

Algoritma dan Pemrograman Lanjut

Pertemuan Ke-14 Arsip (*File*) 3



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*) 3

Deskripsi Materi : Materi ini membahas tipe data file, dan proses-proses konsolidasi, penggabungan (*merging*) dua file, *updating* data file dan *splitting*.

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.

Algoritma Konsolidasi

{Kondisi awal : ArsipMhs sudah berisi NIM dan nilai}

{Kondisi akhir : record sudah dikelompokkan berdasarkan NIM yang sama, dengan nilainya adalah nilai rata-rata}

Deklarasi

Type DataMhs : Record

< NIM : integer

Nilai : real >

ArsipMhs1, ArsipMhs2 : seqFile of DataMhs

Mhs : DataMhs

CurrentNim, JumNil, nMK : integer

Rata : real

Function Mark(Input Mhs : DataMhs) → Boolean

{tanda akhir = <999, 'xxxxx', 99.9>}

Deskripsi

Open(ArsipMhs1,1)

Open(ArsipMhs2,2)

Fread(ArsipMhs1, Mhs)

If (Mark(Mhs) = true) then // if (Mhs.NIM == 9999) then

Output(' Arsip kosong...')

Else

While (Mark(Mhs) = false) do // (Mhs.NIM <> 9999)

JumNil ← 0

nMK ← 1

currentNIM ← Mhs.NIM {record1 dari ArsipMhs1}

Repeat

JumNil ← JumNil + Mhs.Nilai

Fread(ArsipMhs1, Mhs)

nMK ← nMK + 1

Until (currentNIM <> Mhs.NIM)

Rata ← JumNil/nMK

Mhs.NIM ← currentNIM

Mhs.Nilai ← Rata

Fwrite(ArsipMhs2, Mhs)

Output(currentNIM, Rata)

Endwhile

Mhs.NIM ← 9999

Mhs.Nilai ← 999

Fwrite(Arsip2, Mhs) {mark}

EndIf

Close(ArsipMhs1)

Close(ArsipMhs2)

B. Pemrosesan Dua Arsip Beruntun

1. Merging

Merging adalah penggabungan dua buah file yang tipe recordnya sama. Untuk melakukan merging ada dua cara. Cara yang paling sederhana adalah data file kedua ditambahkan setelah record terakhir file pertama, sehingga membentuk file baru. Cara ini tidak dapat dipakai jika field kunci kedua file sudah terurut dan dikehendaki file gabungan yang field kuncinya juga terurut.

Contoh :

NIM	Nama
70001	Adi
70003	Budi
70004	Susi
70006	Anti
70008	Doni

File ArsipMhs1

NIM	Nama
70002	Yudi
70009	Eka
70010	Bima

File ArsipMhs2

NIM	Nama
70001	Adi
70002	Yudi
70003	Budi
70004	Susi
70006	Anti
70008	Doni
70009	Eka
70010	Bima

File ArsipMhs3

Algoritma MergingSambung

```
{Menggabungkan dua buah arsip beruntun yaitu ArsipMhs1 dan
ArsipMhs2, menjadi sebuah arsip baru yaitu ArsipMhs3,
dengan cara semua record arsip kedua disambungkan setelah
record terakhir arsip pertama}
{Kondisi awal : arsip pertama dan kedua sudah berisi data}
{Kondisi akhir : arsip'ketiga berisi hasil sambungan kedua arsip}
```

Deklarasi

```
Type DataMhs : Record
    <NIM : Integer,
    Nama: String>
ArsipMhs1, ArsipMhs2, ArsipMhs3 : SeqFile of DataMhs
Mhs : DataMhs
Function Mark(Input Mhs : DataMhs) → Boolean
```

Deskripsi

```
Open(ArsipMhs1, 1)
Open(ArsipMhs2, 1)
Open(ArsipMhs3, 2)
FRead(ArsipMhs1, Mhs)
While (Mark(Mhs) = false) Do
    FWrite(ArsipMhs3, Mhs)
    FRead(ArsipMhs1, Mhs)
EndWhile
FRead(ArsipMhs2, Mhs)
```

```

While (Mark(Mhs) = false) Do
    FWrite(ArsipMhs3, Mhs)
    FRead(ArsipMhs2, Mhs)
EndWhile
Mhs.NIM ← 99999
Mhs>Nama ← xxxxx
FWrite(ArsipMhs3, Mhs)
Close(ArsipMhs1)
Close(ArsipMhs2)
Close(ArsipMhs3)

