

# Laporan Praktikum Algoritma dan Pemrograman

Semester Genap 2023/2024



<b>NIM</b>	Fransiskus Tedyanto seran
<b>Nama Lengkap</b>	71231006
<b>Minggu ke / Materi</b>	13 / Fungsi Rekursif

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  
YOGYAKARTA  
2023

## BAGIAN 1: MATERI MINGGU INI

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

### MATERI 1

#### Pengertian Rekursif

Rekursif merupakan sebuah konsep dalam pemrograman dimana sebuah fungsi dapat memanggil dirinya sendiri secara berulang. Pada dasarnya sebuah fungsi rekursif terdiri dari dua bagian utama: basis atau kondisi terminasi, dan langkah rekursif. Basis adalah kondisi yang menuntukan kapan fungsi rekursif harus berhenti memanggil dirinya sendiri, dan mengembalikan hasil.

Di dalam fungsi rekursif merupakan fungsi yang harus digunakan secara berhati-hati karena fungsi tersebut dapat bersifat unlimited loop sehingga dapat menyebabkan program hang up. Fungsi rekursif akan terus berjalan sampai kondisi berhenti terpenuhi, oleh karena itu dalam sebuah fungsi rekursif perlu terdapat 2 blok yang penting, yaitu blok yang menjadi titik berhenti dari sebuah proses rekursif dan blok yang memanggil dirinya sendiri. Didalam rekursif ini ada terdapat 2 bagian yaitu:

Base case: dalam rekursif kondisi ini menentukan kapan harus berhenti memanggil dirinya sendiri, dan mengembalikan hasil. Base case merupakan kondisi terminasi yang menjadi titik akhir dari rekursif. Ketika fungsi rekursif dieksekusi, ia akan memeriksa base case terlebih dahulu. Jika base case terpenuhi, artinya situasi yang memerlukan pemanggilan rekursif telah selesai, dan fungsi akan mengembalikan nilai hasil atau melakukan operasi terakhir yang diinginkan. Tanpa adanya base case kasus yang jelas, rekursif akan berlanjut secara tak terbatas dan menyebabkan stack overflow atau kegagalan program.

Rekursif case: rekursif case merupakan bagian dari fungsi rekursif dimana fungsi tersebut memanggil dirinya sendiri dengan argumen yang lebih sederhana atau lebih kecil. Setiap panggilan rekursif membawa masalah tersebut menuju ke ukuran yang lebih kecil atau situasi yang lebih sederhana, sehingga pada akhirnya mencapai base case.

### MATERI 2

#### Kelebihan dan Kekurangan

Fungsi rekursif memiliki kelebihan serta kelemahan yaitu:

Kelebihan:

1. pemecahan masalah yang elegan: fungsi rekursif dapat menyederhanakan pemecahan masalah kompleks menjadi submasalah yang lebih kecil, dan lebih mudah diatasi.
2. Solusi yang fleksibel: rekursif memungkinkan pemanggilan fungsi itu sendiri, sehingga memungkinkan pengulangan, dan iterasi melalui struktur data secara efisien.
3. Pendekatan algoritma yang efisien: beberapa algoritma, seperti algoritma divide and conquer, merge sort, dan quicksort, dapat diimplementasikan secara efisien menggunakan rekursif.

