

# Tugas 3: Magical File Storage

CSGE602040 - Struktur Data & Algoritma (Data Structures & Algorithms) @ Fakultas Ilmu Komputer Universitas Indonesia, Semester Gasal 2017/2018

Nama berkas kode sumber:	SDA<NPM>T3.java
Batas waktu eksekusi program:	3 detik / kasus uji
Batas memori program:	256 MB / kasus uji
Batas waktu pengumpulan kode sumber:	Minggu, 10 Desember 2017 pukul 23.55

"Rejoice, young man. Your wish will finally come true."  
-Kotomine Kirei-

Gudako kini menjadi seorang magus terkenal di Organisasi Clock Tower setelah berhasil menjalankan berbagai tugas yang diberikan kepadanya. Sebagai penghargaan atas dedikasinya, Gudako diberikan akses menuju bagian terlarang dari perpustakaan kuno yang berisikan berbagai gulungan sihir yang sangat kuat. Menyadari bahwa sihir-sihir yang ada terlalu kuat untuk dikuasai manusia dan bisa menyebabkan akhir dunia bila jatuh ke tangan yang salah, Gudako memutuskan untuk membakar seluruh bagian terlarang dari perpustakaan tersebut. Namun sebelum itu, Gudako ingin mendigitalisasi dan menyimpan gulungan tersebut untuk dirinya sendiri. Anda sebagai pengikut setia Gudako diberikan sebuah **tugas terakhir** oleh Gudako, yaitu membuat sebuah File Storage untuk Gudako menyimpan gulungan tersebut.



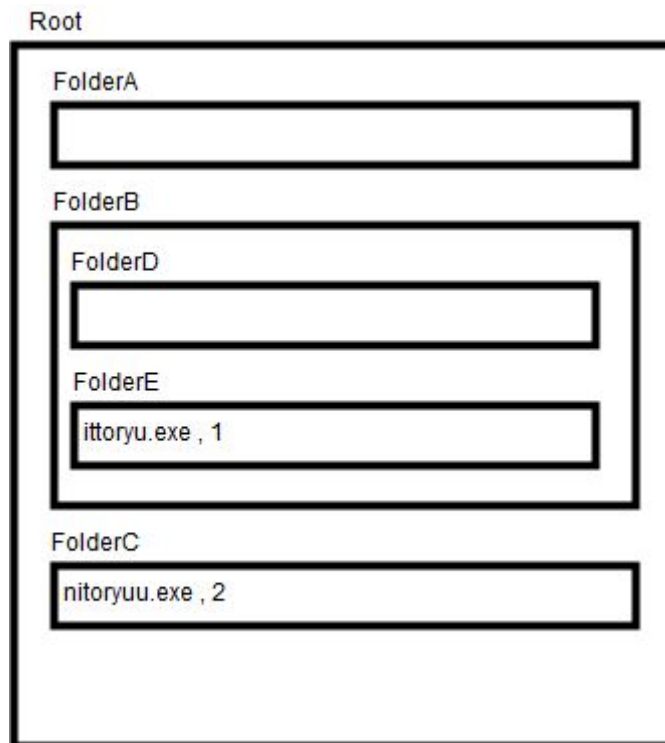
Gambar 1. Ilustrasi salah satu sihir yang tersimpan di perpustakaan kuno  
(Sumber : [https://pbs.twimg.com/media/DE-7C\\_TUQAE9NEL.jpg](https://pbs.twimg.com/media/DE-7C_TUQAE9NEL.jpg))

# File Storage

File Storage yang diinginkan yang diinginkan oleh Gudako terdiri dari 2 komponen, yaitu **Folder** dan **File**. Agar tidak membingungkan, setiap File dan Folder memiliki nama masing-masing. File berisikan gulungan yang ingin disimpan oleh Gudako. Sedangkan, sebuah Folder dapat menampung File ataupun Folder lainnya, namun **tidak bisa menampung keduanya secara bersamaan**. Dengan kata lain, suatu Folder dapat berisi banyak Folder namun tidak terdapat File, serta sebaliknya. Karena itu jika sebuah File **A** dimasukkan ke dalam Folder **B** namun Folder B berisikan Folder lain, maka File A akan dimasukkan ke dalam Folder yang berada di dalam Folder B, hal ini dilakukan terus-menerus hingga tidak ada Folder lagi di dalam sebuah Folder. Folder yang dipilih adalah Folder urutan **pertama** dari urutan **leksikografis** berdasarkan nama Folder tersebut. Sebaliknya, jika suatu Folder **X** dimasukkan ke dalam Folder **Y** tetapi Folder **Y** telah berisikan sebuah File. Maka Folder **X** dimasukkan ke dalam Folder **Y** dan semua File di dalam Folder **Y** dipindahkan ke dalam Folder **X**.

