

# Algoritma dan Pemrograman Lanjut

## Pertemuan Ke-12 Arsip (*File*) 1



Disusun Oleh :  
Wilis Kaswidjanti, S.Si.,M.Kom.

**Jurusan Teknik Informatika  
Fakultas Teknologi Industri  
Universitas Pembangunan Nasional “Veteran”  
Yogyakarta**

# Algoritma dan Pemrograman Lanjut

**Judul Materi** : Arsip (*File*) 1

**Deskripsi Materi** : Materi ini membahas tipe data file, pendeklarasian dan perintah-perintah baku untuk mengolah data pada file

**Tujuan Instruksional Khusus** :

1. Mendefinisikan dan menggunakan tipe data file
2. Mendeskripsikan tipe data file
3. Menggunakan tipe data file
4. Menjelaskan fungsi-fungsi pada tipe data file
5. Mengimplementasikan fungsi-fungsi file untuk memanipulasi file

**Referensi** :

- Buku Teks  
Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 2, Edisi Ketiga, Penerbit Informatika Bandung, Bab 4, hal 115-168.  
Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman II*, Edisi Kedua, Yogyakarta
- Buku Acuan/Referensi  
Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.  
Jarne, Stroustrup B. (1997), *C++ Programming language*, AT &T.  
Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.  
Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.  
Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.

# Arsip (*File*) 1

## PENDAHULUAN

Media Penyimpan :

- Memori Utama (RAM) → tidak menyimpan informasi secara permanen
- Memori Sekunder → media penyimpanan permanen.

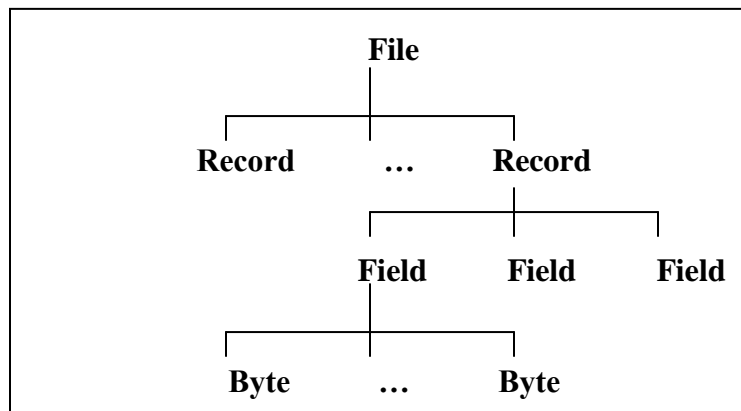
Contoh : disk (floppy disk, hard disk, compact disk, flash disk dll.). Data yang disimpan di dalam penyimpanan sekunder dikelompokkan dalam bentuk arsip/file

## ISI

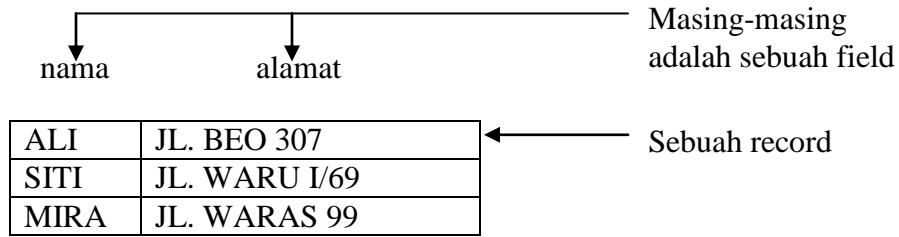
Data yang disimpan di dalam penyimpanan sekunder dikelompokkan dalam bentuk arsip/file. File menyimpan data berkategori sama. Nama arsip/file harus unik.

## Struktur File :

- Suatu file merupakan organisasi dari sejumlah record dapat terdiri dari satu atau beberapa field dan setiap field terdiri dari satu atau beberapa byte.
- Adapun byte merupakan susunan dari 8 bit.



Struktur data dari file



Gambaran record dan field

**Pengorganisasian data di dalam file :**

- Sebuah informasi disebut datum atau record, sedangkan jamaknya disebut data. Semua record di dalam file diorganisasikan penyimpanannya, dan pengaksesan record di dalam file bergantung pada metode perorganisasiannya tsb.
- Ada dua cara : beruntun (sequential) dan acak (random).