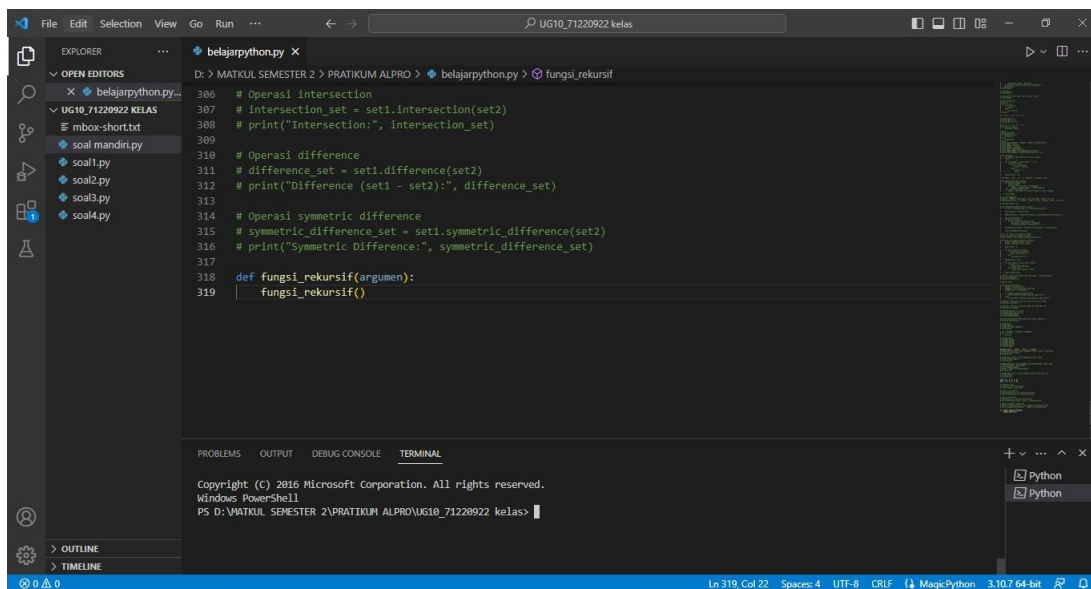
Kelemahan:

1. Kinerja yang kurang efisien: rekursif dapat menyebabkan overhead tambahan karena setiap panggilan rekursif membutuhkan alokasi memori tambahan, dan pengelolaan call stack.
2. Risiko stack overflow: jika rekursif tidak terbatas atau tidak berhenti pada waktu yang tepat, stack overflow dapat terjadi ketika call stack terlalu dalam, dan melebihi batas yang ditentukan.
3. sulit untuk dipahami, dan debug: fungsi rekursif lebih sulit dipahami dibandingkan dengan pendekatan iteratif. Jika tidak diimplementasikan dengan benar, logika rekursif yang rumit dapat menyebabkan kesalahan atau untuk melakukan debugging.
4. penggunaan memori yang lebih tinggi: setiap panggilan rekursif menambahkan overhead memori dengan menyimpan konteks, dan variabel lokal pada setiap level rekursif.

