



Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024

NIM	71231058
Nama Lengkap	Michael Chandra Mahanaim
Minggu ke / Materi	09 / Tipe Data List

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.

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA

BAGIAN 1: MATERI MINGGU INI (40%)

MATERI 1: Sifat-Sifat List

Dalam Python, list adalah rangkaian berbagai nilai yang dapat diakses menggunakan index. List ini dapat berisi campuran Char, Integer, Boolean, etc. Beberapa contoh list dan tipe data yang dapat disimpan dalam list adalah sebagai berikut:

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

Gambar contoh list menyimpan berbagai macam tipe data

List bersifat mutable, yang artinya isinya dapat berubah. Tidak seperti tuple dan list yang isinya sudah paten dan tidak bisa diubah lagi, atau bersifat immutable.

MATERI 2: Operasi isi List

List dapat berfungsi juga dengan beberapa operand, contohnya:

1. Dengan menggunakan operand (+), dapat menggabungkan 2 list menjadi 1

```
>>> bil1 = [1,2,3,4]
>>> bil2 = [5,6,7]
>>> bilTotal = bil1 + bil2
>>> print(bilTotal)
[1, 2, 3, 4, 5, 6, 7]
```

Gambar penggunaan operand + untuk menjumlahkan 2 list menjadi 1

2. Dengan menggunakan operand (*), dapat membuat 1 list baru yang berisi isi list pertama dikali n kali

```
>>> bil1 = [1,2,3,4]
>>> bilTotal = [1,2,3,4]*2
>>> print(bilTotal)
[1, 2, 3, 4, 1, 2, 3, 4]
```

Gambar contoh penggunaan operand * untuk mengalikan list

Isi list dapat diakses menggunakan angka index, contoh penggunaannya sebagai berikut:

```
>>> nama = ["kuncoro", "anton", "dida", "yuan"]
>>> nama[0]
'kuncoro'
>>> nama[3]
'yuan'
```

Gambar contoh pengaksesan isi list menggunakan angka index

Isi dapat juga diakses dari belakang dengan menggunakan angka index negatif seperti:

```
>>> nama[-2]
'dida'
>>> nama[-3]
'anton'
>>> nama[-4]
'kuncoro'
```

Gambar penggunaan index negatif

Jika angka index yang ingin diakses lebih besar dari besar list, maka error seperti berikut akan tampil:

```
>>> nama[4]
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
IndexError: list index out of range
```

Gambar error angka index melebihi jumlah dalam list

Index juga dapat digunakan untuk slicing seperti dalam string. Contoh lebih jelas sebagai berikut:

```
# Seluruh elemen diakses
>>> nama[:]
['kuncoro', 'anton', 'dida', 'yuan']
# elemen diakses pada indeks 1-2
>>> nama[1:3]
['anton', 'dida']
# elemen diakses dari indeks awal hingga 2
>>> nama[:3]
['kuncoro', 'anton', 'dida']
# elemen diakses dari indeks 1 hingga akhir
>>> nama[1:]
['anton', 'dida', 'yuan']
```

Gambar penggunaan slicing untuk mengambil isi tertentu dalam list

Karena list bersifat Mutable, maka isinya dapat diganti. Contoh cara penggantian isi list menggunakan metode slicing adalah seperti berikut:

```
#Penggantian pada indeks elemen 1 dan 2
>>> nama[1:3] = ["felix","ryan"]

>>> nama
['kuncoro', 'felix', 'ryan', 'yuan']
```

Gambar penggunaan slicing untuk mengganti isi list

MATERI 3: Metode dan Fungsi List

List methods mirip seperti string method, yaitu fungsi khusus yang dapat mempermudah untuk manipulasi dan managing isi list. Dari W3school, beberapa list methods adalah seperti tabel di bawah ini:

Method	Description
append()	Menambahkan elemen ke dalam akhir list
clear()	Menghapus semua elemen dalam list
copy()	Mengembalikan sebuah copy dari sebuah list
count()	Mengembalikan jumlah elemen yang bernilai ()
extend()	Memasukkan elemen sebuah list ke list lain
index()	Mengembalikan index dari elemen yang bernilai ()
insert()	Menambah elemen di posisi (posisi, elemen)
pop()	Menghapus elemen yang di posisi index ke - ()
remove()	Menhapus elemen yang bernilai ()
reverse()	Memutar balik urutan list
sort()	Menyortir list

