



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2024/2025

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

NIM	71220829
Nama Lengkap	TIMOTIUS EKANA TEJO KESUMA
Minggu ke / Materi	10 / Tipe Data List

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2025

BAGIAN 1: MATERI MINGGU INI (40%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

MATERI 10.3.1 Sifat-sifat List

List pada Python adalah rangkaian nilai-nilai yang dapat diakses menggunakan satu nama tunggal. List dapat menampung karakter, integer, float, dan tipe data lainnya. List dituliskan di dalam []. Contoh:

```
1 nilai_ujian = [80,75,70,90,81,84,92,71,65,80,70]
2 nama_pahlawan = ['Sukarno', 'Diponegoro', 'Jend. Sudirman', 'Cut Nya Dhien']
3 nilai_campuran = ['Javascript', 20, 34.4, True]
4 list_dalam_list = [23, [22, 20], 45]
```

List bersifat mutable (dapat diubah secara langsung). Contoh:

```
data = [10,20,30,40]
data[0] = 50
print(data)
```

```
[50, 20, 30, 40]
```

Pada list, object yang sama bukanlah sesuatu yang sama. Contoh:

```
a = [1, 2, 3]
b = [1, 2, 3]
```

```
False
```

MATERI 10.3.2 Mengakses dan Mengubah isi List

Untuk mengakses dan mengubah list, dapat menggunakan indeks dari list. Contoh:

```
thislist = ["apple", "banana", "cherry"]
print(thislist[1])
print(thislist[-1])
print(thislist[2:5])
print(thislist[:4])
thislist[1] = "jeruk"
print(thislist)
thislist2 = ["apple", "banana", "cherry", "orange", "kiwi", "mango"]
thislist2[1:3] = ["blackcurrant", "watermelon"]
print(thislist2)
```

```
banana
cherry
['cherry']
['apple', 'banana', 'cherry']
['apple', 'jeruk', 'cherry']
['apple', 'blackcurrant', 'watermelon', 'orange', 'kiwi', 'mango']
```

MATERI 10.3.3 Fungsi-fungsi Untuk List

Ada banyak fungsi yang dapat digunakan untuk list.

```
# 1. append
print("Append")
nama = ['kuncoro', 'felix', 'ryan', 'yuan']
nama.append(['tejo', 'ujo'])
print(nama)

# 2. extend
print("Extend")
nama = ['kuncoro', 'felix', 'ryan', 'yuan']
nama.extend(['tejo', 'ujo'])
print(nama)

# 3. sort
print("Sort")
nama = ['kuncoro', 'felix', 'ryan', 'yuan']
nama.sort()
print(nama)

# 4. pop
print("Pop")
nama = ['kuncoro', 'felix', 'ryan', 'yuan']
dihapus = nama.pop(1)
print(dihapus)
print(nama)

# 5. del
print("Del")
nama = ['kuncoro', 'felix', 'ryan', 'yuan']
del nama[2]
print(nama)

# 6. remove
print("Remove")
nama = ['kuncoro', 'felix', 'ryan', 'yuan']
nama.remove('kuncoro')
print(nama)

# 7. reverse
print("Reverse")
nama = ['kuncoro', 'felix', 'ryan', 'yuan']
nama.reverse()
print(nama)
```

Output:

```
Append
['kuncoro', 'felix', 'ryan', 'yuan', ['tejo', 'ujo']]
Extend
['kuncoro', 'felix', 'ryan', 'yuan', 'tejo', 'ujo']
Sort
['felix', 'kuncoro', 'ryan', 'yuan']
Pop
felix
['kuncoro', 'ryan', 'yuan']
Del
['kuncoro', 'felix', 'yuan']
Remove
['felix', 'ryan', 'yuan']
Reverse
['yuan', 'ryan', 'felix', 'kuncoro']
```

Selain itu, terdapat juga metode built-in:

```
angka = [5, 3, 90, 1]
print("min: ", min(angka))
print("max: ", max(angka))
print("sum: ", sum(angka))
print("len: ", len(angka))
```

```
min: 1
max: 90
sum: 99
len: 4
```

MATERI 10.3.4 List Sebagai Parameter Fungsi

List dapat digunakan sebagai parameter untuk suatu fungsi dan bersifat mutable. Contoh:

```
def ubah(data):
    data[0] = "anton"

data = ['a', 'b', 'c']
ubah(data)
print(data)
```

Output:

```
['anton', 'b', 'c']
```

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

SOAL 1

Buatlah fungsi untuk mencari 3 bilangan terbaik dari sebuah list. Bilangan terbaik adalah bilangan dimulai dari yang paling tinggi.

```
def tiga_terbaik(data):  
    data.sort()  
    data.reverse()  
    print("Tiga nilai terbaik adalah:", data[:3])  
  
data = [10, 5, 20, 8, 15]  
tiga_terbaik(data)
```

Output:

```
Tiga nilai terbaik adalah: [20, 15, 10]
```

Penjelasan:

Fungsi berjalan dengan 1 parameter yaitu data yang berisi list. Data di urutkan dengan fungsi `sort()` kemudian dibalik dengan fungsi `reverse()` kemudian keluarkan output 3 data di awal dengan mengakses indeks `data[:3]`.

