

Skema Standar (Bag. 4): Implementasi File Sekuensial di Python

Tim Pengajar IF1210

Sekolah Teknik Elektro dan Informatika

Translasi ke Python (1)

- Type rekaman dapat didefinisikan sebagai data dalam type dasar atau type bentukan dengan struktur tuple

Notasi Algoritmik	Python
<u>type</u> <i>rekaman</i> : ... { sebuah type terdefinisi untuk setiap rekaman }	# type <i>rekaman</i> : ... # sebuah definisi type untuk # setiap rekaman
<i>NamaArsip</i> : SEQFILE of (*) <i>nama_rek</i> : <i>rekaman</i> (1) <i>mark</i>	# <i>NamaArsip</i> : seqfile of rekaman # mark didefinisikan sebagai # konstanta <i>mark</i> = (...) # <i>nama_rek</i> : <i>rekaman</i>

Translasi ke Python (2)

Notasi Algoritmik	Python
<u>assign</u> (<i>namaArsip</i> , <i>namaFisik</i>)	Bagian dari perintah open
<u>open</u> (<i>namaArsip</i> , <i>nama_rek</i>)	<pre>namaArsip = open(namaFisik) nama_rek = ... #perintah read</pre> <pre>namaArsip = open(namaFisik, 'r') nama_rek = ... #perintah read</pre>
<u>rewrite</u> (<i>namaArsip</i>)	<pre>namaArsip = open(namaFisik, 'w')</pre>
<u>read</u> (<i>namaArsip</i> , <i>nama_rek</i>)	<pre># membaca 1 baris text/data nama_rek = namaArsip.readline()</pre> <pre># membaca n buah character nama_rek = namaArsip.read(n)</pre>
<u>write</u> (<i>namaArsip</i> , <i>rekaman_baru</i>)	<pre>namaArsip.write(rekaman_baru)</pre>
<u>close</u> (<i>namaArsip</i>)	<pre>namaArsip.close()</pre>
EOP	Harus didefinisikan nilai tertentu sebagai mark . EOP dapat didefinisikan sebagai suatu function untuk memeriksa apakah sebuah nilai tertentu adalah mark.

Secara default, Python membaca file dalam **mode text**. Deskripsi di atas adalah untuk file text. Python menyediakan mode untuk membaca file *binary*. Dipersilakan mahasiswa untuk mempelajari secara mandiri.

Contoh-1 (1)

- Translasikan program untuk membaca file dengan skema repeat-until (slide 21 materi Pemrosesan File Sekuensial) ke Python

Contoh-1 (2)

```
# Program BacaText2a
# Membaca sebuah text file diakhiri dengan MARK berupa '.'
# dan menuliskan isinya ke layar.
# Asumsi file dataku.txt ada dan sudah terisi, minimum hanya berisi '.'
# (artinya file kosong).

# KAMUS
# f : SEQFILE of char
# cc : char
# mark = '.'

# ALGORITMA
f = open("dataku.txt",'r')
cc = f.read(1) # First-Elmt, baca 1 karakter
if (cc == '.'):
    print("File kosong")
else: # not EOP(cc), file tidak kosong
    while True:
        print(cc,end='')
        cc = f.read(1)
        if (cc == '.'):
            break
f.close()
```

Contoh-2 (1)

- Translasikan program untuk menulis ke file teks dengan skema while-do (slide 26 materi Pemrosesan File Sekuensial) ke Python

Contoh-2 (2)

```
# Program IsiTeks
# Membaca sejumlah masukan character dari user sampai dimasukkan nilai '.'
# dan menyimpannya ke dalam file teks dataku.txt

# KAMUS
# f : SEQFILE of char
# C1 : char
# mark = '.'

# ALGORITMA
f = open("dataku.txt", 'w')
C1 = input()[0]      # First-Elmt
while (C1 != '.'):
    f.write(C1)       # Proses current elmt
    C1 = input()[0]  # Next-Elmt
# C1 = '.'

# tulis mark di akhir file
f.write('.')

f.close()
```

