

CCH1A4 / Dasar Algoritma & Pemrograman

Yuliant Sibaroni M.T, Abdurahman Baizal M.Kom

KK Modeling and Computational Experiment



File Sekuensial

- ▶ Pendahuluan
- ▶ Primitif-primitif Dalam File Sekuensial
- ▶ Perekaman File Teks
- ▶ Pembacaan File Teks
- ▶ File Bertipe
- ▶ Perekaman File Bertipe
- ▶ Pembacaan File Bertipe
- ▶ Pengolahan Data

Pendahuluan

Definisi

- ▶ File adalah rekaman data yang disimpan dalam komputer dan dapat diakses kembali setelah komputer/program komputer dimatikan. Ini tentunya berbeda dengan saat kita menyimpan data dalam suatu variabel dimana data akan hilang begitu program dimatikan
- ▶ File sekuensial adalah sekumpulan rekaman yang disimpan dalam media penyimpanan sekunder yang dapat diakses secara berurutan mulai dari rekaman pertama sampai rekaman terakhir.
- ▶ Rekaman terakhir adalah rekaman fiktif yang menandai akhir dari file biasa disebut EOF (End Of File).

Pendahuluan

Jenis File

- Setiap rekaman bisa berisi tipe dasar ataupun terstruktur yang telah didefinisikan, setiap rekaman berstruktur sama
- Elemen dalam rekaman disebut ***field***.
- Ada 2 jenis file yang perlu diketahui yaitu **file bertipe** dan **file text**.
- Perbedaan utama file bertipe dan file text terdapat pada ukuran recordnya dan cara aksesnya. File bertipe memiliki panjang record yang sama sedangkan file text memiliki panjang record yang berbeda-beda. File bertipe disimpan dalam format Biner sedangkan file text disimpan dalam format ASCII.

Pendahuluan

Pengaksesan File

- ▶ Pengaksesan file meliputi operasi baca atau tulis dilakukan secara terpisah, tidak bisa dilakukan bersamaan
- ▶ Dalam awal materi ini akan dibahas pemrosesan sekuensial terhadap file text yang nantinya dapat dikembangkan untuk pemrosesan elemen secara beruntun lainnya (tabel, record dll)

Primitif-primitif Dalam File Sekuensial

Menyiapkan dan Membaca File

ASSIGN (Input NamaArsip, NamaFisik)

Arsip sekuensial yang namanya dikenal di dalam program sebagai

NamaArsip,

secara fisik diberi nama **NamaFisik**

RESET (Input NamaArsip)

Menyiapkan rekaman pada posisi awal untuk proses pembacaan file

READ (Input NamaArsip, output VarPenampung)

Membaca file Fisik (namaFisik) yang berasosiasi dengan NamaArsip, hasil pembacaannya akan disimpan dalam VarPenampung. Pembacaan dilakukan pada masing-masing elemen. Pada file teks, elemennya: char, pada file bertipe elemennya: record. Setelah dieksekusi, pointer akan dimajukan satu elemen

Variasi lain: **READLN**

Primitif-primitif Dalam File Sekuensial

Menutup dan Menulis File

CLOSE (Input NamaArsip)

Arsip sekuensial ditutup, tidak dapat diakses maupun ditulisi lagi

REWRITE (Input/Output NamaArsip)

Arsip sekuensial siap untuk direkam dari awal, isi file fisik lama akan dihapus/ditimpa

APPEND(Input/Output NamaArsip)

Arsip sekuensial siap untuk direkam dari posisi terakhir

WRITE (Input NamaArsip, <rekaman>)

Data pada <rekaman> direkam pada posisi aktual arsip.

