LAPORAN PROYEK SHELL SCRIPTING

SISTEM OPERASI

TI-1B

Farrel Augusta Dinata / 2341720081

D4 TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
TAHUN 2024

BAB I
PENDAHULUAN

Program shell scripting adalah sebuah pemrograman yang basis menjalankan programnya adalah shell. Ini biasa ditemukan pada sistem operasi UNIX/Linux. Shell adalah antarmuka pengguna berbasis teks yang memungkinkan pengguna untuk berinteraksi dengan sistem operasi melalui baris perintah. Pengguna bisa membuat program yang disisipi beberapa perintah Linux sehingga ini bisa dijadikan sebuah alat yang bermanfaat karena bisa menjalankan berbagai perintah sekaligus. Tugas-tugas sederhana seperti mengelola file, mengatur alur eksekusi program, atau mengotomatisasi proses-proses rutin adalah contoh beberapa hal yang bisa dilakukan di shell scripting.

Tujuan dari pembuatan proyek ini adalah dalam rangka untuk menuntaskan tugas akhir mata kuliah sistem operasi program studi D4 Teknik Informatika Politeknik Negeri Malang. Proyek ini dikembangkan didasarkan pada materi-materi sistem operasi berbasis Linux yang telah dipelajari selama satu semester ini. Materi yang diajarkan dimulai dari pengenalan beberapa perintah Linux sederhana hingga melakukan pemrograman dengan shell menggunakan teks editor bawaan Linux. Cakupan proyek berupa laporan, program shell yang dibuat, file powerpoint, serta contoh output program berupa *screenshot*. Keseluruhan program dibuat secara individu dan seluruh hasil kerja akan ditaruh di GitHub repository saya.

Program yang saya buat saya namakan “Row”. Program ini rencananya saya tujukan bagi seorang programmer sehingga banyak fitur yang bisa menunjang produktivitas saat melakukan proyek ngoding. Fitur-fitur yang tersedia nantinya adalah 1) melihat alamat direktori saat ini, 2) membuat folder baru, 3) mencari sebuah file, 4) membuat catatan, 5) membaca sebuah file, 6) git commit log, 7) kalkulator, 8) monitoring sistem, 9) timer, dan 10) kustom perintah Linux. Beberapa fitur tersebut saya terinspirasi dari kegiatan sehari-hari saya sebagai mahasiswa programmer. Untuk hal teknis program, saya tak hanya mengandalkan perintah-perintah yang ada di Linux, tapi juga mengkombinasikannya dengan hal lain seperti dengan mekanisme git.

Peralatan yang saya gunakan dalam membuat sebuah program ini tidak terlalu banyak namun sangat berguna dalam menunjang penyelesaian proyek ini. Untuk menulis program ini saya menggunakan perangkat laptop berbasis Windows. Namun, agar program shell bisa dijalankan di Windows, maka bisa menggunakan WSL (Windows Subsystem for Linux). Itu karena secara bawaan, Windows tidak bisa menjalankan program dengan berbasis shell (.sh). Kemudian dalam hal penulisan program saya mengandalkan Microsoft Visual Studio Code (VSCode). Fitur-fitur yang disediakan di VSCode sudah cukup mumpuni untuk membuat program shell scripting. Satu software lagi yang saya gunakan adalah ChatGPT 3.5 untuk membantu memahami dan memberikan referensi pembuatan kode program.

Manfaat utama yang bisa diperoleh dari pembuatan proyek ini adalah bisa memudahkan dan meningkatkan produktivitas sebagai programmer. Selain itu,

sebagai pembuat program, saya bisa mengasah skil keterampilan lain dalam pemrograman. Karena sejatinya membuat sebuah program dengan berbasis shell cukup berbeda dengan program menggunakan bahasa pemrograman yang umum seperti Java, Python, ataupun JavaScript.

Pada laporan ini, saya bertujuan untuk memberikan gambaran yang komprehensif tentang pengembangan proyek shell scripting yang saya kembangkan sebagai bagian dari tugas akhir mata kuliah sistem operasi. Melalui penjelasan tentang tujuan, metodologi, dan hasil yang dicapai, saya berharap laporan ini dapat memberikan wawasan yang bermanfaat bagi pembaca tentang kemampuan saya dalam mengimplementasikan konsep-konsep shell scripting dalam lingkungan praktis.

