

PENGAKSESAN BERKAS (FILE) DALAM C

Arsip/File/Berkas merupakan struktur penyimpanan data didalam memori sekunder seperti hard disk. Struktur arsip memungkinkan data disimpan secara permanen dan diakses kembali jika diperlukan. Setiap item data yang disimpan dalam arsip disebut rekaman (record).

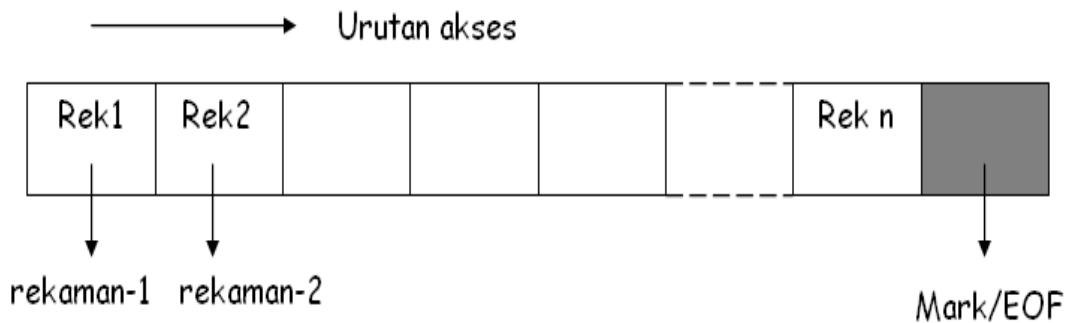
Pengorganisasian data dalam arsip ada 2 cara :

1. Beruntun (sequential)
 - a. metode paling sederhana
 - b. rekaman disimpan secara beruntun (rekaman yg satu setelah yg lain)
 - c. data diakses mulai dari rekaman pertama sampai yang diinginkan
 - d. pada umumnya lambat
 - e. mudah dibuat dan dipelihara
2. Acak (random)/akses-langsung (direct access)
 - a. rekaman diakses secara langsung tanpa perlu mulai dari yang pertama
 - b. pengaksesan data individual lebih cepat
 - c. lebih sulit dibuat dan dipelihara

I. PENGAKSESAN ARSIP BERUNTUN

Arsip beruntun adalah sekumpulan rekaman bertipe sama yang diakses secara beruntun mulai dari rekaman pertama sampai rekaman yang dituju atau sampai dengan rekaman terakhir.

Struktur arsip beruntun tidak jauh berbeda dengan struktur larik. Setiap komponen larik disebut elemen sedangkan pada arsip disebut rekaman. Perbedaan arsip dengan larik hanya pada pilihan cara pengaksesan elemen dan arah pemrosesan. Rekaman dalam arsip beruntun tidak dapat diakses secara langsung karena harus dibaca dari awal rekaman, sementara dalam larik elemennya dapat diakses secara langsung melalui indeksinya.



Struktur arsip beruntun

Sebelum melakukan pemrosesan arsip, maka arsip tersebut harus dideklarasikan terlebih dahulu. Bentuk mum pendeklarasian arsip dalam bahasa C adalah sbb:

```
FILE *arsip;
```

Perintah-perintah baku yang dapat digunakan dalam pemrosesan arsip

Perintah-perintah baku yang dapat digunakan dalam pemrosesan arsip beruntun adalah :

fopen	: menyiapkan arsip untuk penulisan atau pembacaan
fclose	: menutup arsip
fscanf	: membaca rekaman dari arsip
fprintf	: menulis rekaman ke arsip

Contoh 1 : Menulis ke dan membaca dari arsip berurutan

```
#include<iostream.h>
#include<stdio.h>
FILE *fp;
void main(void)
{
    int i=0,k;
    //Menyimpan data ke file
    fp=fopen("d:\\kampus\\program\\c++\\FileIO1\\data1.dat","w");
    if(fp==NULL)cout<<"Error membuka berkas"<<endl;
    else
    {
        for(i=0;i<=10;i++)fprintf(fp,"%d ",i);
        fclose(fp);
    }
    //Membaca data dari file
    fp=fopen("d:\\kampus\\program\\c++\\FileIO1\\data1.dat","r");
    if(fp==NULL)cout<<"Error membuka berkas"<<endl;
    else
    {
        while(!feof(fp))
        {
            fscanf(fp,"%d",&k);cout<<k<<" ";
        }
        cout<<endl;
        fclose(fp);
    }
}
```