<rekaman> :

- kalimat
- variabel yang berisi kalimat

Variasi lain : **WRITELN**

Rekaman File Text

Contoh 14.1

Program Text1

```
{Membuka file data.txt dan menuliskan teks baru ke file tersebut  
(teks lama akan dihapus), Rekaman : kalimat}
```

Kamus

```
T: text;
```

Algoritma

```
assign(T, 'data.txt')  
rewrite(T) { membuat/membuka file siap untuk ditulis}  
write (T, 'saya mahasiswa Ilmu Komputasi')  
  
{menuliskan teks ke file 'data.txt', Rekaman : kalimat}  
  
close(T) {menutup file}
```

Perekaman kalimat : 'saya mahasiswa Ilmu Komputasi' diatas cukup dituliskan dalam satu perintah write (T, 'saya mahasiswa Ilmu Komputasi')

Rekaman File Text

Contoh 14.2

Program Text2

```
{Membuka file data.txt dan menambahkan teks baru ke file tersebut (teks lama tidak dihapus), Rekaman:kalimat }
```

Kamus

T: text

Algoritma

```
assign(T,'data.txt')  
append(T)  
{membuat file siap untuk ditulis}  
write (T,'saya mahasiswa Ilkom')  
  
{menuliskan teks ke file 'data.txt'}  
  
close(T) {menutup file}
```

Perekaman kalimat : 'saya mahasiswa ilkom' diatas cukup dituliskan dalam satu perintah write (T,'saya mahasiswa Ilkom'). Kalimat ini akan ditambahkan pada baris terakhir file data.txt.

Rekaman File Text

Contoh 14.3

Program Text3

Membuka file data.txt dan siap untuk membaca file pada posisi karakter pertama

Kamus

T: text

Algoritma

```
assign(T, 'data.txt')
reset(T) { membuat file siap dibaca pada posisi awal/pertama }
readln(T) {membaca karakter pada posisi yang ditunjuk pointer
           sekarang (pertama), dan memajukan ke karakter
           berikutnya}
close(T) {menutup file}
```

Rekaman File Text

Review Latihan

Program Text5

Kamus

FT : text

Kalimat : string

Algoritma

```
assign(FT, 'data.txt')
```

```
rewrite(FT)
```

```
input(kalimat)
```

```
while kalimat<>' ' do
```

```
    writeln(FT, kalimat)
```

```
    readln(kalimat)
```

```
close(FT)
```

Program Text6

Kamus

FT : text

Kalimat : string

Algoritma

```
assign(FT, 'data.txt')
```

```
append(FT)
```

```
input(kalimat)
```

```
while kalimat<>' ' do
```

```
    writeln(FT, kalimat)
```

```
    readln(kalimat)
```

```
close(FT)
```

Apa perbedaan dua program di atas?

Rekaman File Text

Contoh 14.4

Program Text7

Kamus

T1 : text

kalimat : string

Algoritma

```
assign(T1,'contoh1.txt');
```

```
rewrite(T1);
```

```
input(kalimat) {user diminta memasukkan kalimat tertentu}
```

```
while kalimat<>' ' do
```

```
    write(T1,kalimat)
```

```
    read(kalimat)
```

```
close(T1)
```

Pada program ini, perekaman kalimat yang diinputkan user dilakukan satu-persatu (per-karakter) dengan perintah: `write(T1,kalimat)`. Perintah `read(kalimat)` , digunakan untuk memajukan pointer ke karakter selanjutnya

Pembacaan File Text

Contoh 14.5

```
Program Text8 {membaca dari file text: data.txt}
```

Kamus

```
FTeks : text
```

```
kalimat : string
```

Algoritma

```
assign(FTeks, 'data.txt')
```

```
reset(FTeks)
```

```
while not EOF(FTeks) do
```

```
    read(FTeks, kalimat)
```

```
    output(kalimat)
```

```
close(FTeks)
```

Pada program ini, pembacaan kalimat pada file **FTeks** dilakukan satu-persatu (per-karakter) dengan perintah: **read(FTeks, kalimat)**. Perintah **output(kalimat)**, digunakan untuk menampilkan karakter hasil pembacaan yang disimpan di variabel kalimat

File Sekuensial Pada File Bertipe

Pengertian

- Pada dasarnya pengaksesan file bertipe mirip dengan operasi pada file text. Perbedaan utama adalah berkaitan dengan masalah pendefinisian variabelnya
- Mengapa perlu dilakukan penyimpanan file bertipe? **Agar tipe data tidak mengalami perubahan**
- Bisa saja kita menyimpan variabel yang bertipe: array[1..N] of record dengan file text, tetapi hasil perekamannya akan menjadi file text sehingga operasi-operasi pada data (seperti penjumlahan, perhitungan rata-rata dll) menjadi tidak dapat dilakukan lagi pada file text tersebut.

