

Modul: Implementasi Dictionary di Python

Python dictionary adalah struktur data di Python yang digunakan untuk menyimpan data dalam bentuk pasangan key–value (kunci–nilai).

Dictionary mirip seperti kamus, di mana:

- **key** (kunci) = kata
- **value** (nilai) = definisi kata

Dalam Python, dictionary ditulis menggunakan kurung kurawal { }.

PERCOBAAN 1: FUNDAMENTAL DICTIONARY

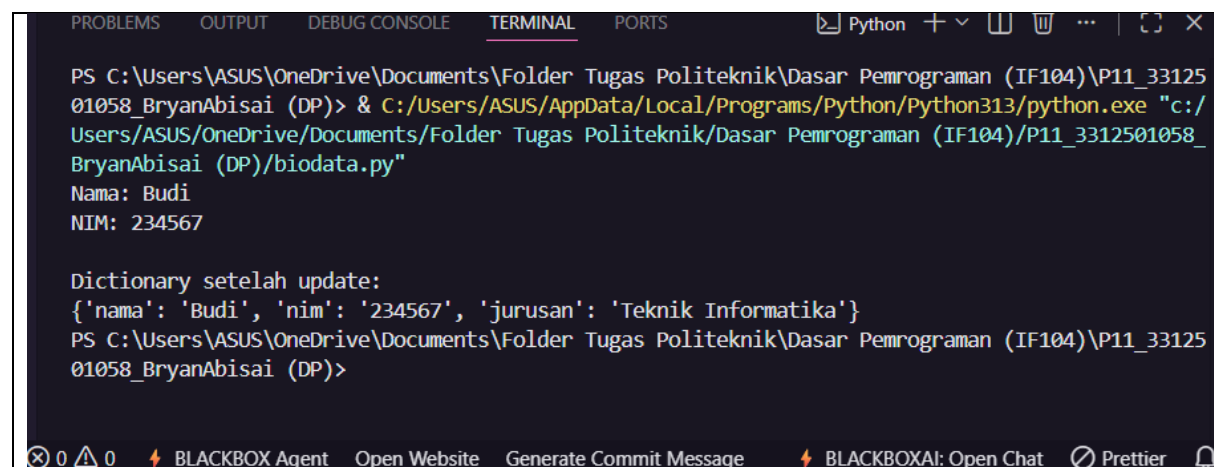
Buat file dengan nama **biodata.py** lalu salin kode berikut ini:

```
# Membuat dictionary awal
biodata = {
    "nama": "Budi",
    "nim": "234567",
    "jurusan": "Teknik Informatika"
}

# Mengakses data menggunakan key
print(f>Nama: {biodata["nama"]}")
print(f"NIM : {biodata["nim"]}")

print("\nDictionary setelah update:")
print(biodata)
```

Screenshot dan beri jawaban anda dibawah ini:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] ... [ ] [ ] X
PS C:\Users\ASUS\OneDrive\Documents\Folder Tugas Politeknik\Dasar Pemrograman (IF104)\P11_33125
01058_BryanAbisai (DP)> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe "c:/
Users/ASUS/OneDrive/Documents/Folder Tugas Politeknik/Dasar Pemrograman (IF104)/P11_3312501058_
BryanAbisai (DP)/biodata.py"
Nama: Budi
NIM: 234567

Dictionary setelah update:
{'nama': 'Budi', 'nim': '234567', 'jurusan': 'Teknik Informatika'}
PS C:\Users\ASUS\OneDrive\Documents\Folder Tugas Politeknik\Dasar Pemrograman (IF104)\P11_33125
01058_BryanAbisai (DP)>
```

Modifikasi kode diatas, tambahkan kode untuk menambahkan data angkatan pada dictionary biodata diatas. Kode apa yang anda buat?

```
"angkatan": "2022"
```

Kode apa yang anda gunakan untuk mengganti nama Budi menjadi Fulan??

