

# Operasi File

Kharisma Monika

Informatika

IT Telkom Surabaya

# Kenapa ada operasi File?

penyimpanan **data** berbasis file akan tersimpan terus walaupun program telah diclose maupun komputer telah di shutdown

# Jenis File di Komputer

1. File teks : file bisa dibaca dengan text editor.  
Contohnya : file txt, file csv, file html, dll.
2. File Binary : file yang tersimpan dalam bentuk biner (0 & 1). Contohnya seperti:  
File .exe dan file .bin

# 3 jenis Operasi File

1. Read (R)
2. Write (W)
3. Append (A)

# Tahapan Operasi File



# Struktur Dasar



path/namafile

```
int main(){  
    FILE *in=fopen(namafile,mode);  
    fclose(in);  
    getchar();  
    return 0;  
}
```

# Parameter Mode

r	Membuka file sesuai namafilennya untuk dibaca, jika file tidak ada maka akan bernilai NULL.
w	Membuka file untuk ditulis, jika sebelumnya sudah ada nama file yang sama, maka akan dilakukan overwrite/menimpa file dengan yang baru.
a	Membuka file sesuai namafilennya untuk ditambahkan datanya dipaling bawah baris terakhir. Jika file tidak ada, maka akan membuat file baru secara otomatis.
r+	Sama dengan “r” tetapi selain file dapat dibaca juga dapat di tulis
w+	Sama dengan “w” tetapi selain file dapat ditulis juga dapat dibaca
a+	Sama dengan ”a” tetapi selain file dapat ditulis file juga dapat dibaca

# Parameter mode file binary

rb	Buka untuk dibaca dalam binary.	return NULL.
wb	Buka untuk ditulis dalam binary	Buat baru atau tulis ulang jika ada
ab	Buka untuk ditambahkan dalam binary	Buat baru atau tulis ulang jika ada
rb+	Buka untuk dibaca dan ditulis dalam binary	returns NULL.
wb+	Buka untuk dibaca dan ditulis dalam binary	Buat baru atau tulis ulang jika ada
ab+	Buka untuk dibaca dan ditambahkan isinya dalam binary	Buat baru atau tulis ulang jika ada



# File Handling in C

File operation	Declaration & Description
<b>fopen() - To open a file</b>	<p>Declaration: FILE *fopen (const char *filename, const char *mode)</p> <p>fopen() function is used to open a file to perform operations such as reading, writing etc. In a C program, we declare a file pointer and use fopen() as below. fopen() function creates a new file if the mentioned file name does not exist.</p> <pre>FILE *fp; fp=fopen ("filename", "mode");</pre> <p>Where,</p> <p>fp - file pointer to the data type "FILE".</p> <p>filename - the actual file name with full path of the file.</p> <p>mode - refers to the operation that will be performed on the file. Example: r, w, a, r+, w+ and a+. Please refer below the description for these mode of operations.</p>
<b>fclose() - To close a file</b>	<p>Declaration: int fclose(FILE *fp);</p> <p>fclose() function closes the file that is being pointed by file pointer fp. In a C program, we close a file as below.</p> <pre>fclose (fp);</pre>
<b>fgets() - To read a file</b>	<p>Declaration: char *fgets(char *string, int n, FILE *fp)</p> <p>fgets function is used to read a file line by line. In a C program, we use fgets function as below.</p> <pre>fgets (buffer, size, fp);</pre> <p>where,</p> <p>buffer - buffer to put the data in.</p> <p>size - size of the buffer</p> <p>fp - file pointer</p>
<b>fprintf() - To write into a file</b>	<p>Declaration:</p> <pre>int fprintf(FILE *fp, const char *format, ...);</pre> <p>fprintf() function writes string into a file pointed by fp. In a C program, we write string into a file as below. fprintf (fp, "some data"); or</p> <pre>fprintf (fp, "text %d", variable_name);</pre>

<https://www.geeksforgeeks.org/basics-file-handling-c/>

# File Handling in C

1. Creation of a new file (**fopen** with attributes as **"a"** or **"a+"** or **"w"** or **"w++"**)
2. Opening an existing file (**fopen**)
3. Reading from file (**fscanf** or **fgets**)
4. Writing to a file (**fprintf** or **fputs**)
5. Moving to a specific location in a file (**fseek**, **rewind**)
6. Closing a file (**fclose**)