Untuk contoh, visualisasi, dan cara guna yang lebih jelas dari beberapa list method ini, penggunaanya seperti berikut:

1. append():

```
>>> nama
['kuncoro', 'felix', 'ryan', 'yuan']
>>> nama.append(['bejo', 'tejo'])
>>> nama
['kuncoro', 'felix', 'ryan', 'yuan', ['bejo', 'tejo']]
```

Gambar contoh penggunaan fungsi list append

2. extend():

```
>>> nama
['kuncoro', 'felix', 'ryan', 'yuan', 'bejo']
>>> nama.extend(["tejo", "ujo"])
>>> nama
['kuncoro', 'felix', 'ryan', 'yuan', 'bejo', 'tejo', 'ujo']
```

Gambar contoh penggunaan fungsi list extend

3. sort():

```
>>> nama.sort()
>>> nama
['bejo', 'felix', 'kuncoro', 'ryan', 'tejo', 'ujo', 'yuan']
```

Gambar contoh penggunaan fungsi list sort

Catatan: Untuk sort(), jika list berisi angka maka akan diurutkan dari yang kecil ke yang terbesar. Jika dalam bentuk string, maka list akan diurutkan berdasarkan alfabet. Jika sebuah angka dalam bentuk string misalkan ['21', '3', '1'], maka setelah di sort() outputnya akan menjadi ['1', '21', '3']. Karena diurutkan secara alfabetikal, maka angka tidak diurutkan berdasarkan besarnya.

4. pop():

```
>>> huruf = ['a', 'b', 'c', 'd']
>>> huruf.pop(3)
'd'
>>> huruf
['a', 'b', 'c']
```

Gambar contoh penggunaan fungsi list pop

5. remove():

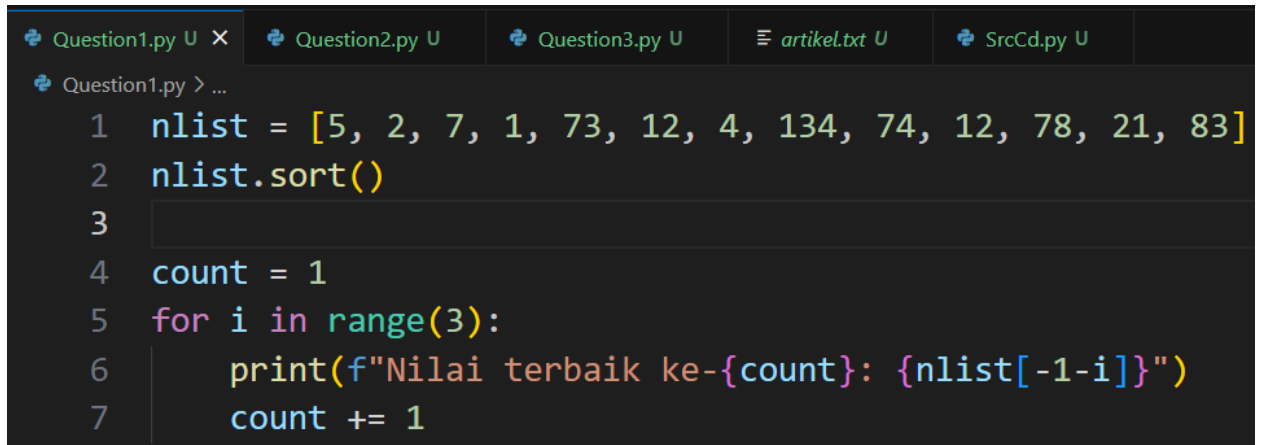
```
>>> huruf = ['a', 'b', 'c', 'd']
>>> huruf.remove("c")
>>> huruf
['a', 'b', 'd']
```

Gambar contoh penggunaan fungsi list remove

BAGIAN 2: LATIHAN MANDIRI (60%)