```
"nama": "Fulan",
```

PERCOBAAN 2: NESTED DICTIONARY

Dictionary dapat berisi tipe data apa pun, termasuk dictionary lain. Struktur ini disebut Nested Dictionary, dan sering digunakan untuk menyimpan data yang lebih kompleks. Buat file dengan nama **mahasiswa.py** lalu salin kode dibawah ini:

```
# Dictionary di dalam dictionary
mahasiswa = {
    "123456": {
        "nama": "Fulan",
        "umur": 18,
        "jurusan": "Teknik Informatika"
    },
    "212121": {
        "nama": "Kirun",
        "umur": 21,
        "jurusan": "Rekayasa Keamanan Siber"
    },
    "767676": {
        "nama": "Wawan",
        "umur": 24,
        "jurusan": "Tata Boga"
    },
    "553322": {
        "nama": "Yanti",
        "umur": 19,
        "jurusan": "Pendidikan Bahasa Zimbabwe"
    }
}
```

Kode apa yang anda perlu implementasikan untuk mengubah jurusan Kirun menjadi jurusan Teknik Elektro??

```
"jurusan": "Teknik Elektro"
```

Bagaimana cara menampilkan seluruh data mahasiswa dengan menggunakan perulangan??

```
for nim, data_mahasiswa in mahasiswa.items():
```

```
print(f"\n--- Data Mahasiswa (NIM: {nim}) ---")
for key, value in data_mahasiswa.items():
    # Mengambil key (nama, umur, jurusan) dan value-nya
    print(f"{key.capitalize(): {value}")
```

PERCOBAAN 3: NILAI MAHASISWA DENGAN DICTIONARY

Buat sebuah file dengan nama **nilai-mahasiswa.py** lalu salin kode berikut ini:

```
surveys = {
    "Fulan": 650,
    "Kirana": 900,
    "Doni": 500,
    "Lisa": 450,
    "Joko": 900,
    "Gilang": 850
}

# Konversi dictionary ke list nilai
nilai = list(surveys.values())

# Tampilkan data setelah dikonversi
print("Data nilai sebelum sorting:", nilai)

# Sorting list dari kecil ke besar
nilai.sort()

# Tampilkan nilai setelah sorting
print("Data nilai setelah sorting :", nilai)

# Ambil nilai tertinggi dan hitung jumlah kemunculannya
nilai_tertinggi = nilai[-1]
jumlah_tertinggi = nilai.count(nilai_tertinggi)

# Cetak hasil
print(f"""
Hasil survey menunjukkan bahwa nilai tertinggi adalah {nilai_tertinggi}.
Nilai tersebut diberikan sebanyak {jumlah_tertinggi} kali
dari total {len(surveys)} responden.
""")
```

Screenshot hasil program anda dibawah ini:

```
23 nilai_tertinggi = nilai[-1]
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + - [] [X] [] [X] [X]

PS C:\Users\ASUS\OneDrive\Documents\Folder Tugas Politeknik\Dasar Pemrograman (IF104)\P11_3312501058_BryanAbisai (DP)> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/ASUS/OneDrive/Documents/Folder Tugas Politeknik/Dasar Pemrograman (IF104)/P11_3312501058_BryanAbisai (DP)/nilai-mahasiswa.py"

Data nilai sebelum sorting: [650, 900, 500, 450, 900, 850]
Data nilai setelah sorting: [450, 500, 650, 850, 900, 900]

Hasil survey menunjukkan bahwa nilai tertinggi adalah 900.
Nilai tersebut diberikan sebanyak 2 kali
dari total 6 responden.

PS C:\Users\ASUS\OneDrive\Documents\Folder Tugas Politeknik\Dasar Pemrograman (IF104)\P11_3312501058_BryanAbisai (DP)>

