File / File Handling

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

file
namafile . fileextension
txt, pdf, mp4, docx
extension

→ akses file by path

Absolute	Relative
C:/Users/Kinara/file.txt	./file.txt
string path = @"C:\\Users\\Kinara\\file.txt";	

\n = new line
\t = 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}";
```

FileStream ⟶ using (FileStream fs = new FileStream("path.txt", FileMode.FileAccess)
{

Dispose ⟶ }

new FileStream("path.txt", FileMode.____, FileAccess.____);

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

FileStream ⟶ using (FileStream fs = new FileStream ("path.txt", FileMode.FileAccess)
{

Dispose ⟶ }

↳ new FileStream ("path.txt", FileMode. ____ , FileAccess. ____);

FileAccess.Write ⟶ cuman nulis

FileAccess.Read ⟶ cuman baca

FileAccess.ReadWrite ⟶ baca / tulis

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

FileShare.None → Another process cant open/write plz

FileShare.Read

FileShare.Write

FileShare.Inheritable

FileShare.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);
}
```

123

- mybytes → bufferdata to be written
- 0 → offset inital point
- mybytes.Length → 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();
}
```

fs.Length → property

GetString(mybytes) → byte → char

FileStream → StreamWriter
FileStream → StreamReader

```
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 );
        1) sw.WriteLine ( "Tengah" );
    }
}
```

Begin
Current
End

WriteLineAsync( )

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