Membaca/Menulis File csv

- Python menyediakan beberapa library untuk membaca file dalam format standar misalnya csv, misalnya dari library csv.
- Akan diberikan pemrosesan file dengan type rekaman yang berbentuk type bentukan (*tuple*) dengan data disimpan dalam file csv
 - Dimungkinkan ada format standar lain dan cara pemrosesan lain, dipersilakan dipelajari mandiri

Contoh-3 (1)

- Translasikan program pemrosesan sekuensial data pada file untuk menghitung nilai rata-rata (slide 28-31) ke Python
- Beberapa adaptasi:
 - Data disimpan dalam file **MHS.csv**
 - **mark** didefinisikan sebagai konstanta (**“99999999”,99**)
 - type **rekamanMHS** didefinisikan sebagai tuple (*NIM,nilai*)
 - Pengecekan mark dilakukan dengan function **EOP**

```
# Program NilaiRataRata
# Membaca data dalam MHS.csv dan menghasilkan nilai rata-rata
# dari semua mahasiswa. MHS.csv diasumsikan sudah isi,
# minimum berisi record ("99999999",99) (mark).
```

```
import csv
```

Import library csv

```
# KAMUS
```

```
# mark
```

```
mark = ("99999999",99)
```

```
# type rekamanMHS : (NIM : string, nilai : int)
```

```
# ArsipMhs : SEQFILE of rekamanMHS
```

```
# reader = hasil pembacaan data csv
```

```
# rekMhs : rekamanMHS
```

```
# SumNil : int # jumlah nilai
```

```
# JumMHS : int # jumlah mahasiswa
```

Function EOP

```
def EOP(rekMHS):
```

```
# menghasilkan true jika RekMHS = mark
```

```
    # Algoritma
```

```
    return (rekMHS[0] == mark[0] and rekMHS[1] == mark[1])
```

```
# ALGORITMA
```

```
...
```

...

ALGORITMA

Buka file csv dan menyiapkan reader

ArsipMHS = open ("MHS.csv", 'r')

reader = csv.reader(ArsipMHS, delimiter=',') # baca semua data dari file csv

baca rekaman pertama dan simpan sebagai tuple rekaman di rekMHS

row = next(reader)

rekMHS = (row[0], int(row[1])) # memindahkan data row ke rekMHS
komponen nilai=row[1], dikonversi mjd int

if (EOP(rekMHS)):

print("Arsip kosong")

else: # File tidak kosong

... # Proses file tidak kosong - next slide

ArsipMHS.close()

Fungsi csv.reader
untuk membaca
seluruh isi file csv
dan menyimpannya
di reader

fungsi next (...) untuk
membaca 1 baris dari hasil
pembacaan file csv. Setiap
komponen hasil
pembacaan diakses spt
mengakses elemen list.

...

```

if (EOP(rekMHS)):
    print("Arsip kosong")
else: # File tidak kosong
    SumNil = 0 # Inisialisasi
    JumMhs = 0 # Inisialisasi
    i = 1
    while True:
        # Proses current elmt
        SumNil = SumNil + rekMHS[1] #rekMHS[1] = nilai
        JumMhs = JumMhs + 1

        # baca next rekaman
        row = next(reader)
        rekMHS = (row[0],int(row[1])) # lihat komentar di atas

        if (EOP(rekMHS)):
            break

    # Terminasi: Tulis rata-rata
    rata = round(SumNil/JumMhs,2)
    print("Rata-rata = %.2f" % rata)

```

...

Contoh-3 (2)

- Berikut adalah contoh pemrosesan sekuensial lain yang juga sering dipakai, menggunakan loop traversal (for)

```

...

# Baca file dan proses
ArsipMHS = open ("MHS.csv",'r')
reader = csv.reader(ArsipMHS,delimiter=',')

# baca setiap rekaman sambil menentukan apakah file kosong
# file kosong adalah jika rekaman hanya tersebut 1 rekaman
# dan rekaman tersebut pasti mark
SumNil = 0
JumMhs = 0
for row in reader: # row adalah 1 baris data di reader
    rekMHS = (row[0],int(row[1]))
    if (not(EOP(rekMHS))):
        SumNil = SumNil + rekMHS[1]
        JumMhs = JumMhs + 1

# cek file kosong dengan JumMhs
if (JumMhs == 0):
    print("File kosong")
else: # JumMhs != 0
    rata = round(SumNil/JumMhs,2)
    print("Rata-rata = %.2f" % rata)

ArsipMHS.close()

