

LAPORAN RESMI
UJIAN AKHIR SEMESTER
PRAKTIKUM SISTEM OPERASI



NIM : 2341720247
Nama : Tiara Mera Sifa
Jurusan : 2341720247
Kelas : TI-1B

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

DESKRIPSI PROGRAM

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
█
```

MeraFS adalah sebuah program manajemen file sederhana yang ditulis dalam Bahasa pemrograman bash untuk digunakan di system operasi berbasis Linux. Program ini memungkinkan pengguna untuk melakukan operasi manajemen file dan direktori dengan mudah. Nama MeraFS diambil dari nama saya, yaitu Mera dan FS merupakan file system.

Terdapat total 10 fitur pada program MeraFS, yaitu:

1. **Buat Direktori**
Fitur ini menggunakan perintah `mkdir` untuk membuat direktori baru. Program meminta pengguna untuk memasukkan nama direktori yang ingin dibuat, lalu membuat direktori tersebut. Misalnya, jika pengguna memasukkan `documents`, direktori `documents` akan dibuat menggunakan `mkdir documents`.
2. **Buat File**
Fitur ini menggunakan perintah `touch` untuk membuat file kosong baru. Program meminta pengguna untuk memasukkan nama file yang ingin dibuat. Misalnya, jika pengguna memasukkan `notes.txt`, file `notes.txt` akan dibuat menggunakan `touch notes.txt`.
3. **Salin File**
Fitur ini menggunakan perintah `cp` untuk menyalin file. Program meminta pengguna untuk memasukkan nama file sumber dan nama file tujuan, kemudian menyalin file sumber ke file tujuan. Misalnya, jika pengguna memasukkan `document.txt` sebagai file sumber dan `document_copy.txt` sebagai file tujuan, perintah `cp document.txt document_copy.txt` akan dijalankan.
4. **Hapus File**
Fitur ini menggunakan perintah `rm` untuk menghapus file. Program meminta pengguna untuk memasukkan nama file yang ingin dihapus. Jika file tersebut ada, file akan dihapus. Misalnya, jika pengguna memasukkan `old_file.txt`, file `old_file.txt` akan dihapus menggunakan `rm old_file.txt`.

5. Baca File

Fitur ini menggunakan perintah `cat` untuk menampilkan isi file di terminal. Program meminta pengguna untuk memasukkan nama file yang ingin dibaca. Jika file tersebut ada, isi file akan ditampilkan. Misalnya, jika pengguna memasukkan `document.txt`, isi dari file `document.txt` akan ditampilkan menggunakan `cat document.txt`.

6. Lihat Isi Direktori

Fitur ini menggunakan perintah `ls` untuk menampilkan isi direktori. Program meminta pengguna untuk memasukkan nama direktori yang ingin dilihat isinya. Jika direktori tersebut ada, daftar file dan sub-direktori di dalamnya akan ditampilkan. Misalnya, jika pengguna memasukkan `documents`, isi dari direktori `documents` akan ditampilkan menggunakan `ls documents`.

7. Ubah Nama File

Fitur ini menggunakan perintah `mv` untuk mengganti nama file. Program meminta pengguna untuk memasukkan nama file lama dan nama file baru. Jika file lama ada, nama file tersebut akan diubah menjadi nama file baru. Misalnya, jika pengguna memasukkan `old_name.txt` sebagai nama file lama dan `new_name.txt` sebagai nama file baru, nama file `old_name.txt` akan diubah menjadi `new_name.txt` menggunakan `mv old_name.txt new_name.txt`.

8. Pindah File

Fitur ini menggunakan perintah `mv` untuk memindahkan file ke direktori lain. Program meminta pengguna untuk memasukkan nama file yang ingin dipindahkan dan nama direktori tujuan. Jika file dan direktori tujuan ada, file akan dipindahkan. Misalnya, jika pengguna memasukkan `file.txt` sebagai nama file dan `documents` sebagai direktori tujuan, file `file.txt` akan dipindahkan ke dalam direktori `documents` menggunakan `mv file.txt documents/`.

9. Ubah Izin File

Fitur ini menggunakan perintah `chmod` untuk mengubah izin akses file atau direktori. Program meminta pengguna untuk memasukkan nama file atau direktori dan izin baru dalam format numerik (misalnya 755). Izin file atau direktori tersebut akan diubah. Misalnya, jika pengguna memasukkan `script.sh` sebagai nama file dan `755` sebagai izin baru, izin file `script.sh` akan diubah menjadi `755` menggunakan `chmod 755 script.sh`.

10. Keluar

Fitur ini menggunakan perintah `exit` untuk menghentikan eksekusi program dan keluar dari sesi terminal. Ketika pengguna memilih opsi keluar, program akan menampilkan pesan "Terima kasih telah menggunakan program manajemen file." dan mengakhiri program.

PENJELASAN PROGRAM

1. Buat Direktori

