

Dasar Pemrograman

07b Struktur data dictionary

AES / 2024 05 22



Struktur data *dictionary*

Dictionary adalah salah satu jenis struktur data dalam bahasa pemrograman python.

Dictionary berisi kumpulan pasangan kunci (*key*) dan nilai (*value*) tertentu, dimana kumpulan *key* dan *value* tersebut merepresentasikan entitas tertentu

Key	Value
nama	Upin
sekolah	Tadika Mesra
usia	5
hobi	Main bola

Key	Value
kode	FTK6201
nama	Dasar Pemrograman
sks	3

Key	Value
nama	ar Rahman
nomor urut	55
jumlah ayat	78
lokasi turun	Madinah

Deklarasi dan inisialisasi *dictionary*

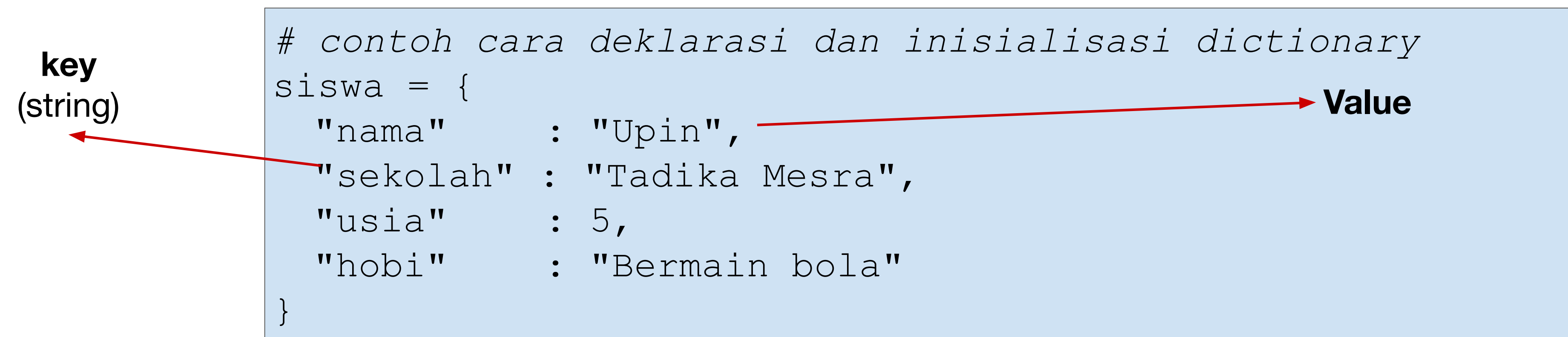
Deklarasi dan inisialisasi variabel dengan tipe dictionary dilakukan dengan menggunakan kurung kurawal (*curly bracket*) “{ }”

Tiap pasangan *key:value* pada sebuah *dictionary* dipisahkan dengan tanda koma

key
(string)

```
# contoh cara deklarasi dan inisialisasi dictionary
siswa = {
    "nama"      : "Upin",
    "sekolah"    : "Tadika Mesra",
    "usia"       : 5,
    "hobi"       : "Bermain bola"
}
```