BAB II
PEMBAHASAN

Sebelum membahas lebih lanjut mengenai fitur yang saya buat, maka saya akan menjelaskan alur kerja dari program yang saya buat saat ini. Di sini saya menggunakan proses looping di shell script agar pengguna bisa melakukan menu berulang kali.

```
ulangiMenu=y
while [ $ulangiMenu == "y" ]; do
    show_menu
    read -p " >>> " pilihan

    case $pilihan in
        1)
```

```
show_menu() {
    echo "======================================================"
    echo "              --[ Jam : $(date +%T) ]--         "
    echo $'\033[0;32m-- Selamat datang di menu serbaguna komputer anda -- \033[0m'
    echo "------------------------------------------------------"
    echo "Silakan pilih salah satu menu berikut"
    echo "  1. Lihat path direktori saat ini"
    echo "  2. Membuat folder baru"
    echo "  3. Cari sebuah file"
    echo "  4. Buat catatan"
    echo "  5. Baca file"
    echo "  6. GitHub repository log"
    echo "  7. Kalkulator"
    echo "  8. System monitoring"
    echo "  9. TIMER!"
    echo "  10. Give me a motivation!"
    echo "  11. ^CUSTOM COMMAND^"
    echo "------------------------------------------------------"
}
```

Di situ saya sudah membagi-bagi beberapa fitur ke dalam beberapa fungsi agar kode bisa lebih dimanajemen dengan baik.

```
show_menu() {
    echo "================================================="
    echo "               --[ Jam : $(date +%T) ]--         "
    echo $'\033[0;32m-- Selamat datang di menu serbaguna komputer anda -- \033[0m'
    echo "-------------------------------------------------"
    echo "Silakan pilih salah satu menu berikut"
    echo "  1. Lihat path direktori saat ini"
    echo "  2. Membuat folder baru"
    echo "  3. Cari sebuah file"
    echo "  4. Buat catatan"
    echo "  5. Baca file"
    echo "  6. GitHub repository log"
    echo "  7. Kalkulator"
    echo "  8. System monitoring"
    echo "  9. TIMER!"
    echo "  10. Give me a motivation!"
    echo "  11. ^CUSTOM COMMAND^"
    echo "-------------------------------------------------"
}

make_dir() {
    read -p "Masukkan nama folder baru: " namaFolder
    if mkdir ../sample/$namaFolder; then
        echo -e "Folder \033[0;33m$namaFolder\033[0m berhasil dibuat."
    else
        echo -e "Gagal membuat folder \033[0;33mT$namaFolder\033[0m. Folder mungkin sudah ada."
    fi
}

search_file() {
    read -p "Masukkan nama file yang ingin dicari: " fileToSearch
    filePath=$(find / -type f -name "$fileToSearch" -print -quit 2> /dev/null)
    if [ -n "$filePath" ]; then
        echo -e "\n\033[0;32mFile ditemukan !\033[0m"
        echo "$filePath"
    else
        echo -e "\033[0;32mFile tidak dapat ditemukan !\033[0m"
    fi
}

make_note() {
    read -p "Masukkan nama file: " fileName
    echo "Masukkan catatan (Ctrl + D to exit)"
```

Rincian penjelasan fitur:

1. Melihat direktori saat ini

```
    read -p " >>> " pilihan

    case $pilihan in
        1)
            pwd
            ;;
        2)
```

```
==================================================
              --[ Jam : 10:20:41 ]--
-- Selamat datang di menu serbaguna komputer anda --
--------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. Kalkulator
  7. Give me a motivation!
--------------------------------------------------
 >>> 1
/mnt/d/03-STORAGE/03-GitHub/01-REPOSITORY/UAS-SISTEMOPERASI/src

--------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

Untuk melihat alamat direktori saat ini secara lengkap, saya menggunakan command **pwd** (print working directory). Di sini saya menggunakan WSL di Windows 11, jadinya alamat direktori diawali dengan `**/mnt/**`.

2. Membuat folder baru

