

Nama : Lu'lu'un Fifalaqisa
NIM : 222410103038
Functional Programming – A

Tugas 1 dan 2 – Kelas Praktikum Functional Programming

Tugas 1

Kode 1

```
1 # Kode 1
2 def sequenceGenerator(start, stop):
3     x = []
4     for i in range(start, stop+1):
5         x.append(i)
6     return x
```

Dari soal yang diberikan, source code tersebut saya ubah menjadi seperti dibawah ini agar menjadi sederhana dalam satu line. Kode yang diubah menggunakan ekspresi **lambda**, fungsi **range()**, dan **list()** untuk menghasilkan hasil yang

sama dengan kode awal, namun dengan functional programming.

```
1 #KODE 1
2 sequenceGenerator_satuline = lambda start, stop: list(range(start, stop+1))
3 result1 = sequenceGenerator_satuline(0, 5)
4 print(result1)
```

Penjelasan Kode 1: Beberapa perubahan dan penambahan code yang dilakukan agar menjadi implementasi functional programming yaitu:

1) Penggunaan Lambda Expression

Kode awal menggunakan definisi fungsi dengan kata kunci def untuk sequenceGenerator. Maka disini saya ubah menggunakan lambda agar code yang dibuat menjadi lebih singkat.

2) Penggunaan list() dan range()

Perulangan for diubah menjadi menggunakan range(). Range() disini berguna untuk menghasilkan urutan bilangan bulat dari start hingga stop+1. Kemudian, hasil dari range() ini akan dimasukkan dalam bentuk list menggunakan function list() tadi.

3) Menghilangkan variabel x:

Awalnya variabel x yang berupa list kosong digunakan sebagai tempat untuk menyimpan elemen yang terurut. Jadi variabel x disini saya hilangkan, dan diubah menggunakan fungsi list() tadi yang gunanya sama.

Contoh Output:

```
result1 = sequenceGenerator_satuline(0, 5)
print(result1)
```

```
[0, 1, 2, 3, 4, 5]
```

Kode 2

```
8 #Kode 2
9 def fizzBuzz(a,b):
10     x = []
11     for num in range(a,b):
12         if num % 3 == 0 and num % 5 == 0:
13             x.append("FizzBuzz")
14         elif num % 3 == 0:
15             x.append("Fizz")
16         elif num % 5 == 0:
17             x.append("Buzz")
18         else:
19             x.append(num)
20     return x
```

Dari soal yang diberikan, source code tersebut saya ubah menjadi seperti dibawah ini agar menjadi sederhana dalam satu line. Kode fizzBuzz telah diubah menjadi satu baris dan menggunakan beberapa metode fungsional programming, seperti **lambda**, **map**, dan **operasi kondisional (ternary)** untuk menggantikan loop for dan pernyataan if-else dalam soal.

```
6 #KODE 2
7 fizzBuzz_satuline = lambda a, b: list(map(lambda num: "FizzBuzz" if num % 3 == 0 and num % 5 == 0 else "Fizz" if num % 3 == 0 else "Buzz" if num % 5 == 0 else num, range(a, b)))
8 result2 = fizzBuzz_satuline(0, 5)
9 print(result2)
```

Penjelasan Kode 2: Beberapa perubahan dan penambahan code yang dilakukan agar menjadi implementasi functional programming yaitu:

1) Penggunaan Lambda Expression

Disini digunakan lambda expression untuk menggantikan fungsi bernama fizzBuzz. Lambda digunakan untuk menentukan bagaimana setiap angka dari rentang a hingga b akan diproses dan diubah.

2) Penggunaan list() dan map():

- map(function, iterable) adalah fungsi bawaan Python yang digunakan untuk menerapkan function ke setiap elemen dalam iterable (misalnya, daftar atau rentang). Penggunaan map untuk menerapkan fungsi lambda pada setiap angka dalam rentang dari a hingga b.
- list() digunakan untuk mengubah objek map menjadi list yang berisi hasil dari aplikasi fungsi lambda pada setiap elemen dalam rentang

3) Penggunaan Ternary (Kondisional) Expression

Pernyataan kondisional (ternary) digunakan untuk mengevaluasi suatu ekspresi berdasarkan kondisi yang diberikan. Kondisional expression digunakan untuk menggantikan pernyataan if-else yang ada pada soal.

- Dalam kondisional expression, memiliki struktur condition ? value_if_true : value_if_false. Setiap elemen dari rentang a hingga b akan diperiksa terlebih dahulu apakah itu merupakan "FizzBuzz" (jika modulus 3 dan 5 sama dengan 0), jika tidak, maka apakah itu "Fizz" (jika hanya modulus 3 = 0), jika tidak, maka apakah itu "Buzz" (jika hanya modulus 5 = 0), dan jika tidak ada kondisi sebelumnya yang terpenuhi, maka elemen tersebut dianggap sebagai angka itu sendiri (num). Ternary expression ini memungkinkan untuk membuat perkondisian secara ringkas dalam satu baris kode.