```

Algoritma Merging2 {versi And}

{Menggabungkan dua buah arsip beruntun yaitu
 ArsipMhs1 dan ArsipMhs2 yang sudah terurut,
 menjadi sebuah arsip baru yaitu ArsipMhs3 yang juga terurut}
 {Kondisi awal : arsip pertama dan kedua sudah berisi data}
 {Kondisi akhir : arsip ketiga berisi hasil merging kedua arsip}

Deklarasi

```

Type DataMhs : Record
    <NIM : Integer,
    Nama : String>
ArsipMhs1, ArsipMhs2, ArsipMhs3 : SeqFile of DataMhs
Mhs1, Mhs2 : DataMhs
Function Mark(Input Mhs : DataMhs) → Boolean

```

Deskripsi

```

Open(ArsipMhs1, 1)
Open(ArsipMhs2, 1)
Open( ArsipMhs3 , 2)
FRead(ArsipMhs1, Mhs1)
FRead(ArsipMhs2, Mhs2)
While (Mark(Mhs1) = false) And (Mark(Mhs2) =false) Do
    If (Mhs1.NIM <= Mhs2.NIM) Then
        FWrite(ArsipMhs3, Mhs1)
        FRead(ArsipMhs1, Mhs1)
    Else
        FWrite(ArsipMhs3, Mhs2)
        FRead(ArsipMhs2, Mhs2)
    Endif
EndWhile
{Mark(Mhs 1) = true atau Mark(Mhs2) = true}

{salin record yg tersisa, pada ArsipMhs 1 atau ArsipMhs2}
{jikayg tersisa adalahArsipMhsI} .
While (Mark(Mhs1) = false) Do
    FWrite(ArsipMhs3, Mhs1)

```

```

        FRead(ArsipMhs1, Mhs1)
    EndWhile
    {jika yg tersisa adalah ArsipMhs2}
    While (Mark(Mhs2) = false) Do
        FWrite(ArsipMhs3, Mhs2)
        FRead(ArsipMhs2, Mhs2)
    EndWhile
    FWrite(ArsipMhs3, <99999, 'xxxxx'>)
    Close(ArsipMhs1)
    Close(ArsipMhs2)
    Close(ArsipMhs3)

```

Algoritma Merging3 {versi Or}

{Menggabungkan dua buah arsip beruntun yaitu
 ArsipMhs1 dan ArsipMhs2 yang sudah terurut,
 Menjadi sebuah arsip baru yaitu ArsipMhs3 yang juga terurut}
 {Kondisi awal : arsip pertama dan kedua sudah berisi data}
 {Kondisi akhir : arsip ketiga berisi hasil merging kedua arsip}

Deklarasi

```

    Type DataMhs : Record
        <NIM : Integer,
        Nama: String>
    ArsipMhs1, ArsipMhs2, ArsipMhs3 : SeqFile of DataMhs
    Mhs 1, Mhs2 : DataMhs
    Function Mark(Input Mhs : DataMhs) → Boolean

```

Deskripsi

```

    Open(ArsipMhs1, 1)
    Open(ArsipMhs2, 1)
    Open(ArsipMhs3, 2)
    FRead(ArsipMhs1, Mhs1)
    FRead(ArsipMhs2, Mhs2)
    While (Mark(Mhs1) = false) Or (Mark(Mhs2) = false) Do
        If (Mhs1.NIM <= Mhs2.NIM) Then
            FWrite(ArsipMhs3, Mhs1)
            FRead(ArsipMhs1, Mhs1)
        Else
            FWrite(ArsipMhs3, Mhs2)
            FRead(ArsipMhs2, Mhs2)
        Endif
    EndWhile {Mark(Mhs 1) = true dan Mark(Mhs2) = true}
    FWrite(ArsipMhs3, <99999, 'xxxxx'>)
    Close(ArsipMhs1)
    Close(ArsipMhs2)
    Close(ArsipMhs3)

```

2. Updating

Updating adalah proses editing harga suatu record (field key tidak diedit) pada file master dengan data dari file transaksi.