**Arsip Beruntun**

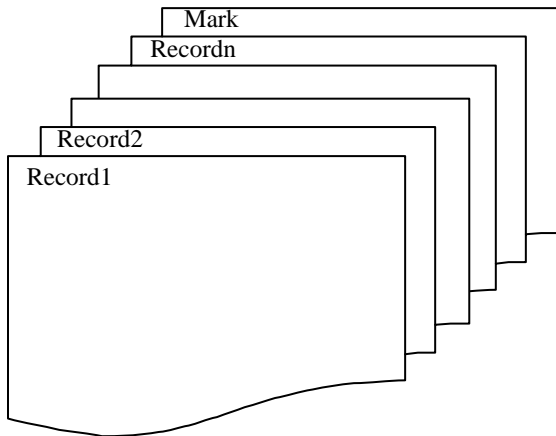
**Pengertian Arsip beruntun :** Sekumpulan record-record terpadu, yang disimpan dalam media penyimpanan sekunder, yang dapat diakses secara berurutan record per record searah mulai dari record pertama.

**File dapat dipandang dengan dua cara :**

1. Secara Tabel

	Field1	Field2	Field3	...	Fieldn
Record1					
Record2					
Record3					
...					
Recordn					
Mark					

## 2. Secara Blanko



Record terakhir adalah record fiktif yang menandai akhir dari file.

### Deklarasi File

(notasi algoritma)

Type NamaRecord : TipeRecord

NamaArsip : SeqFile of TipeRecord

VarRecord : NamaRecord

Contoh :

Type DataMhs : Record

<NIM : Integer,

Nama : String,

IPK : Real >

ArsipMhs : SeqFile of DataMhs

Mhs : DataMhs

Type ArsipBil : SeqFile of Integer

Bil : ArsipBil

i : integer

Type ArsipKar : SeqFile of Char

Kar : ArsipKar

c : Char

### (Bahasa C++)

```
typedef TipeRecord NamaRecord;  
FILE *NamaArsip;  
NamaRecord VarRecord;
```

Contoh :

```
Typedef struct {long NIM;  
                char Nama[25];  
                float IPK;  
                } DataMhs;  
FILE *ArsipMhs;  
DataMhs Mhs;
```

```
FILE *Bil;  
int i;
```

```
FILE *Kar;  
char c;
```

### Perintah Baku

(notasi algoritma)

■ **Open(NamaArsip,kode)**

Contoh :

Open(ArsipMhs,1)

Open(Bil,2)

Open(Kar,1)

■ **FRead(NamaArsip,VarRecord)**

Contoh :

FRead(ArsipMhs,Mhs)

FRead(Bil,i)

FRead(Kar,c)

■ **FWrite>NamaArsip,VarRecord)**

Contoh :

FWrite(ArsipMhs,<12331,'Hanif',3.50)

FWrite(ArsipMhs,<99999,'xxxxx',9.99)

FWrite(Bil,765)

FWrite(Kar,'R')

Input(i)

FWrite(Bil,i)

Input(c)

FWrite(Kar,c)

Input(Mhs.NIM)

Input(Mhs>Nama)

Input(Mhs.IPK)

FWrite(ArsipMhs,Mhs)

■ **Close>NamaArsip)**

Contoh :

Close(ArsipMhs)

**(Notasi bahasa C++)**

- NamaArsip1 = fopen>NamaArsipFisik1,"r"); /\*utk dibaca\*/  
NamaArsip2 = fopen>NamaArsipFisik1,"w"); /\*utk ditulis\*/

Contoh :

ArsipMhs = fopen("C:ArsipMhs.dat","r");

Bil = fopen("A:Bil.dat","w");

Kar = fopen("A:Kar.dat","r");

- fread(VarRecord,sizeof(VarRecord),jum\_record>NamaArsip);

Contoh :

fread(&Mhs,sizeof(Mhs),1,ArsipMhs);

```
fread(&i, sizeof(i), 1, Bil);
```

```
fread(&c, sizeof(c), 1, Kar);
```

- `fwrite(VarRecord, sizeof(VarRecord), jum_record, NamaArsip);`

Contoh :

```
fwrite(&Mhs, sizeof(Mhs), 1, ArsipMhs);
```

```
fwrite(&i, sizeof(i), 1, Bil);
```

```
fwrite(&c, sizeof(c), 1, Kar);
```

```
fwrite(&Kar, 'R')
```

- `fclose>NamaArsip);`

Contoh :

```
fclose(ArsipMhs);
```

### Contoh Program 1:

```
//menyimpan data bilangan integer
#include <stdio.h>
#include <iostream.h>
#include <conio.h>

FILE *Bil;
typedef enum {true=1,false=0} boolean;
boolean Mark(int i);

main()
{
    int i;

    Bil = fopen("Bil.dat","w");
    cout << "Bilangan : "; cin >> i;
    while (Mark(i) != true)
    {
        fwrite(&i, sizeof(i), 1, Bil);
        cout << "Bilangan : "; cin >> i;
    }
    i = 999;
    fwrite(&i, sizeof(i), 1, Bil);
    fclose(Bil);
}

boolean Mark(int i)
{
    return (i == 999);
}
```