BLACKBOX AI: Open Chat Prettier

Modifikasi program dengan menampilkan nilai terendah lalu screenshot kembali program anda dibawah ini:

```
21
22 # Ambil nilai terendah dan hitung jumlah kemunculannya
23 nilai_terendah = nilai[0]
24 jumlah_terendah = nilai.count(nilai_terendah)
25
26 # cetak hasil
27 print(f"""
28 Hasil survey menunjukkan bahwa nilai tertinggi adalah {nilai_tertinggi}.
29 Nilai tersebut diberikan sebanyak {jumlah_terendah} kali
30 dari total {len(surveys)} responden.
31 """)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python + - [] [X] [] [X] [X]

PS C:\Users\ASUS\OneDrive\Documents\Folder Tugas Politeknik\Dasar Pemrograman (IF104)\P11_3312501058_BryanAbisai (DP)> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/ASUS/OneDrive/Documents/Folder Tugas Politeknik/Dasar Pemrograman (IF104)/P11_3312501058_BryanAbisai (DP)/nilai-mahasiswa.py"

Data nilai sebelum sorting: [650, 900, 500, 450, 900, 850]
Data nilai setelah sorting: [450, 500, 650, 850, 900, 900]

Hasil survey menunjukkan bahwa nilai tertinggi adalah 450.
Nilai tersebut diberikan sebanyak 1 kali
dari total 6 responden.

PS C:\Users\ASUS\OneDrive\Documents\Folder Tugas Politeknik\Dasar Pemrograman (IF104)\P11_3312501058_BryanAbisai (DP)>

Ln 31, Col 5 Spaces: 4 UTF-8 CRLF Python 3.13.7 Go Live BLACKBOX AI: Open Chat Prettier

PERCOBAAN 4: MANAJEMEN KONTAK DENGAN DICTIONARY

Pada percobaan ini, anda akan mempelajari metode-metode yang umum digunakan pada dictionary melalui studi kasus manajemen kontak yang meliputi fitur seperti pembuatan kontak, menampilkan isi kontak (key dan value), serta menghapus elemen dari dictionary.

Buat sebuah program dengan nama **manajemen-kontak.py** lalu salin kode berikut ini:

```
contact = {}
```

```

while True:
    p = input("# Ketikkan show/add/del/exit: ")

    match p:
        case "show":
            if len(contact) == 0:
                print("Belum ada kontak")
            else:
                print("\nDaftar Kontak:")
                for key, value in contact.items():
                    print(f"{key} = {value}")
                print()

        case "add":
            no_hp = input("Input no. kontak: ")
            nama_lengkap = input("Input nama lengkap kontak: ")
            contact[no_hp] = nama_lengkap
            print("Data berhasil ditambahkan!\n")

        case "del":
            no_hp = input("Input no. kontak yang ingin dihapus: ")
            if no_hp in contact:
                contact.pop(no_hp)
                print("Data berhasil dihapus!\n")
            else:
                print("Nomor kontak tidak ditemukan!\n")

        case "exit":
            print("Program selesai.")
            break

        case _:
            print("Menu tidak dikenali\n")

```

Modifikasi program diatas dengan menambahkan menu baru dengan nama “edit” untuk mengganti nama lengkap dari isi elemen nomor kontak yang telah ada berdasarkan algoritma berikut ini:

```

INPUT no_kontak
INPUT nama
contact[no_kontak] = nama
OUTPUT "Data {no_kontak} berhasil diubah!"

```

Setelah modifikasi selesai dilakukan, eksekusi kembali program lalu screenshot hasil dibawah ini (pilih menu edit):

```

contact = {}

while True:
    p = input("# Ketikkan show/add/del/edit/exit: ")

    match p:

```

```
case "show":
    if len(contact) == 0:
        print("Belum ada kontak")
    else:
        print("\nDaftar Kontak:")
        for key, value in contact.items():
            print(f"{key} = {value}")
        print()

case "add":
    no_hp = input("Input no. kontak: ")
    nama_lengkap = input("Input nama lengkap kontak: ")
    contact[no_hp] = nama_lengkap
    print("Data berhasil ditambahkan!\n")

case "del":
    no_hp = input("Input no. kontak yang ingin dihapus: ")
    if no_hp in contact:
        contact.pop(no_hp)
        print("Data berhasil dihapus!\n")
    else:
        print("Nomor kontak tidak ditemukan!\n")

case "edit":
    no_hp = input("Input no. kontak yang ingin diubah: ")
    if no_hp in contact:
        nama_baru = input("Input nama lengkap kontak yang baru: ")
        contact[no_hp] = nama_baru
        print(f"Data {no_hp} berhasil diubah!\n")
    else:
        print("Nomor kontak tidak ditemukan!\n")