```

Contoh-4 (1)

- Buatlah program untuk membaca data bertipe rekamanMHS (seperti pada slide sebelumnya) dan menyimpannya dalam suatu file csv bernama **MHS2.csv**

```
... # definisikan judul dan kamus program

# Baca file dan proses
ArsipMHS = open ("MHS2.csv",'w',newline='')
writer = csv.writer(ArsipMHS,delimiter=',')

# baca masukan dari user dan bentuk menjadi tuple
nim = input()
nilai = int(input())
rekMHS = (nim,nilai)

while (not(EOP(rekMHS))):
    writer.writerow(rekMHS) # tulis rekMHS ke file
    nim = input()
    nilai = int(input())
    rekMHS = (nim,nilai)

# tulis mark
writer.writerow(mark)

ArsipMHS.close()
```

Siap mengisi file MHS2.csv. **newline=""** memastikan bhw antar baris tidak akan ada tambahan newline.

End of File

- Semua pembahasan dan contoh dalam materi kuliah ini mengasumsikan dibutuhkan mark khusus
- Bahasa pemrograman menyediakan mekanisme untuk mendeteksi akhir file atau *End Of File* (EOF) secara otomatis
 - Di Python, EOF dideteksi jika perintah **read()** atau **readline()** menghasilkan *empty string*

Contoh-5 (1)

- Contoh persoalan:
 - Diketahui file puisi.txt dengan isi berikut:

```
aku ingin mencintaimu dengan sederhana  
dengan kata yang tak sempat diucapkan  
kayu kepada api yang menjadikannya abu
```

- Baca isi file dan tuliskan setiap baris satu per satu ke layar.
- Pembacaan dihentikan jika ditemukan EOF
- Jika file kosong, tidak dilakukan apa pun

Contoh-5 (2)

Alternatif-1:

- Dibaca satu per satu dengan perintah **readline()**

```
# Program BacaFile
# Membaca isi puisi.txt dan menuliskan
# setiap kalimat ke layar
# Kamus
# f : file of string
# baris : string
# Algoritma
f = open("puisi.txt", 'r')

baris = f.readline()
while (baris):
    print(baris.rstrip())
    baris = f.readline()
f.close()
```

```
# Program BacaFile
# Membaca isi puisi.txt dan menuliskan
# setiap kalimat ke layar
# Kamus
# f : file of string
# baris : string
# Algoritma
f = open("puisi.txt", 'r')
while True:
    baris = f.readline()
    if not baris:
        break
    print(baris.rstrip())
f.close()
```

Contoh-5 (2)

Alternatif-1:

- Dibaca satu per satu dengan perintah **readline()**

```
# Program BacaFile
# Membaca isi puisi.txt satu baris per baris
# setiap kalimat k
# Kamus
# f : file of string
# baris : string
# Algoritma
f = open("puisi.txt", 'r')

baris = f.readline()
while (baris):
    print(baris.rstrip())
    baris = f.readline()
f.close()
```

rstrip dipakai untuk menghilangkan karakter newline '\n' di sebelah kanan string. Mengapa diperlukan? Pelajari pula **lstrip** dan **strip**.

```
# Program BacaFile
# Membaca isi puisi.txt satu baris per baris
# setiap kalimat k
# Kamus
# f : file of string
# baris : string
# Algoritma
f = open("puisi.txt", 'r')
while True:
    baris = f.readline()
    if not baris:
        break
    print(baris.rstrip())
f.close()
```

Menggunakan skema iterate-stop

Contoh-5 (3)

Alternatif-2

- Menggunakan **readlines()** untuk membaca seluruh baris secara langsung

Alternatif-3

- Menggunakan file object untuk mengiterasi isi file

```
# Alternatif-2
# ... Header sama dengan slide sblmnya
# Algoritma
f = open("puisi.txt", 'r')
puisi = f.readlines()
for baris in puisi:
    print(baris.rstrip())
f.close()
```

```
# Alternatif-3
# ... Header sama dengan slide sblmnya
# Algoritma
f = open("puisi.txt", 'r')
for baris in puisi:
    print(baris.rstrip())
f.close()
```

Selamat belajar