


<p>Nama: Aini Rihhadatul Aisy</p> <p>NIM: 064102400024</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<p>MODUL 7</p> <p>Nama Dosen: Binti solihah, S.T, M.KOM</p>
<p>Hari/Tanggal: Jumat, 08 November 2024</p>		<p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none"> 1. Yustianas Rombon - 064002300015 2. Vira Aditya Kurniawan - 065002300012

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 memanggilya. 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 gerbong 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))
```

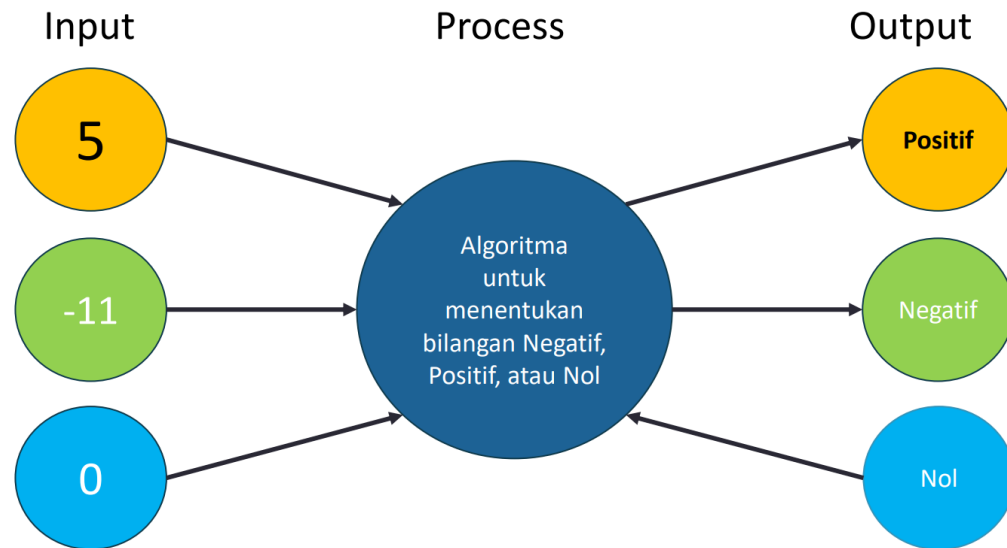
IPO (Input Process Output)

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

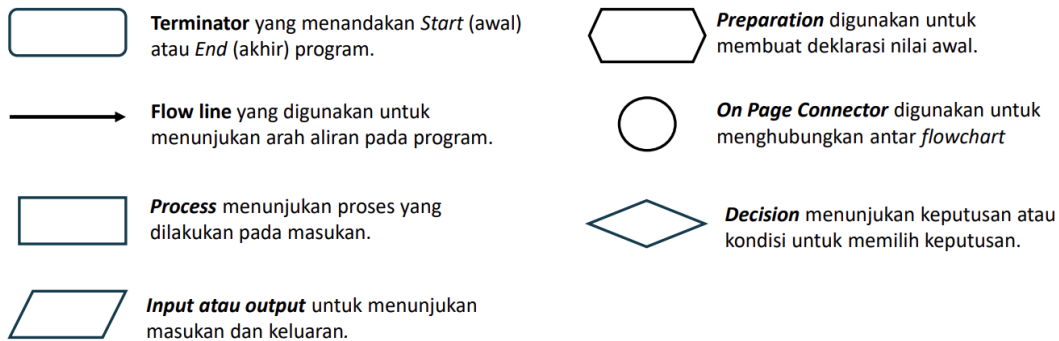
1. Flowchart adalah representasi visual atau diagram alir yang digunakan untuk menggambarkan langkahlangkah dan urutan proses suatu algoritma atau program.
2. Flowchart menyajikan langkah-langkah dalam bentuk simbol-simbol grafis yang saling terhubung, membantu dalam memvisualisasikan bagaimana informasi 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.

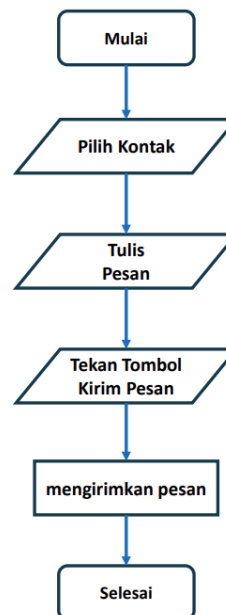


... Simbol-simbol pada Flowchart



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)



Input:

- Pengguna memasukkan satu bilangan integer. Dengan Tipe data Integer