SOAL 2

Buatlah program yang dapat menerima input bilangan-bilangan dari pengguna dan kemudian jika pengguna memasukkan kata "done" maka program akan menampilkan bilangan rata-ratanya.

```
1  data = []  
2  
3  while True:  
4      user_input = input("Masukkan bilangan (atau ketik 'done' bila sudah selesai): ")  
5      if user_input.lower() == "done":  
6          break  
7      number = int(user_input)  
8      data.append(number)  
9  
10 if data:  
11     rata_rata = sum(data) / len(data)  
12     print(f"Rata-rata: {rata_rata}")
```

Output:

```
Masukkan bilangan (atau ketik 'done' bila sudah selesai): 1
Masukkan bilangan (atau ketik 'done' bila sudah selesai): 2
Masukkan bilangan (atau ketik 'done' bila sudah selesai): 3
Masukkan bilangan (atau ketik 'done' bila sudah selesai): 4
Masukkan bilangan (atau ketik 'done' bila sudah selesai): done
Rata-rata: 2.5
```

Penjelasan:

Program berjalan dengan inisiasi tampungan list yaitu `data = []`, kemudian dilakukan perulangan jika benar, maka perulangan akan terus berjalan untuk menerima inputan pengguna, jika inputan adalah bilangan, maka program akan terus menambahkan data bilangan ke dalam list. Ketika inputan adalah 'done' maka program akan berhenti meminta inputan dari pengguna dan akan melakukan kalkulasi untuk rata-rata dengan memanfaatkan fungsi `sum(data)` untuk menjumlahkan semua yang ada di dalam list dan kemudian dibagi dengan `len(data)` yang berarti banyaknya data yang ada di dalam list.

SOAL 3

Buatlah program yang menerima sebuah file teks yang berisi sebuah artikel atau berita. Kemudian program akan memeriksa semua kata yang ada di dalam artikel teks tersebut, kemudian program akan menampilkan semua kata yang unik dari artikel tersebut.

```
def get_unique_words(filename):
    with open(filename, 'r', encoding='utf-8') as file:
        text = file.read()

        tandabaca = '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
        translator = str.maketrans('', '', tandabaca)
        words = text.translate(translator).lower().split()

        word_count = {}
        for word in words:
            word_count[word] = word_count.get(word, 0) + 1

        unique_words = [word for word, count in word_count.items() if count == 1]
        return unique_words

filename = input("Masukkan nama file teks: ")
try:
    unique_words = get_unique_words(filename)
    print("Kata-kata unik (hanya muncul sekali) dalam file:")
    for word in sorted(unique_words):
        print(word)
except FileNotFoundError:
    print("File tidak ditemukan.")
```

Output:

```
Masukkan nama file teks: taksepadan.txt
Kata-kata unik (hanya muncul sekali) dalam file:
ahasvéros
anwar
api
baik
beginilah
beranak
berbahagia
buta
chairil
ciptaan
dan
dikutuk
dinding
eros
ini
jadi
jadinya
kan
karena
kawin
kira
kita
mengembara
merangkaki
nanti
padami
pintu
rangka
satu
sedang
sepadan
serupa
sumpahi
terbuka
terpanggang
tidak
tinggal
unggunan
```

Penjelasan:

Program dimulai dengan mendefinisikan sebuah fungsi untuk menampung proses, yaitu `get_unique_words(filename)`. Di dalam fungsi ini, file teks dibuka dan seluruh isinya dibaca. Kemudian, dilakukan pembersihan tanda baca pada teks agar hanya tersisa kata-kata saja, lalu semua huruf diubah menjadi huruf kecil dan dipisahkan berdasarkan spasi sehingga menjadi list kata. Selanjutnya, dibuat sebuah dictionary untuk menghitung kemunculan setiap kata. Program melakukan perulangan pada setiap kata dalam list, jika kata tersebut sudah ada di dictionary maka nilainya akan bertambah satu, jika belum maka akan diinisialisasi dengan nilai satu. Setelah semua kata dihitung, program membuat list baru yang hanya berisi kata-kata yang muncul satu kali saja (unik). Di bagian utama, program meminta nama file dari pengguna. Jika file ditemukan, maka fungsi `get_unique_words` dijalankan dan hasilnya (kata-kata unik) akan ditampilkan satu per satu secara urut. Jika file tidak ditemukan, maka program akan menampilkan pesan bahwa file tidak ditemukan.

Link GitHub:

https://github.com/TimotiusEkanaT/Alpro/tree/main/Laporan_10