Contoh 2 : Menulis ke dan membaca dari arsip berurutan

```
#include<iostream.h>
#include<stdio.h>
FILE *fp;
void main(void)
{
    int i,j,jml_dat,k[100];
    //Menyimpan data ke file
    fp=fopen("d:\\kampus\\program\\c++\\FileIO2\\data1.dat","w");
    if(fp==NULL)cout<<"Error membuka berkas"<<endl;
    else
    {
        for(i=0;i<=10;i++)fprintf(fp,"%d ",i);fclose(fp);
    }
    //Membaca data dari file
    fp=fopen("d:\\kampus\\program\\c++\\FileIO2\\data1.dat","r");
    if(fp==NULL)cout<<"Error membuka berkas"<<endl;
```

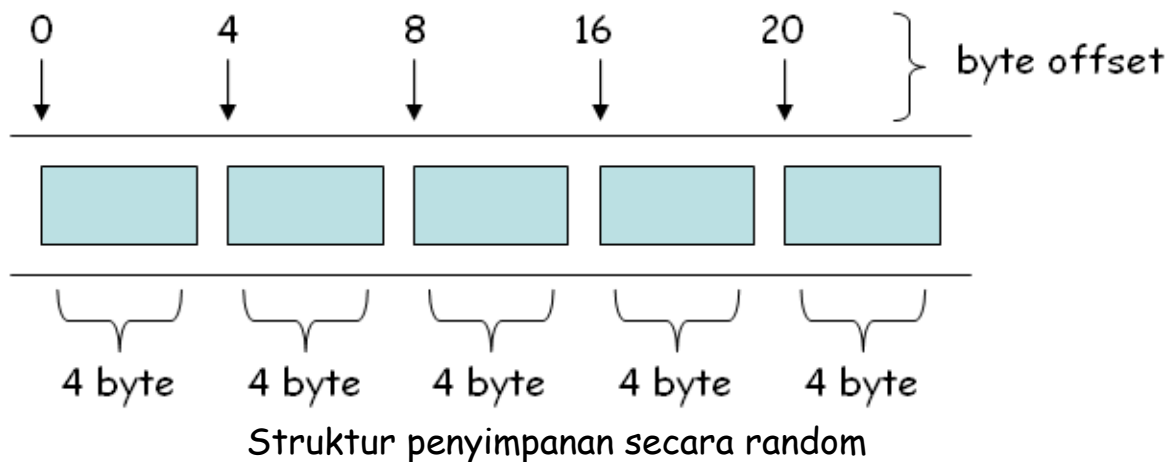
```

fp=fopen("d:\\kampus\\program\\c++\\FileIO2\\data1.dat","r");
if(fp==NULL)cout<<"Error membuka berkas"<<endl;
else
{
    i=0;
    while(!feof(fp))
    {
        fscanf(fp,"%d",&k[i]);i++;
    }
    jml_dat=i;fclose(fp);
}
for(j=0;j<jml_dat;j++)cout<<k[j]<<" ";cout<<endl;
}

```

II. PENGAKSESAN FILE SECARA RANDOM

Panjang rekaman dalam file yang diakses secara random tetap dan dapat diakses secara langsung tanpa melalui pencarian melalui rekaman lain. Oleh karena itu file yang diakses secara random cocok digunakan dalam bidang reservasi pesawat terbang, sistem perbangkan, aplikasi lain yang memerlukan pengaksesan data secara cepat terhadap data tertentu. Karena panjang rekaman bersipat tetap maka lokasi dari suatu rekaman relatif terhadap awal berkas dapat dihitung sebagai fungsi dari kunci rekaman.