```
# Membuat direktori baru
create_directory() {
    echo "Masukkan nama direktori baru:"
    read directory_name
    mkdir "$directory_name"
    echo "Direktori '$directory_name' telah berhasil dibuat."
}
```

Fitur ini menggunakan perintah `mkdir` untuk membuat direktori baru. Program meminta pengguna untuk memasukkan nama direktori yang ingin dibuat melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama direktori, program membaca input tersebut dan menyimpannya dalam variabel `directory_name`. Kemudian, perintah `mkdir` dijalankan dengan nama direktori yang diberikan pengguna. Jika pengguna memasukkan nama `documents`, direktori `documents` akan dibuat menggunakan `mkdir documents`. Setelah itu, program menampilkan pesan konfirmasi bahwa direktori telah berhasil dibuat.

2. Buat File

```
# Membuat file baru
create_file() {
    echo "Masukkan nama file baru:"
    read file_name
    touch "$file_name"
    echo "File '$file_name' telah berhasil dibuat."
}
```

Fitur ini menggunakan perintah `touch` untuk membuat file kosong baru. Program meminta pengguna untuk memasukkan nama file yang ingin dibuat melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama file, program membaca input tersebut dan menyimpannya dalam variabel `file_name`. Kemudian, perintah `touch` dijalankan dengan nama file yang diberikan pengguna. Jika pengguna memasukkan nama `notes.txt`, file `notes.txt` akan dibuat menggunakan `touch notes.txt`. Setelah itu, program menampilkan pesan konfirmasi bahwa file telah berhasil dibuat.

3. Salin File

```
# Menyalin file
copy_file() {
    echo "Masukkan nama file yang akan disalin:"
    read source_file
    echo "Masukkan nama tujuan file yang baru:"
    read destination_file
    cp "$source_file" "$destination_file"
    echo "File '$source_file' telah berhasil disalin menjadi '$destination_file'."
}
```

Fitur ini menggunakan perintah `cp` untuk menyalin file. Program meminta pengguna untuk memasukkan nama file sumber dan nama file tujuan melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama file sumber dan tujuan, program

membaca input tersebut dan menyimpannya dalam variabel `source_file` dan `destination_file`. Kemudian, perintah `cp` dijalankan dengan nama file sumber dan tujuan yang diberikan pengguna. Jika pengguna memasukkan nama `document.txt` sebagai file sumber dan `document_copy.txt` sebagai file tujuan, perintah `cp document.txt document_copy.txt` akan dijalankan untuk menyalin file tersebut. Setelah itu, program menampilkan pesan konfirmasi bahwa file telah berhasil disalin.

4. Hapus File

```
# Menghapus file
delete_file() {
    echo "Masukkan nama file yang akan dihapus:"
    read file_name

    if [ -f "$file_name" ]; then
        rm "$file_name"
        echo "File '$file_name' telah berhasil dihapus."
    else
        echo "File '$file_name' tidak ditemukan."
    fi
}
```

Fitur ini menggunakan perintah `rm` untuk menghapus file. Program meminta pengguna untuk memasukkan nama file yang ingin dihapus melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama file, program membaca input tersebut dan menyimpannya dalam variabel `file_name`. Kemudian, program memeriksa apakah file dengan nama yang diberikan ada. Jika file ada, perintah `rm` dijalankan untuk menghapus file tersebut. Jika pengguna memasukkan nama `old_file.txt`, file `old_file.txt` akan dihapus menggunakan `rm old_file.txt`. Setelah itu, program menampilkan pesan konfirmasi bahwa file telah berhasil dihapus. Jika file tidak ditemukan, program menampilkan pesan bahwa file tidak ditemukan.

5. Baca File

