

LAPORAN
PRAKTIKUM ALGORITMA DAN PEMROGRAMAN
SEMESTER GENAP 2019/2020

PERTEMUAN 13
PENGANTAR BAHASA PEMROGRAMAN PYTHON



DISUSUN OLEH:
Mardonius Riel (71180293)

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2020

BAGIAN 1: MATERI PRAKTIKUM

Rekursi adalah proses pengulangan sesuatu dengan cara kesamaan-diri. Sebagai contohnya, saat dua cermin berada paralel antara satu dengan yang lain, gambar yang tertangkap adalah suatu bentuk rekursi tak-terbatas. Istilah ini memiliki makna beragam bergantung kepada ragam disiplin mulai dari linguistik sampai logika. Penggunaan paling umum dari rekursi yaitu dalam matematika dan ilmu komputer, yang mengacu kepada suatu metode mendefinisikan fungsi yang mana fungsi tersebut menggunakan definisinya sendiri. Secara spesifik hal ini mendefinisikan suatu instansi tak-terbatas (nilai fungsi), menggunakan ekspresi terbatas dengan beberapa instansi bisa merujuk ke instansi lainnya, tetapi dengan suatu cara sehingga tidak ada perulangan atau keterkaitan tak-terbatas dapat terjadi. Istilah ini juga digunakan secara umum untuk menjelaskan suatu proses pengulangan objek dengan cara kesamaan-diri.

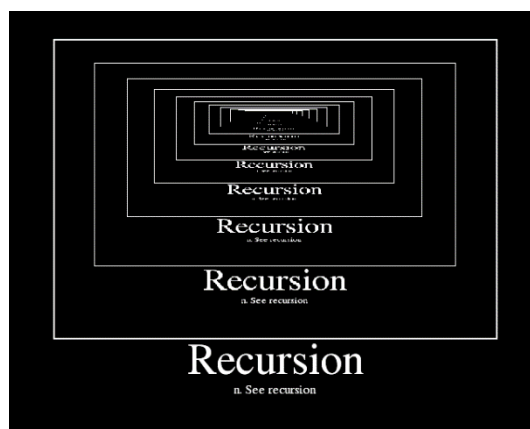
Pemanggilan prosedur atau fungsi ke dirinya sendiri bisa berarti proses yang berulang yang tidak bisa diketahui kapan akan berakhir. Dalam pemakaian sehari-hari, rekursi merupakan teknik pemrograman yang berdaya guna untuk digunakan pada pekerjaan pemrograman dengan mengeksperiskannya ke dalam suku-suku dari program lain dengan menambahkan langkahlangkah sejenis.

KELEBIHAN

- Program lebih singkat.
- Pada beberapa kasus, lebih mudah menggunakan fungsi rekursif, contohnya: pangkat, factorial, dan fibonacci, dan beberapa proses deret lainnya.
- Lebih efisien dan cepat dibandingkan proses secara iteratif.

KEKURANGAN

- Memakan memori lebih besar, karena setiap bagian dari dirinya dipanggil, akan membutuhkan sejumlah ruang memori untuk penyimpanan.
- Rekursif sering kali tidak bisa berhenti sehingga memori akan terpakai habis dan program bisa hang.



BAGIAN 2: JAWABAN SOAL LATIHAN MANDIRI

1. Program untuk mengecek prima atau bukan



```
8.py > ...
1  angka = 0
2  def cek(pembagi):
3      if(pembagi == 1):
4          return("prima")
5      elif(angka % pembagi == 0 ):
6          return("bukan prima")
7      else:
8          return cek(pembagi-1)
9  angka = 11
10 print(cek(angka-1))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS D:\Coding\Python\Python\Pertemuan 14 minggu> & 'C:\Users\Riel\.vscode\extensions\ms-python.python-2020.5.80290\pythonFiles\pythonFileLauncher\Pertemuan 14 minggu\8.py'
prima
PS D:\Coding\Python\Python\Pertemuan 14 minggu> cd 'd:\Coding\Python\Python37-32' & 'c:\Users\Riel\.vscode\extensions\ms-python.python-2020.5.80290\pythonFiles\pythonFileLauncher\Pertemuan 14 minggu\8.py'
bukan prima
PS D:\Coding\Python\Python\Pertemuan 14 minggu> 
```