Perintah-perintah baku yang dapat digunakan dalam pemrosesan arsip beruntun adalah :

<code>fopen</code>	: menyiapkan arsip untuk penulisan atau pembacaan
<code>fclose</code>	: menutup arsip
<code>fread</code>	: membaca rekaman dari arsip
<code>fwrite</code>	: menulis rekaman ke arsip
<code>fseek</code>	: menempatkan pointer pada posisi tertentu

Contoh 1 : Menulis ke dan membaca dari arsip secara random

```

//Penyimpanan & Pembacaan data secara random
#include<iostream.h>
#include<stdio.h>
FILE *fp;
void main(void)
{
    int i,j,jml_dat;

```

```

{
    int i,j,jml_dat;
    int k[100];
    //Menulis data ke arsip
    fp=fopen("d:\\kampus\\program\\c++\\FileIO3\\data1.dat","w");
    if(fp==NULL)cout<<"Error membuka file"<<endl;
    else
    {
        for(i=0;i<=10;i++)fwrite(&i,sizeof(i),1,fp);
        fclose(fp);
    }
    //Membaca seluruh data data dari arsip
    fp=fopen("d:\\kampus\\program\\c++\\FileIO3\\data1.dat","r");
    if(fp==NULL)cout<<"Error membuka file"<<endl;
    else
    {
        i=0;
        while(!feof(fp))
        {
            fread(&k[i],sizeof(int),1,fp);
            i++;
        }
        fclose(fp);
    }
    //Mencetak selauruh data ke layar
    jml_dat=i-1;
    cout<<"Jumlah data = "<<jml_dat<<endl;
    for(j=0;j<jml_dat;j++)cout<<k[j]<<" ";
    cout<<endl;
    fp=fopen("d:\\kampus\\program\\c++\\FileIO3\\data1.dat","r");
    if(fp==NULL)cout<<"Error membuka file"<<endl;
    else
    {
        //Baca data ke-0
        fseek(fp,0*sizeof(j),SEEK_SET);
        fread(&j,sizeof(j),1,fp);
        cout<<"Data ke-0 = "<<j<<endl;
        //Baca data ke-1
        fseek(fp,1*sizeof(j),SEEK_SET);
        fread(&j,sizeof(j),1,fp);
        cout<<"Data ke-1 = "<<j<<endl;
        //Baca data ke-5
        fseek(fp,5*sizeof(j),SEEK_SET);
        fread(&j,sizeof(j),1,fp);
        cout<<"Data ke-5 = "<<j<<endl;
        //Baca data ke-0
        rewind(fp);
        fread(&j,sizeof(j),1,fp);
        cout<<"Data ke-0 = "<<j<<endl;
    }
}

```

Contoh 2 : Menulis ke dan membaca dari arsip secara random

//Penyimpanan & Pembacaan data secara random, data ditulis sekaligus

#include<iostream.h>

#include<stdio.h>

```

// Menyimpan dan membaca data secara Random, data ditulis sekaligus
#include<iostream.h>
#include<stdio.h>
FILE *fp;
void main(void)
{
    int i,j,jml_dat, k[100];
    fp=fopen("d:\\kampus\\program\\c++\\FileIO4\\data1.dat","w");
    if(fp==NULL)cout<<"Error membuka berkas"<<endl;
    else
    {
        for(i=0;i<=10;i++)k[i]=i;
        fwrite(k,sizeof(i),11,fp);
        fclose(fp);
    }
    fp=fopen("d:\\kampus\\program\\c++\\FileIO4\\data1.dat","r");
    if(fp==NULL)cout<<"Error membuka berkas"<<endl;
    else
    {
        i=0;
        rewind(fp);
        while(!feof(fp))
        {
            fread(&k[i],sizeof(int),1,fp);i++;
        }
        fclose(fp);
    }
    jml_dat=i-1;
    cout<<"Jumlah data = "<<jml_dat<<endl;
    for(j=0;j<jml_dat;j++)cout<<k[j]<<" ";
    cout<<endl;
}

```

STUDY KASUS

Buatlah program C/C++ yang memberikan tampilan sbb :

M E N U P I L I H A N

-
1. Masukan Data
 2. Simpan Data
 3. Baca Data
 4. Tampilkan Data Asal
 5. Urutkan Data Secara Menaik
 6. Tampilkan Data Terurut
 7. Keluar
-

Masukan Nomor Pilihan(1...7)

Ket :

Pilihan 1 : Memasukan sejumlah data dari keyboard

Pilihan 2 : Menyimpan seluruh data kedalam sebuah arsip

Pilihan 3 : Membaca data dari arsip

Pilihan 4 : Menampilkan data

Pilihan 5 : Mengurutkan data

Pilihan 6 : Menampilkan data yang telah diurutkan