

<p>Nama: Dimas Alip Priyono</p> <p>NIM: 064102400032</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<h1>MODUL 7</h1> <p>Nama Dosen: Binti solihah, S.T, M.KOM</p> <p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none">Yustianas Rombon - 064002300015Vira Aditya Kurniawan - 065002300012
<p>Hari/Tanggal: Hari, Tanggal Bulan 2024</p>		

Latihan Fungsi (Function)

1. Teori Singkat

Fungsi

Fungsi merupakan suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilnya. Pada pembuatan program yang kompleks dan memiliki banyak fitur, kita diharuskan menggunakan fungsi. Fungsi diperlukan agar mempermudah kita dalam membaca sebuah kode program dan mempermudah untuk merawatnya. Dengan adanya sebuah fungsi kita tidak perlu menulis kode sepanjang gerbang kereta api di program utama dan kita dapat memecah atau membaginya lalu tinggal memanggil fungsinya saja didalam program utama. Dan itu juga akan mempersingkat penulisan dari program utama tersebut nantinya

Fungsi memiliki parameter yang dapat dipergunakan untuk memasukkan atau menampung variabel kedalam sebuah fungsi. Dalam pendeklarasian parameter pada sebuah fungsi kita juga dapat memasukkan *default argument/parameter*.

Fungsi juga dapat mengembalikan nilai dengan cara menggunakan keyword *return* yang di taruh didalam blok program fungsi itu sendiri untuk mengembalikan nilai yang ingin kita kembalikan kedalam program utama.



Membuat Fungsi pada Python

Fungsi pada Python, dibuat dengan kata kunci *def* kemudian diikuti dengan nama fungsinya.

```
# Pendeklarasian sebuah fungsi
def nama_fungsi():
    print("Hello Ini Fungsi")

# Pemanggilan sebuah fungsi
nama_fungsi()
```

Fungsi dengan parameter pada Python

Memasukkan parameter kedalam sebuah fungsi



```
# pendeklarasian fungsi
def luas_persegi(sisi):
    luas = sisi * sisi
    return luas

# pemanggilan fungsi
print("Luas persegi: ", luas_persegi(6))
```

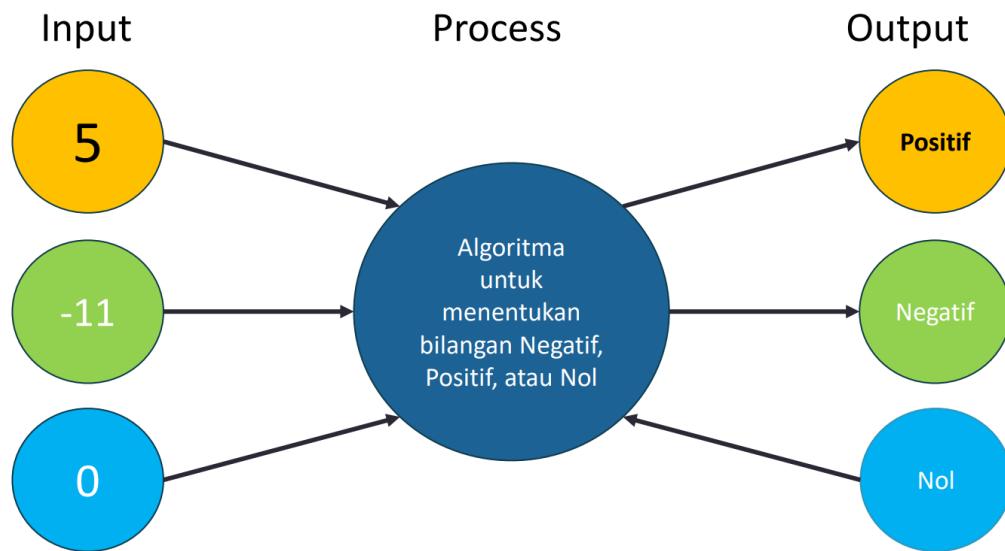
IPO (Input Process Output)

Konsep Dasar Input, Process, dan Output (IPO)

- Konsep input, process, dan output adalah prinsip dasar dalam pemrograman dan pengembangan algoritma.
- Setiap algoritma melibatkan tiga tahap utama: mengambil data masukan (input), melakukan operasi atau pengolahan data (process), dan menghasilkan hasil akhir (output).
- Konsep ini menggambarkan bagaimana algoritma beroperasi untuk memproses informasi.