Contoh Output:

```
result2 = fizzBuzz_satuline(0, 5)
print(result2)
```

```
['FizzBuzz', 1, 2, 'Fizz', 4]
```

Kode 3

```
22 #Kode 3
23 def twoNumber(l):
24     res = []
25     for i in l:
26         if l.index(i) == len(l)-1:
27             break
28         z = i + l[l.index(i)+1]
29         res.append(z)
30     return res
```

Dari soal yang diberikan, source code tersebut saya ubah menjadi seperti dibawah ini agar menjadi sederhana dalam satu line. **Lambda** digunakan untuk menggambarkan operasi yang akan diaplikasikan pada setiap elemen dalam list, **map()** digunakan untuk menerapkan operasi ini pada setiap elemen dalam **range()**, dan **list()** digunakan

untuk mengubah hasilnya menjadi list yang berisi hasil perhitungan.

```
11 #KODE 3
12 twoNumber_satuline = lambda l: list(map(lambda i: l[i] + l[l.index(i)+1], range(len(l)-1)))
13 result3 = twoNumber_satuline([1, 2, 3, 4, 5])
14 print(result3)
```

Penjelasan Kode 3: Beberapa perubahan dan penambahan code yang dilakukan agar menjadi implementasi functional programming yaitu:

1) Penggunaan Lambda Expression:

- Lambda expression digunakan untuk membuat fungsi yang menerima argumen i (indeks) dan mengembalikan hasil penjumlahan antara elemen l[i] dan l[i+1].
- Fungsi lambda digunakan dalam konteks map untuk mengaplikasikan fungsi ini pada setiap elemen dalam range(len(l)-1). l merupakan suatu list yang menjadi parameter inputan.

2) Penggunaan list():

- Fungsi list() digunakan untuk mengonversi hasil dari operasi map menjadi list yang berisi hasil perhitungan yang dihasilkan oleh pemanggilan fungsi lambda pada setiap elemen dalam rentang yang diberikan.

3) Penggunaan map():

- map() adalah fungsi yang digunakan untuk menerapkan fungsi tertentu pada setiap elemen dalam iterable (dalam hal ini, iterable adalah range(len(l)-1)).
- Penggunaan map untuk menerapkan fungsi lambda pada setiap indeks i dalam rentang range(len(l)-1). Hasil dari map adalah objek iterable yang berisi hasil perhitungan dari fungsi lambda ini.

4) Penggunaan range():

- range() adalah fungsi yang digunakan untuk menghasilkan rentang angka. Penggunaan range(len(l)-1) untuk membuat rentang angka dari 0 hingga panjang list l dikurangi 1. Rentang ini digunakan sebagai indeks saat mengakses elemen dalam list l.

Contoh Output:

```
result3 = twoNumber_satuline([1, 2, 3, 4, 5])
print(result3)
```

```
[3, 5, 7, 9]
```

Tugas 2

Berikut ini adalah code python yang sudah diubah tadi lalu diubah kembali menggunakan bahasa dart. Untuk penulisannya sebagai berikut:

```
2 // KODE 1
3 List<int> sequenceGeneratorSatuLine(int start, int stop) => List<int>.generate(stop - start + 1, (index) => start + index);
4 List<int> result1 = sequenceGeneratorSatuLine(0, 5);
5
6
7 // KODE 2
8 List<String> fizzBuzzSatuLine(int a, int b) => List<String>.generate(b - a, (index) {
9   int num = a + index;
10  return (num % 3 == 0 && num % 5 == 0) ? "FizzBuzz" : (num % 3 == 0) ? "Fizz" : (num % 5 == 0) ? "Buzz" : num.toString();
11 });
12 List<String> result2 = fizzBuzzSatuLine(0, 5);
13
14
15 // KODE 3
16 List<int> twoNumberSatuLine(List<int> l) => List<int>.generate(l.length - 1, (index) => l[index] + l[index + 1]);
17 List<int> result3 = twoNumberSatuLine([0, 1, 2, 3, 4, 5]);
18
19
20 //Output Kode 1, 2 dan 3
21 void main(){
22   print(result1);
23   print(result2);
24   print(result3);
25 }
```

Penjelasan KODE 1:

- sequenceGeneratorSatuLine adalah sebuah fungsi yang menghasilkan daftar (List) berisi urutan bilangan bulat dalam satu baris.
- Fungsi ini mengambil dua parameter, start dan stop, yang menentukan batas awal dan batas akhir dari urutan.
- Menggunakan metode List.generate untuk membuat daftar bilangan bulat dari start hingga stop.

Penjelasan KODE 2:

- fizzBuzzSatuLine adalah sebuah fungsi yang menghasilkan daftar (List) berisi deret FizzBuzz dalam satu baris.
- Fungsi ini mengambil dua parameter, a dan b, yang menentukan batas awal dan akhir untuk deret FizzBuzz.
- Menggunakan metode List.generate untuk membuat daftar deret FizzBuzz dari a hingga b.
- Fungsi ini memeriksa apakah suatu angka adalah kelipatan 3, 5, atau keduanya, dan mengembalikan "Fizz", "Buzz", "FizzBuzz", atau angka itu sendiri sesuai dengan kondisi.

Penjelasan KODE 3:

- twoNumberSatuLine adalah sebuah fungsi yang menghasilkan daftar (List) berisi penjumlahan dua angka berurutan dalam satu baris.
- Fungsi ini mengambil satu parameter, yaitu daftar bilangan bulat l.
- Menggunakan metode List.generate untuk membuat daftar baru yang berisi penjumlahan dari dua angka berurutan dalam daftar l.

Output Kode 1, 2, dan 3:

```
[0, 1, 2, 3, 4, 5]
[FizzBuzz, 1, 2, Fizz, 4]
[1, 3, 5, 7, 9]
```

Pada bagian main, tiga hasil (result1, result2, dan result3) dicetak ke layar dengan perintah print. result1 berisi urutan bilangan bulat dari 0 hingga 5, result2 berisi deret FizzBuzz dari 0 hingga 5, dan result3 berisi hasil penjumlahan dua angka berurutan dari daftar yang diberikan.