



# pengelolaan data siswa

putri salsabila  
24343020



# Input Siswa

Proses membaca data siswa dari file dan menyimpannya kembali ke file lain.

1

Definisi Struktur Data

2

Membaca Data dari File

3

Menampilkan dan Menyimpan Data

4

Waktu Eksekusi dan Penutupan







# Definisi Struktur Data

Menyimpan  
informasi siswa

```
typedef struct Siswa {  
    char nama[50];  
    char nisan[20];  
    char kelas[20];  
    char ttl[30];  
    char jenisKelamin[2];  
    char noHp[15];  
    char hobi[30];  
    char alamat[100];  
} Siswa;   ← → menyimpan informasi data siswa
```



# Membaca Data dari File

## Langkah 1: Membuka File

- Program membuka file data\_siswa.txt untuk dibaca.
- Menggunakan fgets() untuk membaca baris dan sscanf() untuk parsing data.

```
inputFile = fopen("data_siswa.txt", "r");
```

## Langkah 2: Menyimpan Data

- Data yang diparsing disimpan dalam array Siswa.







## Menampilkan dan Menyimpan Data

### Menampilkan Data

Data siswa ditampilkan dalam format tabel dengan menggunakan printf().

```
tampilkanData(data, jumlahSiswa);
```

### Menyimpan Data

Data yang sudah ditampilkan disimpan kembali ke file output\_data\_siswa.txt.

```
simpanData(data, jumlahSiswa,  
outputFile);
```



# Waktu Eksekusi dan Penutupan

## Menghitung Waktu Eksekusi

- Waktu eksekusi program dihitung menggunakan clock().

## Membebaskan Memori

- Memori yang digunakan untuk menyimpan data siswa dibebaskan dengan free(data)

```
double cpu_time_used = ((double)(end - start)) /  
CLOCKS_PER_SEC;
```





# Kesimpulan



Program ini berhasil membaca data siswa dari file, menampilkan data dalam format tabel, dan menyimpannya kembali ke file output. Proses utama meliputi:

1. Membaca Data: Program membuka file dan mem-parsing data siswa menggunakan format yang telah ditentukan.
2. Menyimpan Data: Data yang telah dibaca disimpan dalam array dinamis, kemudian ditampilkan dalam format yang rapi dan disimpan kembali ke file lain.
3. Pengelolaan Memori: Memori yang digunakan untuk menyimpan data siswa dikelola secara efisien dengan malloc() dan realloc(), serta dibebaskan setelah selesai.
4. Efisiensi Waktu: Waktu eksekusi program dihitung untuk mengevaluasi kinerja program.

Secara keseluruhan, program ini menyediakan solusi untuk memproses data siswa secara otomatis, efisien, dan terstruktur.







Thank  
you!

TERIMA  
KASIH