<https://www.geeksforgeeks.org/basics-file-handling-c/>

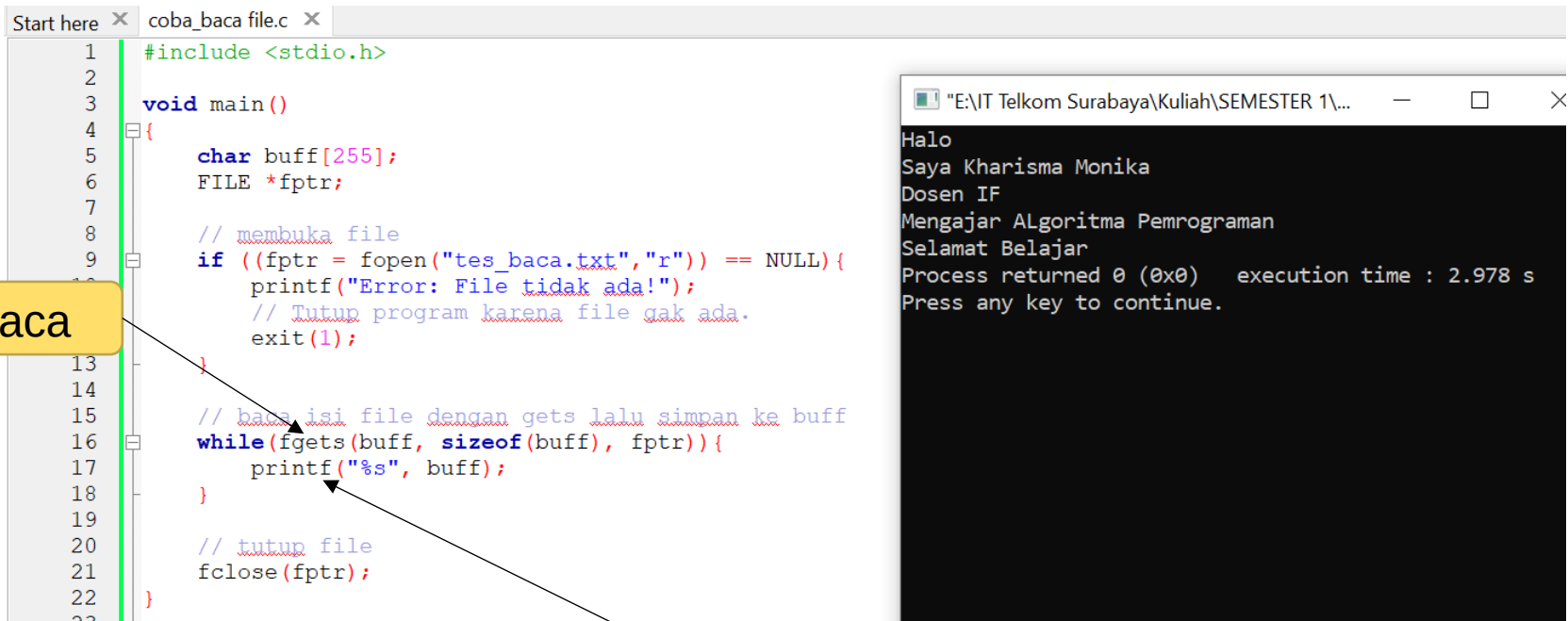
# Membaca File

File path atau  
nama file yang  
akan dibuka

Mode

```
fopen("filepath", "r");
```

# Contoh Baca File



The image shows a C program in a text editor and its execution in a console window. The program, named 'coba\_baca file.c', includes `<stdio.h>` and defines a `main()` function. It attempts to open a file named 'tes\_baca.txt' in read mode. If the file is not found, it prints an error message and exits. If the file is found, it reads the file line by line using `fgets()` and prints each line to the console. The console output shows the program successfully reading and printing the contents of 'tes\_baca.txt', which contains a greeting and course information. The execution time is 2.978 seconds.

```
1 #include <stdio.h>
2
3 void main()
4 {
5     char buff[255];
6     FILE *fptr;
7
8     // membuka file
9     if ((fptr = fopen("tes_baca.txt", "r")) == NULL) {
10         printf("Error: File tidak ada!");
11         // Tutup program karena file gak ada.
12         exit(1);
13     }
14
15     // baca isi file dengan gets lalu simpan ke buff
16     while (fgets(buff, sizeof(buff), fptr)) {
17         printf("%s", buff);
18     }
19
20     // tutup file
21     fclose(fptr);
22 }
23
```

Halo  
Saya Kharisma Monika  
Dosen IF  
Mengajar ALgoritma Pemrograman  
Selamat Belajar  
Process returned 0 (0x0) execution time : 2.978 s  
Press any key to continue.

Baca

Langsung  
dicetak di  
console

# Menulis File

Kumpulan karakter  
atau teks yang akan ditulis

pointer file

`fputs ( 'teks' , fptr );`

# Contoh Write ke File

```
char buff[255];  
char text[255];  
FILE *fptr;
```

```
// membuka file untuk ditulis  
fptr = fopen("coba_tulis.txt", "w");
```

```
for(int i = 0; i < 5; i++){
```

```
    // mengambil input dari user  
    printf("Isi baris ke-%d: ", i+1);  
    fgets(text, sizeof(text), stdin);
```

```
    // menulis ke text ke file  
    fputs(text, fptr);
```

```
}
```

```
printf("File berhasil ditulis\n");
```

```
// tutup file setelah ditulis  
fclose(fptr);
```

Membuka file, dengan mode w

Membaca inputan user lalu  
disimpan di array

Menulis ke File

Menutup File

# Contoh write lainnya

```
int main()
{
    int num[5];
    FILE *fptr;

    fptr = fopen("cobatulis2.txt", "w");

    if(fptr == NULL)
    {
        printf("Error!");
        exit(1);
    }

    for(int i = 0; i < 5; i++){

        // mengambil input dari user
        printf("Isi number baris ke-%d: ", i+1);
        scanf("%d", &num[i]);

        // menulis ke text ke file
        fprintf(fptr, "%d", num[i]);
    }

    fclose(fptr);

    return 0;
}
```

Dengan scanf

Dengan fprintf

# Append File

```
// membuka file untuk ditulis  
fptr = fopen("coba_tulis.txt", "a");
```

Menggunakan mode a



# Contoh Program Append

```
char buff[255];
char text[255];
FILE *fptr;

// membuka file untuk ditulis
fptr = fopen("coba_tulis.txt", "a");

for(int i = 0; i < 5; i++){

    // mengambil input dari user
    printf("Isi baris ke-%d: ", i+1);
    fgets(text, sizeof(text), stdin);

    // menulis ke text ke file
    fputs(text, fptr);
}

printf("File berhasil ditulis\n");

// tutup file setelah ditulis
fclose(fptr);

// buka kembali file untuk dibaca
fptr = fopen("coba_tulis.txt", "r");

// baca isi file dengan gets lalu simpan ke buff
while(fgets(buff, sizeof(buff), fptr)){
    printf("Isi filenya sekarang: %s", buff);
}

// tutup file
fclose(fptr);
```



# Terima Kasih