

# **Laporan Praktikum**

## **Sistem Operasi**

### **Modul 9**



**Nama : Asep haryana saputra**

**NIM : 20230810043**

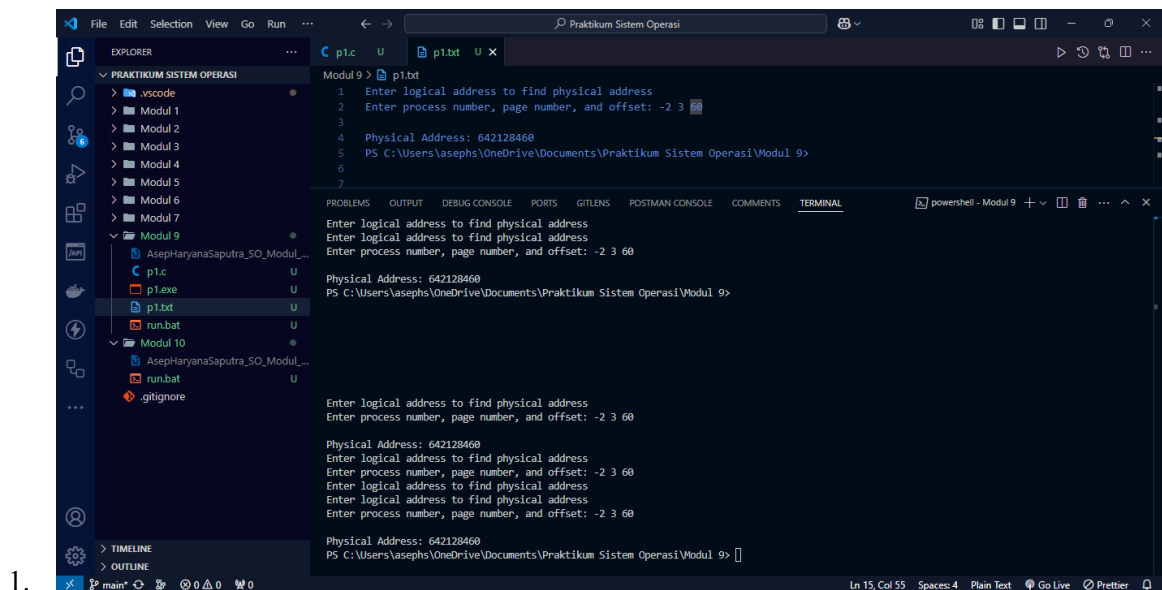
**Kelas : TINFC-2023-04**

**Teknik Informatika**  
**Fakultas Ilmu Komputer**  
**Universitas Kuningan**

## Pretest

1. Paging adalah teknik manajemen memori yang membagi memori logis menjadi halaman (page) dan memori fisik menjadi bingkai (frame) dengan ukuran yang sama. Saat program dijalankan, halaman-halaman dari memori logis dimuat ke dalam bingkai-bingkai yang tersedia di memori fisik. Hubungannya, halaman adalah unit data dari memori logis, sementara bingkai adalah tempat penyimpanan halaman di memori fisik, memungkinkan eksekusi program tanpa perlu memuat keseluruhan program ke memori secara bersamaan.

## Praktikum



Kode:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%209/p1.c>

Log:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%209/p1.txt>

## Posttest

1. program ini mengimplementasikan simulasi manajemen memori berbasis paging, di mana memori dibagi menjadi halaman-halaman dan setiap proses diberikan sejumlah halaman sesuai kebutuhan. Program menghitung dan memvalidasi alokasi memori untuk setiap proses, serta dapat menerjemahkan alamat logis ke alamat fisik berdasarkan tabel halaman yang diberikan. Program ini juga mengatasi keterbatasan memori dengan memberikan pesan kesalahan jika memori tidak cukup atau jika input tidak valid.

## Tugas

1. Membedakan antara teknik alokasi memori paging dan segmentasi?

- Paging: Membagi memori menjadi potongan-potongan berukuran sama yang disebut page.
- Segmentasi: Membagi memori menjadi potongan-potongan berukuran tidak sama yang disebut segment.

2. Apa tujuan dari tabel halaman/page table?

- Untuk menyimpan informasi tentang lokasi page di memori fisik.
- Informasi tersebut meliputi alamat fisik dari page, bit validitas, bit modifikasi, dan bit akses.

