FILE PROCESSING

Tipe-tipe file:

- Text files
  - ✓ Biasanya memiliki ekstensi .txt
  - ✓ Kita dapat melihat langsung isi konten apabila membuka file tersebut dalam bentuk teks (plain text)
- Binary files
  - ✓ Biasanya disimpan dalam eksnetensi .bin
  - ✓ Data tidak disimpan dalam bentuk teks biasa (plain text), melainkan dalam bentuk binary (0 dan 1)

Operasi pada File

1) Create new file
2) Open existing file
3) Close file
4) Read from and write to file

Open and Create File

```
FILE *fptr; //Membuat pointer yang akan menunjuk suatu file
*fptr = fopen("testdata.in","r"); //Membuka file
//fopen("filepath","mode")
```

Mode	Meaning of Mode	During Inexistence of file
r	Open for reading.	If the file does not exist, fopen() returns NULL.
rb	Open for reading in binary mode.	If the file does not exist, fopen() returns NULL.
w	Open for writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb	Open for writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a	Open for append. Data is added to the end of the file.	If the file does not exist, it will be created.
ab	Open for append in binary mode. Data is added to the end of the file.	If the file does not exist, it will be created.
r+	Open for both reading and writing.	If the file does not exist, fopen() returns NULL.
rb+	Open for both reading and writing in binary mode.	If the file does not exist, fopen() returns NULL.
w+	Open for both reading and writing.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
wb+	Open for both reading and writing in binary mode.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
a+	Open for both reading and appending.	If the file does not exist, it will be created.
ab+	Open for both reading and appending in binary mode.	If the file does not exist, it will be created.

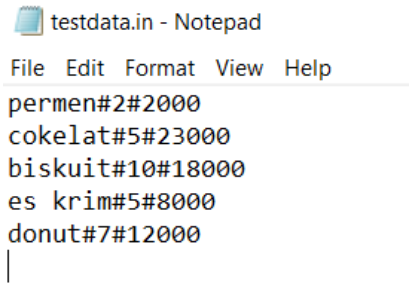
FILE PROCESSING

Close File

```
fclose(fptr);
//Menutup file yang sudah dibuka
```

Read File

Misalkan terdapat file dengan nama testdata.in



Membaca data dari file 'testdata.in'

```
#include<stdio.h>

struct data{
    long int price;
    char name[20];
    int qty;
};
data brg[5];

int main(){

    //Membuka file untuk mode baca (r)
    FILE *fp;
    fp = fopen("testdata.in", "r");

    int i = 0;
    //Membaca file selama masih ada data (!= EOF) menggunakan fscanf()
    while(fscanf(fp,"%[^#]#%d#%ld\n", brg[i].name, &brg[i].qty, &brg[i].price)!= EOF){
        printf("%d %s @%ld\n", brg[i].qty, brg[i].name, brg[i].price);
    }

    //Menutup file
    fclose(fp);

    return 0;
}
```

FILE PROCESSING

Write File

Misalkan kita ingin menulis data ke dalam file ‘output.txt’

```c
#include<stdio.h>
#include<string.h>

struct data{
    long int price;
    char name[20];
    int qty;
};
data brg[5];

void insertData(int idx, char *name, int qty, int price){
    strcpy(brg[idx].name, name);
    brg[idx].qty = qty;
    brg[idx].price = price;
}

int main(){

    insertData(0, "permen", 2, 2000);
    insertData(1, "cokelat", 5, 23000);
    insertData(2, "biskuit", 10, 18000);
    insertData(3, "es krim", 5, 8000);
    insertData(4, "donut", 7, 12000);

    //Membuka file untuk mode baca (tulis)
    FILE *fp;
    fp = fopen("output.txt", "w");

    //Menulis file
    for(int i = 0; i < 5; i++){
        fprintf(fp, "%d %s @%ld\n", brg[i].qty, brg[i].name, brg[i].price);
    }

    //Menutup file
    fclose(fp);

    return 0;
}
```