```
# Membaca isi file
read_file() {
    echo "Masukkan nama file yang akan dibaca:"
    read file_name

    if [ -f "$file_name" ]; then
        echo "Isi dari file '$file_name':"
        cat "$file_name"
    else
        echo "File '$file_name' tidak ditemukan."
    fi
}
```

Fitur ini menggunakan perintah `cat` untuk menampilkan isi file di terminal. Program meminta pengguna untuk memasukkan nama file yang ingin dibaca melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama file, program membaca input tersebut dan menyimpannya dalam variabel `file_name`. Kemudian, program

memeriksa apakah file dengan nama yang diberikan ada. Jika file ada, perintah `cat` dijalankan untuk menampilkan isi file tersebut. Jika pengguna memasukkan nama `document.txt`, isi dari file `document.txt` akan ditampilkan menggunakan `cat document.txt`. Setelah itu, program menampilkan pesan bahwa isi file telah berhasil ditampilkan. Jika file tidak ditemukan, program menampilkan pesan bahwa file tidak ditemukan.

6. Lihat Isi Direktori

```
# Melihat isi direktori
list_directory() {
    echo "Masukkan nama direktori yang akan dilihat isinya:"
    read directory_name

    if [ -d "$directory_name" ]; then
        echo "Isi dari direktori '$directory_name':"
        ls "$directory_name"
    else
        echo "Direktori '$directory_name' tidak ditemukan."
    fi
}
```

Fitur ini menggunakan perintah `ls` untuk menampilkan isi direktori. Program meminta pengguna untuk memasukkan nama direktori yang ingin dilihat isinya melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama direktori, program membaca input tersebut dan menyimpannya dalam variabel `directory_name`. Kemudian, program memeriksa apakah direktori dengan nama yang diberikan ada. Jika direktori ada, perintah `ls` dijalankan untuk menampilkan isi direktori tersebut. Jika pengguna memasukkan nama `documents`, isi dari direktori `documents` akan ditampilkan menggunakan `ls documents`. Setelah itu, program menampilkan pesan bahwa isi direktori telah berhasil ditampilkan. Jika direktori tidak ditemukan, program menampilkan pesan bahwa direktori tidak ditemukan.

7. Ubah Nama File

```
# Mengubah nama file
rename_file() {
    echo "Masukkan nama file yang akan diubah:"
    read old_name
    echo "Masukkan nama file baru:"
    read new_name

    if [ -f "$old_name" ]; then
        mv "$old_name" "$new_name"
        echo "File '$old_name' telah berhasil diubah namanya menjadi '$new_name'."
    else
        echo "File '$old_name' tidak ditemukan."
    fi
}
```

Fitur ini menggunakan perintah `mv` untuk mengganti nama file. Program meminta pengguna untuk memasukkan nama file lama dan nama file baru melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama file lama dan baru, program membaca input tersebut dan menyimpannya dalam variabel `old_name` dan

`new_name`. Kemudian, program memeriksa apakah file lama dengan nama yang diberikan ada. Jika file lama ada, perintah `mv` dijalankan untuk mengganti nama file lama menjadi nama file baru. Jika pengguna memasukkan nama `old_name.txt` sebagai nama file lama dan `new_name.txt` sebagai nama file baru, perintah `mv old_name.txt new_name.txt` akan dijalankan untuk mengganti nama file tersebut. Setelah itu, program menampilkan pesan konfirmasi bahwa nama file telah berhasil diubah. Jika file lama tidak ditemukan, program menampilkan pesan bahwa file lama tidak ditemukan.

8. Pindah File