### Contoh Program 2 :

```
//membaca data bilangan integer yang ada di
//file dari contoh program 1
#include <stdio.h>
#include <iostream.h>
#include <conio.h>

FILE *Bil;
typedef enum {true=1,false=0} boolean;
boolean Mark(int i);

main()
{
    int i,jum;
    jum=0;
    Bil = fopen("Bil.dat","r");
    fread(&i,sizeof(i),1,Bil);
    if (Mark(i))
        cout << "Arsip kosong\n";
    else
    {
        do
        {
            jum=jum+i;
            //cout << i <<endl;
            fread(&i,sizeof(i),1,Bil);
        }
        while (!Mark(i));
    }
    cout <<jum;
    fclose(Bil);

    return 0;
}

boolean Mark(int i)
{
    return (i == 999);
}
```

### Contoh program 3 :

```
//program menyimpan data mahasiswa dan mencetak
//data mahasiswa sesuai dengan nim yang diinginkan
#include <stdio.h>
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <iomanip.h>
```

```

typedef struct { long NIM;
                char Nama[25];
                char KodeMK[5];
                int SKS;
                char Indeks;
                } DataMhs;

FILE *ArsipMhs;
typedef enum {true=1,false=0} boolean;
void RekamDataMahasiswa();
void CetakDaftarNilai(long KodeNIM);
boolean Mark(DataMhs Mhs);

main()
{
    long KodeNIM;

    RekamDataMahasiswa();
    clrscr();
    cout << " NIM : "; cin >> KodeNIM;
    clrscr();
    CetakDaftarNilai(KodeNIM);
    return 0;
}

void RekamDataMahasiswa()
{
    DataMhs Mhs;

    ArsipMhs = fopen("Mhs.dat","w");
    cout << "NIM : "; cin >> Mhs.NIM;
    while (Mark(Mhs) != true)
    {
        cout << "Nama : "; cin >> Mhs.Nama;
        cout << "Kode MK : "; cin >> Mhs.KodeMK;
        cout << "SKS : "; cin >> Mhs.SKS;
        cout << "Nilai : "; cin >> Mhs.Indeks;
        fwrite(&Mhs,sizeof(Mhs),1,ArsipMhs);
        cout << "NIM : "; cin >> Mhs.NIM;
    }
    Mhs.NIM = 9999;
    strcpy(Mhs.Nama, "xxxxx");
    strcpy(Mhs.KodeMK, "xxxxx");
    Mhs.SKS = 9;
    Mhs.Indeks = 'x';
    fwrite(&Mhs,sizeof(Mhs),1,ArsipMhs);
    fclose(ArsipMhs);
}

void CetakDaftarNilai(long KodeNIM)
{
    int no;

```

```

DataMhs Mhs;
boolean ketemu;
boolean stop;
ArsipMhs = fopen("Mhs.dat","r");
stop = false;
fread(&Mhs,sizeof(Mhs),1,ArsipMhs);
if (Mark(Mhs))
    cout << "Arsip kosong\n";
else
{ //cari code NIM
    ketemu = false;
    do
    {
        if (Mhs.NIM == KodeNIM)
            ketemu = true;
        else
            fread(&Mhs,sizeof(Mhs),1,ArsipMhs);
    }
    while (!ketemu && !Mark(Mhs));
    if (ketemu)
    {
        cout << "Daftar Nilai Mata Kuliah\n";
        cout << " NIM      : " << Mhs.NIM << endl;
        cout << " Nama    : " << Mhs>Nama << endl;
        cout << "-----\n";
        cout << "No. Mata Kuliah      SKS      Nilai\n";
        cout << "-----\n";
        no = 0;
        do
        {
            no++;
            cout << no << setw(7) << Mhs.KodeMK;
            cout << setw(13) << Mhs.SKS << setw(9);
            cout << Mhs.Indeks << endl;
            fread(&Mhs,sizeof(Mhs),1,ArsipMhs);
        }
        while (Mhs.NIM == KodeNIM && !Mark(Mhs));
        cout << "-----\n";
    }
    else
    {
        cout << "Data mahasiswa dengan NIM = ";
        cout << KodeNIM << " tidak ada";
    }
}
}
fclose(ArsipMhs);
}

boolean Mark(DataMhs Mhs)
{
    return (Mhs.NIM == 9999);
}

```

## **PENUTUP**

File dengan memanfaatkan media penyimpan permanen, berfungsi untuk menyimpan data secara permanen juga. Data yang disimpan dapat bertipe apa saja.

## **SOAL-SOAL**

Modifikasi contoh program 3 di atas agar program tersebut tidak hanya mencetak daftar nilai seorang mahasiswa tetapi mencetak semua data mahasiswa, dan juga mencetak daftar mhs berdasarkan nilai yang diperoleh.