kalkulator
Masukkan angka pertama: 2
Masukkan operator (+, -, /, \*): +
Masukkan angka kedua: 3
2 + 3 = 5
Hitungan yang lain (y/n): y
Kalkulator
Masukkan angka pertama: 4
Masukkan operator (+, -, /, \*): \*
Masukkan angka kedua: 4
4 * 4 = 16
Hitungan yang lain (y/n): y
Kalkulator
Masukkan angka pertama: 7
Masukkan operator (+, -, /, \*): &
Masukkan angka kedua: 7
Operator tidak valid
Hitungan yang lain (y/n):
```

Penutup

Dalam pembelajaran matakuliah system operasi, tugas akhir praktikum ini sangat efektif untuk membantu mahasiswa menggunakan shell script yang telah dipelajari pada pertemuan sebelumnya. Dapat disimpulkan bahwa dalam semester ini, matakuliah system operasi menekankan mahasiswa untuk mempelajari Sistem Operasi linux dan berbagai macam hal yang dapat diakses disana, salah satunya adalah Shell Script.

Demikian laporan yang saya buat berdasarkan praktikum yang saya kerjakan. Mohon maaf apabila terdapat kekurangan atau ketidak sempurnaan dalam pengerjaan proyek ini. Terimakasih.