```
# Memindahkan file
move_file() {
    echo "Masukkan nama file yang akan dipindahkan:"
    read file_name
    echo "Masukkan direktori tujuan:"
    read destination_directory

    if [ -f "$file_name" ]; then
        if [ -d "$destination_directory" ]; then
            mv "$file_name" "$destination_directory"
            echo "File '$file_name' telah berhasil dipindahkan ke direktori '$destination_directory'"
        else
            echo "Direktori '$destination_directory' tidak ditemukan."
        fi
    else
        echo "File '$file_name' tidak ditemukan."
    fi
}
```

Fitur ini menggunakan perintah `mv` untuk memindahkan file ke direktori lain. Program meminta pengguna untuk memasukkan nama file yang ingin dipindahkan dan nama direktori tujuan melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama file dan direktori tujuan, program membaca input tersebut dan menyimpannya dalam variabel `file_name` dan `destination_directory`. Kemudian, program memeriksa apakah file dengan nama yang diberikan ada. Jika file ada, program memeriksa apakah direktori tujuan dengan nama yang diberikan ada. Jika direktori tujuan ada, perintah `mv` dijalankan untuk memindahkan file ke direktori tujuan. Jika pengguna memasukkan nama `file.txt` sebagai nama file dan `documents` sebagai direktori tujuan, perintah `mv file.txt documents/` akan dijalankan untuk memindahkan file tersebut. Setelah itu, program menampilkan pesan konfirmasi bahwa file telah berhasil dipindahkan. Jika file atau direktori tujuan tidak ditemukan, program menampilkan pesan yang sesuai.

9. Ubah Izin File

```
# Mengubah izin file
change_permissions() {
    echo "Masukkan nama file atau direktori yang izinnya akan diubah:"
    read file_name
    echo "Masukkan izin baru (misalnya 755):"
    read permissions

    if [ -e "$file_name" ]; then
        chmod "$permissions" "$file_name"
        echo "Izin untuk '$file_name' telah diubah menjadi '$permissions'."
    else
        echo "File atau direktori '$file_name' tidak ditemukan."
    fi
}
```

Fitur ini menggunakan perintah `chmod` untuk mengubah izin akses file atau direktori. Program meminta pengguna untuk memasukkan nama file atau direktori dan izin baru dalam format numerik (misalnya 755) melalui pesan yang ditampilkan. Setelah pengguna memasukkan nama file atau direktori dan izin baru, program membaca input tersebut dan menyimpannya dalam variabel `file_name` dan `permissions`. Kemudian, program memeriksa apakah file atau direktori dengan nama yang diberikan ada. Jika file atau direktori ada, perintah `chmod` dijalankan untuk mengubah izin file atau direktori tersebut. Jika pengguna memasukkan nama `script.sh` sebagai nama file dan `755` sebagai izin baru, perintah `chmod 755 script.sh` akan dijalankan untuk mengubah izin file tersebut. Setelah itu, program menampilkan pesan konfirmasi bahwa izin telah berhasil diubah. Jika file atau direktori tidak ditemukan, program menampilkan pesan bahwa file atau direktori tidak ditemukan.

10. Keluar

```
10) echo "Terima kasih telah menggunakan program manajemen file." ; exit ;;
*) echo "Pilihan tidak valid. Silakan masukkan pilihan yang benar." ;;
```

Pilihan ini menangani proses keluar dari program manajemen file. Ketika pengguna memilih opsi ini, program menampilkan pesan "Terima kasih telah menggunakan program manajemen file." sebagai ucapan terima kasih kepada pengguna. Setelah itu, perintah `exit` dieksekusi untuk menghentikan eksekusi skrip dan keluar dari program. Pilihan ini memberikan pengguna cara untuk mengakhiri sesi mereka dengan program secara bersih dan teratur.

PENGUJIAN PROGRAM

1. Buat Direktori

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
1
Masukkan nama direktori baru:
documents
Direktori 'documents' telah berhasil dibuat.
```

Saat pengguna memasukkan nama direktori yang akan dibuat, program akan menampilkan pesan yang menyatakan bahwa direktori telah berhasil dibuat dengan nama yang sesuai dengan input pengguna.