```bash
make_dir() {
    read -p "Masukkan nama folder baru: " namaFolder
    if mkdir $namaFolder; then
        echo -e "Folder \033[0;33m$namaFolder\033[0m berhasil dibuat."
    else
        echo -e "Gagal membuat folder \033[0;33mT$namaFolder\033[0m. Folder mungkin sudah ada."
    fi
}
```

```
==================================================
              --[ Jam : 10:25:43 ]--
-- Selamat datang di menu serbaguna komputer anda --
--------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. Kalkulator
  7. Give me a motivation!
--------------------------------------------------
 >>> 2
Masukkan nama folder baru: folder-rahasia
Folder folder-rahasia berhasil dibuat.

--------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

Untuk membuat sebuah folder baru, maka di sini saya menggunakan command **mkdir** (make directory). Pembuatan folder ini berdasarkan pada nama variabel `**namaFolder**` yang diinputkan pengguna. Untuk memastikan

bahwa folder sudah berhasil dibuat, maka dibuat sebuah pengecekan kondisi jika folder sudah berhasil dibuat dengan memanfaatkan statement **if-else**.

3. Mencari sebuah file

```
search_file() {
    read -p "Masukkan nama file yang ingin dicari: " fileToSearch
    filePath=$(find / -type f -name "$fileToSearch" -print -quit 2> /dev/null)
    if [ -n "$filePath" ]; then
        echo -e "\n\033[0;32mFile ditemukan !\033[0m"
        echo "$filePath"
    else
        echo -e "\033[0;32mFile tidak dapat ditemukan !\033[0m"
    fi
}
```

```
==================================================
              --[ Jam : 11:07:30 ]--
-- Selamat datang di menu serbaguna komputer anda --
--------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. Kalkulator
  7. Give me a motivation!
--------------------------------------------------
 >>> 3
Masukkan nama file yang ingin dicari: note2.txt

File ditemukan !
/mnt/d/03-STORAGE/03-GitHub/01-REPOSITORY/UAS-SISTEMOPERASI/sample/notes/note2.txt

--------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

Kode utama yang saya buat tersebut berada pada baris **filePath=$(find / -type f -name “$fileToSearch” -print -quit 2> /dev/null).** Kode tersebut akan mencari sebuah file dari input pengguna. `find / -type f` berarti akan mencari sebuah file di seluruh direktori perangkat pengguna. Kemudian `-print -quit` akan menghentikan proses pencarian ketika sudah ditemukan dan menyimpan alamat file di variabel **filePath**.

Untuk memastikan bahwa file sudah ditemukan, maka perlu dilakukan pengecekan kondisi apakah variabel **filePath** kosong atau tidak pada kode berikut: -n “$filePath”.

4. Membuat catatan

```
make_note() {
    read -p "Masukkan nama file: " fileName
    echo "Masukkan catatan (Ctrl + D to exit)"
    echo "----------------------------------------------------"
    cat - > ../sample/notes/$fileName
    echo "----------------------------------------------------"
    if [ $? -eq 0 ]; then
        echo "Catatan berhasil dibuat!"
    else
        echo "Error: catatan gagal dibuat!"
    fi
}
```

```
====================================================
               --[ Jam : 11:21:58 ]--
-- Selamat datang di menu serbaguna komputer anda --
----------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. Kalkulator
----------------------------------------------------
 >>> 4
Masukkan nama file: secret.txt
Masukkan catatan (Ctrl + D to exit)
----------------------------------------------------
Nothing. Keep it secret!
----------------------------------------------------
Catatan berhasil dibuat!

----------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

Di sini saya membuat fitur catatan agar pengguna bisa menyimpan catatan-catatan penting dengan mudah. Untuk menginputkan teks catatan, hanya diperlukan input keyboard di terminal. Hasilnya nanti akan disimpan pada direktori ../sample/notes/$fileName.

Untuk memastikan bahwa file sudah berhasil dibuat, maka bisa dilakukan dengan mengecek system exit dari proses yang baru saja dibuat, yaitu dengan menggunakan **$? -eq 0**. Jika file berhasil dibuat, maka status system exit hasilnya 0 dan akan mencetak output “Catatan berhasil dibuat!”. Jika file tidak berhasil dibuat, maka akan mencetak “Error: catatan gagal dibuat”.

