

Laporan Praktikum

Sistem Operasi

Modul 10



Nama : Asep haryana saputra

NIM : 20230810043

Kelas : TINFC-2023-04

Teknik Informatika
Fakultas Ilmu Komputer
Universitas Kuningan

Pretest

1. Jelaskan perbedaan antara struktur direktori satu tingkat dan struktur direktori dua tingkat. Sertakan keuntungan dan kelemahan dari masing-masing struktur.

- a. Struktur Direktori Satu Tingkat

Pengertian:

- Semua file disimpan dalam direktori utama, tanpa ada sub-direktori.

Keuntungan:

- Sederhana dan mudah dipahami.
- Tidak membutuhkan waktu lama untuk membuat struktur direktori.

Kelemahan:

- Sulit untuk mengorganisasikan file yang banyak.
- Sulit untuk menemukan file yang spesifik.
- Risiko kehilangan file lebih tinggi karena tidak terorganisir dengan baik.

2. Struktur Direktori Dua Tingkat

Pengertian:

- Memiliki direktori utama yang berisi sub-direktori yang mengorganisir file.

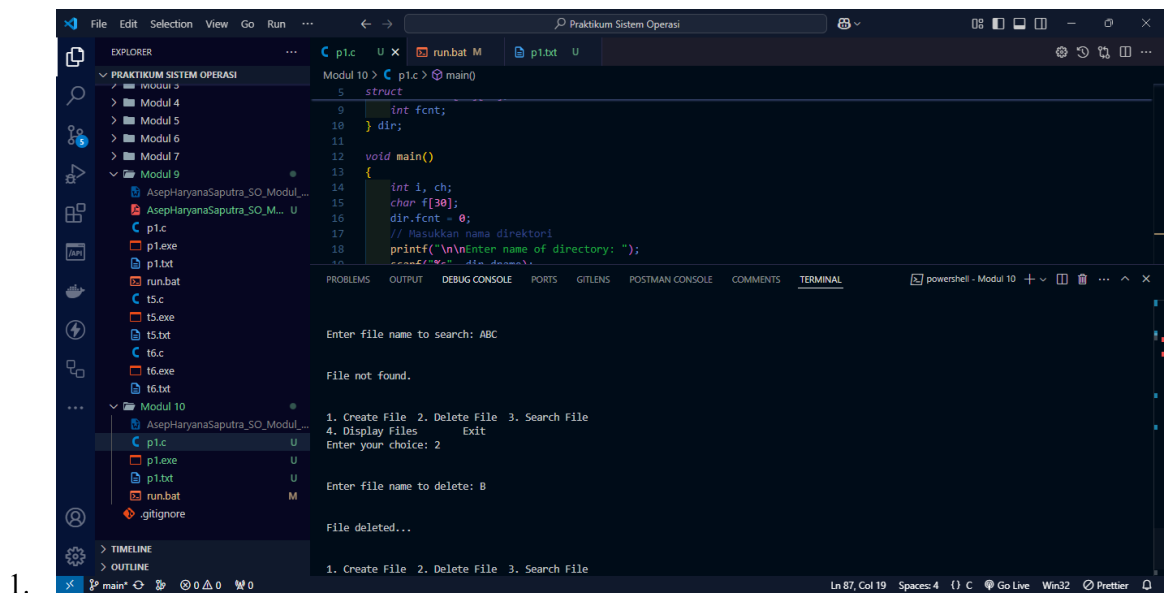
Keuntungan:

- Memudahkan pengorganisasian file yang banyak.
- Memudahkan pencarian file yang spesifik.
- Risiko kehilangan file lebih rendah karena lebih terorganisir.

Kelemahan:

- Lebih kompleks dibandingkan dengan struktur direktori satu tingkat.
- Membutuhkan waktu lebih lama untuk membuat struktur direktori.

Praktikum



1.

Kode:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%2010/p1.c>

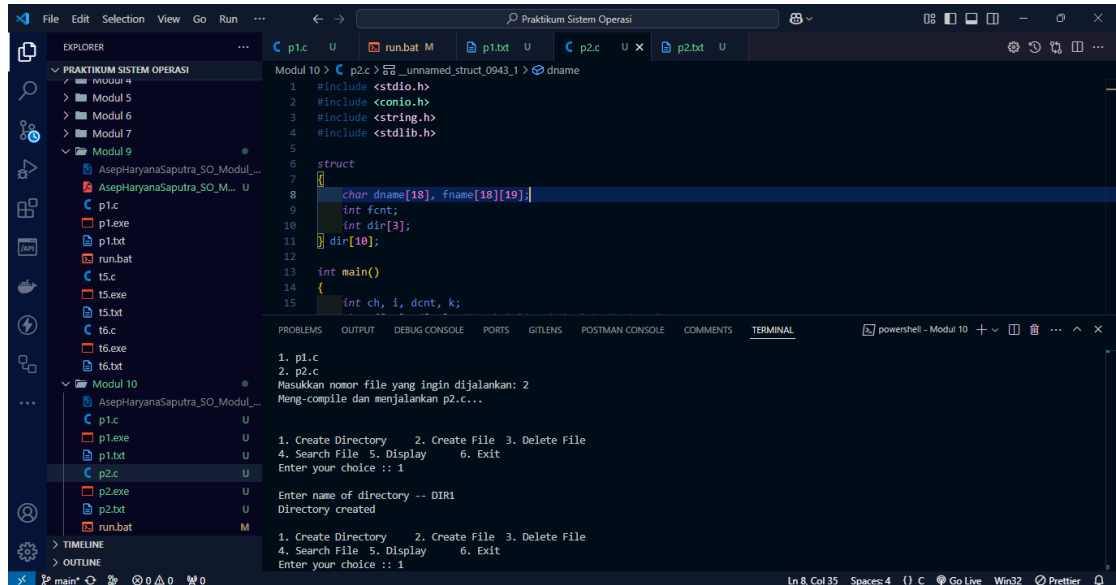
Log:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%2010/p1.txt>

Penjelasan:

Program ini bertujuan untuk mengelola file dalam sebuah direktori sederhana, termasuk operasi pembuatan, penghapusan, pencarian, dan tampilan daftar file. Pengguna dapat memasukkan nama direktori, kemudian melakukan operasi sesuai pilihan yang tersedia. Setiap file baru yang dibuat akan disimpan dalam array, file dapat dihapus berdasarkan nama, dan pencarian file dilakukan dengan mencocokkan nama file. Program juga memungkinkan untuk menampilkan seluruh file yang ada dalam direktori. Hasil yang diperoleh menunjukkan bagaimana pengguna dapat membuat file, mencari, menampilkan, dan menghapus file, serta menangani kasus ketika file tidak ditemukan, seperti yang terjadi saat mencoba mencari file "ABC" yang tidak ada, atau menghapus file "B".

2.



```
Modul 10 > C p2.c > 50 _unnamed_struct_0943_1 > dname
1 #include <stdio.h>
2 #include <conio.h>
3 #include <string.h>
4 #include <stdlib.h>
5
6 struct
7 {
8     char dname[18], fname[18][10];
9     int fcnt;
10    int dir[3];
11    dir[10];
12
13    int main()
14    {
15        int ch, i, dcnt, k;
```

1. p1.c 2. p2.c
Masukkan nomor file yang ingin dijalankan: 2
Meng-compile dan menjalankan p2.c...

1. Create Directory 2. Create File 3. Delete File
4. Search File 5. Display 6. Exit
Enter your choice :: 1
Enter name of directory -- DIR1
Directory created

1. Create Directory 2. Create File 3. Delete File
4. Search File 5. Display 6. Exit
Enter your choice :: 1

Kode:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%2010/p2.c>

Log:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%2010/p2.txt>

Penjelasan:

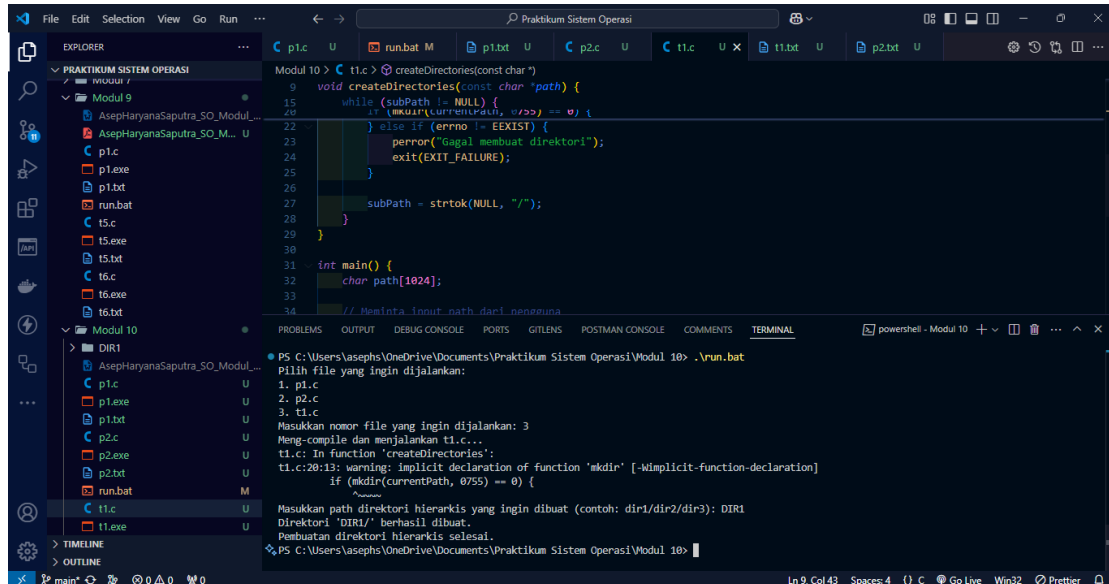
Program ini bertujuan untuk mengelola direktori dan file dalam sistem sederhana. Pengguna dapat membuat direktori, menambahkan file ke direktori tertentu, menghapus file dari direktori, mencari file, dan menampilkan daftar semua direktori beserta file di dalamnya. Program ini mencatat setiap operasi dan memberikan umpan balik jika file atau direktori tidak ditemukan. Hasil eksekusi menunjukkan bahwa pengguna berhasil membuat direktori "DIR1" dan "DIR2", menambahkan file "A1" dan "A2" ke "DIR1", serta file "B1" ke "DIR2". Ketika daftar direktori ditampilkan, program menunjukkan semua direktori dan file yang sesuai. Hal ini menggambarkan pengelolaan file dan direktori secara efisien dalam skenario simulasi.

Posttest

1.

Kedua program tersebut menunjukkan simulasi pengelolaan file dan direktori sederhana dengan operasi dasar seperti pembuatan, penghapusan, pencarian, dan penampilan daftar file atau direktori. Program pertama berfokus pada pengelolaan file dalam satu direktori, sedangkan program kedua mencakup pengelolaan beberapa direktori dengan file di dalamnya. Hasil eksekusi masing-masing program menegaskan kemampuan untuk menangani kasus-kasus seperti file yang tidak ditemukan, serta memberikan umpan balik yang jelas terhadap setiap operasi. Simulasi ini memberikan gambaran implementasi dasar manajemen file dan direktori secara terstruktur dan efisien.

Tugas



```
Modul 10 > C t1.c > createDirectories(const char*)
9 void createDirectories(const char *path) {
10     while (subPath != NULL) {
11         if (mkdir(currentPath, 0755) == 0) {
12             subPath = strtok(NULL, "/");
13         } else if (errno != EEXIST) {
14             perror("Gagal membuat direktori");
15             exit(EXIT_FAILURE);
16         }
17     }
18     subPath = strtok(NULL, "/");
19 }
20
21 int main() {
22     char path[1024];
23     // Meminta input path dari pengguna
24
25     PROBLEMS OUTPUT DEBUG CONSOLE PORTS GITLINS POSTMAN CONSOLE COMMENTS TERMINAL
26 PS C:\Users\asephs\OneDrive\Documents\Praktikum Sistem Operasi\Modul 10> .run.bat
27 Pilih file yang ingin dijalankan:
28 1. p1.c
29 2. p2.c
30 3. t1.c
31 Masukkan nomor file yang ingin dijalankan: 3
32 Meng-compile dan menjalankan t1.c...
33 t1.c: In function 'createDirectories':
34 t1.c:20:13: warning: implicit declaration of function 'mkdir' [-Wimplicit-function-declaration]
35     if (mkdir(currentPath, 0755) == 0) {
36         ^~~~~~
37 Masukkan path direktori hierarkis yang ingin dibuat (contoh: dir1/dir2/dir3): DIR1
38 Direktori "DIR1/" berhasil dibuat.
39 Pembuatan direktori hierarkis selesai.
40 PS C:\Users\asephs\OneDrive\Documents\Praktikum Sistem Operasi\Modul 10>
```

Kode:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%2010/t1.c>

Log:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%2010/t1.txt>

Penjelasan:

Program ini bertujuan untuk membuat struktur direktori hierarkis berdasarkan path yang diberikan pengguna. Dengan memanfaatkan fungsi mkdir, program menciptakan setiap level direktori satu per satu, memastikan bahwa direktori baru dibuat hanya jika belum ada. Hasil eksekusi menunjukkan bahwa direktori "DIR1" berhasil dibuat, dan proses pembuatan direktori hierarkis selesai tanpa error.