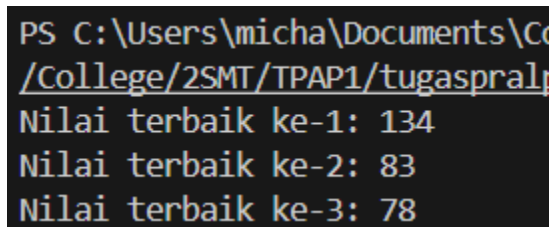
SOAL 1

A. Source Code



```
Question1.py U X Question2.py U Question3.py U artikel.txt U SrcCd.py U
Question1.py > ...
1 nlist = [5, 2, 7, 1, 73, 12, 4, 134, 74, 12, 78, 21, 83]
2 nlist.sort()
3
4 count = 1
5 for i in range(3):
6     print(f"Nilai terbaik ke-{count}: {nlist[-1-i]}")
7     count += 1
```

B. Output Result



```
PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugasprali>
Nilai terbaik ke-1: 134
Nilai terbaik ke-2: 83
Nilai terbaik ke-3: 78
```

C. Explanation

Kode di atas menggunakan fungsi list `sort()` untuk mengurutkan list yang berisi angka. Setelah diurutkan, menggunakan for loop untuk mengeluarkan 3 nilai terbaik. Nilai terbaik diambil dari list yang sudah diurutkan dan mengambil 3 index paling belakang dengan angka negatif.

SOAL 2

A. Source Code

```
Question1.py U Question2.py U x Question3.py U artikel.txt U
Question2.py > ...
1 print("Masukkan angka!")
2
3 nlist = []
4 chk = True
5 while chk:
6     try:
7         n = input(">")
8         if n == "done":
9             chk = False
10        else:
11            n = int(n)
12            nlist.append(n)
13        except:
14            print("Invalid")
15    mx = max(nlist)
16    mn = min(nlist)
17    print(f"Angka terbesar adalah: {mx}")
18    print(f"Angka terkecil adalah: {mn}")
```

B. Output Result

```
PS C:\Users\micha\Documents>
/College/2SMT/TPAP1/tugas
Masukkan angka!
>5
>6
>4
>3
>1
>9
>done
Angka terbesar adalah: 9
Angka terkecil adalah: 1
```


C. Explanation

Kode di atas list kosong dan variable chk untuk sebaga pengecek jika 'done' sudah diketik. Menggunakan while loop supaya user bisa memasukkan input berkali-kali, dan try except untuk mengatasi input selain angka supaya tidak error. Jika angka diinputkan, maka akan dimasukkan ke list "nlist" dengan fungsi list .append(). Lalu menggunakan fungsi min dan max untuk mencari angka dengan nilai terbesar dan terkecil dari dalam list.

SOAL 3

A. Source Code

```
Question1.py U Question2.py U Question3.py U X artikel.txt U SrcCd.py U
Question3.py > ...
1 handle = open("artikel.txt", "r")
2 |
3 handle_list = handle.readlines()
4 kata = []
5 for i in range(len(handle_list)):
6     list_word = handle_list[i].split()
7     for item in list_word:
8         word = ''.join(j for j in item if j.isalpha())
9         lword = word.lower()
10        if lword not in kata:
11            kata.append(lword)
12 print(kata)
13
14 handle.close()
```

Fimela.com, Jakarta Meyakini bahwa setiap individu memiliki kepribadian yang unik dengan mengetahui tipe kepribadiannya, dapat membantumu dalam memahami diri sendiri dan orang lain.
Salah satu alat untuk mengetahui tipe kepribadian adalah MBTI.

Ketika digunakan dalam konteks hubungan, tes MBTI dapat membantu mengidentifikasi chemistry antara dua individu dan memberikan wawasan yang berharga dalam membangun dan memelihara hubungan yang sehat.

Berikut adalah beberapa manfaat melakukan MBTI Chemistry Test, lengkap dengan rekomendasi website terbaik untuk melakukan tes chemistry pada MBTI pasangan maupun teman satu cirklemu!

Sebelum beranjak ke pembahasan tentang website terbaik untuk melakukan MBTI Chemistry Test, ada baiknya kamu memahami esensi dari tes itu sendiri. Berikut adalah beberapa manfaat tes chemistry.

Pemahaman Diri yang Lebih Baik: Dengan mengetahui tipe kepribadianmu, kamu dapat memahami kekuatan, kelemahan, preferensi, dan kecenderungan dalam berbagai aspek kehidupan Anda.
Hal ini dapat membantumu mengenali pola-pola perilaku dan respon emosional yang mungkin kamu alami dalam hubungan, serta memberikan landasan untuk pertumbuhan pribadi.

