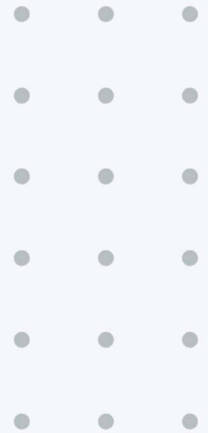




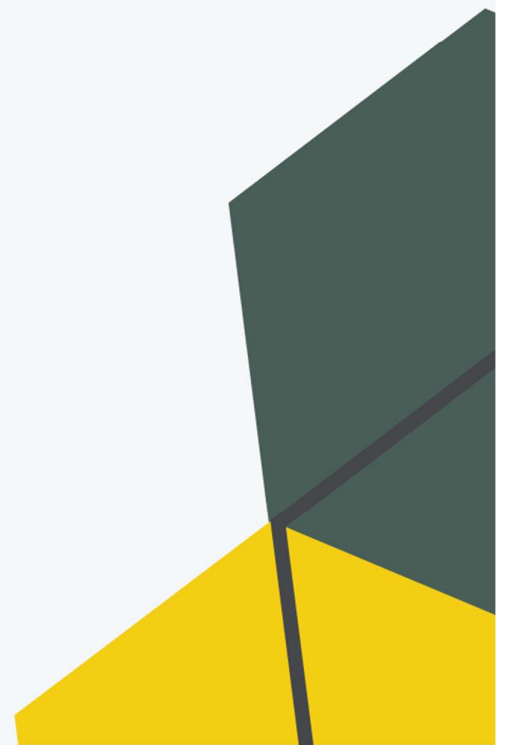
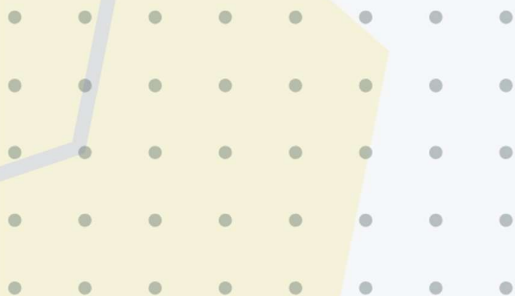
TEKNIK
INFORMARTIKA



MODUL PRAKTIKUM

PEMROGRAMAN BERBASIS OBJEK

Pertemuan 1
Python



A. Python

Python adalah pemrograman yang lebih simple daripada C++, karena banyak library yang bisa digunakan disini. Biasanya python digunakan untuk Machine Learning, Data Engineering, Scripting, dan masih banyak lagi

B. Proses Instalasi

Kalian bisa mendownload python melalui website ini <https://www.python.org/downloads/> dan ikuti step by step nya.



Jangan lupa ceklis seperti gambar diatas, agar masuk ke PATH

Setelah diinstall, kalian bisa cek apakah python nya masuk ke dalam path dengan perintah `py --list` atau `python --version`

```
PowerShell 7.5.0
PS C:\Users\akhda> py --list
-V:3.13 *      Python 3.13 (64-bit)
-V:3.12       Python 3.12 (64-bit)
-V:3.10       Python 3.10 (64-bit)
PS C:\Users\akhda> _
```

Kalo ada error, berarti belum masuk di PATH

C. Tipe Data

| | |
|------------------|------------------------------|
| Text | str |
| Numerik | int, float, complex |
| Urutan | list, tuple, range |
| Pemetaan/Mapping | dict |
| Set | set, frozenset |
| Boolean | bool |
| Biner | bytes, bytearray, memoryview |
| None | NoneType |

Tipe Data di python itu langsung di set tanpa perlu di deklarasi seperti di C++

```
1 x = "Hello, World" # strint
2 x = 3 # int
3 x = 1.412 # float
4 x = True # bool
5 x = [1,2,3,4,5] # list
```

Kalian juga bisa Casting atau mengkonversi tipe data seperti dibawah ini

```
1 int_x = int(input("Masukan bilangan bulat"))
2 float_x = float(input("Masukan bilangan pecahan"))
3
```

Fungsi input() itu selalu mengembalikan string, makanya harus dikonversi dengan casting seperti diatas.

D. Kondisional

Kalo di C++ if else nya seperti ini

```
1 if(x == 1) {
2     // do something
3 } else {
4     // do something
5 }
```

Kalo di python cuma gini, simple kan?

```
1  if x:  
2    # do something  
3  else :  
4    # do something
```

Kalian juga bisa pake OR atau AND untuk logic nya seperti ini

```
1  s = input("Do you agree? ")  
2  # Check whether agreed  
3  if s == "Y" or s == "y":  
4    print("Agreed.")  
5  elif s == "N" or s == "n":  
6    print("Not agreed.")
```

E. Pengulangan

for loops di python itu seperti ini, ga ribet

```
1  # Better design  
2  for i in range(3):  
3    print("meow")
```

for loops juga bisa dipakai untuk list seperti di bawah ini

```
1  # Print all scores value  
2  scores = [72, 73, 33]  
3  for score in scores:  
4    print(score);
```

F. List dan Dictionaries

List atau array dalam C++ adalah struktur data bawaan untuk menyimpan beberapa nilai di dalam [].

Salah satu contoh penggunaan list adalah seperti ini:

```

1 # Getting the scores value and take the average value of them
2 scores = []
3 for i in range(3):
4     x = int(input("Input the number"))
5     scores.append(x)
6
7 # Print the average
8 average = sum(scores) / len(scores)
9 print(f"Average: {average}")

```

Kalian bisa belajar lebih lanjut tentang list di [dokumentasi Python](#).