## MATERI 3

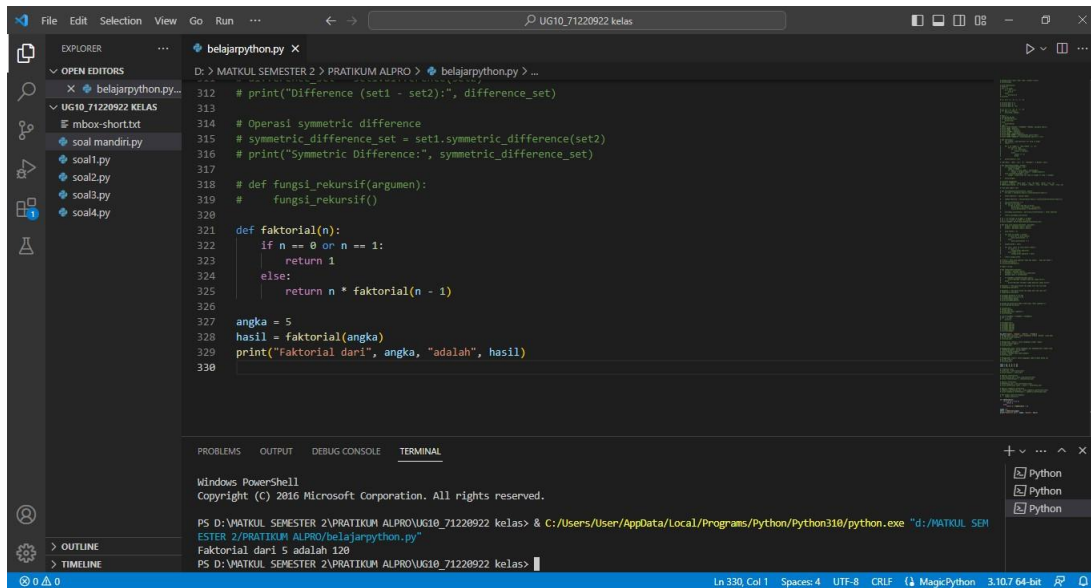
### Bentuk Umum dan Studi Kasus

Bentuk umum fungsi rekursif pada python:



```
File Edit Selection View Go Run ... UG10_71220922 kelas
EXPLORER
  OPEN EDITORS
    belajarpython.py
  UG10_71220922 KELAS
    mbox-short.txt
    soal_mandiri.py
    soal1.py
    soal2.py
    soal3.py
    soal4.py
  belajarpython.py
    306 # Operasi Intersection
    307 # intersection_set = set1.intersection(set2)
    308 # print("Intersection:", intersection_set)
    309
    310 # Operasi difference
    311 # difference_set = set1.difference(set2)
    312 # print("Difference (set1 - set2):", difference_set)
    313
    314 # Operasi symmetric difference
    315 # symmetric_difference_set = set1.symmetric_difference(set2)
    316 # print("Symmetric Difference:", symmetric_difference_set)
    317
    318 def fungsi_rekursif(argumen):
    319     fungsi_rekursif()
  PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
  Copyright (C) 2016 Microsoft Corporation. All rights reserved.
  Windows PowerShell
  PS D:\MATKUL SEMESTER 2\PRATIKUM ALPRO\UG10_71220922 kelas>
  Ln 319, Col 22 Spaces: 4 UTF-8 CR/LF MagicPython 3.10.7 64-bit
```

Berikut adalah contoh penggunaan rekursif menghitung perkalian deret angka:



```
File Edit Selection View Go Run ...
UG10_71220922 kelas

EXPLORER
  OPEN EDITORS
    belajarpython.py...
  UG10_71220922 KELAS
    mbox-short.txt
    soal_mandiri.py
    soal1.py
    soal2.py
    soal3.py
    soal4.py

belajarpython.py
312 # print("Difference (set1 - set2):", difference_set)
313
314 # Operasi symmetric difference
315 # symmetric_difference_set = set1.symmetric_difference(set2)
316 # print("Symmetric Difference:", symmetric_difference_set)
317
318 # def fungsi_rekursif(argumen):
319 #     fungsi_rekursif()
320
321 def faktorial(n):
322     if n == 0 or n == 1:
323         return 1
324     else:
325         return n * faktorial(n - 1)
326
327 angka = 5
328 hasil = faktorial(angka)
329 print("Faktorial dari", angka, "adalah", hasil)
330

TERMINAL
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS D:\MATKUL SEMESTER 2\PRATIKUM ALPRO\UG10_71220922 kelas> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/MATKUL SEMESTER 2/PRATIKUM ALPRO/belajarpython.py"
Faktorial dari 5 adalah 120
PS D:\MATKUL SEMESTER 2\PRATIKUM ALPRO\UG10_71220922 kelas>
```

