

**Nama : Ghina Khoerunnisa**  
**Program : Data Scientist Intern**

### **Introduction: Basic Syntax, Data Type, and Variable**

- **Tipe data dasar** : Integer (bilangan bulat), Float (bilangan desimal), String (kumpulan karakter, diapit oleh petik satu atau petik dua serta untuk string multi line diapit oleh petik tiga), Boolean (bernilai True atau False). Untuk mengecek tipe data dari suatu value atau variabel dapat menggunakan sintaks `type()`.
- **Variable assignment**, memberi value pada suatu variabel yang ditandai dengan “=”.
- **Pemberian nama variabel**, variabel dapat diberi nama yang terdiri dari huruf besar, huruf kecil, angka, dan garis bawah. Tapi, karakter pertama nama variabel tidak boleh diberi angka. Kemudian nama variabel bersifat case sensitive. Variabel “Age” dan “age” diperlakukan sebagai dua variabel yang berbeda. Terdapat tiga cara untuk menuliskan nama variabel, yaitu camelCase, PascalCase, dan snake\_case yang dapat dipilih salah satu dan digunakan secara konsisten. Tetapi, pada python pemberian nama variabel lebih baik menggunakan snake\_case karena mayoritas pengguna menggunakan penulisan tersebut.
- **Operator dan Expression**, operator merupakan simbol yang menunjukkan komputasi yang dilakukan pada operand. Contohnya operator aritmatika dan operator perbandingan. Yang termasuk pada operator aritmatika adalah penambahan (+), pengurangan (-), perkalian (\*), pembagian (/), modulus/hasil sisa bagi (%), perpangkatan (\*\*). Sedangkan operator perbandingan contohnya adalah sama dengan (==), tidak sama dengan (!=), lebih besar dari (>), lebih kecil dari (<), lebih besar atau sama dengan (>=), lebih kecil atau sama dengan (<=). Pada ekspresi perbandingan hasil outputnya berupa boolean.
- **Manipulasi String**, Penambahan pada string akan menyambungkan string-string yang ditambah, Perkalian string dengan suatu bilangan n akan menduplikat string sebanyak n kali (dalam satu string). Operator “in” di python berguna untuk mengecek subset dari suatu value. Case Conversion berfungsi mengubah string, contohnya `capitalize()` yang berfungsi untuk mengubah string menjadi huruf kapital, `lower()` yang berfungsi untuk mengubah string menjadi huruf kecil, `upper()` yang berfungsi untuk mengubah string menjadi huruf besar, `swapcase()` yang berfungsi untuk mengubah huruf kecil menjadi huruf besar dan sebaliknya, `title()` yang berfungsi untuk mengubah huruf pertama di setiap kata menjadi huruf besar.
- Untuk memanggil banyak variabel pada suatu string dapat menggunakan format “Jumlah pengeluaran bulan {bulan} adalah {pengeluaran}”.`format(bulan=b, pengeluaran=p)` dimana b dan p merupakan variabel yang telah ada kemudian diassign ke dalam variabel bulan dan pengeluaran.
- **List**, kumpulan objek arbitrary yang mirip dengan array tetapi di list ini dapat terdiri dari berbagai tipe data bahkan list juga. List ditandai dengan tanda kurung siku dan setiap objek dipisah dengan koma.
- **Karakteristik List**, list bersifat berurutan sehingga jika ada dua list dengan urutan elemen yang berbeda di dalamnya walaupun dari segi konten sama akan tetapi pada python kedua list tidak sama (urutan diperhatikan). Kemudian elemen pada list dapat diakses dengan indexnya. Index dimulai dari 0 sehingga jika ingin mengakses elemen

kedua pada list `a = ["aku", "kamu", 0]` maka sintaksnya `a[1]` (elemen pertama "aku" memiliki index 0, elemen kedua "kamu" memiliki index 1, dan elemen ketiga yaitu 0 memiliki index 2). Untuk mengakses elemen terakhir juga dapat dilakukan dengan sintaks `a[-1]`. Selain itu, jika ingin mengakses lebih dari satu elemen dapat menggunakan slicing. Contoh penggunaannya adalah `a[0:2]`, sintaks tersebut akan menampilkan elemen list `a` dengan index 0 hingga 1 yaitu `["aku", "kamu"]`.

- **Concatenation dan Replication di List**, sama seperti pada manipulasi string, concat antar dua list atau lebih dapat dilakukan dengan menggunakan simbol tambah (+). Begitu juga dengan replication di list dapat menggunakan simbol perkalian (\*).
- Pada python untuk menghitung panjang list dapat menggunakan `len()`, untuk mencari nilai maksimum dapat menggunakan `max()`, dan untuk mencari nilai minimum dapat menggunakan `min()`.
- Update elemen list dapat dilakukan dengan pengindeksan (bisa juga dengan slicing) dan assignment. Contohnya index 1 di list `a` akan diubah menjadi "dia" maka sintaksnya adalah `a[1] = "dia"`. Sedangkan untuk menghapus elemen dapat dilakukan dengan `del` dan pengindeksan. Contohnya elemen ketiga (indexnya 2) akan dihapus maka sintaksnya `del a[2]`.
- **Tuple**, kumpulan objek yang berurutan juga sama seperti list tetapi tuple ditandai dengan tanda kurung (()) dan elemen-elemen tuple tidak dapat diubah-ubah. Contohnya tuple `b = ("aku", "kamu", "dia", "mereka")`.
- **Dictionary**, tipe data yang terdiri dari pasangan key dan value. Dictionary ditandai dengan kurung kurawal ({}). Jika pada list pengindeksan dilakukan dengan indexnya maka pada dictionary pengindeksan dapat dilakukan dengan keynya. Contohnya adalah dictionary `peserta = {"nama": "Ghina", "program": "data scientist intern", "batch": 1}` sehingga jika ingin mengakses "Ghina" maka sintaksnya `peserta["nama"]`.
- Untuk menambahkan pasangan key dan value baru dapat dilakukan dengan cara `peserta["univ"] = "Telkom University"`. Untuk update value dari suatu key yang telah ada dapat dilakukan dengan `peserta["nama"] = "Ghina Khoerunnisa"`. Untuk menghapus pasangan key dan value dapat dilakukan dengan `del` dan pengindeksan seperti `del peserta["batch"]`.
- Built-in Method di dictionary, `peserta.items()` akan mengembalikan list tuple pasangan key dan valuenya `[("nama", "Ghina Khoerunnisa"), ("program", "data scientist intern"), ("batch", 1), ("univ", "Telkom University")]`. `peserta.keys()` akan mengembalikan list semua key pada dictionary `peserta` `["nama", "program", "batch", "univ"]`. `peserta.values()` akan mengembalikan list semua value pada dictionary `peserta` `["Ghina Khoerunnisa", "data scientist intern", 1, "Telkom University"]`.
- Tambahan : Untuk menghitung frekuensi kata pada suatu string dapat dilakukan dengan `defaultdict()` dari `collections`.

```
from collections import defaultdict
default_d = defaultdict(int)
string = "saya saya dua saya tiga"
for i in string.split():
    default_d[i] += 1
```

```
default_d
defaultdict(int, {'dua': 1, 'saya': 3, 'tiga': 1})
```