Selain nama, sebuah File memiliki tipe. Sebuah folder hanya bisa berisikan **satu** tipe File saja. Jika sebuah File bertipe M dimasukkan ke dalam sebuah Folder J yang sudah berisikan File bertipe N, maka File bertipe M akan dimasukkan ke dalam Folder urutan **selanjutnya** dari urutan **leksikografis** berdasarkan nama Folder tersebut, hal ini dilakukan terus-menerus hingga menemukan folder kosong atau berisikan File bertipe sama. Jika Folder selanjutnya berisi Folder lainnya, maka dimasukkan ke dalam Folder yang ada di dalamnya. Jika tidak ada Folder selanjutnya alias Folder terakhir dalam urutan leksikografis, maka Folder yang menjadi Folder selanjutnya adalah Folder pertama dalam urutan leksikografis alias **memutar**. Jika setelah 1 kali putaran dan tetap tidak ditemukan Folder kosong atau berisikan File bertipe sama, maka naik ke Folder di atasnya lalu dimasukkan ke Folder selanjutnya. Jika di seluruh File Storage tidak ada Folder Kosong atau berisikan File bertipe sama, maka File tidak jadi dimasukkan.

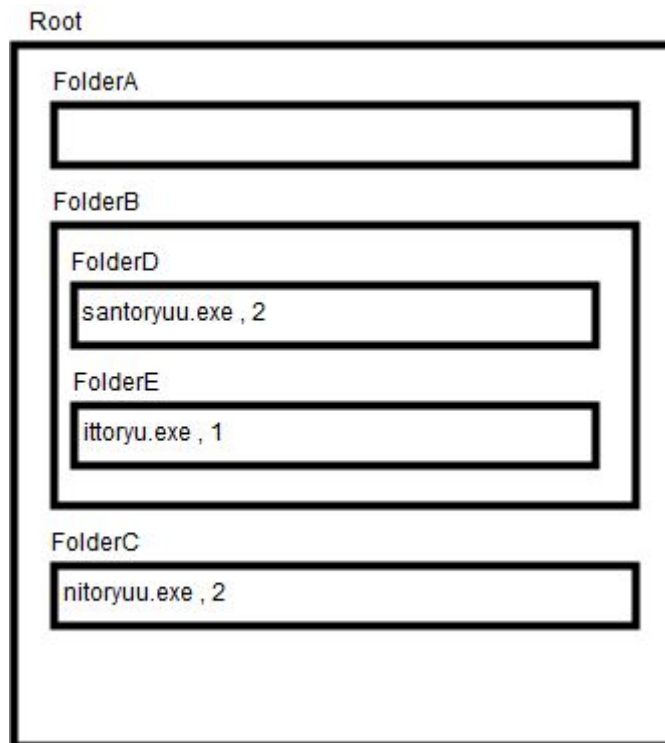
Setiap File memiliki ukuran tersendiri, sedangkan ukuran Folder adalah 1 ditambah dengan ukuran seluruh File dan/atau Folder yang ada di dalamnya. Berikut merupakan contoh ilustrasi dari File Storage :



