**Rangkuman Berinteraksi dengan Data**

Kita sudah berada di penghujung materi Berinteraksi dengan Data. Sampai sini, Anda telah memiliki pemahaman mendasar mengenai data dalam Python. Mari kita rangkum secara saksama.

**Abstraksi Data**

Abstraksi data merupakan kemampuan Anda untuk mengerti konteks dan merepresentasikannya menjadi bentuk lain sesuai dengan konteks masalahnya.

Ketika menuliskan data dalam pemrograman, komputer tidak akan mengetahui data yang dimaksud hingga Anda mendeklarasikan tipe datanya.

**Data Typing**

Dalam pemrograman, Anda harus mengenali istilah deklarasi dan inisialisasi yang umum ditemui ketika membuat sebuah program.

**Deklarasi dan Inisialisasi**

Deklarasi merujuk pada pembuatan variabel dengan menentukan tipe data dan nama variabelnya. Contohnya seperti berikut yang merupakan deklarasi dalam pemrograman C/C++.

1. int age;
2. float salary;

Inisialisasi merujuk kepada pemberian nilai awal pada variabel yang sebelumnya telah dideklarasikan. Contohnya seperti berikut yang merupakan deklarasi dalam pemrograman C/C++.

1. int age = 17;
2. float salary = 5000000;

Dalam Python, Anda tidak diharuskan mendeklarasikan tipe data variabel. Anda dapat langsung melakukan inisialisasi variabel.

1. age = 17
2. salary = 5000000.0
4. print(type(age))
5. print(type(salary))
7. """
8. Output:
9. <class 'int'>
10. <class 'float'>
12. """

**Tipe Data**

Sebuah data memiliki tipe yang berbeda-beda. Dalam Python, Tipe data dapat dikelompokkan menjadi **tipe data primitif** dan **tipe data collection**.

**Tipe Data Primitif**

Tipe data primitif merupakan jenis data yang paling dasar dalam pemrograman. Tipe data ini menyimpan *single value*. Beberapa tipe data primitif sebagai berikut.

1. **Numbers**
   1. Integer: Bilangan bulat positif atau negatif dan tidak memiliki angka desimal. Contoh: 1; -20; 999; dan 0.
   2. Float: Bilangan riil yang dapat mewakili bilangan bulat atau bilangan desimal. Contoh: 3.14; 1; dan 4.01E+1
   3. Complex: Bilangan kompleks. **(Kita tidak akan menggunakannya di kelas ini.)**Contoh: 1+2j
2. **Boolean**  
   Boolean merupakan tipe data yang hanya bernilai TRUE atau FALSE. Tipe data ini merepresentasikan nilai kebenaran (*truth values*).

|  |
| --- |
| True False |

1. Sebenarnya, setiap variabel yang memiliki nilai bisa dievaluasi dan menghasilkan nilai *true*. Hanya ada beberapa nilai yang akan dianggap bernilai false, yakni berikut.
   1. Nilai yang sudah didefinisikan bernilai salah: **None dan False**.
   2. Angka nol dari semua tipe numerik: **0, 0.0, 0j, Decimal(0), Fraction(0,1)**.
   3. Urutan (*sequence*) dan koleksi (*collection*) yang kosong: **"", (), {}, set(), range(0)**.
2. **String**  
   String merupakan karakter yang berurutan. Ketika Anda membuat variable bernilai string maka akan diawali dengan *single quote*('') atau *double quote*("").
   1. "Dicoding Indonesia"

**Tipe Data Collection**

Selain tipe data primitif yang menyimpan *single value*, ada tipe data lain, yakni **tipe data collection**. Tipe data ini menyimpan satu atau lebih data primitif sebagai satu kelompok. Berikut adalah tipe data yang masuk ke dalam tipe data collection.

1. List  
   List merupakan jenis kumpulan data terurut (ordered sequence) dan salah satu tipe data yang sering digunakan pada Python.  
     