5. Membaca sebuah file

```
read_file() {
    echo "Daftar file catatan yang sudah dibuat:"
    ls ../sample/notes > ls_output.txt

    while IFS= read -r line; do
        echo -e "\e[33m$line\e[0m"
    done < ls_output.txt

    read -p "Masukkan catatan yang ingin dibaca: " fileToRead
    cat ../sample/notes/$fileToRead
    rm ls_output.txt
}
```

```
=====================================================
               --[ Jam : 11:27:06 ]--
-- Selamat datang di menu serbaguna komputer anda --
-----------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. Kalkulator
-----------------------------------------------------
 >>> 5
Daftar file catatan yang sudah dibuat:
myNotes.txt
note2.txt
secret.txt
Masukkan catatan yang ingin dibaca: myNotes.txt
Halo, kawanku!
Aku adalah seorang pelaut!
Bagaimana kabarmu?

-----------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

File yang bisa dibaca ini adalah file yang ada di folder **sample/notes**. Di sini saya akan menyimpan output dari ls folder sample/notes ke sebuah file txt. Hasil yang ada di file txt akan saya tampilkan ke terminal dan beri warna kuning. Dari situ, pengguna bisa memilih file mana yang akan dilihat dengan memasukkan nama file yang sudah terdaftar. Untuk membaca file bisa menggunakan command **cat**.

6. Git commit log

```bash
show_git_log() {
    echo "=================================================="
    # Alamat direktori disimpan di dalam sebuah variabel
    parentClonedGithubRepoPath=""
    if [ ! -d "$parentClonedGithubRepoPath" ]; then
        echo "Masukkan alamat parent kloningan repositori GitHub: "
        read parentClonedGithubRepoPath
    fi

    echo "Kloningan repositori yang anda miliki: "
    # Cek apakah subdirektori dari parent adalah kloningan github repository
    # Kemudian akan dicetak
    for dir in "$parentClonedGithubRepoPath"/*/ ; do
        if [ -d "$dir/.git" ]; then
            echo -e "\033[1;37;43m $(basename "$dir") \033[0m"
        fi
    done

    echo "--------------------------------------------------"
    read -p "Pilih direktori: " selectedDirectory
    # Melakukan pengecekan tracking git commit yang telah dilakukan
    git --git-dir="$parentClonedGithubRepoPath/$selectedDirectory/.git" log --oneline
    echo "--------------------------------------------------"
    read -p "Simpan pesan commit (y/n)? " isWantToSaveCommitLog
    # Menyimpan hasil dari git log ke dalam sebuah file
    if [[ "$isWantToSaveCommitLog" == "y"  ||  "$isWantToSaveCommitLog" == "Y" ]]; then
        read -p "Nama file: " fileGitlogName
        git --git-dir="$parentClonedGithubRepoPath/$selectedDirectory/.git" log --oneline > ../sample/gitlog/$fileGitlogNam
        if [ $? -eq 0  ]; then
            echo "Git commit log telah berhasil dibuat!"
        else
            echo "Git commit log gagal dibuat!"
        fi
    fi
}
```