Gambaran IPO (Menentukan Bilangan)



Notasi Algoritma Flowchart

1. Flowchart adalah representasi visual atau diagram alir yang digunakan untuk menggambarkan langkah-langkah dan urutan proses suatu algoritma atau program.
2. Flowchart menyajikan langkah-langkah dalam bentuk simbol-simbol grafis yang saling terhubung, membantu dalam memvisualisasikan bagaimana informasi mengalir dan bagaimana proses dilakukan.
3. Dalam kaitannya dengan notasi deskriptif, notasi algoritma yang menggunakan flowchart dapat lebih cepat dibaca dan dilihat alur dan hubungannya.

Simbol-simbol pada Flowchart

1. Setiap elemen flowchart dihubungkan oleh garis aliran bertanda panah
2. Garis aliran dimulai dari atas symbol dan keluar dari bagian bawah, kecuali symbol keputusan yang alirannya keluar dari bawah atau samping
3. Aliran bergerak dari atas ke bawah
4. Proses awal dan akhir menggunakan symbol terminal.



... Simbol-simbol pada Flowchart



Terminator yang menandakan *Start* (awal) atau *End* (akhir) program.



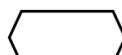
Flow line yang digunakan untuk menunjukkan arah aliran pada program.



Process menunjukkan proses yang dilakukan pada masukan.



Input atau output untuk menunjukkan masukan dan keluaran.



Preparation digunakan untuk membuat deklarasi nilai awal.



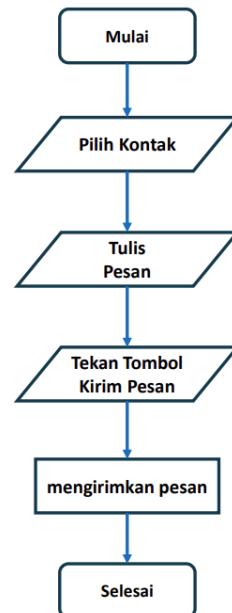
On Page Connector digunakan untuk menghubungkan antar *flowchart*.



Decision menunjukkan keputusan atau kondisi untuk memilih keputusan.

**Contoh sederhana
Penggunaan *flowchart*
untuk menunjukkan algoritma**

**Kasus/Aliran:
Mengirim pesan WhatsApp**



2. Alat dan Bahan

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

3. Elemen Kompetensi

a. Latihan pertama

Buatlah sebuah program yang bisa menentukan apakah itu bilangan prima atau bukan.

Gunakan 2 fungsi beserta implementasikan parameternya.

IPO (Input Process Output)