Memahami Kecocokan dengan Pasangan: Menggunakan MBTI untuk mengevaluasi chemistry antara kamu dan pasangan dapat memberikan wawasan yang berharga tentang kesesuaian dan kompatibilitas kepribadian.
Dengan memahami perbedaan dan kesamaan tipe kepribadian, kamu dapat menghargai dan menghormati perspektif dan preferensi satu sama lain, serta menemukan cara untuk saling melengkapi.

Meningkatkan Komunikasi: Mengetahui tipe kepribadian pasangan dapat membantu meningkatkan komunikasi dalam hubungan.
Dengan memahami preferensi komunikasi dan gaya pengambilan keputusan masing-masing, kamu dapat menyesuaikan cara berkomunikasi dan menanggapi kebutuhan satu sama lain dengan lebih efektif.

Membangun Koneksi yang Lebih Dalam: Memiliki pemahaman yang lebih dalam tentang kepribadian pasangan dan teman dapat membantumu membangun koneksi yang lebih dalam dan lebih bermakna.
Dengan memahami nilai-nilai, motivasi, dan tujuan satu sama lain, kamu pun dapat mendukung pertumbuhan dan perkembangan hubunganmu dengan cara yang positif.

B. Output Result

```
PS C:\Users\micha\Documents\College\2SMT\TPAP1\tugaspralpro9> & C:/Users/micha/PycharmProjects/Firsthello/venv/Scripts/python.exe c:/Users/micha/Documents/College/2SMT/TPAP1/tugaspralpro9/Question3.py
['finelacom', 'jakarta', 'meyakini', 'bahwa', 'setiap', 'individu', 'memiliki', 'kepribadian', 'yang', 'unik', 'dengan', 'mengetahui', 'tipe', 'kepribadian', 'nya', 'dapat', 'membantumu', 'dalam', 'memahami', 'diri', 'sendiri', 'dan', 'orang', 'lain', 'salah', 'satu', 'alat', 'untuk', 'adalah', 'mbti', 'ketika', 'digunakan', 'konteks', 'hubungan', 'tes', 'membantu', 'mengidentifikasi', 'chemistry', 'antara', 'dua', 'memberikan', 'wawasan', 'berharga', 'membangun', 'memelihara', 'sehat', 'berikut', 'beberapa', 'manfaat', 'melakukan', 'test', 'lengkap', 'rekomendasi', 'website', 'terbaik', 'melakukan', 'pada', 'pangan', 'maupun', 'teman', 'ciclemu', 'sebelum', 'beranjak', 'ke', 'pembahasan', 'tentang', 'ada', 'baiknya', 'kamu', 'esensi', 'dari', 'itu', 'berikut', 'pemahaman', 'lebih', 'baik', 'kepribadianmu', 'kekuatan', 'kelemahan', 'preferensi', 'kecenderungan', 'berbagai', 'aspek', 'kehidupan', 'anda', 'hal', 'ini', 'mengenali', 'polapola', 'perilaku', 'respon', 'emosional', 'mungkin', 'alami', 'serta', 'landasan', 'pertumbuhan', 'pribadi', 'kecocokan', 'menggunakan', 'mengevaluasi', 'kesesuaian', 'kompatibilitas', 'perbedaan', 'kesamaan', 'menghargai', 'menghormati', 'perspektif', 'sama', 'menemukan', 'cara', 'saling', 'melengkapi', 'meningkatkan', 'komunikasi', 'gaya', 'pengambilan', 'keputusan', 'masingmasing', 'menyesuaikan', 'berkomunikasi', 'menanggapi', 'kebutuhan', 'efektif', 'koneksi', 'bermakna', 'nilainilai', 'motivasi', 'tujuan', 'pun', 'mendukung', 'perkembangan', 'hubunganmu', 'positif']
```

C. Explanation

Kode di atas menggunakan open untuk membuka file .txt dan fungsi .readlines() untuk membuat list dari baris per baris dari file txt. Setelah itu for loop digunakan untuk menggunakan index untuk mengambil setiap baris dari handle_list yang kemudian di split untuk membuat list per kata dari baris file. for loop ke dua digunakan untuk mengambil setiap kata dari dalam list_word yang kemudian akan menghapus tanda baca menggunakan ".join()" dan di .lower() agar setiap kata sama. Lalu dengan IF, jika kata belum masuk list kosong, maka kata akan di tambah ke list.