Result : 11 adalah prima dan 6 bukan prima

angka = 0

def cek(pembagi):

if(pembagi == 1):

return("prima")

elif(angka % pembagi == 0):

return("bukan prima")

else:

```
    return cek(pembagi-1)
```

```
angka = 6
```

```
print(cek(angka-1))
```

2. Soal Palindrome menggunakan rekursif

```
19.py 16.py 12.py 8.py
soal2.py > ...
1 def is_palindrome(s):
2     if len(s) < 1:
3         return True
4     else:
5         if s[0] == s[-1]:
6             return is_palindrome(s[1:-1])
7         else:
8             return False
9 a=str(input("Kalimat :"))
10 if(is_palindrome(a)==True):
11     print("palindrome")
12 else:
13     print("bukan palindrome")
```

```
s:\Python\Python37-32\python.exe 'c:\Users\Riel\.vscode\python\bugpy\launcher' '56944' '--' 'd:\Coding\Python\Python\Pertemuan 14 minggu> 
Kalimat :kasur ini rusak
palindrome
PS D:\Coding\Python\Python\Pertemuan 14 minggu> 
```

```
def is_palindrome(s):
    if len(s) < 1:
        return True
    else:
        if s[0] == s[-1]:
            return is_palindrome(s[1:-1])
        else:
            return False
a=str(input("Kalimat :"))
if(is_palindrome(a)==True):
    print("palindrome")
else:
```

```
print("bukan palindrome")
```

#Source code diambil dari <https://www.sanfoundry.com/python-program-check-string-palindrome-using-recursion/>

1. pengguna harus memasukkan string.
2. string dilewatkan sebagai argumen untuk fungsi rekursif.
3. dalam fungsi, jika panjang string kurang dari 1, True dikembalikan.
4. jika huruf terakhir sama dengan huruf pertama, fungsinya disebut rekursif dengan argumen sebagai daftar slicenya dengan karakter pertama dan karakter terakhir dihapus, yang lain mengembalikan false.
5. pernyataan if digunakan untuk memeriksa apakah nilai yang dikembalikan adalah true atau false dan hasil akhirnya dicetak.

3. Hitung jumlah deret

```
UG_Rekursif_71180293 > 71180293-C-4.py > ganjil
1  def ganjil(b):
2      if b==1:
3          print(b)
4          return b
5      else:
6          if b % 2 != 0 :
7              print(b)
8              return b + ganjil(b-1)
9          else:
10             return ganjil(b-1)
11
12  print(ganjil(10))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Kalimat :kasur ini rusak
palindrome
PS D:\Coding\Python\Python\Pertemuan 14 minggu> cd '
s\Python\Python37-32\python.exe' 'c:\Users\Riel\.vsco
bugpy\launcher' '57066' '--' 'd:\Coding\Python\Python
9
7
5
3
1
25
PS D:\Coding\Python\Python\Pertemuan 14 minggu> |
```

```
def ganjil(b):
    if b==1:
        print(b)
        return b
    else:
        if b % 2 != 0 :
            print(b)
            return b + ganjil(b-1)
```

else:

 return ganjil(b-1)

print(ganjil(10))

Alur programnya adalah mencari angka ganjil didalam range pada parameter yang dikirim(10), mencari angka ganji adalah parameter % 2 !=0 (line 6)

10 % 2 = Genap

9 % 2 = Ganjil

8 % 2 = Genap

7 % 2 = Ganjil

Dst..

Nah proses ini adalah proses perulangan, pada saat kita menguji ganjil genapnya bisa dilakukan berulang-ulang pada fungsi yang sama(rekursifnya) itu ada pada line 6 – line 10.

4. Fungsi Rekursif dari jumlah digit suatu bilangan

```
19.py 16.py 12.py
soal4.py > ...
1 def sum(aa):
2     if len(aa) == 1:
3         return aa[0]
4     else:
5         return aa[0] + sum(aa[1:])
6
7 print(sum([5,7,3,8,10]))
```

```
PS D:\Coding\Python\Python\Pertemuan 14 minggu> cd
s\Python\Python37-32\python.exe 'c:\Users\Riel\.vscode\
bugpy\launcher' '57423' '--' 'd:\Coding\Python\Python
33
PS D:\Coding\Python\Python\Pertemuan 14 minggu> 
```

$5 + 7 + 3 + 8 + 10 = 33$

```
def sum(aa):
```

```
    if len(aa) == 1:
```

```
        return aa[0]
```

```
    else:
```

```
        return aa[0] + sum(aa[1:])
```

```
print(sum([5,7,3,8,10]))
```

Jika panjang list mengembalikan list (kondisi berhenti). Yang lain, mengembalikan elemen dan panggilan ke jumlah fungsi () minus satu elemen dari daftar. Jika semua panggilan dieksekusi, maka akan kembali mencapai kondisi penghentian dan mengembalikan jawabannya.

