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	**71241078**
Nama Lengkap	**Kevin Setiadi Wijaya**
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%)

Sifat-sifat List

List adalah suatu tempat penyimpanan dalam Pytho nyang dapat diakses menggunakan suatu fungsi build in atau fungsi buatan, list juga dapat diakses menggunakan satu nama tunggal. List mempunyai kembaran tapi beda, yaitu string bedanya string hanya menyimpan karakter namun list dapat menyimpan berbagai tipe data (integer, float, Boolean, character, string, list). Rangkaian input tersebut dapat dituliskan menggunakan [] seperti contoh 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]
```

List juga memiliki satu perbedaan dengan string, yaitu string bersifat immutable sedangkan List Mutable, Mutable artinya data dalam list dapat diganti seperti contoh berikut:

Contoh List:

```
# definisikan list berisi 4 nilai
data = [10,20,30,40]
# ubah nilai index ke-0 menjadi 50
data[0] = 50
# isinya sekarang: 50, 20, 30, 40
print(data)
```

Contoh String:

```
# definisikan sebuah string
nama = 'Antonius Rachmat'
# ubah karakter index ke-0 menjadi Z
nama[0] = 'Z'
# tidak akan mencapai baris ini karena muncul error
# TypeError: 'str' object does not support item assignment
print(nama)
```

Perbedaan selanjutnya adalah, jika terdapat dua string dengan isi yang sama, keduanya akan merujuk pada objek yang sama. Namun, pada list, dua list dengan isi yang sama tetap merupakan dua objek yang berbeda. Perilaku ini dapat dibuktikan melalui contoh di Python shell berikut:

```
>>> a = 'banana'
>>> b = 'banana'
>>> a is b
True

>>> a = [1, 2, 3]
>>> b = [1, 2, 3]
>>> a is b
False
```

Mengakses dan Mengubah isi List

Untuk mengakses List kita bisa menggunakan indexing (menunjuk kedalam listnya) saat kita menunjuk kita juga dapat mengubah isinya. 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)
```

Dalam contoh diatas kita menyimpan data list dalam variabel thislist lalu kita menampilkan List index ke (1,-1,2:5,:4) yang hasilnya akan seperti ini:

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

Kita juga dapat mengambil bagian yang kita inginkan saat melakukan print dengan format seperti ini:

Nama_Variabel = [Start:Stop:Step]

Namun kita juga dapat menggunakan nya seperti ini:

Nama_Variabel[:3],	yang berarti 3 dari depan
Nama_Variabel[-1:],	Untuk menggambil angka paling belakang
Nama_Variabel[1:],	Start dari index 1 sampai selesai

Fungsi-fungsi untuk List

Penambahan list juga dapat dilakukan menggunakan operator +, sedangkan pengulangan elemen dalam list dapat dilakukan dengan operator *. Berikut contohnya:

```
thislist = [1,2,3,4,5]
thislist2 = [6,7,8]

listbaru = thislist + thislist2
listbaru2 = thislist * 2

print(thislist)
print(thislist2)
print(listbaru)
print(listbaru2)
```

Hasilnya:

```
[1, 2, 3, 4, 5]
[6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8]
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5]
```

Metode dalam Python untuk Manipulasi List

Python menyediakan berbagai metode yang dapat digunakan untuk memanipulasi data dalam list. Berikut beberapa metode umum:

append(), Menambahkan satu elemen ke akhir list sebagai satu kesatuan.

Contoh:

mylist = [1, 2, 3]

mylist.append(4) # mylist menjadi [1, 2, 3, 4]

extend(), Menambahkan beberapa elemen ke list, dan memperlakukan setiap elemen sebagai elemen individu dalam list.

Contoh:

mylist = [1, 2, 3]

mylist.extend([4, 5]) # mylist menjadi [1, 2, 3, 4, 5]

sort(), Mengurutkan elemen-elemen dalam list secara ascending (menaik).

Contoh:

```
mylist = [3, 1, 4, 2]

mylist.sort()  # mylist menjadi [1, 2, 3, 4]
```

pop([index]), Menghapus elemen dari list berdasarkan indeks dan mengembalikan nilainya. Jika tidak diberikan indeks, akan menghapus elemen terakhir.