   Melakukan inisialisasi list pada Python cukup mudah: menggunakan kurung siku "[]" dan setiap elemennya dipisahkan dengan koma.
   1. x = [1, 2.2, "Dicoding"]

Perlu diingat bahwa nilai yang ada dalam sebuah list selalu dimulai dari indeks ke-0. Artinya, nilai "1" pada list di atas merupakan indeks ke-0.

1. **Tuple**  
   Tuple merupakan jenis dari list yang tidak dapat diubah elemennya. Umumnya, tuple digunakan untuk data yang bersifat sekali deklarasi dan dapat dieksekusi lebih cepat. Anda dapat mendeklerasikan tuple dengan menggunakan tanda kurung "()" dan setiap elemen di dalamnya dipisahkan dengan koma.
   1. x = (1, "Dicoding", 1+3j)
2. **Set**  
   Set merupakan kumpulan item bersifat unik dan tanpa urutan (unordered collection). Anda dapat melakukan inisialisasi variabel set dengan menggunakan tanda kurawal "{}" dan setiap elemennya dipisahkan dengan koma.
   1. x = {1, 2, 3 , 7, 13}
3. **Dictionary**  
   Dictionary pada Python merupakan kumpulan pasangan key-valueyang bersifat tidak berurutan. Dictionary dapat digunakan untuk menyimpan data kecil hingga besar. Pada Python, dictionary didefinisikan dengan kurawal dan tambahan definisi berikut.
   1. Setiap elemen pasangan *key-value* dipisahkan dengan koma (,).
   2. Key dan valuedipisahkan dengan titik dua (:).
   3. Key dan valuedapat berupa tipe variabel/objek apa pun.
      1. x = { 'name': 'Perseus Evans', 'age': 20, 'isMarried': False }

**Konversi antara Tipe Data**

Anda pun dapat melakukan konversi antar tipe data dengan menggunakan beberapa fungsi yang tersedia pada Python. Fungsi merupakan blok kode yang dapat dipanggil untuk melakukan tugas tertentu. Beberapa fungsi yang dapat digunakan untuk mengonversi tipe data pada Python sebagai berikut.

1. Konversi integer ke float: **float().**
2. Konversi float ke integer: **int().**
3. Konversi dari-dan-ke string: **str(), float(), int().**

**Transformasi Angka, Karakter, dan String**

Khusus tipe data string, terdapat berbagai fungsi untuk mentransformasikan tipe data string menjadi bentuk lain. Beberapa di antaranya berikut.

1. **Mengubah Huruf Besar/Kecil**  
   Mengubah string menjadi UPPERCASE atau lowercase.
   1. upper()
   2. lower()
2. **Awalan dan Akhiran**  
   Menghapus karakter *whitespace*pada suatu string.
   1. rstrip()
   2. lstrip()
   3. strip()
   4. startswith()
   5. endswith()
3. **Memisah dan Menggabung String**  
   Fungsi-fungsi untuk memisahkan dan menggabungkan string.
   1. join()
   2. split()
4. **Mengganti Elemen String**  
   Metode yang bertujuan untuk mengganti elemen string dengan elemen string lainnya.
   1. replace()
5. **Pengecekan String**  
   Fungsi-fungsi yang bertujuan untuk melakukan pengecekan pada string dan mengembalikan nilai Boolean.
   1. isupper()
   2. islower()
   3. isalpha()
   4. isalnum()
   5. isdecimal()
   6. isspace()
   7. istitle()
6. **Formatting pada String**  
   Fungsi-fungsi untuk pemformatan string.
   1. zfill()
   2. rjust()
   3. ljust()
   4. center()
7. **String Literals**  
   String literals adalah serangkaian karakter yang diapit oleh tanda kutip baik tunggal (') maupun ganda (").  
     
