

# main

October 5, 2022

## 1 Condition

```
[1]: if True:
      print("Hello")

b = 5
if 3 < b:
    print(f'3<{b}')
```

  

```
c = 5
d = 5
if c<=d:
    print(f'nilai c adalah {c} dan nilai d adalah {d}')
```

Hello

3<5

nilai c adalah 5 dan nilai d adalah 5

## 2 Latihan

- nilai 80-100 -> A
- nilai 70-80 -> B
- nilai 60-70 -> C
- nilai 50-60 -> D
- nilai <50 -> E

```
[2]: def NilaiGrade(nilai):
      # guard clause
      if nilai < 0 or nilai > 100:
          return "error"

      if nilai >= 80 and nilai <= 100:
          return "A"
      if nilai >= 70 and nilai <= 80:
          return "B"
      if nilai >= 60 and nilai < 70:
          return "C"
      if nilai >= 50 and nilai < 60:
```

```

        return "D"
    if nilai < 50:
        return "E"

print(NilaiGrade(101), "expected: error")
print(NilaiGrade(100), "expected: A")
print(NilaiGrade(80), "expected: A")
print(NilaiGrade(79), "expected: B")
print(NilaiGrade(70), "expected: B")
print(NilaiGrade(69), "expected: C")
print(NilaiGrade(60), "expected: C")
print(NilaiGrade(59), "expected: D")
print(NilaiGrade(50), "expected: D")
print(NilaiGrade(49), "expected: E")
print(NilaiGrade(0), "expected: E")
print(NilaiGrade(-1), "expected: error")

```

```

error expected: error
A expected: A
A expected: A
B expected: B
B expected: B
C expected: C
C expected: C
D expected: D
D expected: D
E expected: E
E expected: E
error expected: error

```

### 3 In

```

[3]: kota = ['A', 'B', 'C']
     if 'A' in kota:
         print('A merupakan bagian dari kota')

```

```
A merupakan bagian dari kota
```

```

[4]: a = 1

     while a < 10:
         print(f'berhasil nilai a adalah {a}')
         a += 1

```

```

berhasil nilai a adalah 1
berhasil nilai a adalah 2
berhasil nilai a adalah 3
berhasil nilai a adalah 4

```

```
berhasil nilai a adalah 5
berhasil nilai a adalah 6
berhasil nilai a adalah 7
berhasil nilai a adalah 8
berhasil nilai a adalah 9
```

## 4 For Loop

```
[5]: kota = ['A','B','C']
for nama_kota in kota:
    print(f'{nama_kota} merupakan bagian dari kota')

for c in range (1,3):
    print(f'berhasil nilai c adalah {c}')
```

```
A merupakan bagian dari kota
B merupakan bagian dari kota
C merupakan bagian dari kota
berhasil nilai c adalah 1
berhasil nilai c adalah 2
```

```
[6]: a = 1
b = 1
while a < 5:
    while b < 6:
        print(f'{a} x {b} = {a * b}')
        b += 1
    a += 1
    b = 1
    print('*'*20)
```

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
*****
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
*****
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
```

```

*****
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
*****

```

```

[7]: for a in range(1,5):
      for b in range(1,6):
          print(f'{a} x {b} = {a * b}')
      print('*'*20)

```

```

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
*****
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
*****
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
*****
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
4 x 4 = 16
4 x 5 = 20
*****

```

## 5 Flow Control Statement

1. Continue
2. Break

```

[8]: for b in range (1, 6):
      if b in (1,2):
          continue
      if b == 3:
          continue

```

```
if b == 5:
    break
print(f'nilai {b}')
```

nilai 4

## 6 Latihan

- nilai 80-100 -> A
- nilai 70-80 -> B
- nilai 60-70 -> C
- nilai 50-60 -> D
- nilai <50 -> E

```
[9]: def NilaiGrade(nilai: int):
    # guard clause
    if nilai < 0 or nilai > 100:
        return "error"

    if nilai >= 80 and nilai <= 100:
        return "A"
    if nilai >= 70 and nilai <= 80:
        return "B"
    if nilai >= 60 and nilai < 70:
        return "C"
    if nilai >= 50 and nilai < 60:
        return "D"
    if nilai < 50:
        return "E"

kamus={'Raka':100, 'Hana':90, 'Dian':55}

for key in kamus:
    print(f'{key} grade: {NilaiGrade(kamus[key])}')
```

Raka grade: A  
Hana grade: A  
Dian grade: D

```
[10]: print("luas dari Persegi: %d" % 25)
```

luas dari Persegi: 25

## 7 Latihan

```
[11]: nama_kota: list[str] = []

def insertKota(data: str):
    for kota in nama_kota:
        if kota == data:
            print("error")
            return "error"
    nama_kota.append(data)

insertKota("Malang")
insertKota("Malang")
print(nama_kota)

def insertKotaMany(*kotas: tuple[str]):
    for kota in kotas:
        insertKota(kota)

def insertKotaList(kotas: list[str]):
    for kota in kotas:
        insertKota(kota)

insertKotaMany('Jakarta', 'Riau')
print(nama_kota)

insertKotaList(['Surabaya', 'Batam'])
print(nama_kota)

def removeKota(kota: str):
    nama_kota.remove(kota)

removeKota('Jakarta')
print(nama_kota)

nama_kota.pop(0)
print(nama_kota)

error
['Malang']
['Malang', 'Jakarta', 'Riau']
['Malang', 'Jakarta', 'Riau', 'Surabaya', 'Batam']
['Malang', 'Riau', 'Surabaya', 'Batam']
['Riau', 'Surabaya', 'Batam']
```

## 8 Embedding method to an existing class

```
[12]: class DuplicateError(BaseException): ...

class Kota(list[str]):

    def insertKota(self, input: str = "Jakarta"):
        for kota in self:
            if kota == input:
                return DuplicateError("duplicate")
        self.append(input)

d = Kota()

d.insertKota(input = "Malang")
print(d)

d.insertKota()
print(d)

err = d.insertKota("Malang")
if err != None:
    print(err)
print(d)

['Malang']
['Malang', 'Jakarta']
duplicate
['Malang', 'Jakarta']
```

```
[13]: dir()
```

```
[13]: ['DuplicateError',
      'In',
      'Kota',
      'NilaiGrade',
      'Out',
      '_',
      '__',
      '___',
      '__annotations__',
      '__builtin__',
      '__builtins__',
      '__doc__',
      '__loader__',
      '__name__',
      '__package__',
      '__spec__']
```

```
'_dh',  
'_i',  
'_i1',  
'_i10',  
'_i11',  
'_i12',  
'_i13',  
'_i2',  
'_i3',  
'_i4',  
'_i5',  
'_i6',  
'_i7',  
'_i8',  
'_i9',  
'_ih',  
'_ii',  
'_iii',  
'_oh',  
'a',  
'b',  
'c',  
'd',  
'err',  
'exit',  
'get_ipython',  
'insertKota',  
'insertKotaList',  
'insertKotaMany',  
'kamus',  
'key',  
'kota',  
'nama_kota',  
'quit',  
'removeKota']
```