type **rekaman** : <.....> sebuah type terdefinisi untuk setiap rekaman
namaArsip : **SEQFILE** of
 (*) <nama_rek> : **rekaman**
 (1) <**mark**>

File Sekuensial Pada File Bertipe

Deklarasi File Bertipe

```
type Rekaman : <.....>  
{sebuah tipe terdefinisi bernama Rekaman yang bertipe record}  
namaArsip : File of Rekaman  
{Merupakan file yang dikenali dalam eksekusi program}
```

File Sekuensial Pada File Bertipe

Contoh 14.6

Deklarasi file bertipe

```
type Rekaman : < NIM:integer, Nama : string, IPK : real>  
{sebuah tipe terdefinisi bernama Rekaman yang elemennya  
terdiri : Nim,Nama dan IPK}  
FileMhs : File of Rekaman  
{Merupakan file yang dikenali dalam eksekusi program}  
{ FileMhs ini sejenis dengan dengan variabel yang bertipe: array[]  
of Rekaman }
```


File Sekuensial Pada File Bertipe

Contoh 14.7

Misal diketahui data **nilaiMhs** sebagai berikut :

nama	nilai
Andi	8
Budi	9
Seno	8
Anton	7
Parto	5

Buatlah program untuk :

- Menyimpan/Merekam data nilaiMhs
- Membaca File bertipe
- Program untuk menghitung rata-rata nilai dari File Bertipe

Perekaman File Bertipe

Contoh 14.7.a

Procedure RekamFileBertipe

Kamus

Type Mhs : <nama: string, nilai: integer>

F : File of Mhs

NilaiMhs: array [1..1000] of Mhs

i : integer

Algoritma

assign(F, 'Fnilai.txt')

{Membuat/menyiapkan file Fnilai.txt }

rewrite(F)

input(jml)

for i ← 1 to jml do

input(NilaiMhs[i].nama)

input(NilaiMhs[i].nilai)

write(F, NilaiMhs[i])

 {perekaman dari NilaiMhs[i] ke F dilakukan disini}

close(F)

Pembacaan File Bertipe

Contoh 14.7.b

Procedure BacaFileBertipe

Kamus

Type Mhs : <nama: string, nilai: integer>

F : File of Mhs

NilaiMhs : array [1..1000] of Mhs

i : integer

Algoritma

assign(F, 'Fnilai.txt')

{Membuat/menyiapkan file Fnilai.txt }

reset(F) { membuat file siap dibaca pada posisi awal }

i ← 0

While not EOF(F) do

 i ← i+1

read(F, NilaiMhs[i])

 {file dibaca dari F disimpan di NilaiMhs[i]}

output(NilaiMhs[i].nama, ' ', NilaiMhs[i].nilai)

close(F)

Pengolahan Data

Contoh 14.7.c

Untuk program ketiga yang dibuat adalah pengolahan data. Pengolahan data yang dimaksud disini adalah, berdasarkan dari hasil pembacaan file, akan dilakukan pengolahan data yaitu berupa perhitungan nilai rata-ratanya.

Langkah-langkahnya:

1. Dari pembacaan file, maka variabel **NilaiMhs** akan berisi nilai sama seperti file Fnilai (proses ini sama seperti yang dilakukan pada program bacaFileBertipe)
2. Perhitungan nilai rata-rata dilakukan terhadap variabel **NilaiMhs**

Pengolahan Data

Contoh 14.7.c

Procedure OlahFileBertipe

Kamus

Type Mhs : <nama: string, nilai: integer>

F : File of Mhs

NilaiMhs : array [1..1000] of Mhs

i, j, jumlah : integer

Algoritma

assign(F, 'FNilai')