5. Rekursif hitung kombinasi

```
soal5.py > ...
1  def komb(a,b):
2      hasil=faktorial(a)/(faktorial(a-b)*faktorial(b))
3      return hasil
4
5
6  def faktorial(n):
7      if n==0 or n==1:
8          hasil=1
9          return hasil
10     else:
11         hasil=faktorial(n-1) * n
12         return hasil
13
14  print(komb(10,3))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved

Try the new cross-platform PowerShell https://aka.ms/powershell

PS D:\Coding\Python\Python\Pertemuan 14 minggu> & 'C:\Users\user\AppData\Local\Microsoft\WindowsApps\Microsoft.Windows.Common-Stack-1.0.0.16398\pythonFile
hon\Pertemuan 14 minggu\soal5.py'
120.0
PS D:\Coding\Python\Python\Pertemuan 14 minggu> |
```

kombinasi

alur program :

- 1 membuat fungsi yang memasukan parameter untuk perhitungan kombniasi.
2. karena rumus kombniasi memakai perhitungan faktorial, maka dari fungsi perhitungan kombinasi tinggal memanggil perhitungan faktorial.

BAGIAN 3 : Soal sendir

1. Fibonnaci

```
sendiri1.py > ...
1  def recur_fibo(n):
2      if n <= 1:
3          return n
4      else:
5          return(recur_fibo(n-1) + recur_fibo(n-2))
6
7  nterms = 10 #inputnya
8
9  # check if the number of terms is valid
10 if nterms <= 0:
11     print("bilangan positif")
12 else:
13     print("Fibonacci sequence:")
14     for i in range(nterms):
15         print(recur_fibo(i))
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Fibonacci sequence:
0
1
1
2
3
5
8
13
21
34
PS D:\Coding\Python\Python\Pertemuan 14 minggu> 
```

```
def recur_fibo(n):
```

```
    if n <= 1:
```

```
        return n
```

```
    else:
```

```
    return(recur_fibo(n-1) + recur_fibo(n-2))
```

```
nterms = 10 #inputnya
```

```
# check if the number of terms is valid
```

```
if nterms <= 0:
```

```
    print("bilangan positif")
```

```
else:
```

```
    print("Fibonacci sequence:")
```

```
    for i in range(nterms):
```

```
        print(recur_fibo(i))
```

#source code <https://www.programiz.com/python-programming/examples/fibonacci-recursion>

2. Faktorial

```
sendiri1.py > ...  
1  def factorial(x):  
2      if x == 1:  
3          return 1  
4      else:  
5          return (x * factorial(x-1))  
6  
7  num = 3  
8  print("Faktorial dari", num, "adalah", factorial(num))
```

```
PS D:\Coding\Python\Python\Pertemuan 14 minggu> cd  
s\Python\Python37-32\python.exe' 'c:\Users\Riel\.vs  
ugpy\launcher' '50847' '--' 'd:\Coding\Python\Pyth  
Faktorial dari 3 adalah 6  
PS D:\Coding\Python\Python\Pertemuan 14 minggu> 
```

```
def factorial(x):
```

```
    if x == 1:
```

```
        return 1
```

```
    else:
```

```
        return (x * factorial(x-1))
```

```
num = 3
```

```
print("Faktorial dari", num, "adalah", factorial(num))
```


BAGIAN 4: CREDITS / REFERENSI / DAFTAR PUSTAKA

Pada bagian ini tuliskan referensi-referensi yang anda gunakan dalam menyusun laporan praktikum ini. Referensi bisa berupa buku, website atau sumber-sumber lainnya. Jika anda meminta bantuan dari teman anda, tuliskan nama dan NIM teman anda di sini (supaya anda terhindar dari plagiasi).

<https://www.duniaikom.com/tutorial-belajar-python-tipe-data-dictionary-dalam-bahasa-python/>

<https://www.w3resource.com/python-exercises/tuple/python-tuple-exercise-16.php>

<https://www.sanfoundry.com/python-program-check-string-palindrome-using-recursion/>