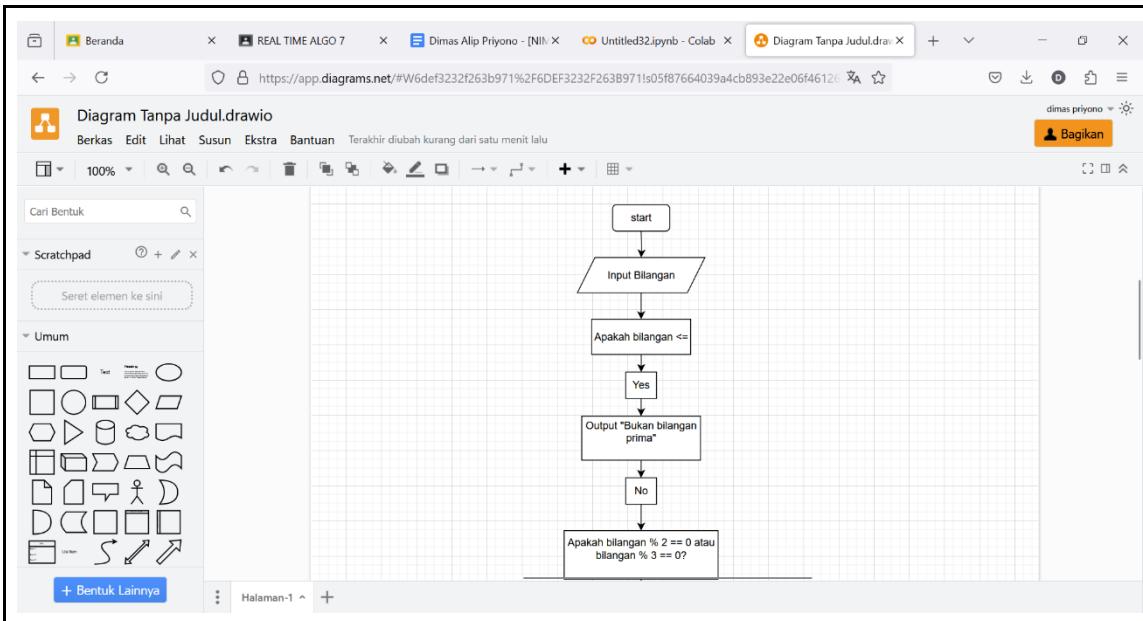


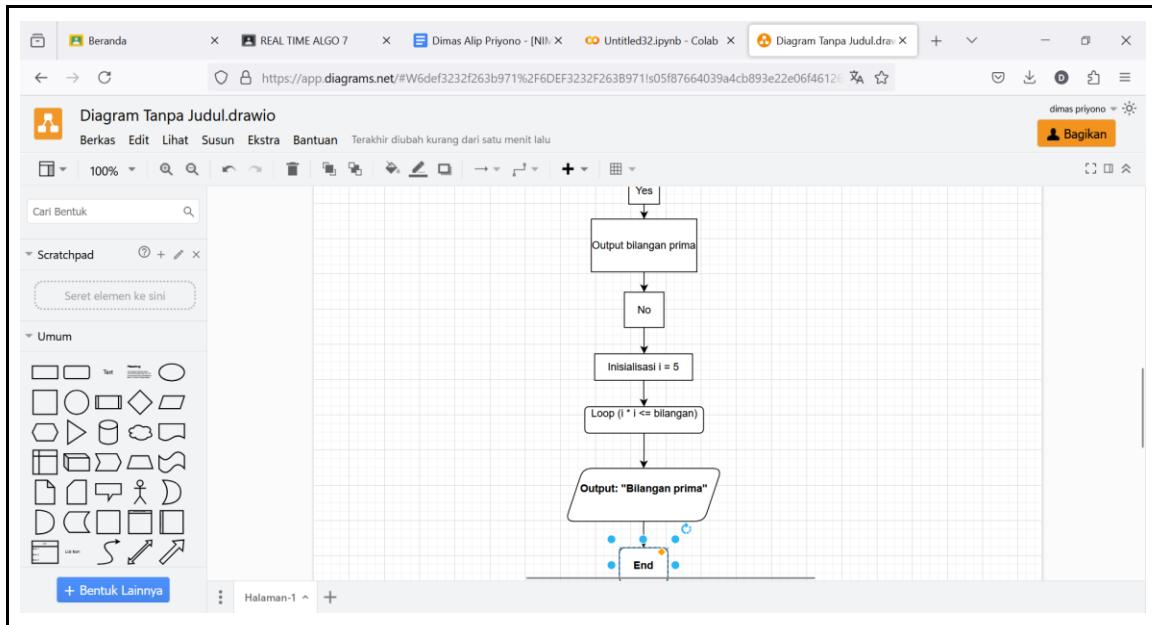
```
def bilangan_prima(bilangan):
    if bilangan <= 0:
        return False
    if bilangan <= 1:
        return True
    if bilangan <= 3:
        return True
    if bilangan % 2 == 0 or bilangan % 3 == 0:
        return False
    i = bilangan
    while i * i <= bilangan:
        if bilangan % i == 0 or bilangan % (i + 2) == 0:
            return False

    return True
bilangan = int(input("Masukkan bilangan: "))
if bilangan_prima(bilangan):
    print(f"{bilangan} adalah bilangan prima.")
else:
    print(f"{bilangan} bukan bilangan prima.")

Masukkan bilangan: 7
7 adalah bilangan prima.
```