```
==================================================
               --[ Jam : 08:00:42 ]--
-- Selamat datang di menu serbaguna komputer anda --
--------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. GitHub repository log
  7. Kalkulator
  8. System monitoring
  9. TIMER!
  10. Give me a motivation!
--------------------------------------------------
 >>> 6
==================================================
Masukkan alamat parent kloningan repositori GitHub:
/mnt/d/03-STORAGE/03-GitHub/01-REPOSITORY
Kloningan repositori yang anda miliki:
 AlgoritmaDanStrukturData
 Belajar-GitHub
 Campus-Coder-MEVN
 Change-Languange-Feature-JAVA
 Complicated-IF-JAVA
 Cuy-University-5-Python
 Daspro-Project-ATM
 DicodingxIDCamp-Front-End-Web
 FarrelAD
 Ngampus-POLINEMA
 Ngekost-Aja
 Ngekost-aja [Not active]
```

Pada fitur ini saya mencoba untuk mengkombinasikan mekanisme yang ada di Git, yaitu bisa melihat log/riwayat commit sebelumnya. Command yang saya gunakan sebenarnya hanya seperti ini: **git log –oneline**. Namun, saya ingin bisa mengecek commit log meski tidak berada di direktori tersebut. Maka dari itu, saya menggunakan perintah berikut: **git –git-dir=/path/directory/.git log –oneline**.

Di sini saya juga menambahkan opsi untuk menyimpan commit log tersebut ke dalam sebuah file atau tidak. Di sini mekanismenya adalah menggunakan **redirection output** dari proses pemanggilan git commit log sebelumnya ke sebuah file yang dikehendaki pengguna. File yang disimpan akan berada di folder sample/gitlog.

7. Kalkulator sederhana

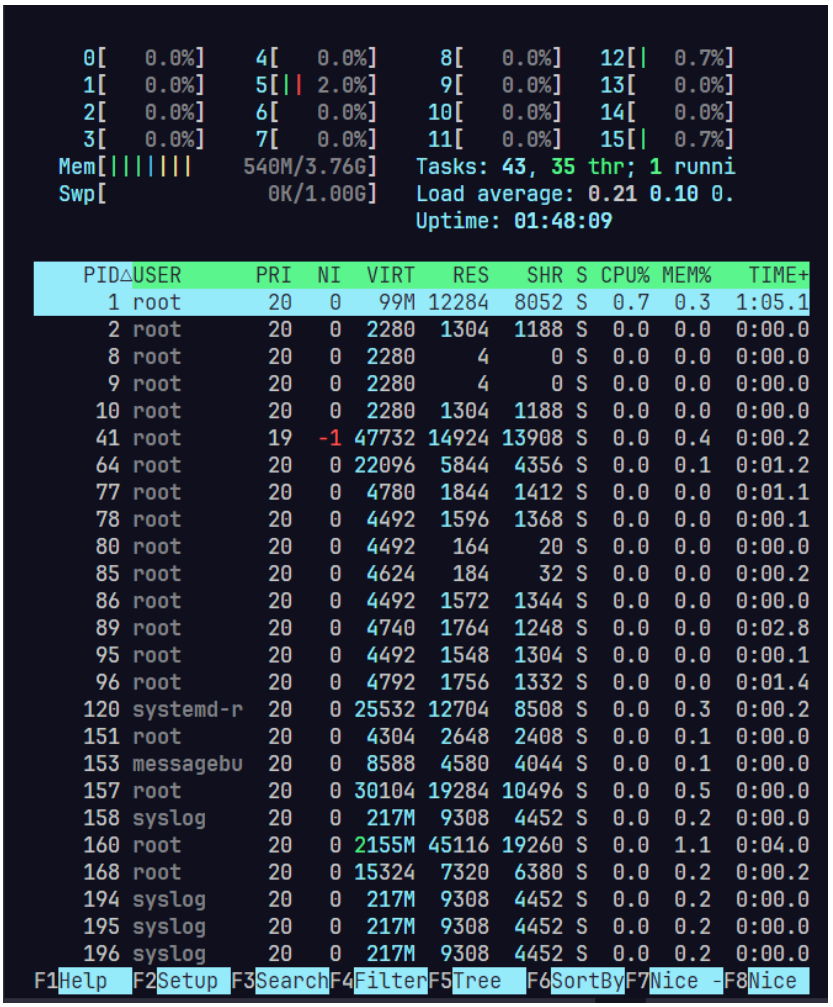
```
calculator() {
    echo -e "\n=================================================="
    echo "Pilih opsi menu"
    echo "--------------------------------------------------"
    echo "  1. Penjumlahan"
    echo "  2. Pengurangan"
    echo "  3. Perkalian"
    echo "  4. Pembagian"
    read -p " >>> " pilihan

    read -p "Masukkan bilangan pertama: " number1
    read -p "Masukkan bilangan kedua: " number2

    unset result
    case $pilihan in
        1)
            result=$(($number1 + $number2))
            ;;
        2)
            result=$(($number1 - $number2))
            ;;
        3)
            result=$(($number1 * $number2))
            ;;
        4)
            result=$(awk "BEGIN { print $number1 / $number2 }")
            ;;
        *)
            echo "Pilihan tidak valid!"
            ;;
    esac
    echo -e "Hasilnya: \033[0;33m$result\033[0m"
}
```