   String dapat ditulis mudah dalam Python dengan cara diapit oleh tanda petik tunggal. Namun, dalam kondisi tertentu, dibutuhkan petik tunggal di tengah string (misalnya struktur kepemilikan dalam Bahasa Inggris—Dicoding's Cat atau penyebutan Jum'at pada hari dalam bahasa Indonesia).  
     
   Solusinya adalah menggunakan *escape character* yang memungkinkan Anda untuk menggunakan karakter yang sebelumnya tidak bisa dimasukkan dalam string. Umumnya diawali dengan *backslash* (\) dan diikuti karakter tertentu yang diinginkan. Beberapa contoh *escape character*sebagai berikut.
   1. \' Single quote
   2. \" Double quote
   3. \t Tab
   4. \n Newline (*line break*)
   5. \\ Backslash
8. **Raw String**  
   Merupakan cara untuk mencetak string sesuai dengan apa pun input atau teks yang diberikan. Umumnya digunakan untuk regex atau beberapa implementasi lain yang sangat bergantung pada keberadaan backslash. Untuk mengimplementasikan raw strings, sisipkan huruf r sebelum pembuka string.
   1. print(r'Dicoding\tIndonesia')
   3. """
   4. Output:
   5. Dicoding\tIndonesia
   6. """

**Operasi pada List, Set, dan String**

Ada beberapa fungsi untuk melakukan operasi pada list, set, dan string. Berikut beberapa di antaranya.

1. **len()**  
   Fungsi yang bertujuan untuk menghitung panjang atau banyaknya elemen dari list, set, dan string.
   1. contoh\_list = [1, 3, 3, 5, 5, 5, 7, 7, 9]
   3. print(contoh\_list)
   4. print(len(contoh\_list))
   6. """
   7. Output:
   8. [1, 3, 3, 5, 5, 5, 7, 7, 9]
   10. 9
   11. """
2. **min() dan max()**  
   Fungsi yang digunakan untuk mengetahui nilai minimum dan maksimum dari suatu list.
   1. angka = [13, 7, 24, 5, 96, 84, 71, 11, 38]
   2. print(min(angka))
   3. print(max(angka))
   5. """
   6. Output:
   7. 5
   8. 96
   9. """
3. **count()**  
   Fungsi yang digunakan untuk mengetahui berapa kali suatu objek muncul dalam list.
   1. genap = [2, 4, 4, 6, 6, 6, 8, 10, 10]
   2. print(genap.count(6))

   5. """
   6. Output:
   7. 3
   8. """
4. **In dan Not In**  
   **In** dan **not in** merupakan operator yang diperuntukkan untuk mengetahui nilai atau objek yang ada dalam list. Anda bisa menggunakan operator ini untuk memastikan suatu nilai ada dalam list bahkan dalam string. Operatori **in** dan **not in**akan mengembalikan nilai boolean True atau False.
   1. kalimat = "Belajar Python di Dicoding sangat menyenangkan"
   2. print('Dicoding' in kalimat)
   3. print('tidak' in kalimat)
   4. print('Dicoding' not in kalimat)
   5. print('tidak' not in kalimat)
   7. """
   8. Output:
   9. True
   10. False
   11. False
   12. True
   13. """
5. **Memberikan Nilai untuk Multiple Variable**Anda dapat memberikan nilai untuk multiple variable dengan cara berikut.
   1. data = ['shirt', 'white', 'L']
   2. apparel, color, size = data
   4. print(data)
   6. """
   7. Output:
   8. ['shirt', 'white', 'L']
   9. """
6. **sort()**  
   Fungsi sort() digunakan untuk mengurutkan angka atau huruf. Berikut contoh implementasinya.
   1. kendaraan = ['motor', 'mobil', 'helikopter', 'pesawat']
   2. kendaraan.sort()
   4. print(kendaraan)
   6. """
   7. Output:
   8. ['helikopter', 'mobil', 'motor', 'pesawat']
   9. """