# PRAKTIKUM ALGORITMA dan PEMROGRAMAN

# PRAKTIKUM 6: DATA DAN OPPERATOR


**Disusun Oleh:**

**AS'AD NIROT AHMADI**

**L200220155**

**PROGRAM STUDI TEKNIK INFORMATIKA**

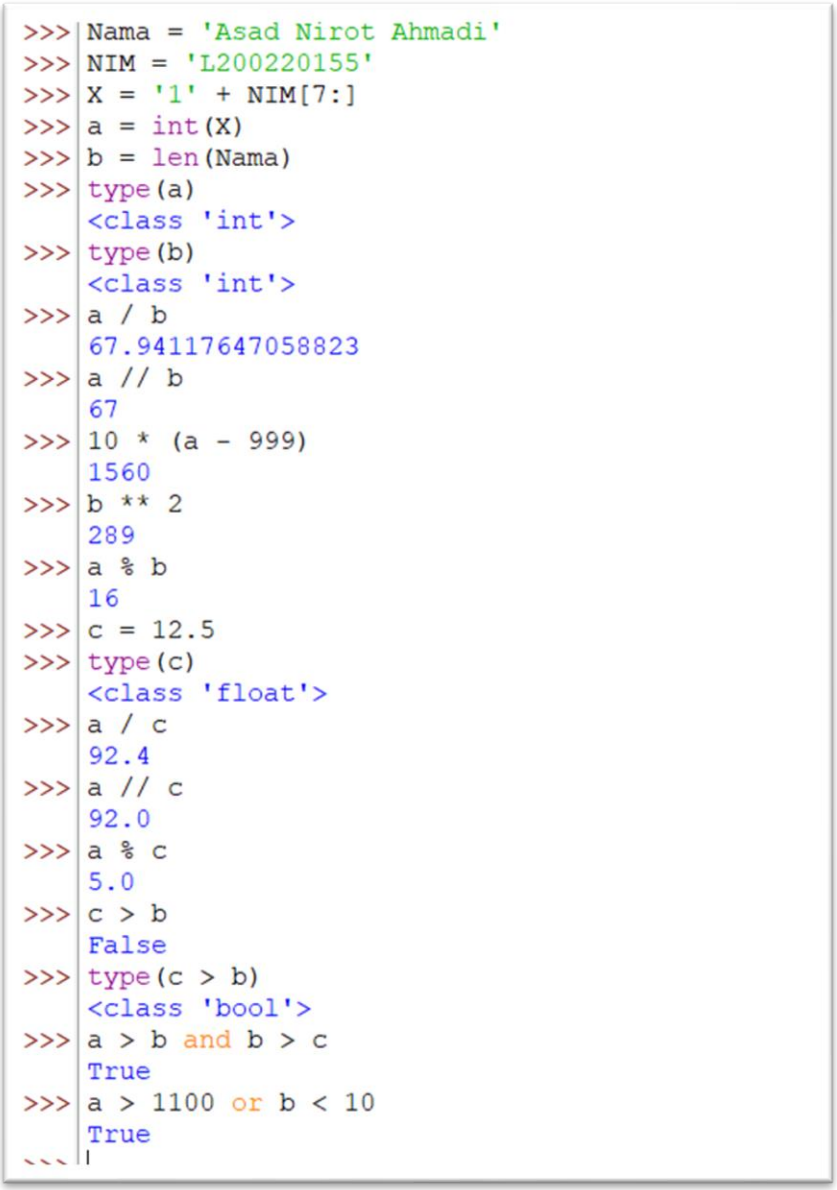
**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**TAHUN 2022/2023**

**Kegiatan 3. Operator**

Berikut adalah screenshoot dari IDLE pyhton saya



```
>>> Nama = 'Asad Nirot Ahmadi'
>>> NIM = 'L200220155'
>>> X = '1' + NIM[7:]
>>> a = int(X)
>>> b = len(Nama)
>>> type(a)
<class 'int'>
>>> type(b)
<class 'int'>
>>> a / b
67.94117647058823
>>> a // b
67
>>> 10 * (a - 999)
1560
>>> b ** 2
289
>>> a % b
16
>>> c = 12.5
>>> type(c)
<class 'float'>
>>> a / c
92.4
>>> a // c
92.0
>>> a % c
5.0
>>> c > b
False
>>> type(c > b)
<class 'bool'>
>>> a > b and b > c
True
>>> a > 1100 or b < 10
True
>>>
```

Komentar:

**Kegiatan 3.2**

type(a) >> Python mengecek tipe data variable a

type(b) >> Python mengecek tipe data variable b

a / b >> Python diberikan perintah untuk membagi a bagi b

a // b >> Python diberikan perintah untuk melakukan pembagian dengan hasilnya dibulatkan a dibagi b

10 * (a – 999) >> Python diberikan perintah perkalian dan pengurangan dengan pengurangannya didahulukan karena didalam kurung

b ** 2 >> Python diberikan perintah untuk melakukan perpangkatan yaitu b pangkat 4

a % b >> Python diberikan perintah untuk mencari sisa hasil bagi dari a bagi b

**Kegiatan 3.2**

C = 12.5 >> menambahkan data c

type (c) >> Python mengecek tipe data variable c

a / c >> Python diberikan perintah untuk membagi a bagi c

a // c >> Python diberikan perintah untuk melakukan pembagian dengan hasilnya dibulatkan a dibagi c

a % c >> Python diberikan perintah untuk mencari sisa hasil bagi dari a bagi c

**Kegiatan 3.4**

c > b >> Mencari nilai kebenaran dengan operator perbandingan lebih dari

type(c > b) >> Python mengecek tipe data dari hasil operator perbandingan c > b

a > b and b > c >> Operator perbandingan didalam operator logika. Operator logika ”and” dan jika hasil dari keduanya bener maka outputnya True

a > 1100 or b < 10 >> Operator perbandingan didalam operator logika, Operator logika”or” dan jika hasilnya salah satunya ataupun semuanya True maka hasil akhirnya True dan jika keduanya False maka hasil akhirnya False

**Kegiatan 4. Tipe Data**

Berikut adalah screenshoot dari IDLE pyhton saya

```
>>> Nama = 'Asad Nirot Ahmadi'
>>> NIM = 155
>>> Tinggi = 1.7
>>> Berat = 60
>>> TahunLahir = 2004
>>> Aku = (TahunLahir, Berat, Tinggi, NIM, Nama)
>>> Data = [TahunLahir, Berat, Tinggi, NIM, Nama]
>>> type(Aku)
    <class 'tuple'>
>>> Aku[0]
    2004
>>> a = NIM % 4; Aku[a]
    155
>>> type(Aku[a])
    <class 'int'>
>>> Aku[a:4]
    (155,)
>>> type(Aku[4])
    <class 'str'>
>>> Aku[0] = 'ok'
    Traceback (most recent call last):
      File "<pyshell#13>", line 1, in <module>
        Aku[0] = 'ok'
    TypeError: 'tuple' object does not support item assignment
>>> type(Data)
    <class 'list'>
>>> type(Data[4])
    <class 'str'>
>>> Data[4][5]
    'N'
>>> Data[4][a:6]
    'd N'
>>> Data[0] = 'ok'; Data
    ['ok', 60, 1.7, 155, 'Asad Nirot Ahmadi']
>>> Data[-a]
    1.7
>>> range(a)
    range(0, 3)
>>>
```

Komentar:

**Kegiatan 4.2**

type(Aku) >> Python mengecek tipe data variabel Aku

Aku[0] >> Mengakses variable Aku indeks ke 0

a = NIM % 4; Aku[ a ] >> Mendeklarasikan variable a dengan nilai hasil operasi bilangan, lalu mengakses Aku indeks ke-a

type(Aku[ a ]) >> Python mengecek tipe data variable Aku indeks ke-4

Aku[ a:4 ] >> Mengakses variable Aku indeks ke-a dan seterusnya dengan batas indeks ke-4

type( Aku[4] ) >> Python mengecek tipe data variable Aku indeks ke-4

Aku[ 0 ] = ‘ok’ >> Python menganalisa pendeskripsian tuple tetapi Aku[0] adalah list

**Kegiatan 4.3**

type(Data) >> Python mengecek tipe data variable Data

type(Data[4]) >> Python mengecek tipe variable Data indeks ke-4

Data[4] [5] >> Mengakses data pada indeks 4 yang dimana merupakan string yang akan diakses pada 5

Data [4] [ a:6 ] >> Mengakses data pada indeks 4 yang dimana merupakan string yang akan diakses pada indeks a hingga sebelum indeks 6

Data[0] = ‘ok’; Data >> Menuliskan ulang variable Data pada indeks ke 0, lalu menampilkan isi variable data

Data[ -a ] >> Mengakses variable Data pada indeks ke 2 dari belakang

Range(a) >> mencari batas jarak pada variable a