Flowchart





Source Code

```
def bilangan_prima(bilangan):
    if bilangan <= 0:
        return False
    if bilangan <= 1:
        return True
    if bilangan <= 3:
        return True
    if bilangan % 2 == 0 or bilangan % 3 == 0:
        return False
    i = bilangan
    while i * i <= bilangan:
        if bilangan % i == 0 or bilangan % (i + 2) == 0:
            return False

    return True
bilangan = int(input("Masukkan bilangan: "))
if bilangan_prima(bilangan):
    print(f"{bilangan} adalah bilangan prima.")
else:
    print(f"{bilangan} bukan bilangan prima.")
```



Output

```
Masukkan bilangan: 7
7 adalah bilangan prima.
```

b. Latihan Kedua

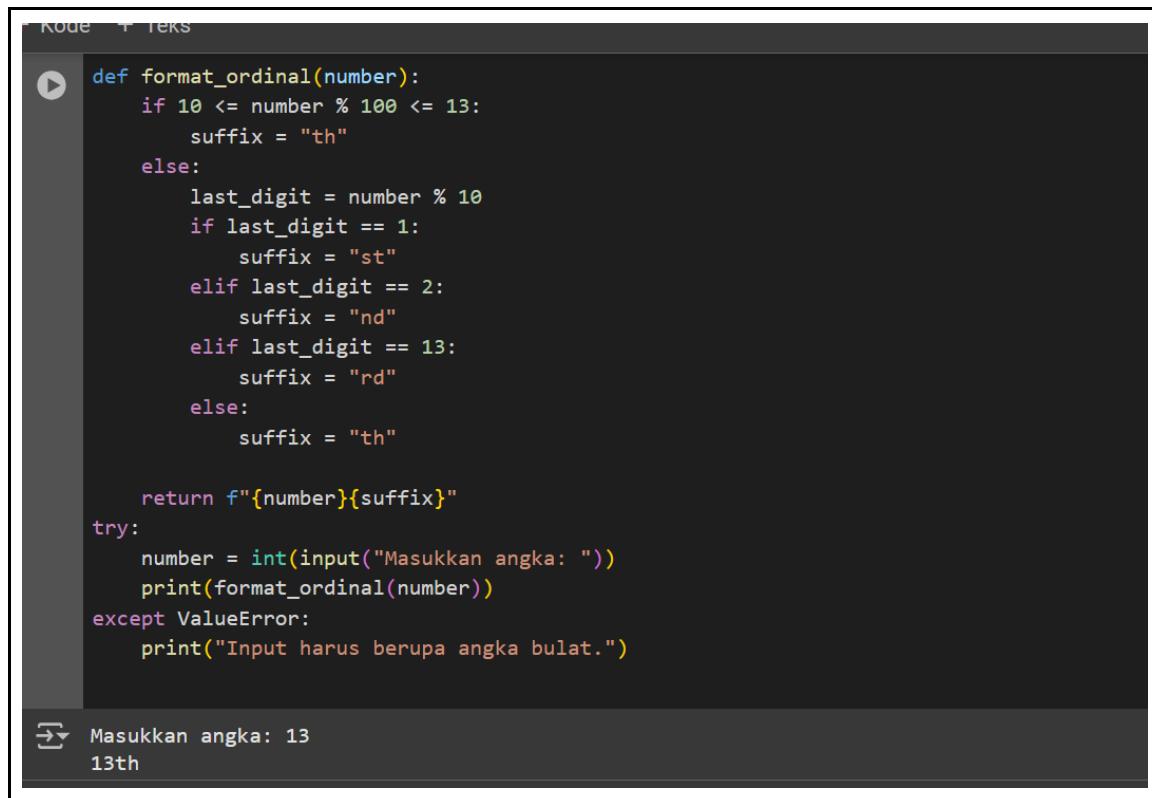
Buatlah sebuah program yang bisa menampilkan output seperti berikut ini:

```
>>1 //Input
>>1st //output
>>2 //input
>>2nd //output.
>>13 //input
>>13rd //output
```

Dst. Menggunakan implementasi fungsi dengan parameter.