Value

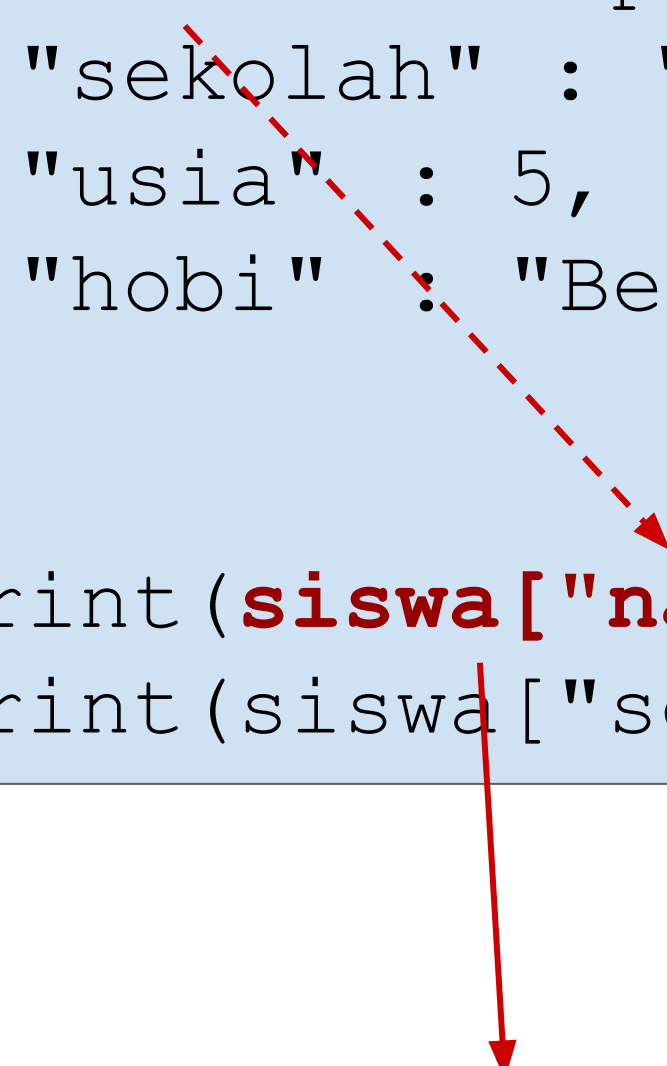


Latihan: buatlah *dictionary* dari entitas mata kuliah Dasar Pemrograman dan surat *ar Rahman* dengan pasangan *key* dan *value* yang ada pada halaman sebelum ini

Akses dictionary

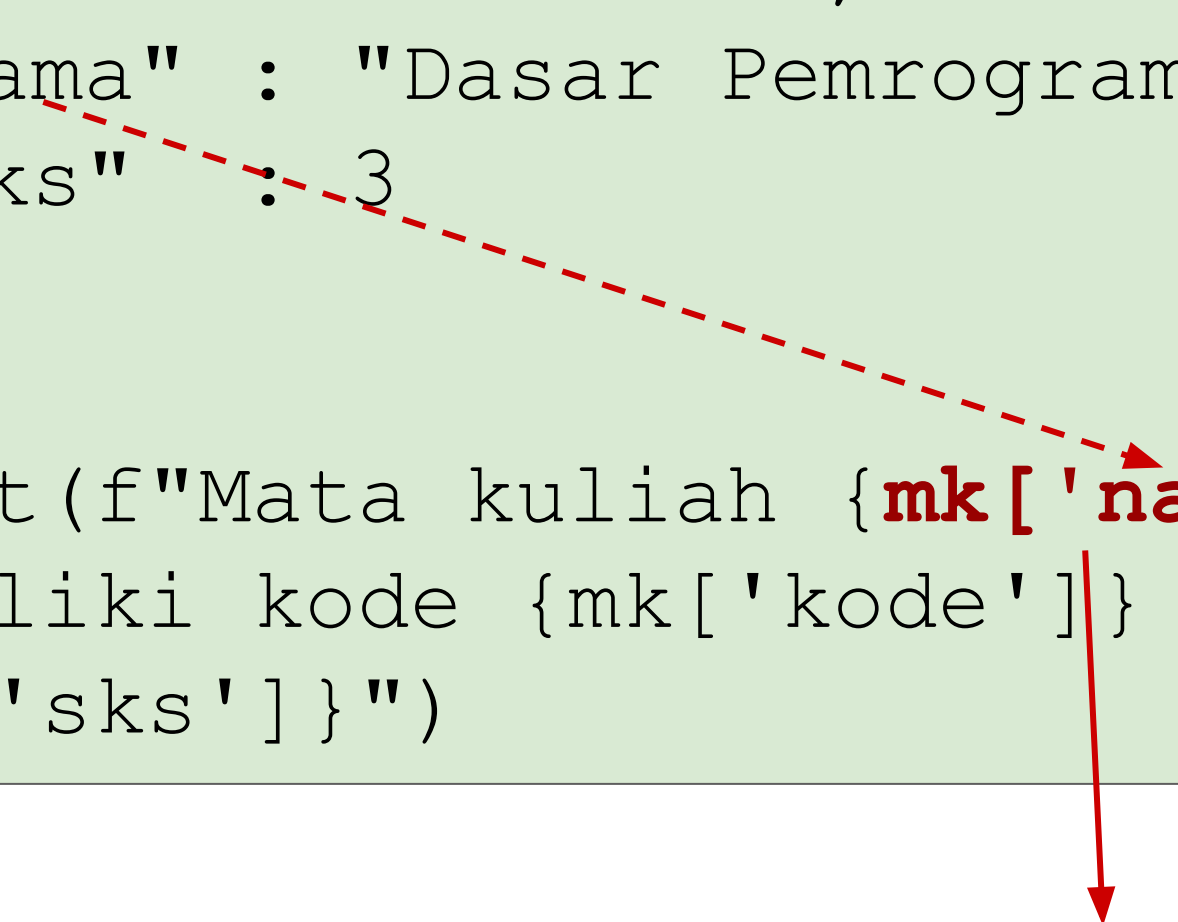
elemen suatu *value*(nilai) pada sebuah dictionary dapat diakses menggunakan *key* yang bersesuaian

```
siswa = {  
    "nama" : "Upin",  
    "sekolah" : "Tadika Mesra",  
    "usia" : 5,  
    "hobi" : "Bermain bola"  
}  
  
print(siswa["nama"])  
print(siswa["sekolah"])
```