Contoh:

```
mylist = [1, 2, 3]

removed = mylist.pop(1)  # removed = 2, mylist = [1, 3]
```

del, Menghapus elemen dari list berdasarkan indeks, tetapi tidak mengembalikan nilai.

Contoh:

```
mylist = [1, 2, 3]

del mylist[1]  # mylist menjadi [1, 3]
```

remove(), Menghapus elemen dari list berdasarkan nilai.

Contoh:

```
mylist = [1, 2, 3]

mylist.remove(2)  # mylist menjadi [1, 3]
```

reverse(), Membalik urutan elemen dalam list.

Contoh:

```
mylist = [1, 2, 3]

mylist.reverse()  # mylist menjadi [3, 2, 1]
```

Fungsi Built-in untuk List

Python juga memiliki beberapa fungsi bawaan yang umum digunakan dengan list:

min(), Mengembalikan nilai terkecil dari elemen list.

Contoh:

```
angka = [10, 20, 5, 8]
print(min(angka))  # Output: 5
```

max(), Mengembalikan nilai terbesar dari elemen list.

Contoh:

```
print(max(angka))  # Output: 20
```

sum(), Menghitung jumlah seluruh elemen dalam list.

Contoh:

```
print(sum(angka))  # Output: 43
```

len(), Mengembalikan jumlah elemen dalam list.

Contoh:

```
print(len(angka))  # Output: 4
```

List Comprehension

List comprehension adalah cara singkat untuk membuat list berdasarkan list lain, dengan sintaks yang ringkas dan efisien.

Contoh 1: Elemen yang mengandung substring "or"

```
fruits = ["anton", "funcoro", "buntoro", "yuworo", "margono"]
hasil = [item for item in fruits if "or" in item]
```

Contoh 2: Mengubah elemen menjadi huruf besar

```
hasil = [item.upper() for item in fruits]
```

Contoh 3: Memilih elemen yang panjangnya lebih dari 6 huruf

```
hasil = [item for item in fruits if len(item) > 6]
```

List sebagai parameter Fungsi

Jika sebuah list digunakan sebagai parameter dalam fungsi, maka ia bersifat mutable. Artinya, perubahan yang dilakukan terhadap list di dalam fungsi akan berdampak langsung pada list aslinya di luar fungsi, Contoh:

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

data = ["a", "b", "c"]
ubah(data)
print(data)  # Output: ['anton', 'b', 'c']
```

Keuntungan Menggunakan List sebagai Parameter:

1. Fleksibilitas:
   Fungsi dapat digunakan untuk memproses berbagai jenis data dalam list.
2. Reusabilitas:
   Fungsi dapat dipakai kembali dengan list yang berbeda.
3. Pembagian Tugas:

   List membantu membagi data menjadi bagian-bagian yang lebih kecil untuk diproses.

4. Pengorganisasian Data:
   List mempermudah dalam mengelompokkan data sebelum diproses oleh fungsi.

Contoh Lain: Menghapus Elemen Pertama dari List

```
def hapus(data):
    return data[1:]

data = ["anton", "b", "c"]
data = hapus(data)
print(data)  # Output: ['b', 'c']
```

BAGIAN 2: LATIHAN MANDIRI (60%)

Link Git: https://github.com/IS-Miyuki/LaporanMingguKeSepuluh.git

SOAL 1

Source Code:

```python
# Latihan 10.1: Mencari 3 bilangan terbaik dari sebuah list
def find_top_three(numbers):
    if len(numbers) < 3:
        return "List harus memiliki minimal 3 bilangan"
    sorted_numbers = sorted(numbers)
    return sorted_numbers[-3:]
```

Contoh Penggunaan:

```python
# Contoh penggunaan Latihan 10.1
numbers = [5, 2, 8, 1, 9, 3, 7, 4, 6]
print("Latihan 10.1 - 3 bilangan terbaik:")
print(find_top_three(numbers))
```

```
Latihan 10.1 - 3 bilangan terbaik:
[7, 8, 9]
```

Penjelasan:

1. Buat fungsi yang akan menerima variabel.
2. Jika Panjang data kurang dari 3 maka akan mengembalikan string yang berisi error code.
3. Urutkan data.
4. Ambil 3 dari belakang, maka angka itu adalah angka terbesar.

SOAL 2

Source Code:

```python
# Latihan 10.2: Program untuk menghitung rata-rata bilangan yang diinput user
def calculate_average():
    numbers = []
    while True:
        user_input = input("Masukkan bilangan (atau 'done' untuk selesai): ")
        if user_input.lower() == 'done':
            break
        try:
            number = float(user_input)
            numbers.append(number)
        except ValueError:
            print("Input tidak valid. Masukkan bilangan atau 'done'")
    average = sum(numbers) / len(numbers)
    return f"Rata-rata dari bilangan yang dimasukkan: {average}"
```

Contoh Penggunaan:

```python
# Contoh penggunaan Latihan 10.2
print("\nLatihan 10.2 - Hitung rata-rata:")
print(calculate_average())
```

```
Latihan 10.2 - Hitung rata-rata:
Masukkan bilangan (atau 'done' untuk selesai): 1
Masukkan bilangan (atau 'done' untuk selesai): 2
Masukkan bilangan (atau 'done' untuk selesai): 3
Masukkan bilangan (atau 'done' untuk selesai): 4
Masukkan bilangan (atau 'done' untuk selesai): done
Rata-rata dari bilangan yang dimasukkan: 2.5
```

Penjelasan:

1. Buat variabel yang tidak menerima parameter apapun karena semua data berasal dari input user.
2. Buat list kosong untuk menampung semua input user.
3. Buat perulangan yang akan di break loop Ketika user memasukkan string “done”.
4. Buat variabel user_input untuk menerima inputan user.
5. Buat if statement jika input_user == “done” maka loop akan di break.
6. Lalu gunakan error handling supaya saat jika input user tidak berupa string angka maka tidak merusak program.
7. Ubah input_user menjadi float, lalu masukkan kedalam variabel number.
8. Buat variabel average yang berisi nilai total dari variabel number dan frekuensi dari variabel number.

SOAL 3

Source Code:

```python
# Latihan 10.3: Program untuk menemukan kata-kata unik dalam sebuah file teks
def find_unique_words(file_path):

    with open(file_path, 'r') as file:
        text = file.read()

    words = text.lower().split()
    words = [word.strip('.,!?()[]{}"\'') for word in words]
    unique_words = sorted(set(words))

    print(f"\nTotal kata dalam file: {len(words)}")
    print(f"Jumlah kata unik: {len(unique_words)}")
    print("\nDaftar kata unik:")

    half_length = (len(unique_words) + 1) // 2
    for i in range(half_length):
        word1 = unique_words[i]
        word2 = unique_words[i + half_length] if i + half_length < len(unique_words) else ""
        print(f"{i+1}. {word1:<20} {i+1+half_length}. {word2}")
    return unique_words
```

Contoh Penggunaan:

```python
# Contoh penggunaan Latihan 10.3
print("\nLatihan 10.3 - Kata-kata unik dalam file:")
file_path = "artikel.txt"
find_unique_words(file_path)
```

artikel.txt

```
1  Python adalah bahasa pemrograman yang populer dan mudah dipelajari.
2  Python digunakan untuk berbagai keperluan seperti pengembangan web,
3  analisis data, kecerdasan buatan, dan banyak lagi.
```

```
Latihan 10.3 - Kata-kata unik dalam file:

Total kata dalam file: 24
Jumlah kata unik: 22

Daftar kata unik:
1. adalah               12. keperluan
2. analisis             13. lagi
3. bahasa               14. mudah
4. banyak               15. pemrograman
5. berbagai             16. pengembangan
6. buatan               17. populer
7. dan                  18. python
8. data                 19. seperti
9. digunakan            20. untuk
10. dipelajari           21. web
11. kecerdasan           22. yang
```

Penjelasan:

1. Buat fungsi yang akan menerima Lokasi file.
2. Buka dan baca file.
3. Ubah semua kata menjadi huruf kecil dan pisah spasi
4. Hilangakan semua symbol di words
5. Buat variable yang akan menampung dan mengurutkan semua kata unik dengan menggunakan set karena di dalam set tidak ada kalimat yang di double.
6. Tampilkan total kata dalam file
7. Tampilkan total kata untik dalam file.
8. Gunakan perulangan untuk setengah Panjang dari kata unik agar bisa ditampilkan dalam 2 kolom
9. Buat variable untuk menghitung nilai Tengah data agar bisa menampilkan dalam 2 kolom, lalu print.