IPO (Input Process Output)





```

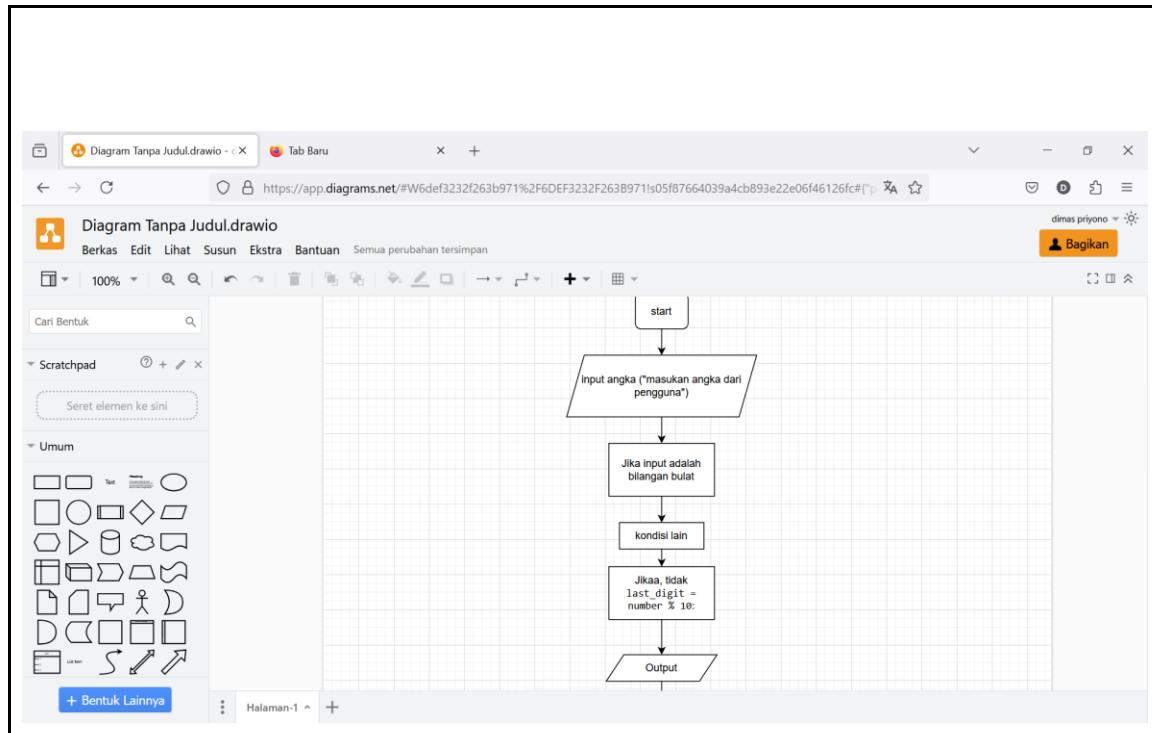
def format_ordinal(number):
    if 10 <= number % 100 <= 13:
        suffix = "th"
    else:
        last_digit = number % 10
        if last_digit == 1:
            suffix = "st"
        elif last_digit == 2:
            suffix = "nd"
        elif last_digit == 3:
            suffix = "rd"
        else:
            suffix = "th"

    return f"{number}{suffix}"
try:
    number = int(input("Masukkan angka: "))
    print(format_ordinal(number))
except ValueError:
    print("Input harus berupa angka bulat.")