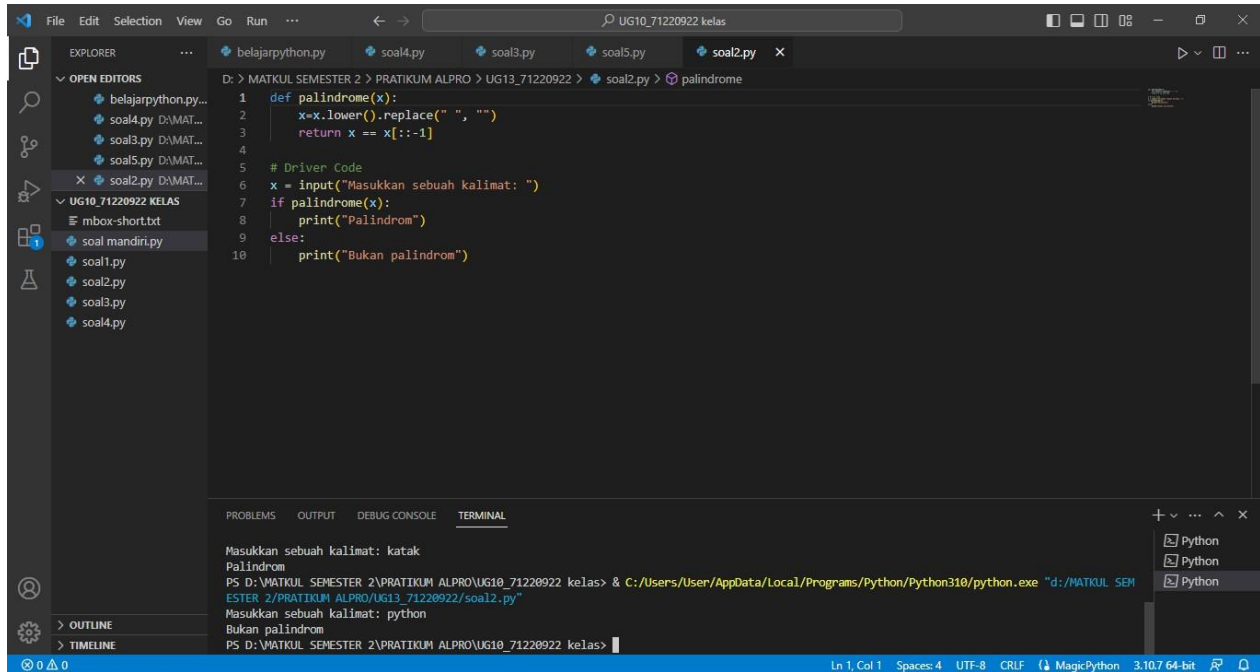
Program di atas fungsi faktorial() merupakan fungsi rekursid yang menghitung faktorial dari bilangan n. pada basis kasus,yaitu ketika n adalah 0 atau 1, fungsi mengembalikan nilai 1 karena faktorial dari 0, dan 1 adalah 1. Lalu pada langkah rekursif, jika n lebih besar dari 1, fungsi rekursif memanggil dirinya sendiri dengan argumen n-1, kemudian mengkalikan n dengan hasil rekursif tersebut. berarti fungsi akan mengulangi proses perkalian n dengan faktorial dari n-1, dan berlanjut hingga mencapai basis kasus ketika n menjadi 0, atau 1.

## BAGIAN 2: LATIHAN MANDIRI

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

### SOAL 1

### SOAL 2



```
1 def palindrome(x):
2     x=x.lower().replace(" ", "")
3     return x == x[::-1]
4
5 # Driver Code
6 x = input("Masukkan sebuah kalimat: ")
7 if palindrome(x):
8     print("Palindrome")
9 else:
10    print("Bukan palindrome")
```

Terminal Output:

```
Masukkan sebuah kalimat: katak
Palindrome
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/UG10_71220922/soal2.py"
Masukkan sebuah kalimat: python
Bukan palindrome
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
```