Akan menghasilkan
"Upin"

```
mk = {  
    "kode" : "FTK6201",  
    "nama" : "Dasar Pemrograman",  
    "sks" : 3  
}  
  
print(f"Mata kuliah {mk['nama']}  
memiliki kode {mk['kode']} dan SKS  
{mk['sks']}")
```




Akan menghasilkan
"Dasar Pemrograman"

Update nilai pada dictionary

Nilai tertentu pada dictionary dapat diupdate/diberi nilai baru sebagaimana variabel biasa.
Sintaks yang digunakan `dictionary[key] = nilai_baru`

Kondisi awal

```
mk = {  
    "kode" : "FTK6201",  
    "nama" : "Dasar Pemrograman",  
    "sks"   : 3  
}
```

 key sks memiliki nilai 3

Ganti nilai untuk key tertentu (sks)

```
mk["sks"] = 2
```

Kondisi baru

```
mk = {  
    "kode" : "FTK6201",  
    "nama" : "Dasar Pemrograman",  
    "sks"   : 2  
}
```

Menambah pasangan *key-value* nilai pada dictionary

Pasangan key-value pada dictionary dapat ditambah. Caranya sama dengan saat melakukan update yakni dengan menggunakan sintaks

```
dictionary[key_baru] = nilai_baru
```

Bedanya, kali ini key yang digunakan belum ada sebelumnya pada dictionary

Kondisi awal

```
mk = {  
    "kode" : "FTK6201",  
    "nama" : "Dasar Pemrograman",  
    "sks"   : 3  
}
```

Tambahkan pasangan *key-value* baru

```
mk["wajib"] = True
```

Kondisi baru

```
mk = {  
    "kode" : "FTK6201",  
    "nama" : "Dasar Pemrograman",  
    "sks"   : 2,  
    "wajib" : True  
}
```

Menghapus pasangan *key-value* nilai pada dictionary

Pasangan key-value pada dictionary juga dapat dihapus. Keyword yang digunakan adalah keyword *del*. Cara penggunaannya dapat dilihat pada contoh berikut.

Kondisi awal

```
mk = {  
    "kode"    : "FTK6201",  
    "nama"    : "Dasar Pemrograman",  
    "sks"     : 2,  
    "wajib"   : True  
}
```

Hapus pasangan *key-value* tertentu

```
del mk["wajib"]
```

Kondisi baru

```
mk = {  
    "kode"    : "FTK6201",  
    "nama"    : "Dasar Pemrograman",  
    "sks"     : 2  
}
```

Loop pada dictionary

Contoh *dictionary*

```
mk = {  
    "kode"    : "FTK6201",  
    "nama"    : "Dasar Pemrograman",  
    "sks"     : 2,  
    "wajib"   : True  
}
```

Loop dengan *for* untuk mendapatkan seluruh *key* pada *dictionary*

```
for k in mk:  
    print(k)
```

Loop dengan *for* untuk mendapatkan seluruh pasangan *key-value* pada *dictionary*

```
for k,v in mk.items():  
    print(f"Key {k} pada dictionary bernilai: {v}")
```


Memeriksa keberadaan suatu *key* pada dictionary dengan *in*

Keyword *in* dapat digunakan untuk memeriksa apakah suatu nilai ada dalam *dictionary*.

```
mk = {  
    "kode" : "FTK6201",  
    "nama" : "Dasar Pemrograman",  
    "sks" : 3,  
    "wajib" : True  
}  
print("nama" in mk)  
print("dosen" in mk)
```

menghasilkan ***True*** karena *key nama*
ada pada dictionary *mk*

menghasilkan ***False*** karena *key dosen*
tidak ada pada dictionary *mk*

Kombinasi penggunaan struktur data: *dictionary* sebagai isi *list*

Struktur data *dictionary* dapat diletakkan sebagai elemen dari *list*, seperti pada contoh berikut ini:

```
daftar_mhs = [  
    {  
        "nim" : "13424001",  
        "nama" : "Soleh Rahmat",  
        "thn_masuk" : 2024  
    },  
    {  
        "nim" : "13424002",  
        "nama" : "Dinda Ceria",  
        "thn_masuk" : 2024  
    },  
    {  
        "nim" : "13424003",  
        "nama" : "Syakila Onila",  
        "thn_masuk" : 2024  
    },  
    {  
        "nim" : "13424004",  
        "nama" : "Fernando Maldini",  
        "thn_masuk" : 2024  
    },  
]  
  
for mhs in daftar_mhs:  
    print(f"{mhs['nama']} dengan NIM {mhs['nim']} masuk tahun {mhs['thn_masuk']}")
```