```
==================================================
               --[ Jam : 11:35:14 ]--
-- Selamat datang di menu serbaguna komputer anda --
--------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. Kalkulator
--------------------------------------------------
 >>> 6

==================================================
Pilih opsi menu
--------------------------------------------------
  1. Penjumlahan
  2. Pengurangan
  3. Perkalian
  4. Pembagian
 >>> 4
Masukkan bilangan pertama: 1928
Masukkan bilangan kedua: 34
Hasilnya: 56.7059

--------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

Kalkulator di sini bisa digunakan untuk melakukan operasi aritmatika sederhana penjumlahan, pengurangan, perkalian, dan pembagian. Ketika pengguna memilih opsi menu ke-6, maka akan diarahkan ke fungsi **calculator**. Untuk operasi penjumlahan, pengurangan, dan perkalian bisa langsung menggunakan tanda (+, -, *). Sedangkan untuk pembagian agak berbeda. Agar hasil dari pembagian bisa lebih presisi, maka saya menggunakan command **awk**. Command tersebut akan melakukan operasi aritmatika dan hasil yang didapat bisa berupa **float**/ bilangan desimal.

8. Sistem monitoring

```
    ;;
8)
    htop
    ;;
```



Perintah yang digunakan cukup sederhana saja, yaitu menggunakan perintah bawaan dari Linux, yaitu **htop**. htop artinya: Hisham table of process. Hisham adalah pembuat dari perintah ini. Data yang ditampilkan di htop adalah statistik penggunaan komputer saat itu. Contohnya adalah status memory, penggunaan CPU, proses yang dijalankan dan masih banyak lagi.

9. Timer

```
run_timer() {
    read -p "Set timer (detik): " timerSecondSet
    read -p "Pesan timer ketika selesai: " timerMessage

    start_timer() {
        sleep $timerSecondSet
        echo -e "\n=======================TIMER=======================\033[0;33m$timerMessage\033[0m"
        echo "==================================================="
    }

    start_timer &
    echo "Timer dimulai selama $timerSecondSet detik. PID: $!"
}
```

```
=====================================================
                --[ Jam : 18:25:09 ]--
-- Selamat datang di menu serbaguna komputer anda --
-----------------------------------------------------
Silakan pilih salah satu menu berikut
  1. Lihat path direktori saat ini
  2. Membuat folder baru
  3. Cari sebuah file
  4. Buat catatan
  5. Baca file
  6. Kalkulator
  7. System monitoring
  8. TIMER!
  9. Give me a motivation!
-----------------------------------------------------
 >>> 8
Set timer (detik): 5
Pesan timer ketika selesai: It's time to sleep!
Timer dimulai selama 5 detik. PID: 33986

-----------------------------------------------------
Ingin kembali ke menu? (y/n):
=======================TIMER=======================
It's time to sleep!
=====================================================
```

Fitur timer ini memanfaatkan teknik **background process** yang ada di Linux. Ini membuat program yang seharusnya muncul di terminal bisa dijalankan di belakang layar. Sehingga pengguna bisa melakukan aktivitas lain sembari menunggu proses background selesai.

Untuk detail teknisnya adalah pengguna dimungkinkan untuk input seberapa lama timer akan berjalan dengan format detik. Kemudian pesan yang ingin ditambahkan saat timer selesai. Ketika itu semua selesai, maka program timer akan dijalankan. Untuk menjalankan prosesnya di background bisa menggunakan **“&”**.