Berikut ini adalah algoritma umum untuk meremajakan record pada file master (bersifat beruntun, nilai field keynya terurut naik tetapi bisa tidak unik).

Artinya bahwa suatu record pada file master dapat mengalami satu atau beberapa kali peremajaan.

Contoh :

Peremajaan File Saldo tabungan pada Bank, dengan perjanjian jika nilai uang pada file update berharga negatif, berarti pengambilan, tetapi jika nilai uang pada File Update berharga positif, berarti transaksi penabungan.

File Master		File Transaksi		File New Master	
NoRek	Saldo	NoRek	Saldo	NoRek	Saldo
001	100000	001	+5000	001	103000
003	150000	001	-2000	003	151000
004	175000	003	+1000	004	176000
006	180000	004	-1500	006	185500
007	210000	004	+10000	007	207000
008	112000	004	-7500	008	112000
009	135000	006	+10500	009	135000
010	210000	006	-5000	010	210000
999	0	007	-3000	999	0
		999	0		

Contoh 1 :

Updating {judul algoritma}

{Mengubah salah satu isi field dari file master
berdasarkan data dari file transaksi lalu simpan hasil editing ke file new master}
{Kondisi awal : suatu variabel sudah bernilai isi dari field key pada posisi record yg
akan diubah, field key sudah terurut naik}
{kondisiakhir : file new master sudah berisi data dari hasil editing file master
berdasarkan file transaksi1}

Deklarasi

Type DataSaldo : Record

< NoRek : Integer,
Saldo : Longint >

Master, Transaksi, NewMaster : SeqFile of DataSaldo

Nasabah1, Nasabah2 : DataSaldo

NewSaldo : Longint

FunctionMark(InputNasabah : DataSaldo) → Boolean

Deskripsi

Open(Master, 1)

Open(Transaksi, 1)

Open(NewMaster, 2)

FRead(Master, Nasabah1)

FRead(Transaksi, Nasabah2)

While (Mark(Nasabah1) = false) Do

While (Nasabah2.NoRek < Nasabah1.NoRek)

and (Mark(Nasabah2) = false) Do

 FRead(Transaksi, Nasabah2) {skip record dari file trans }

EndWhile

If (Nasabah2.NoRek = Nasabah1.NoRek) Then

 NewSaldo ← Nasabah1.Saldo {yg akan diedit}

While (Nasabah2.NoRek = Nasabah1.NoRek)

and (Mark(Nasabah2) = false) Do

 NewSaldo ← NewSaldo+ Nasabah2.Saldo

 FRead(Transaksi, Nasabah2)

EndWhile

 Nasabah1.Saldo ← NewSaldo

 FWrite(NewMaster, Nasabah1)

Else

 FWrite(NewMaster, Nasabah1) .

Endif

 FRead(Master, Nasabah1)

EndWhile

FWrite(NewMaster, <999, 0>)

Close(Master)

Close(Transaksi)

Close(NewMaster)

Contoh2 :

Misalkan akan diubah IPK mahasiswa dengan NIM tertentu. Nilai IPK yg baru dibaca dari piranti masukan. Langkah-langkahnya adalah sebagai berikut :

- (i). Buka arsip Master untuk dibaca
- (ii). Buka arsip Sementara untuk ditulis

- (iii). Baca dan salin record dari arsip Master sampai ditemukan NIM dari mahasiswa yg IPK-nya akan diubah.
- (iv). Baca pada arsip Master setiap record yg tersisa, lalu tulis ke arsip Sementara.
- (v). Buka arsip Sementara untuk dibaca.
- (vi). Buka arsip Master untuk dituEs.
- (vii). Baca seluruh record dari arsip Sementara, lalu tuliskan ke arsip Master.