```

Masukkan angka: 13
13th

Flowchart



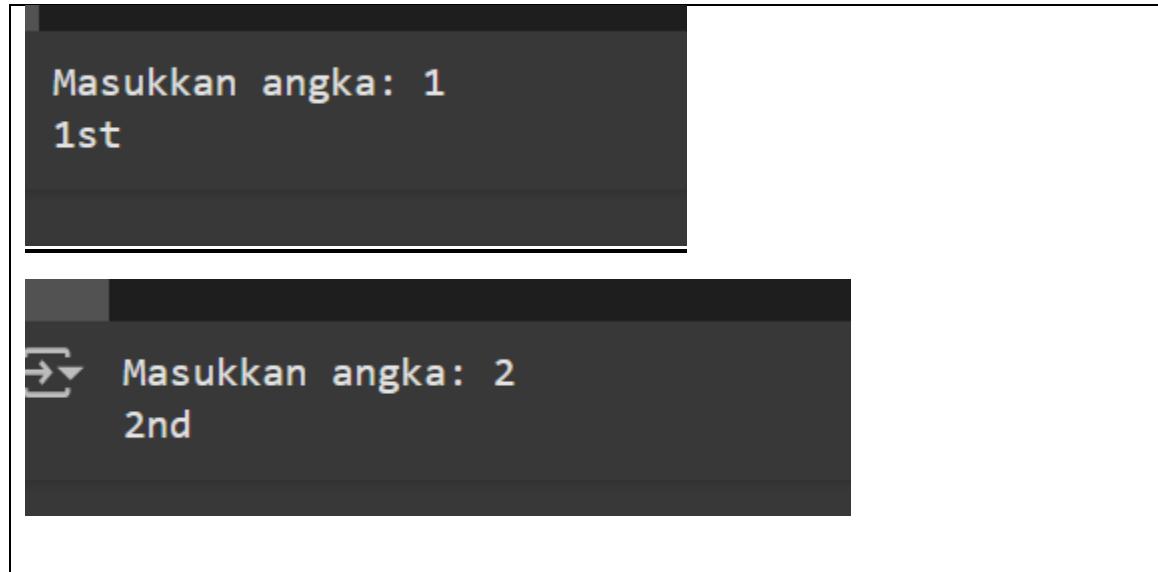
Source Code

```
def format_ordinal(number):
    if 10 <= number % 100 <= 13:
        suffix = "th"
    else:
        last_digit = number % 10
        if last_digit == 1:
            suffix = "st"
        elif last_digit == 2:
            suffix = "nd"
        elif last_digit == 3:
            suffix = "rd"
        else:
            suffix = "th"

    return f"{number}{suffix}"

try:
    number = int(input("Masukkan angka: "))
    print(format_ordinal(number))
except ValueError:
    print("Input harus berupa angka bulat.")
```

Output



```
Masukkan angka: 1
1st

→ Masukkan angka: 2
2nd
```



4. File Praktikum

Github Repository:

<https://github.com/Alip1023/Prak7Algoo.git>

5. Soal Latihan

Soal:

1. Jika sebuah deklarasi fungsi memiliki parameter a “*def nama_fungsi(a):*”, tetapi ketika dipanggil pada program utama kita mengisi fungsi tersebut dengan nama variabel b “*nama_fungsi(b)*” berbeda dengan nama parameter yang sebelumnya dideklarasikan, apakah program tetap akan berjalan dengan sesuai? dan jika sesuai mengapa demikian dan jika tidak mengapa demikian?
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!

Jawaban:

1. Ya, program tetap akan berjalan sesuai. Parameter dalam deklarasi fungsi hanya sebagai penampung sementara nilai yang diberikan saat pemanggilan fungsi. Jadi, meskipun nama variabel berbeda, nilai variabel tetap akan diteruskan dengan benar ke dalam fungsi.
2. Program ini bertujuan untuk menambahkan akhiran ordinal ("st", "nd", "rd", "th") pada angka yang diinputkan pengguna dan menampilkannya. Program menggunakan fungsi dengan parameter untuk mengelola input pengguna dan menghasilkan output yang sesuai

6. Kesimpulan

- a. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- b. Kita dapat mengetahui Program ini menggunakan dua fungsi: satu untuk menentukan apakah angka merupakan bilangan prima (*cek_prima*), dan satu lagi untuk menambahkan akhiran ordinal (*tambah_akhiran_ordinal*). Fungsi pertama mengecek keunikan faktor dari angka, sementara fungsi kedua menambahkan akhiran sesuai aturan ordinal.

7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai



1.	Latihan Pertama	✓	
2.	Latihan Kedua	✓	

8. Formulir Umpam Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	60 Menit	Baik
2.	Latihan Kedua	60 Menit	Baik

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang