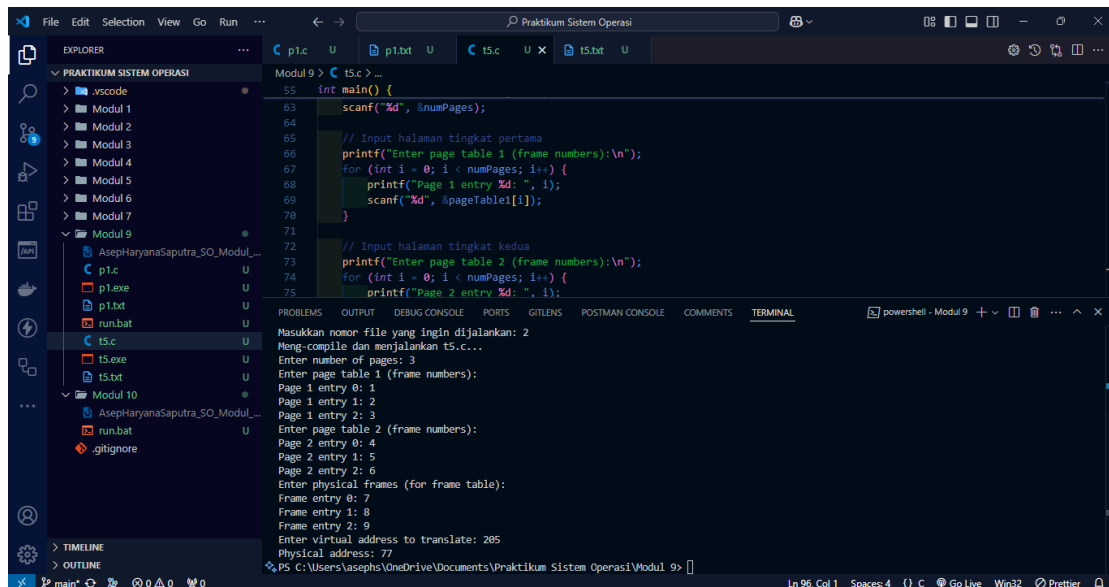
3. Apakah teknik manajemen memori paging menderita masalah fragmentasi internal atau eksternal. Mengapa?

- Teknik manajemen memori paging menderita masalah fragmentasi eksternal.
- Hal ini terjadi karena page-page yang tidak digunakan akan menyisakan ruang kosong di antara page-page yang digunakan.

4. Apa efek dari paging pada keseluruhan waktu pengalihan konteks!

- Paging dapat mempercepat waktu pengalihan konteks.
- Hal ini karena paging dapat mempermudah proses penyalinan data dari memori utama ke cache, karena data yang dibutuhkan sudah berada di page yang sama.

5. Tulis program C untuk mensimulasikan teknik paging dua tingkat!



```

55 int main() {
56     scanf("%d", &numPages);
57
58     // Input halaman tingkat pertama
59     printf("Enter page table 1 (frame numbers):\n");
60     for (int i = 0; i < numPages; i++) {
61         printf("Page 1 entry %d: ", i);
62         scanf("%d", &pageTable1[i]);
63     }
64
65     // Input halaman tingkat kedua
66     printf("Enter page table 2 (frame numbers):\n");
67     for (int i = 0; i < numPages; i++) {
68         printf("Page 2 entry %d: ", i);
69         scanf("%d", &pageTable2[i]);
70     }
71
72     // Masukan nomor file yang ingin dijalankan: 2
73     printf("Meng-compile dan menjalankan t5.c...\n");
74     printf("Enter number of pages: 3\n");
75     printf("Enter page table 1 (frame numbers):\n");
76     printf("Page 1 entry 0: 1\n");
77     printf("Page 1 entry 1: 2\n");
78     printf("Page 1 entry 2: 3\n");
79     printf("Enter page table 2 (frame numbers):\n");
80     printf("Page 2 entry 0: 4\n");
81     printf("Page 2 entry 1: 5\n");
82     printf("Page 2 entry 2: 6\n");
83     printf("Enter physical frames (for frame table):\n");
84     printf("Frame entry 0: 7\n");
85     printf("Frame entry 1: 8\n");
86     printf("Frame entry 2: 9\n");
87     printf("Enter virtual address to translate: 205\n");
88     printf("Physical address: 77\n");
89 }
  