10. Custom command Linux

```bash
create_custom_command() {
    shopt -s expand_aliases

    read -p "Masukkan command yang ingin digunakan: " usedCommand
    read -p "Masukkan alias: " aliasCommand
    echo "-----------------------------------------------------"

    # Menambahkan custom command ke ~/.bashrc agar ketika program berhenti
    # Custom command tetap bisa dijalankan
    echo "alias $aliasCommand='$usedCommand'" >> ~/.bashrc
    source ~/.bashrc

    # Mengecek apakah alias sudah benar-benar dibuat
    if grep -q "^alias $aliasCommand='$usedCommand'" ~/.bashrc; then
        echo "Custom command telah sukses dibuat!"
    else
        echo "Custom command gagal dibuat"
    fi
}
```

```
=====================================================
  1. Buat custom command
  2. Lihat custom command yang telah dibuat
  3. Hapus custom command
 >>> 1
-----------------------------------------------------
Masukkan command yang ingin digunakan: clear
Masukkan alias: c
Custom command telah sukses dibuat!

-----------------------------------------------------
Ingin kembali ke menu? (y/n): y_
```

```bash
show_custom_command() {
    echo "Berikut adalah custom command yang telah anda buat:"
    grep "^alias" ~/.bashrc
}
```

```
=====================================================
  1. Buat custom command
  2. Lihat custom command yang telah dibuat
  3. Hapus custom command
 >>> 2
-----------------------------------------------------
Berikut adalah custom command yang telah anda buat:
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo term
inal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*
//;s/[;&|]\s*alert$//'\'')"'
alias c='clear'

-----------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

```bash
delete_custom_command() {
    read -p "Masukkan command yang ingin dihapus: " deletedCommand
    read -p "Masukkan alias yang ingin dihapus: " deletedAlias
    echo "------------------------------------------------"

    if grep -q "^alias $deletedAlias='$deletedCommand'" ~/.bashrc; then
        # Menghapus custom command yang telah dibuat
        sed -i "/^alias $deletedAlias='$deletedCommand'/d" ~/.bashrc
        echo "Alias '$deletedAlias' untuk command '$deletedCommand' telah dihapus
        ~/.bashrc."
    else
        echo "Alias '$deletedAlias' untuk command '$deletedCommand' tidak
        ditemukan di  ~/.bashrc."
    fi

    source ~/.bashrc
}
```

```
======================================================
  1. Buat custom command
  2. Lihat custom command yang telah dibuat
  3. Hapus custom command
 >>> 3
------------------------------------------------------
Masukkan command yang ingin dihapus: clear
Masukkan alias yang ingin dihapus: c
Alias 'c' untuk command 'clear' telah dihapus ~/.bashrc.

------------------------------------------------------
Ingin kembali ke menu? (y/n): _
```

Di sini saya membuat 3 subfitur yang tersedia di bagian custom command Linux. Pengguna bisa melakukan penambahan custom command baru, melihat daftar custom command yang telah dibuat, dan yang terakhir adalah menghapus custom command yang telah dibuat. Secara umum mekanismenya adalah menggunakan perintah **alias**. Pengguna hanya perlu menginputkan nama command yang ada dan juga nama alias/custom command ketika dijalankan. Keseluruhan itu akan disimpan di file **.bashrc** agar custom command bisa dijalankan meski program shell script dihentikan.

BAB III
PENUTUP

A. KESIMPULAN

Secara keseluruhan program yang saya buat berjalan dengan baik. Fitur yang saya buat sejumlah 10 saya kira sudah cukup untuk membantu produktivitas bagi programmer secara mendasar. Fitur yang paling menarik bagi saya adalah git commit log dan juga custom command. Dari situ, saya belajar program shell script dengan lebih kompleks. Apalagi mengkombinasikan dengan git yang belum pernah saya temui sebelumnya. Saya juga berusaha untuk membuat kode program dengan lebih rapi dengan membagi beberapa proses ke dalam fungsi.

Tantangan terbesar adalah memikirkan fitur-fitur yang bakal disediakan oleh program ini. Bagaimana caranya membuat sebuah fitur yang juga tak hanya memanfaatkan mekanisme shell scripting dan juga bermanfaat adalah hal yang cukup sulit.

Memang fitur-fitur tersebut tidak sepenuhnya meningkatkan produktivitas programmer, tapi setidaknya dari sini bisa belajar bahwa saya bisa membuat sebuah program yang berguna dengan memanfaatkan mekanisme shell scripting. Apa yang telah saya kerjakan dalam proyek ini adalah kemampuan yang saya bisa dari materi-materi sistem operasi yang telah dipelajari dan dari internet. Ini adalah program pertama saya di dalam shell scripting sehingga mungkin memang dirasa banyak kesalahan. Untuk itu, saya menerima kritik dan saran yang membangun dari pembaca.