{Membuat/menyiapkan file FNilai }

reset(F) { membuat file siap dibaca pada posisi awal }

i ← 0

While not EOF(F) do

 i ← i + 1

read(F, NilaiMhs[i])

output(NilaiMhs[i].nama, ' ', NilaiMhs[i].nilai)

close(F)

jumlah ← 0 {tambahan langkah perhitungan : mulai dari baris ini: teks hijau}

for j ← 1 to i do

 jumlah ← jumlah + NilaiMhs[j].Nilai

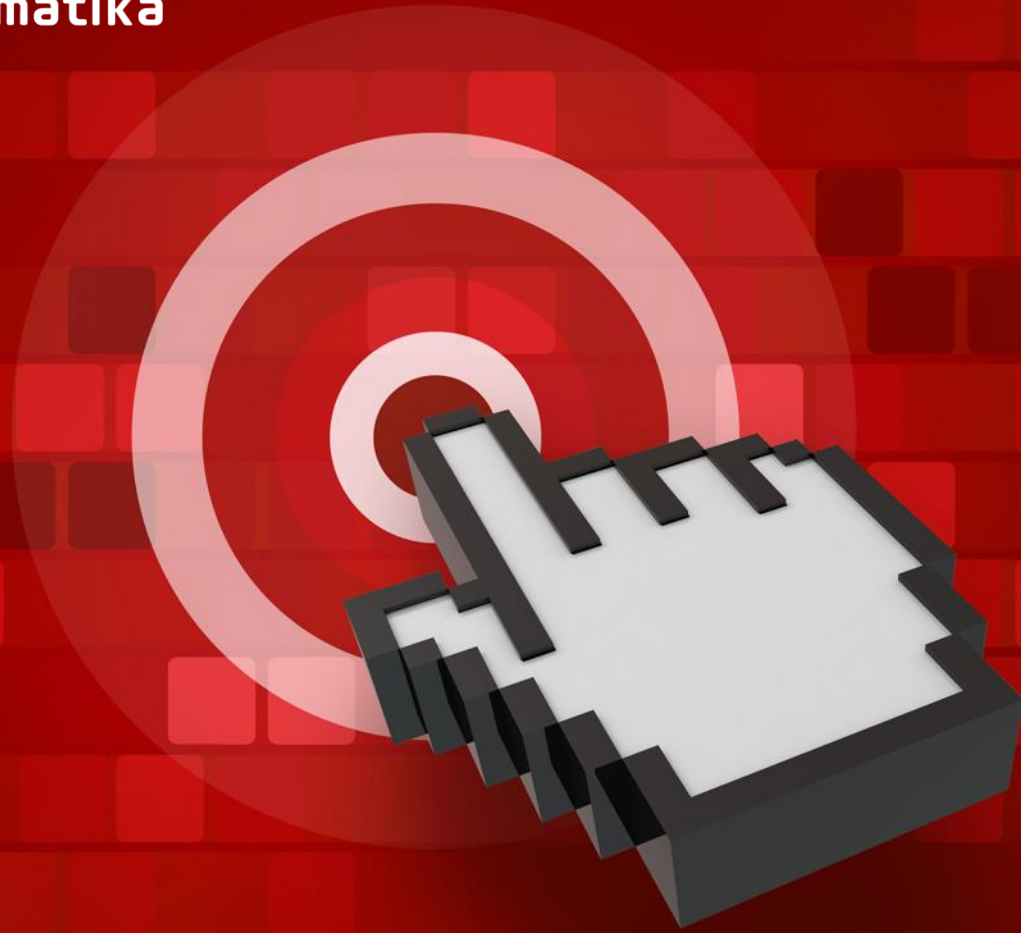
output(jumlah/i) {rata-rata ditampilkan ke layar}

Referensi

- ▶ Inggriani Liem, Diktat Kuliah IF223 Algoritma Dan Pemrograman, Jurusan Teknik Informatika Bandung, 1999
- ▶ Rinaldi Munir, Algoritma dan Pemrograman Dalam Bahasa Pascal dan C edisi ke-3", penerbit Informatika 2005
- ▶ Agoes Soehianie, " Belajar dengan cepat dan mudah TURBO PASCAL 4.0/5.0/5.5 ", Elex Media Komputindo, 1990



Fakultas Informatika
School of Computing
Telkom University



THANK YOU