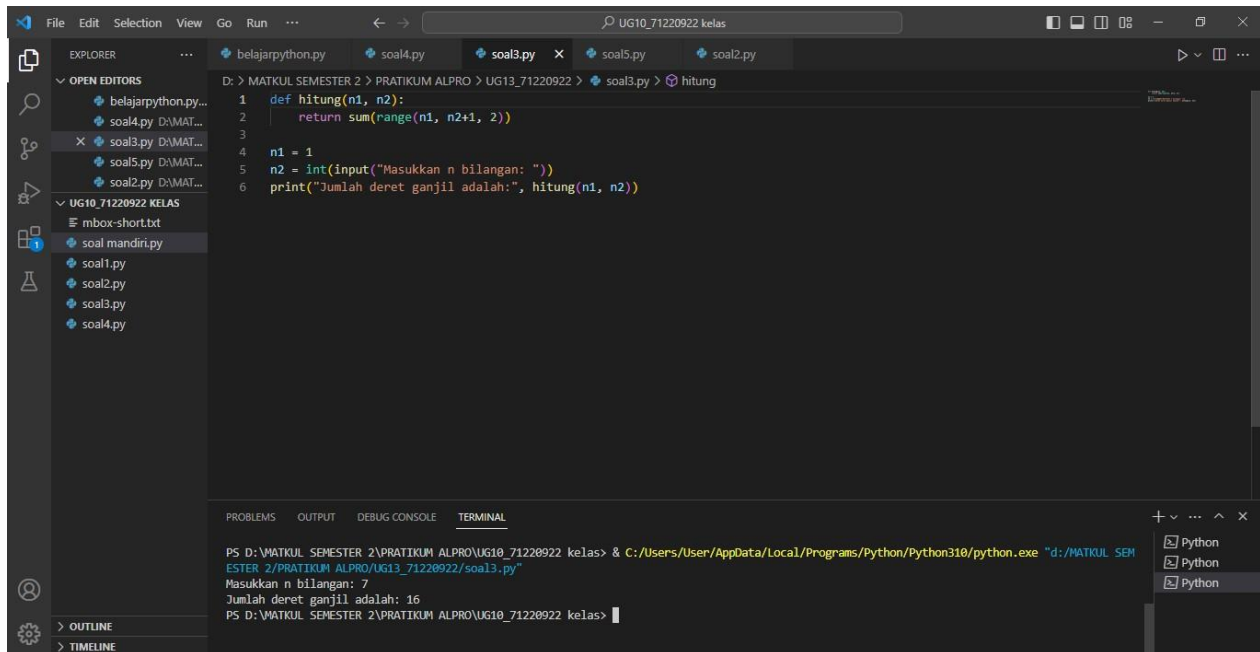
Program di atas adalah sebuah program Python yang menguji apakah sebuah kalimat atau frase merupakan palindrom atau bukan. Berikut adalah langkah-langkah program tersebut:

Program dimulai dengan mendefinisikan sebuah fungsi bernama `palindrome(x)`. Fungsi ini menerima satu parameter yaitu `x`, yang merupakan kalimat atau frase yang akan diperiksa. Di dalam fungsi `palindrome()`, langkah pertama yang dilakukan adalah mengubah `x` menjadi huruf kecil menggunakan metode `lower()`. Hal ini dilakukan untuk memastikan bahwa perbandingan berjalan secara case-insensitive (tidak memperhatikan perbedaan huruf besar dan kecil). Kemudian, menggunakan metode `replace(" ", "")`, semua spasi dalam kalimat dihapus. Hal ini dilakukan agar pengujian palindrom tidak dipengaruhi oleh adanya spasi.

Selanjutnya, fungsi melakukan perbandingan antara `x` dengan `x[::-1]`. Ekspresi `x[::-1]` digunakan untuk membalikkan urutan karakter dalam `x`. Jika `x` sama dengan `x[::-1]`, artinya kalimat tersebut palindrom, dan fungsi mengembalikan nilai `True`. Jika `x` tidak sama dengan `x[::-1]`, artinya kalimat tersebut bukan palindrom, dan fungsi mengembalikan nilai `False`. Setelah itu, program meminta pengguna untuk memasukkan sebuah kalimat dengan menggunakan fungsi `input()`. Kalimat tersebut disimpan dalam variabel `x`. Program kemudian memanggil fungsi `palindrome(x)` untuk memeriksa apakah kalimat tersebut palindrom atau bukan. Berdasarkan hasil yang dikembalikan oleh fungsi

`palindrome(x)`, program mencetak pesan "Palindrom" jika hasilnya True, atau mencetak pesan "Bukan palindrom" jika hasilnya False.

### SOAL 3



The screenshot shows a Python IDE with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'UG10\_71220922 KELAS' with several files including 'soal1.py', 'soal2.py', 'soal3.py', and 'soal4.py'. The code editor displays the following Python code:

```
1 def hitung(n1, n2):
2     return sum(range(n1, n2+1, 2))
3
4 n1 = 1
5 n2 = int(input("Masukkan n bilangan: "))
6 print("Jumlah deret ganjil adalah:", hitung(n1, n2))
```

The terminal at the bottom shows the execution of the program:

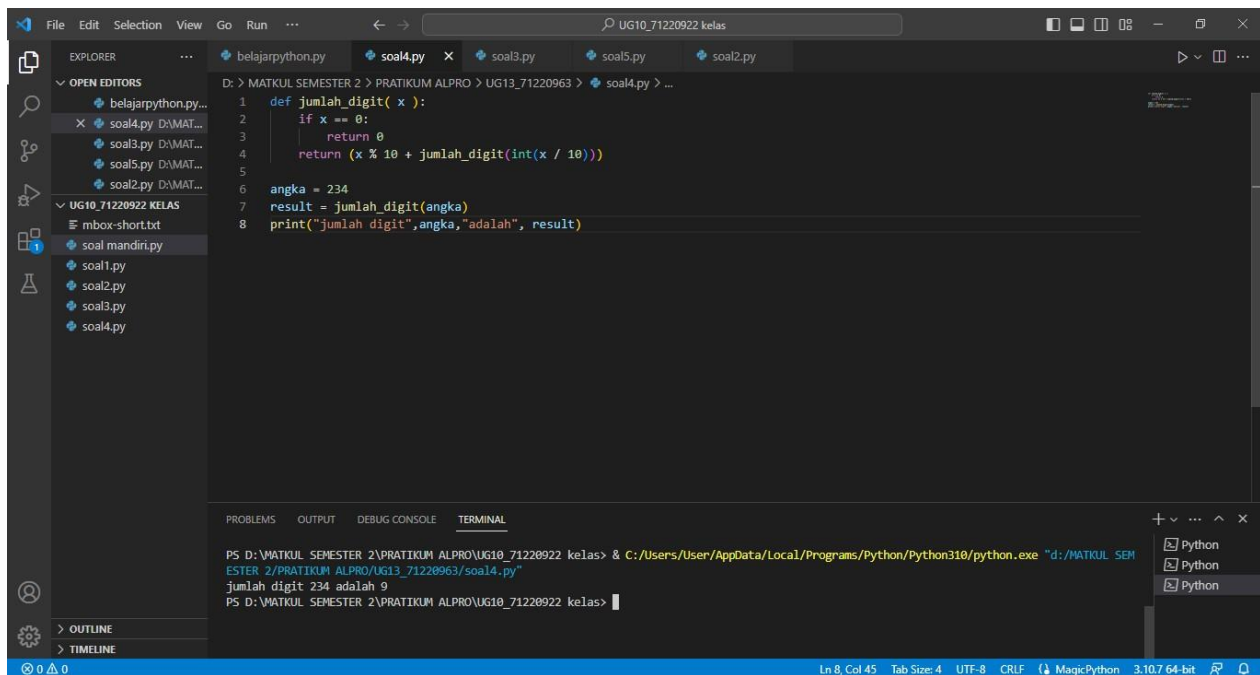
```
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/UG10_71220922/soal3.py"
Masukkan n bilangan: 7
Jumlah deret ganjil adalah: 16
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
```

Program di atas adalah sebuah program Python yang menghitung jumlah deret bilangan ganjil dalam rentang n1 hingga n2. Berikut adalah langkah-langkah program tersebut:

Program dimulai dengan mendefinisikan sebuah fungsi bernama `hitung(n1, n2)`. Fungsi ini menerima dua parameter, yaitu `n1` dan `n2`, yang merupakan batas awal dan batas akhir rentang bilangan. Di dalam fungsi `hitung()`, menggunakan fungsi `range(n1, n2+1, 2)`, dibentuk sebuah range dari n1 hingga n2 dengan langkah 2. Hal ini dilakukan untuk memperoleh semua bilangan ganjil dalam rentang tersebut. Kemudian, menggunakan fungsi `sum()`, dilakukan penjumlahan semua bilangan dalam range tersebut. Hasil penjumlahan tersebut dikembalikan sebagai nilai hasil fungsi.

Setelah itu, program meminta pengguna untuk memasukkan sebuah bilangan dengan menggunakan fungsi `input()`. Bilangan tersebut diubah menjadi integer menggunakan `int()` dan disimpan dalam variabel `n2`. Variabel `n1` diinisialisasi dengan nilai 1. Program kemudian memanggil fungsi `hitung(n1, n2)` untuk menghitung jumlah deret bilangan ganjil dalam rentang n1 hingga n2. Hasil penjumlahan tersebut dicetak dengan pesan "Jumlah deret ganjil adalah:", menggunakan fungsi `print()`.

## SOAL 4



The screenshot shows a Python IDE with the following code in `soal4.py`:

```
1 def jumlah_digit( x ):
2     if x == 0:
3         return 0
4     return (x % 10 + jumlah_digit(int(x / 10)))
5
6 angka = 234
7 result = jumlah_digit(angka)
8 print("jumlah digit",angka,"adalah", result)
```

The terminal output shows the execution of the program:

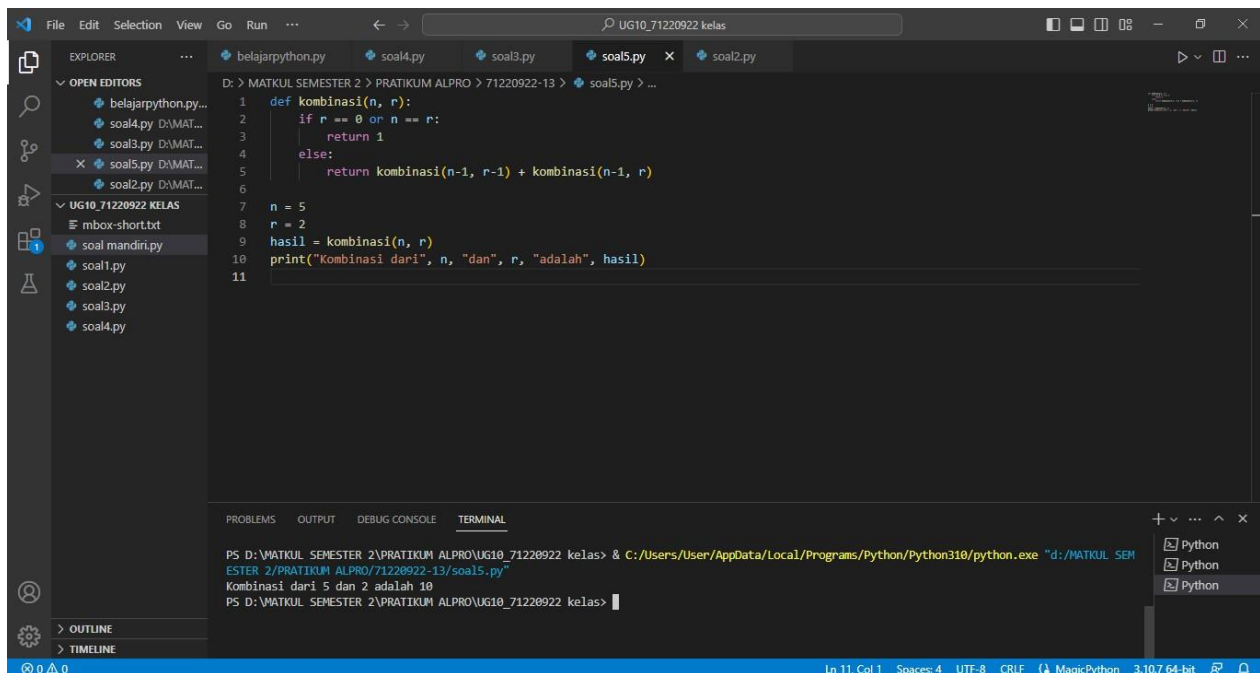
```
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/UG10_71220922/soal4.py"
jumlah digit 234 adalah 9
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
```

Program di atas adalah sebuah program Python yang menghitung jumlah digit dalam suatu bilangan. Berikut adalah langkah-langkah program tersebut:

Program dimulai dengan mendefinisikan sebuah fungsi bernama `jumlah_digit(x)`. Fungsi ini menerima satu parameter `x`, yang merupakan bilangan yang akan dihitung jumlah digitnya. Di dalam fungsi `jumlah_digit()`, terdapat kondisi basis yaitu jika `x` sama dengan 0, fungsi mengembalikan nilai 0. Hal ini menandakan bahwa bilangan telah habis diproses dan tidak memiliki digit lagi. Pada langkah rekursif, fungsi membagi bilangan `x` dengan 10 menggunakan operasi `x / 10` untuk menghapus digit paling kanan dari `x`. Kemudian, menggunakan operasi `x % 10`, fungsi mengambil digit paling kanan yang telah dihapus sebelumnya.

