

# File / File Handling

- ↳ kumpulan data + yg tersimpan
- ↳ tempat data
- ↳ integral fungsi komputer
- ↳ berkas

file  
name file

txt, pdf, mp4  
file extension — docx  
extension

↳ akses file by path

Absolute

Relative

C:/Users/Kunara/file.txt

./file.txt

string path = @ "C://Users//Kunara//file.txt" ;

~~n~~ = new line  
~~A~~ = tab  
~~/u~~

String x = "hello";

int y = 2;

String result = String.Format("{0} my age is {1}", x, y);

String result = \$" {x} my age is {y}";

File Stream → using ( FileStream fs = new FileStream ("path.txt", FileMode.Files) )

↓  
new FileStream ("path.txt", FileMode. \_\_\_\_\_ , FileAccess. \_\_\_\_\_ );

Open → }

FileMode.CreateNew → Create new file, if file exist, exception

FileMode.Create → —||—, if file exist, replace

FileMode.Open → Open file, if file not exist, exception

FileMode.OpenOrCreate → —||—, —||—, create new file

FileMode.Append → FileMode.Open, append data only, seeks end, if not exist, create file

FileMode.Truncate → FileMode.Open, reset all data to 0 bytes

File Stream → using ( FileStream fs = new FileStream ("path.txt", FileMode.Create) )

↳ new FileStream ("path.txt", FileMode. File Access );

File Access. Write → cuman tulis

File Access. Read → cuman baca

File Access. ReadWrite → baca / tulis

new FileStream("path.txt", FileMode. —, FileAccess. —, FileShare. —)

File Share. None → Another process cant open/write file

File Share. Read

File Share. Write

File Share. Inheritable

File Share. Delete

using (FileStream fs = new FileStream(@"./createFile.txt", FileMode.CreateNew, FileAccess.Write)

{ string toWrite = "Hello"

byte[] mybytes = Encoding.UTF8.GetBytes(toWrite);

fs.Write(mybytes, 0, mybytes.Length);

↓  
buffer data  
to be written

↓  
offset  
initial point

↓  
count

}  
  
Encoding

→ ASCII

→ UTF8 → A . . . . .

→ UTF7

→ UTF16 → UTF8 + emoji

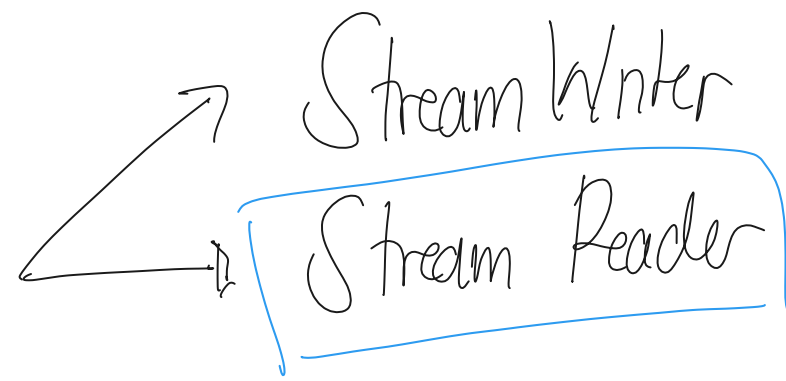
byte →

char

```
using (FileStream fs = new FileStream(@"./createFile.txt", FileMode.Open, FileAccess.Read)
{
    byte[] mybytes = new byte[ fs.Length ];
    fs.Read ( mybytes, 0, fs.Length );
    String result = Encoding.UTF8.GetString ( mybytes );
    result.Dump();
}
```

↳ property

byte → char

File Stream  Stream Writer  
Stream Reader

fs.Seek(150, SeekOrigin.Begin);

```
using (var fs = new FileStream(@"path.txt", FileMode, FileAccess))  
{  
    using (var sw = new StreamWriter(fs))  
    {  
        fs.Seek(150, SeekOrigin.Begin);
```

 sw.WriteLine("Tengah");

 WriteLineAsync()

```
}  
await sw.WriteLineAsync(" ");
```