Gambar 2. Ilustrasi File Storage

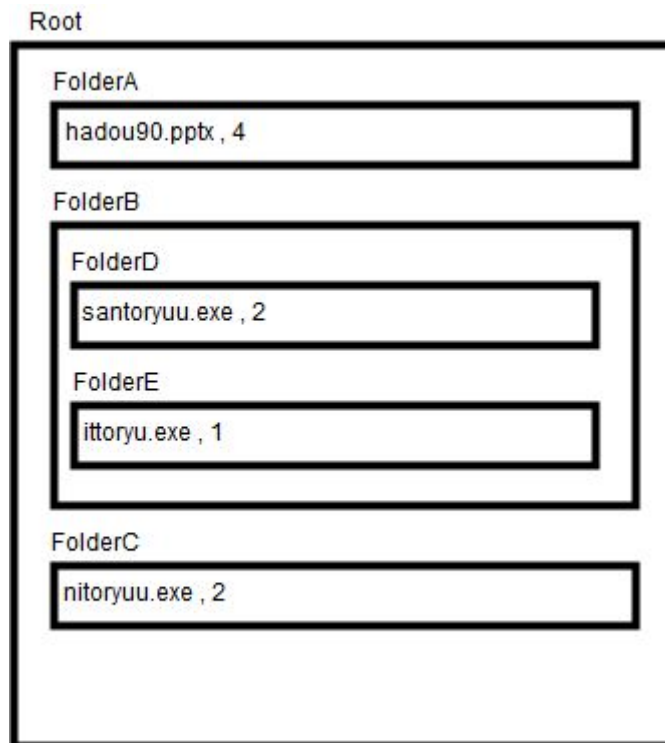
**Keterangan :**

- Jika File dengan nama santoryuu yang bertipe exe (santoryuu.exe) dengan ukuran 2 dimasukkan ke dalam FolderB, maka santoryuu.exe tidak berakhir di FolderB (karena FolderB berisi Folder lainnya), melainkan pada FolderD (karena secara leksikografis lebih awal daripada FolderE)



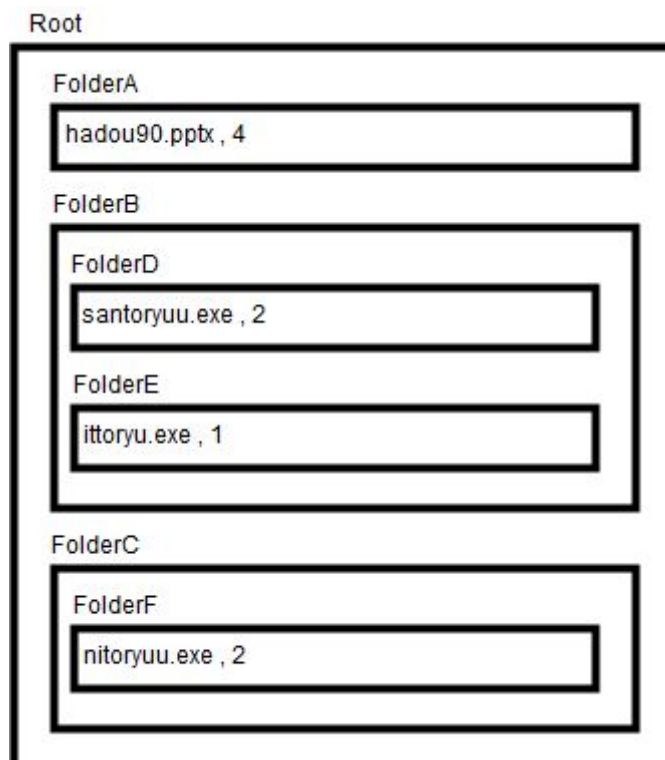
Gambar 3. Ilustrasi kondisi File Storage setelah meng-insert santoryuu.exe

- Jika File dengan nama hadou90 yang bertipe pptx (hadou90.pptx) dengan ukuran 4 dimasukkan ke dalam FolderC, maka hadou90.pptx tidak dimasukkan di FolderC (karena FolderB berisi File yang tipenya berbeda dengan hadou90), melainkan FolderA (karena tidak ada Folder selanjutnya setelah FolderC, maka dimasukkan ke dalam Folder pertama dalam urutan leksikografis)



Gambar 4. Ilustrasi kondisi File Storage setelah meng-insert hadou90.pptx

- Jika FolderF dimasukkan ke dalam FolderC, maka nitoryuu.exe berada di dalam FolderF



Gambar 5. Ilustrasi kondisi File Storage setelah meng-insert FolderF

## Input dan Output :

Program akan menerima input untuk *command* seperti di bawah ini hingga *End of File*:

### 1. Basic Operation

#### 1. add A B

Memasukkan Folder A ke dalam Folder B  
e.g. add folderA root

Perintah ini tidak memiliki keluaran

#### 2. insert A.B C D

Memasukkan File A bertipe B dengan ukuran C ke dalam Folder D  
Jika operasi gagal, tidak mengeluarkan apapun  
e.g. insert santoryuu.exe 5 root

Keluaran:

"A.B added to X"

X adalah nama Folder tempat A akhirnya dimasukkan

#### 3. remove A

Menghapus File/Folder dengan nama A  
Jika A adalah Folder, maka hapus seluruh isinya juga  
Jika A adalah File, maka hapus semua File yang memiliki nama sama  
e.g. remove santoryuu