Di sisi lain, dictionary atau dict adalah kumpulan pasangan **key-value**. Contohnya, implementasi pencarian berdasarkan nama:

```

1 people = [
2     {"name": "Jokowi", "status": "President"},
3     {"name": "Prabowo", "status": "Minister of Defense"},
4     {"name": "Ceb", "status": "Just s normal person"}
5 ]
6
7 # Search for a name
8 name = input("Name: ")
9 for person in people:
10     if person['name'] == name:
11         print(f"Found {person['name']} : {person['status']}")
12         break
13 else:
14     print("Not found")

```

Atau mungkin kita bisa merombaknya, kita tidak perlu **name** dan **status**. Kita bisa menyederhanakannya seperti ini:

```

1 people = {
2     "Jokowi": "President",
3     "Prabowo": "Minister of Defense",
4     "Ceb": "Just a normal person"
5 }
6
7 # We can iterate by the object dictionary rather than object list of dictionary
8 name = input("Name: ")
9 if name in people:
10     print(f"Found {name} : {people[name]}")
11 else:
12     print("Not found")

```

Kalian bisa belajar lebih lanjut tentang **dictionary** di [dokumentasi Python](#)

G. Command Line Arguments

Satu hal tentang menjalankan program Python adalah kalian juga bisa memasukkan argumen tambahan ke dalam program. Misalnya, Kalian bisa menambahkan informasi seperti data.csv atau information.txt sebagai argumen saat menjalankan program. Dengan begitu, argumen tambahan ini bisa digunakan untuk berbagai operasi.

```
1 python main.py data.csv
```

Kalian bisa melihat argumen pada program seperti ini. Secara default, jika file tidak ada, program akan membuat file baru atau langsung menimpa file yang sudah ada.

```
1 # Prints a command-line argument
2
3 from sys import argv
4
5 if len(argv) == 2:
6     print(f"hello, {argv[1]}")
7 else:
8     print("hello, world")
```

H. File I/O

Salah satu hal yang bisa kita manfaatkan dengan Python adalah operasi **File I/O**. Ada banyak operasi yang bisa dilakukan dengan **File I/O**, tetapi bagian ini hanya akan membahas cara menggunakannya pada file teks.

Hal pertama yang harus dilakukan adalah membuka file itu sendiri.

```
1 open("readme.txt", "w")
```

Fungsi open() biasanya menerima 2 parameter, yaitu **file** dan **mode**.

Parameter pertama adalah **nama file**. Jika file belum ada, maka akan dibuat file baru.

Parameter kedua menentukan **mode** yang digunakan untuk membuka file.

- "r" - **Read** - Nilai default. Membuka file untuk membaca, akan error jika file tidak ada.
- "a" - **Append** - Membuka file untuk menambahkan data di akhir file, akan membuat file jika belum ada.
- "w" - **Write** - Membuka file untuk menulis ulang, akan membuat file jika belum ada (jika ada, isinya akan ditimpa).

- "x" - **Create** - Membuat file baru, akan error jika file sudah ada.

Langkah selanjutnya adalah menulis isi file menggunakan `write()` dan `writelines()`.

```
1 with open('readme.txt', 'w') as f:
2     f.write('readme')
```

Jika kalian ingin menulis karakter **UTF-8** seperti emoji atau huruf Tiongkok, kalian bisa menambahkan langkah tambahan ini:

```
1 quote = '成功を収める人とは人が投げってきたレンガでしっかりした基盤を築くことができる人のことである。'
2
3 with open('quotes.txt', 'w', encoding='utf-8') as f:
4     f.write(quote)
```

I. Third Party Library

Salah satu keunggulan Python adalah komunitas pengguna yang besar serta banyaknya **library pihak ketiga**. Pada bagian ini, kita akan menggunakan `qrcode library`.

Namun, langkah pertama adalah menginstal library tersebut dengan kode berikut. Dokumentasi lengkapnya bisa ditemukan di [sini](#).

“pip install qrcode”

Secara default, library ini akan diinstal **secara global** daripada **lokal**. Jika kamu ingin menginstalnya secara lokal, ada beberapa metode yang bisa digunakan, seperti `venv`. Detail tentang metode ini tidak dibahas di sini, tetapi kamu bisa merujuk pada dokumentasi berikut:

- **Install Package** – docs.python.org
- **Venv** – docs.python.org

Langkah selanjutnya adalah menggunakan library ini. Berikut adalah contoh dasar cara membuat **QR Code** untuk sebuah video YouTube tertentu:

```
1 import qrcode
2 img = qrcode.make('https://www.youtube.com/watch?v=dQw4w9WgXcQ&pp=ygUlcmlja3JvbGw=')
3 type(img) # qrcode.image.pil.PilImage
4 img.save("some_file.png")
```

Jalankan programnya, dan **Voila!** 🙌



J. Study Reference

Karena output dari kursus ini adalah membuat game dengan **pygame**, mungkin kalian bisa cek referensi ini dan mulai nyoba. Biar tubes bisa nyantai 😊

Referensi Pygame:

- Pengenalan Pygame di YouTube – [Tonton di sini](#)
- Dokumentasi Pygame – [Baca di sini](#)

Referensi Python Resmi:

- Tutorial Resmi Python – [Lihat di sini](#)

Semoga game kalian gacorr! 🚀 🔥