

## Dasar Teori

3 jenis operasi File : read (r) , write (w), append (a)

Struktur Dasar :

```
FILE *fp; //pointer file fp
fp = fopen("data.txt", w); //membuka file fp mode w

statement;

fclose(fp); //menutup file
```

## Parameter Mode

r	Membuka file sesuai nama filenya 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 nama filenya 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

## Membaca File

Contoh Program Membaca File dengan fgetc :

1. Buat File C, tulis program dibawah ini. Simpan

```
#include <stdio.h>

void main()
{
    char buff[255];
    FILE *fptr;

    // membuka file
    if ((fptr = fopen("tes_baca.txt", "r")) == NULL){
        printf("Error: File tidak ada!");
        // Tutup program karena file gak ada.
    }
}
```

```

    exit(1);
}

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

// tutup file
fclose(fptr);
}

```

2. Buat file yang diberi nama tes\_baca.txt . Tulis apapun (maks 255 karakter). Simpan
3. Jalankan program.
4. Program akan mengeluarkan hasil sesuai dengan isi file

## Menulis File

1. Salin program dibawah ini.

```

#include <stdio.h>

void main()
{
    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);
}

```

```

// 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);
}

```

2. Jalankan program
3. Masukkan inputan
4. Program akan mencetak isi file ke console/terminal
5. Tutup console
6. Buka file coba\_tulis.txt isinya sama dengan inputan terakhir

## Append ke File

Append File hampir sama dengan write. Hanya saja mode write akan mereplace data yang sudah ada. Sedangkan mode append akan menambah baris data ke file.

Contoh kode program input angka

```

#include <stdio.h>
#include <stdlib.h>

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

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

    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;
}
```

SOAL LATIHAN :

1. Buat program dimana membaca inputan nilai dari user. Program dapat memisahkan angka ganjil dan genap ke dalam file yang berbeda. Misalnya ganjil.txt untuk menyimpan nilai ganjil. Genap.txt untuk menyimpan nilai genap. Program menulis ke file menggunakan mode append.
2. Buat program untuk mengcopy isi file. Program membaca isi file a.txt lalu menuliskan ke file b.txt
3. Buat program untuk menginputkan data mahasiswa ke file  
Format Data :  
Nama : Fajrul Fallah  
NIM : 12345678  
Alamat : Caruban Kab. Madiun

Penulisan file menggunakan mode append