Kombinasi penggunaan struktur data: *dictionary* di dalam *dictionary*

Suatu *dictionary* juga dapat diletakkan sebagai isi *dictionary* lain, seperti pada contoh berikut ini:

```
daftar_mhs = {  
    "13424001" : {  
        "nama" : "Soleh Rahmat",  
        "thn_masuk" : 2024  
    },  
    "13424002" : {  
        "nama" : "Dinda Ceria",  
        "thn_masuk" : 2024  
    },  
    "13424003" : {  
        "nama" : "Syakila Onila",  
        "thn_masuk" : 2024  
    },  
    "13424004": {  
        "nama" : "Fernando Maldini",  
        "thn_masuk" : 2024  
    },  
}  
  
for k,v in daftar_mhs.items():  
    print(f"{v['nama']} dengan NIM {k} masuk tahun {v['thn_masuk']}")
```

Perhatikan bahwa pada contoh ini kode NIM dijadikan sebagai *key* untuk tiap-tiap mahasiswa

Contoh soal 1

Berikut adalah tabel berisi mahasiswa dan nilai UTSnya. Buatlah program untuk menemukan mahasiswa dengan nilai UTS terbesar.

Nama	Nilai UTS
Upin	75
Jarjit	64
Meimei	80
Susanti	82
Ehsan	78
Zoro	43

Petunjuk
Data masing-masing mahasiswa dapat disimpan dalam struktur data *dictionary*. Seluruh *dictionary* tersebut kemudian diletakkan di dalam sebuah *list*. Mahasiswa dengan nilai UTS terbesar dapat dicari dengan menggunakan perulangan terhadap elemen *list* tersebut.

Contoh soal 2

Body Mass Index (indeks massa tubuh), atau BMI dihitung dengan rumus sebagai berikut.

$$\text{BMI} = \text{bb} / \text{tb}^2$$

dimana,

bb: berat badan dalam satuan kilogram

tb: tinggi badan dalam satuan meter

Berikut adalah tabel berisi beberapa orang beserta jenis kelamin, berat badan, dan tinggi badannya.

Buatlah program untuk menentukan:

1. Siswa dengan BMI terbesar
2. Siswa dengan BMI terkecil
3. Rata-rata BMI siswa laki-laki

Nama	Jenis Kelamin	Berat (kg)	Tinggi (cm)
Dinara Kamelia	P	55	161
Rustam Perkasa	L	69	174
Ridwan Ali	L	60	169
Ricardo Salami	L	79	170
Ningsih Ayu	P	58	163
Laetisia Cantika	P	49	158
Roronoa Zoro	L	85	181

Petunjuk

Data tiap siswa dapat disimpan dalam struktur data *dictionary*. Seluruh *dictionary* tersebut kemudian diletakkan di dalam sebuah *list*.

Dengan menggunakan perulangan terhadap elemen *list* tersebut, indikator-indikator terkait BMI siswa tersebut dapat dicari.

Contoh soal 3

Berikut adalah data Badan Pusat Statistik yang menampilkan persentase penduduk <=18 tahun yang merokok menurut jenis kelamin pada tahun 2021, 2022, dan 2023.

Buat sebuah variabel yang mengandung seluruh informasi di dalam tabel tersebut.

Petunjuk: Gunakan kombinasi struktur data di dalam struktur data (*dictionary* di dalam *list* atau *dictionary* di dalam *dictionary*)

Jenis Kelamin + Nasional	Persentase Merokok Pada Penduduk Usia ≤ 18 Tahun, Menurut Jenis Kelamin (Persen)		
	2021	2022	2023
Laki Laki	7,14	6,54	7,01
Perempuan	0,09	0,16	0,15
Nasional	3,69	3,44	3,65

Refleksi

Setelah mempelajari dan mempraktikkan materi pada slide ini anda diharapkan dapat:

1. Melakukan operasi-operasi dasar yang melibatkan *dictionary*
2. Menggunakan *dictionary* dalam persoalan-persoalan pemrograman yang anda jumpai