**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 ekpresi 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 mengeksperisikannya ke dalam suku-suku dari program lain dengan menambahkan langkahlangkah sejenis.

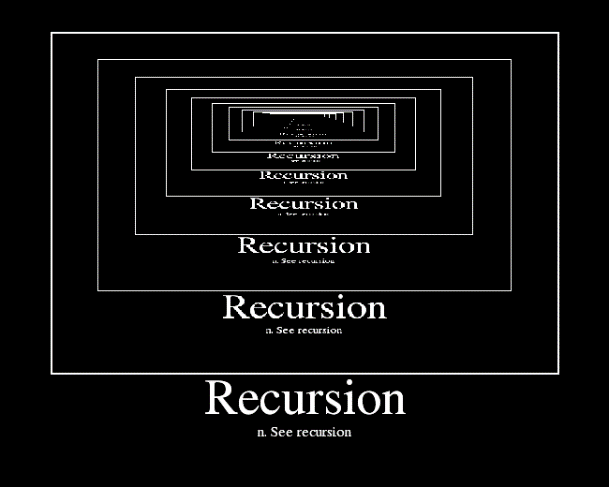
# KELEBIHAN

# Program lebih singkat.

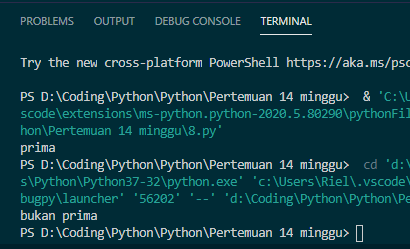
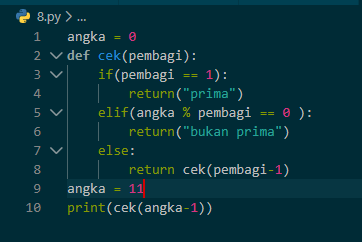
# Pada beberapa kasus, lebih mudah menggunakan fungsi rekursif, contohnya: pangkat, factorial, dan fibonacci, dan beberapa proses deret lainnya.

# 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 ngecek prima atau bukan

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")

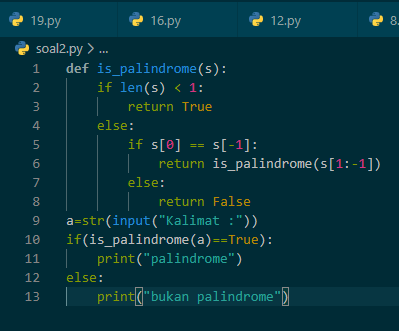
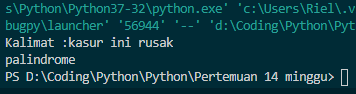
else:

return cek(pembagi-1)

angka = 6

print(cek(angka-1))

1. Soal Palindrome menggunakan rekursif



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.

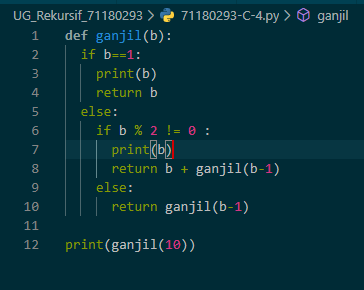
2. string dilewatkan sebagai argumen untuk fungsi rekursif.

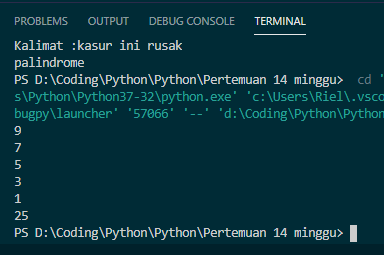
3. dalam fungsi, jika panjang string kurang dari 1, True dikembalikan.

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.

1. Hitung jumlah deret





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

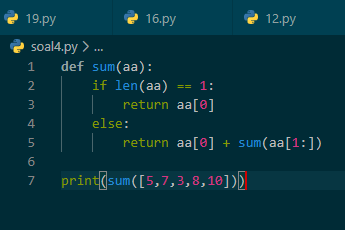
9 % 2 = Ganjil

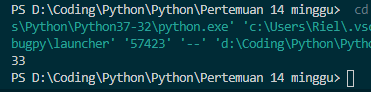
8 % 2 = Genap

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.

1. Fungsi Rekursif dari jumlah digit suatu bilangan



5 + 7 + 3 + 8 + 10 = 33

def sum(aa):

if len(aa) == 1:

return aa[0]

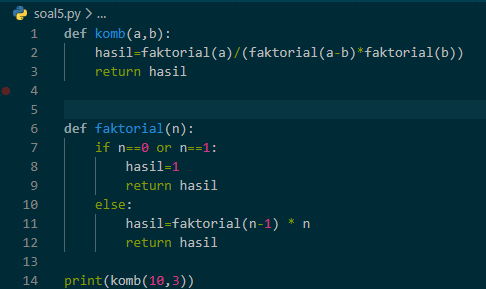
else:

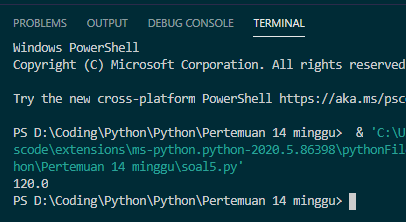
return aa[0] + sum(aa[1:])

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.

1. Rekurisf hitung kombinasi





# kombinasi

alur program :

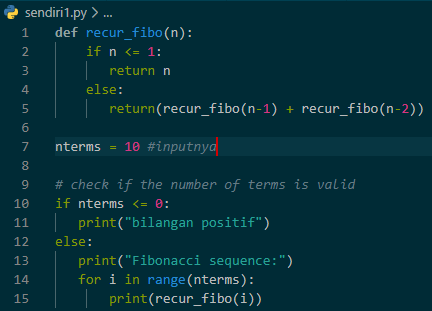
1 membuat fungsi yang memasukan parameter untuk perhitungan kompbinasi.

2. karena rumus kombnasi memakai perhitungan faktorial, maka dari fungsi perhitungan

kombinasi tinggal memanggil perhitungan faktorial.

# BAGIAN 3 : Soal sendir

1. Fibonnaci



# 

# 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:

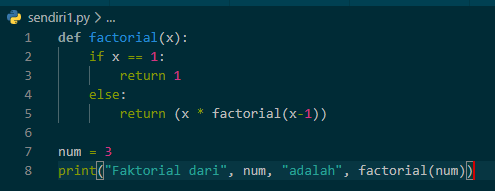
# print("Fibonacci sequence:")

# for i in range(nterms):

# print(recur\_fibo(i))

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

1. Faktorial



# 

# 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.duniailkom.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/>