Deklarasi

Type DataMhs: Record
 < NIM : Integer,
 Nama: String,
 IPK : Real>
 Master, Sementara : SeqFile of DataMhs
Function Mark(Input Mhs : DataMhs) → Boolean

Procedure MutakhirkanData(Input x : Integer)

{Kondisi awal : x sudah berisi NIM mahasiswa yang akan diubah IPK-nya, x diisi dari piranti masukan}
 {Kondisi akhir : IPK yang NIM-nya = x, sudah diubah menjadi IPK yang baru}

Deklarasi lokal

Mhs : DataMhs
 IPKbaru : Real

Deskripsi

Open(Master, 1)
 Open(Sementara, 2)
 FRead(Master, Mhs) {baca record pertama dari file Master}
 {cari record dengan NIM = x}
While (Mhs.NIM <> x) And (Mark(Mhs) = false) Do
 FWrite(Sementara, Mhs) {catat record yg <> x }
 FRead(Mater, Mhs) {ambillagi record yg lain}
EndWhile
If (Mhs.NIM = x) Then
 {tampilkan data yg lama}
 Output(Mhs.NIM, Mhs>Nama, Mhs.IPK)
 Input(IPKBaru) {masukkan IPK yg baru}
 Mhs.IPK ← IPKBaru
 FWrite(Sementara, Mhs)
 FRead(Master, Mhs) {baca record selanjutnya}
Else
 Output(x, , tidak ada pada arsip master')
Endif
 {salin record yg terletak sesudah record x}

```

While (Mark(mhs) = false) Do
    FWrite(Sementara, Mhs)
    FRead(Master, Mhs)
EndWhile
Close(Master)
Close(Sementara)
{ buka kembali kedua arsip untuk keperluan berbeda,
yaitu untuk nmenyalin isi arsip Sennentara ke arsip Master }
Open(Sementara, 1)
Open(Master2 )
FRead(Sementara, Mhs)
While (Mark(mhs) = false) Do
    FWrite(Master, Mhs)
    FRead(Sementara, Mhs)
EndWhile
FWrite(Master, <99999, 'xxxxx', 9.99>)
Close(Sementara)
Close(Master)

```

3. Splitting

Splitting adalah pemecahan sebuah file menjadi dua atau lebih file baru. Algoritmanya tergantung dari kriteria pemecahannya.

Contoh :

Memisahkan file nilai suatu matakuliah suatu kelas berdasarkan yang nilai ≥ 55 dan yang nilai < 55 .

NIM	Nilai
-	80
-	42
-	60
-	56
-	55
-	71
	38
	65
	40
	54
99999	999

NIM	Nilai
-	80
-	60
-	56
-	55
-	71
-	65
99999	999

NIM	Nilai
-	42
-	38
-	40
-	54
99999	999

PENUTUP

Algoritma Konsolidasi adalah pengelompokan data dengan kunci yang sama yang harus diproses sebagai satu kesatuan.

Merging adalah penggabungan dua buah file yang tipe recordnya sama.

Updating adalah proses editing harga suatu record (field key tidak diedit) pada file master dengan data dari file transaksi.

Splitting adalah pemecahan sebuah file menjadi dua atau lebih file baru. Algoritmanya tergantung dari kriteria pemecahannya.

SOAL-SOAL

Implementasikan algoritma konsolidasi, merging, updating dan splitting dengan bahasa pemrograman C++.

Algoritma Spliting

Deklarasi

```
Type DataMhs : Record
    <NIM : Integer,
      Nilai: Integer>
ArsipMhs1, ArsipMhs2, ArsipMhs3 : SeqFile of DataMhs
Mhs : DataMhs
Function Mark(Input Mhs : DataMhs) → Boolean
```

Deskripsi

```
Open(ArsipMhs1, 1)
Open(ArsipMhs2, 2)
Open(ArsipMhs3, 2)
FRead(ArsipMhs1, Mhs)
While (Mark(Mhs) = false) Do
    If (Mhs.Nilai >= 55) Then
        FWrite(ArsipMhs2, Mhs)
        FRead(ArsipMhs1, Mhs)
    Else
        FWrite(ArsipMhs3, Mhs)
        FRead(ArsipMhs1, Mhs)
    Endif
EndWhile
Mhs.NIM ← 99999
Mhs.Nilai ← 999
FWrite(ArsipMhs2, Mhs)
FWrite(ArsipMhs3, Mhs)
Close(ArsipMhs1)
Close(ArsipMhs2)
Close(ArsipMhs3)
```