```

Kode:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%209/t5.c>

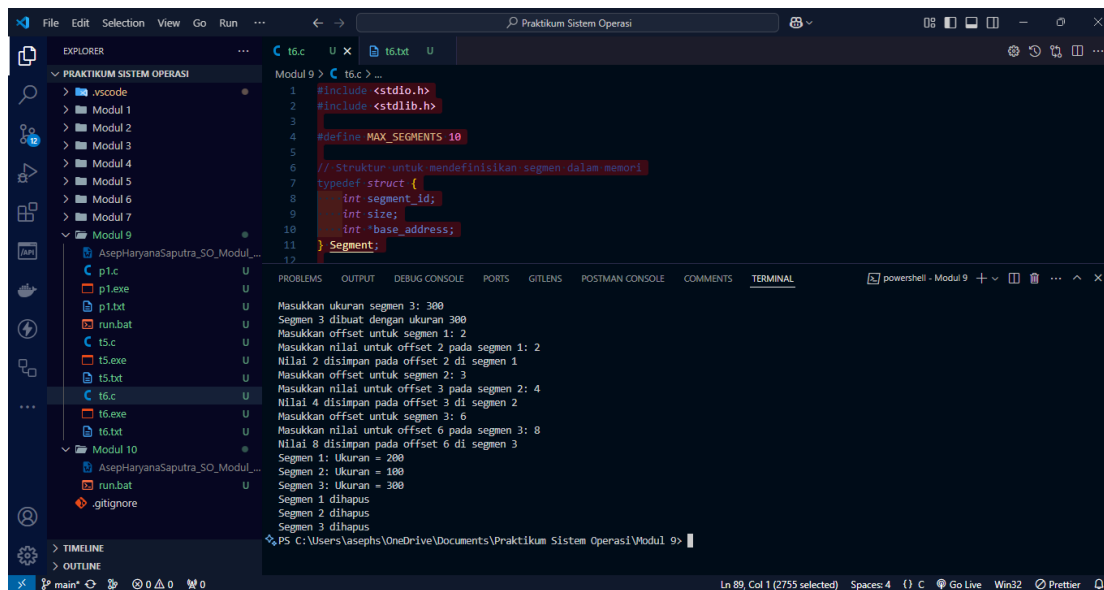
Log:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%209/t5.txt>

Penjelasan:

Program ini bertujuan untuk menerjemahkan alamat virtual ke alamat fisik menggunakan sistem dua tingkat tabel halaman. Pengguna memasukkan jumlah halaman, nomor entri untuk dua tabel halaman, nomor frame untuk tabel frame fisik, dan alamat virtual yang ingin diterjemahkan. Program ini memecah alamat virtual menjadi tiga bagian: indeks direktori halaman, indeks tabel halaman, dan offset. Setelah memvalidasi entri tabel halaman, program menghitung alamat fisik yang sesuai dengan mengakses frame dari tabel frame dan menambahkan offset. Hasil dari contoh yang diberikan adalah alamat fisik 77 untuk alamat virtual 205, yang dihitung berdasarkan pengaturan tabel yang dimasukkan.

6. Tulis program C untuk mensimulasikan teknik manajemen memori segmentasi!



```
Modul 9 > t6.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define MAX_SEGMENTS 10
5
6 // Struktur untuk mendefinisikan segmen dalam memori
7 typedef struct {
8     int segment_id;
9     int size;
10    int *base_address;
11 } Segment;
12
```

```
PROBLEMS OUTPUT DEBUG CONSOLE PORTS GITLENS POSTMAN CONSOLE COMMENTS TERMINAL
Masukkan ukuran segmen 3: 300
Segmen 3 dibuat dengan ukuran 300
Masukkan offset untuk segmen 1: 2
Masukkan nilai untuk offset 2 pada segmen 1: 2
Nilai 2 disimpan pada offset 2 di segmen 1
Masukkan offset untuk segmen 2: 3
Masukkan nilai untuk offset 3 pada segmen 2: 4
Nilai 4 disimpan pada offset 3 di segmen 2
Masukkan offset untuk segmen 3: 6
Masukkan nilai untuk offset 6 pada segmen 3: 8
Nilai 8 disimpan pada offset 6 di segmen 3
Segmen 1: Ukuran = 200
Segmen 2: Ukuran = 100
Segmen 3: Ukuran = 300
Segmen 1 dihapus
Segmen 2 dihapus
Segmen 3 dihapus
PS C:\Users\asephs\OneDrive\Documents\Praktikum Sistem Operasi\Modul 9>
```

Kode:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%209/t6.c>

Log:

<https://github.com/MythEclipse/Praktikum-Sistem-Operasi/blob/main/Modul%209/t6.txt>

Penjelasan:

Program ini bertujuan untuk mengelola segmen-segmen memori, termasuk membuat, mengakses, menampilkan, dan menghapus segmen. Pengguna dapat menentukan jumlah segmen dan ukuran masing-masing segmen, kemudian mengakses dan menyimpan nilai pada offset tertentu dalam segmen-segmen tersebut. Program akan menampilkan informasi ukuran setiap segmen dan memastikan bahwa nilai disimpan pada posisi yang valid dalam segmen. Setelah selesai, segmen-segmen yang telah dibuat akan dihapus untuk membebaskan memori. Hasilnya, program berhasil membuat dan mengelola tiga segmen dengan ukuran yang berbeda, menyimpan nilai pada offset tertentu, menampilkan informasi segmen, dan menghapus segmen-segmen tersebut setelah selesai.