2. Buat File

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
2
Masukkan nama file baru:
notes.txt
File 'notes.txt' telah berhasil dibuat.
```

Setelah pengguna memasukkan nama file yang akan dibuat, program akan memberikan konfirmasi bahwa file tersebut berhasil dibuat sesuai dengan nama yang telah diinput.

3. Salin File

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
3
Masukkan nama file yang akan disalin:
notes.txt
Masukkan nama tujuan file yang baru:
mera.txt
File 'notes.txt' telah berhasil disalin menjadi 'mera.txt'.
```

Setelah pengguna memasukkan nama file sumber dan tujuan, program akan memberikan pesan yang menyatakan bahwa file sumber telah berhasil disalin ke file tujuan yang baru.

4. Hapus File

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
4
Masukkan nama file yang akan dihapus:
mera.txt
File 'mera.txt' telah berhasil dihapus.
```

Jika nama file yang dimasukkan oleh pengguna ada, program akan memberikan pesan bahwa file tersebut telah berhasil dihapus. Namun, jika file tidak ditemukan, pesan yang ditampilkan adalah bahwa file tidak ditemukan.

5. Baca File

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
5
Masukkan nama file yang akan dibaca:
hello.txt
Isi dari file 'hello.txt':
saya mera oke
```

Setelah pengguna memasukkan nama file yang ingin dibaca, program akan menampilkan isi dari file tersebut. Namun, jika file tidak ditemukan, program akan memberikan pesan bahwa file tidak ditemukan.

6. Lihat Isi Direktori

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
6
Masukkan nama direktori yang akan dilihat isinya:
documents
Isi dari direktori 'documents':
helloworld.cpp
```

Setelah pengguna memasukkan nama direktori yang ingin dilihat isinya, program akan menampilkan daftar file yang ada di dalam direktori tersebut. Jika direktori tidak ditemukan, program akan memberikan pesan bahwa direktori tidak ditemukan.

7. Ubah Nama File

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
7
Masukkan nama file yang akan diubah:
uas.sh
Masukkan nama file baru:
UAS_TIARAMERAS.sh
File 'uas.sh' telah berhasil diubah namanya menjadi 'UAS_TIARAMERAS.sh'.
```

Setelah pengguna memasukkan nama file lama dan baru, program akan memberikan pesan bahwa file lama berhasil diubah namanya menjadi file baru. Namun, jika file lama tidak ditemukan, pesan yang ditampilkan adalah bahwa file tidak ditemukan.

8. Pindah File

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
8
Masukkan nama file yang akan dipindahkan:
hello.txt
Masukkan direktori tujuan:
documents
File 'hello.txt' telah berhasil dipindahkan ke direktori 'documents'.
```

Setelah pengguna memasukkan nama file dan direktori tujuan, program akan memberikan konfirmasi bahwa file telah berhasil dipindahkan ke direktori tujuan yang baru. Namun, jika file atau direktori tujuan tidak ditemukan, program akan memberikan pesan bahwa file atau direktori tidak ditemukan.

9. Ubah Izin File

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
9
Masukkan nama file atau direktori yang izinnya akan diubah:
documents
Masukkan izin baru (misalnya 755):
755
Izin untuk 'documents' telah diubah menjadi '755'.
```

Setelah pengguna memasukkan nama file atau direktori dan izin baru, program akan memberikan konfirmasi bahwa izin untuk file atau direktori telah berhasil diubah sesuai dengan input pengguna. Namun, jika file atau direktori tidak ditemukan, program akan memberikan pesan bahwa file atau direktori tidak ditemukan.

10. Keluar

```
===== MeraFS =====
===== Pilih Menu =====
1. Buat direktori
2. Buat file
3. Salin file
4. Hapus file
5. Baca file
6. Lihat isi direktori
7. Ubah nama file
8. Pindah file
9. Ubah izin file
10. Keluar
=====
Masukkan pilihan:
10
Terima kasih telah menggunakan program manajemen file.
```

Program akan menampilkan pesan "Terima kasih telah menggunakan program manajemen file." dan kemudian keluar dari program menggunakan perintah exit.