case "exit":
    print("Program selesai.")
    break

case _:
    print("Menu tidak dikenali\n")
```



```

        "jawaban_benar": "a"
    },
    "soal3": {
        "pertanyaan": "Ular mana yang merupakan bahasa pemrograman?",
        "bidang": "IT",
        "poin": 35,
        "a": "sanca",
        "b": "kobra",
        "c": "sawah",
        "d": "python",
        "jawaban_benar": "d"
    },
}

```

Lanjutkan koleksi bank soal diatas dengan menambahkan **3 soal lagi** (bebas).

Setelah total 5 soal selesai dibuat, lanjutkan program kuis dengan memberikan pengguna seluruh pertanyaan yang ada kemudian pengguna harus menjawab pilihan yang diberikan.

Output yang ditampilkan dari program ini adalah nilai akhir yang diperoleh dari poin berdasarkan jawaban benar dan salah. Hasil akhir yang diharapkan adalah sebagai berikut:

```

PS D:\python\P09> python .\t02-kuis.py

soal1.      1+1 = ...
a. 0.5
b. 1
c. 2
d. 3

Pilih jawaban anda: (a/b/c/d): c

soal2.      Jika ada teman yg sakit maka kita harus ...
a. menjenguknya
b. menjauhinya
c. berolahraga
d. waspada

Pilih jawaban anda: (a/b/c/d): d

soal3.      Ular mana yang merupakan bahasa pemrograman...
a. sanca
b. kobra
c. sawah
d. python

Pilih jawaban anda: (a/b/c/d): a
Selamat! Skor akhir anda adalah 40
PS D:\python\P09> 

```


Salin kode jawaban anda dibawah ini:

```
bank_soal = {
  "soal1": {
    "pertanyaan": "1+1 = ...",
    "bidang": "mtk",
    "poin": 30,
    "a": "0.5",
    "b": "1",
    "c": "2",
    "d": "3",
    "jawaban_benar": "c"
  },
  "soal2": {
    "pertanyaan": "Jika ada teman yg sakit maka kita harus ...",
    "bidang": "kwn",
    "poin": 25,
    "a": "menjenguknya",
    "b": "menjauhinya",
    "c": "berolahraga",
    "d": "waspada",
    "jawaban_benar": "a"
  },
  "soal3": {
    "pertanyaan": "Ular mana yang merupakan bahasa pemrograman?",
    "bidang": "IT",
    "poin": 25,
    "a": "sanca",
    "b": "kobra",
    "c": "sawah",
    "d": "python",
    "jawaban_benar": "d"
  },
  "soal4": {
    "pertanyaan": "Jika bertemu orang hitam kita harus ...",
    "bidang": "keberagaman",
    "poin": 10,
    "a": "menghina",
    "b": "menerima dengan baik",
    "c": "menjauhi",
    "d": "out aja lu suki",
    "jawaban_benar": "b"
  },
  "soal5": {
    "pertanyaan": "Jika melihat orang makassar kita harus ...",
```

```

        "bidang": "daerah",
        "poin": 10,
        "a": "lempar coto makassar",
        "b": "menghina sulawesi",
        "c": "menerima dengan baik",
        "d": "melayu-kan dia",
        "jawaban_benar": "c"
    }
}

# Inisialisasi skor
skor_akhir = 0

# Loop melalui setiap soal dalam bank_soal
for soal_key, data_soal in bank_soal.items():
    # 1. Tampilkan pertanyaan
    print(f"\n{soal_key}: {data_soal['pertanyaan']}")

    # 2. Tampilkan pilihan jawaban
    print(f"a. {data_soal['a']}")
    print(f"b. {data_soal['b']}")
    print(f"c. {data_soal['c']}")
    print(f"d. {data_soal['d']}")

    # 3. Minta jawaban pengguna (diubah ke huruf kecil untuk konsistensi)
    jawaban_user = input("Pilih jawaban anda: (a/b/c/d): ").lower()

    # 4. Cek jawaban dan hitung skor
    if jawaban_user == data_soal['jawaban_benar']:
        # Jawaban benar, tambahkan poin
        skor_akhir += data_soal['poin']

# 5. Tampilkan hasil akhir
print("\nSelamat! Skor akhir anda adalah", skor_akhir)

```