Keluaran:

Jika A adalah Folder

"Folder A removed"

Jika A adalah File

"X File A removed"

X adalah jumlah File A yang berhasil dihapus

#### 4. search A

Mencari semua File/Folder dengan nama A di dalam File Storage  
e.g. search KongGun

Keluaran:

> root

> folderA

> KongGun.docx

> folderB

> KongGun.xlsx

Cetak semua path dari menuju File/Folder yang dicari

Jika yang dicari adalah File, cetak juga tipe Filenya

## 5. **print A**

Mencetak isi dari Folder dengan nama A  
Setiap Folder/File yang berada 1 level lebih dalam dicetak dengan indentasi **2 spasi lebih kanan** dari Folder di atasnya.  
e.g. print root

Keluaran:

```
> root 9
  > folderA 8
    > nihilego.exe 3
    > santoryuu.exe 4
```

## 2. **Advanced Operation (BONUS)**

### 1. **recommend A B**

Mencari semua File/Folder yang memiliki prefix A di dalam Folder B  
e.g. recommend fold root

Hiraukan semua File/Folder yang tidak berada langsung di dalam Folder B

Keluaran:

```
> folderA
> folderB
> folderC
```

folderD, folderE, dan folderF tidak dicetak karena tidak berada langsung di dalam folder root

Contoh output mengacu pada kondisi File Storage di Gambar 5

### 2. **move A B**

Memindahkan Folder dengan nama A ke dalam Folder B  
e.g. move FolderA root

Jika B berada di dalam A secara langsung maupun tidak langsung, batalkan operasi, lalu keluarkan :  
"B is inside A"

### 3. **cut A B C**

Memindahkan semua File dengan nama A yang berada di dalam Folder B ke dalam Folder C  
e.g. cut santoryuu FolderA root

Keluaran:

"X File A moved to C"

X adalah jumlah File A yang berhasil dipindahkan

### **Hal-Hal yang perlu diperhatikan**

- **Setiap perintah dipastikan akan valid (semua Folder dan File yang menjadi input dijamin ada)**

- Folder terbesar yang telah ada sejak awal bernama root dan tidak ada perintah remove root
- Tidak ada duplikasi nama Folder, namun File dan ukurannya mungkin ada duplikasi
- Memungkinkan File dengan nama sama memiliki tipe yang berbeda
- Semua nama folder dan file hanya terdiri dari 1 kata alfanumerik
- Semua pencarian dan/atau pencetakan dilakukan secara terurut berdasarkan leksikografis
- Jika nama sama, urutkan berdasarkan tipe, lalu ukuran

#### Contoh Masukan 1:

```
add folderA root
add folderB root
add folderD folderA
add folderE folderA
add folderF root
add folderG root
add folderC root
insert ittoryuu.exe 1 folderE
insert nitoryuu.exe 2 folderB
insert santoryuu.exe 2 folderC
print root
```

#### Contoh Keluaran 1:

```
ittoryu.exe added to folderE
nitoryu.exe added to folderB
santoryu.exe added to folderC
> root 13
  > folderA 4
    > folderD 1
    > folderE 2
      > ittoryu.exe 1
  > folderB 3
    > nitoryu.exe 2
  > folderC 3
    > santoryu.exe 2
  > folderF 1
  > folderG 1
```

#### Contoh Masukan 2:



```
add folderA root
add folderB root
add folderD folderA
add folderE folderA
add folderF root
add folderG root
add folderC root
insert ittoryuu.exe 1 folderE
insert nitoryuu.exe 2 folderB
insert santoryuu.exe 2 folderC
insert haki.pptx 3 folderE
print root
remove folderB
search haki
```

### Contoh Keluaran 2:

```
ittoryu.exe added to folderE
nitoryu.exe added to folderB
santoryu.exe added to folderC
haki.pptx added to folderD
> root 16
  > folderA 7
    > folderD 4
      > haki.pptx 3
    > folderE 2
      > ittoryu.exe 1
  > folderB 3
    > nitoryu.exe 2
  > folderC 3
    > santoryu.exe 2
  > folderF 1
  > folderG 1
Folder folderB removed
> root
  > folderA
  > folderD
    > haki.pptx
```

**“This might be your final trial,  
but your real epic has just started”  
-PS-**