Fungsi mengembalikan hasil penjumlahan antara digit paling kanan dengan hasil rekursif dari bilangan yang telah dikurangi satu digit. Setelah itu, program menginisialisasi variabel `angka` dengan nilai 234. Program memanggil fungsi `jumlah_digit(angka)` untuk menghitung jumlah digit dalam bilangan `angka`. Hasil jumlah digit tersebut disimpan dalam variabel `result`. Terakhir, program mencetak hasil dengan pesan "jumlah digit {angka} adalah {result}" menggunakan fungsi `print()`.

## SOAL 5



The screenshot shows a Python IDE with a dark theme. The Explorer panel on the left shows a project named 'UG10\_71220922 KELAS' with several files including 'soal1.py' through 'soal5.py'. The main editor window displays the code for 'soal5.py'. The code defines a recursive function 'kombinasi(n, r)' that calculates the number of combinations. It includes a base case where if 'r' is 0 or 'n' is 'r', it returns 1. Otherwise, it returns the sum of 'kombinasi(n-1, r-1)' and 'kombinasi(n-1, r)'. Below the function definition, the variables 'n' and 'r' are set to 5 and 2 respectively, the function is called, and the result is printed. The terminal at the bottom shows the command to run the script and the output 'Kombinasi dari 5 dan 2 adalah 10'.

```
1 def kombinasi(n, r):
2     if r == 0 or n == r:
3         return 1
4     else:
5         return kombinasi(n-1, r-1) + kombinasi(n-1, r)
6
7 n = 5
8 r = 2
9 hasil = kombinasi(n, r)
10 print("Kombinasi dari", n, "dan", r, "adalah", hasil)
11
```

```
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/MATKUL SEMESTER 2/PRATIUM ALPRO/71220922-13/soal5.py"
Kombinasi dari 5 dan 2 adalah 10
PS D:\MATKUL SEMESTER 2\PRATIUM ALPRO\UG10_71220922 kelas>
```

Program di atas adalah sebuah program Python yang menghitung kombinasi dari dua bilangan  $n$  dan  $r$ . Berikut adalah langkah-langkah program tersebut:

Program dimulai dengan mendefinisikan sebuah fungsi bernama `kombinasi(n, r)`. Fungsi ini menerima dua parameter, yaitu  $n$  dan  $r$ , yang merupakan bilangan yang akan digunakan dalam perhitungan kombinasi. Di dalam fungsi `kombinasi()`, terdapat kondisi basis yaitu jika  $r$  sama dengan 0 atau  $n$  sama dengan  $r$ , fungsi mengembalikan nilai 1. Hal ini menandakan bahwa dalam kasus tersebut tidak ada pilihan yang harus dibuat, sehingga hasil kombinasi adalah 1. Pada langkah rekursif, fungsi memanggil dirinya sendiri dua kali dengan argumen yang berbeda. Pertama, dengan argumen  $(n-1, r-1)$ , yang berarti mengurangi satu angka dari  $n$  dan  $r$  untuk mempertahankan relasi kombinasi. Kedua, dengan argumen  $(n-1, r)$ , yang berarti mengurangi satu angka dari  $n$  tetapi  $r$  tetap sama.

Fungsi menggabungkan kedua hasil rekursif tersebut dengan operasi penjumlahan. Hasil perhitungan kombinasi dikembalikan sebagai nilai hasil fungsi. Setelah itu, program menginisialisasi variabel  $n$  dengan nilai 5 dan  $r$  dengan nilai 2. Program memanggil fungsi `kombinasi(n, r)` untuk menghitung kombinasi dari  $n$  dan  $r$ . Hasil perhitungan kombinasi disimpan dalam variabel `hasil`. Terakhir, program mencetak hasil dengan pesan "Kombinasi dari {n} dan {r} adalah {hasil}" menggunakan fungsi `print()`.