Proses:

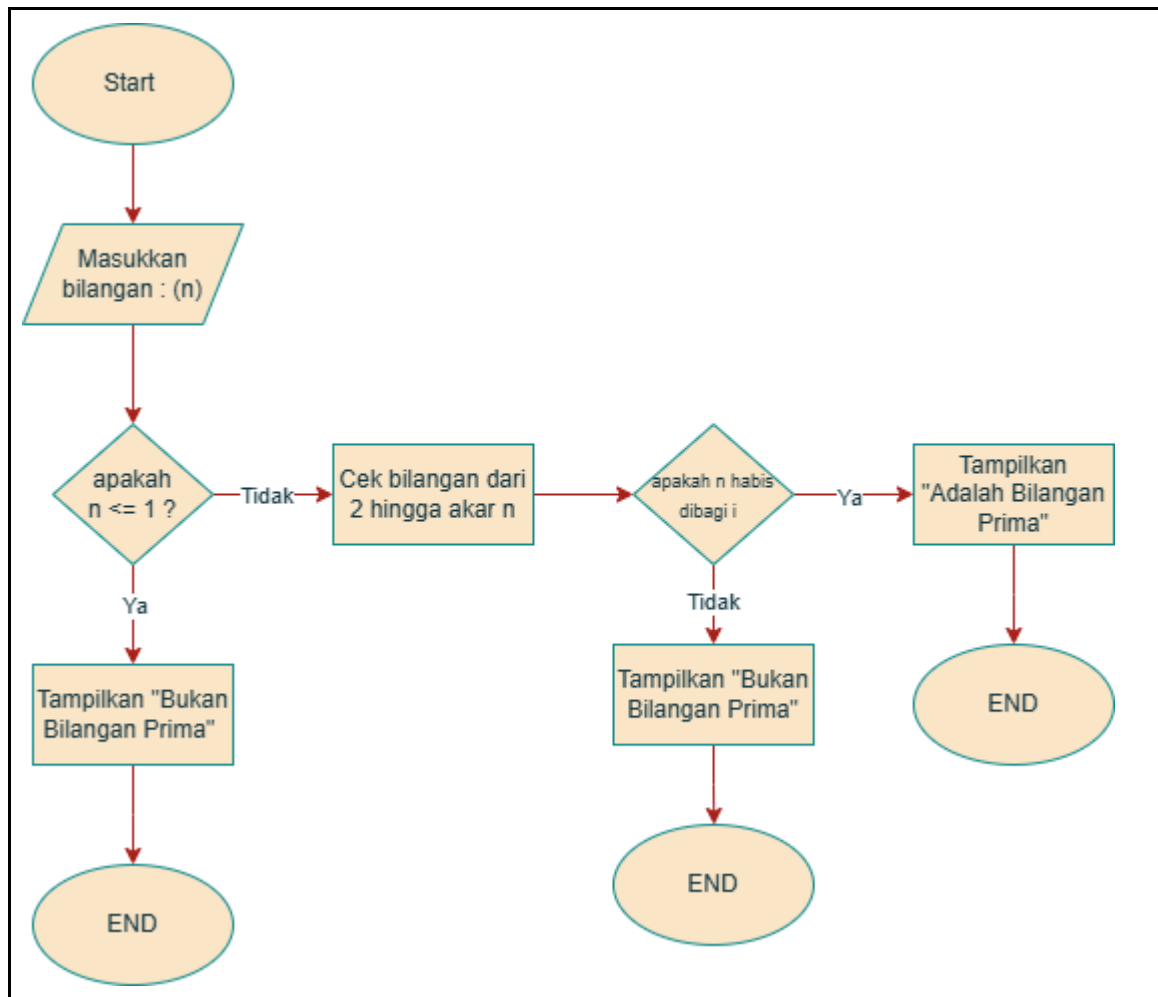
- Fungsi cek_prima(n):
 - Jika $n \leq 1$, kembalikan **False**.
 - Periksa apakah ada pembagi n dari 2 hingga akar kuadrat dari n. Jika ada, kembalikan **False** (bukan bilangan prima).
 - Jika tidak ada pembagi, kembalikan **True** (bilangan prima).
- Fungsi tampilkan_hasil(n):
 - Panggil cek_prima(n).
 - Jika hasilnya **True**, tampilkan bahwa n adalah bilangan prima.
 - Jika hasilnya **False**, tampilkan bahwa n bukan bilangan prima.

Output:

- Menampilkan pesan yang memberitahukan apakah bilangan tersebut prima atau bukan. Dengan tipe data string
 - Jika input 5, output: "5 adalah bilangan prima".
 - Jika input 10, output: "10 bukan bilangan prima".

Flowchart





Source Code



```

6s def cek_prima(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False # bukan bilangan prima
    return True # adalah bilangan prima

def tampilkan_hasil(n):
    if cek_prima(n):
        print(f"{n} adalah bilangan prima")
    else:
        print(f"{n} bukan bilangan prima")

n = int(input("Masukkan bilangan: "))
tampilkan_hasil(n)

```

Output

```

➡ Masukkan bilangan: 11
11 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)

Input:

- User memasukkan angka (misalnya 1, 2, 13).

Proses:

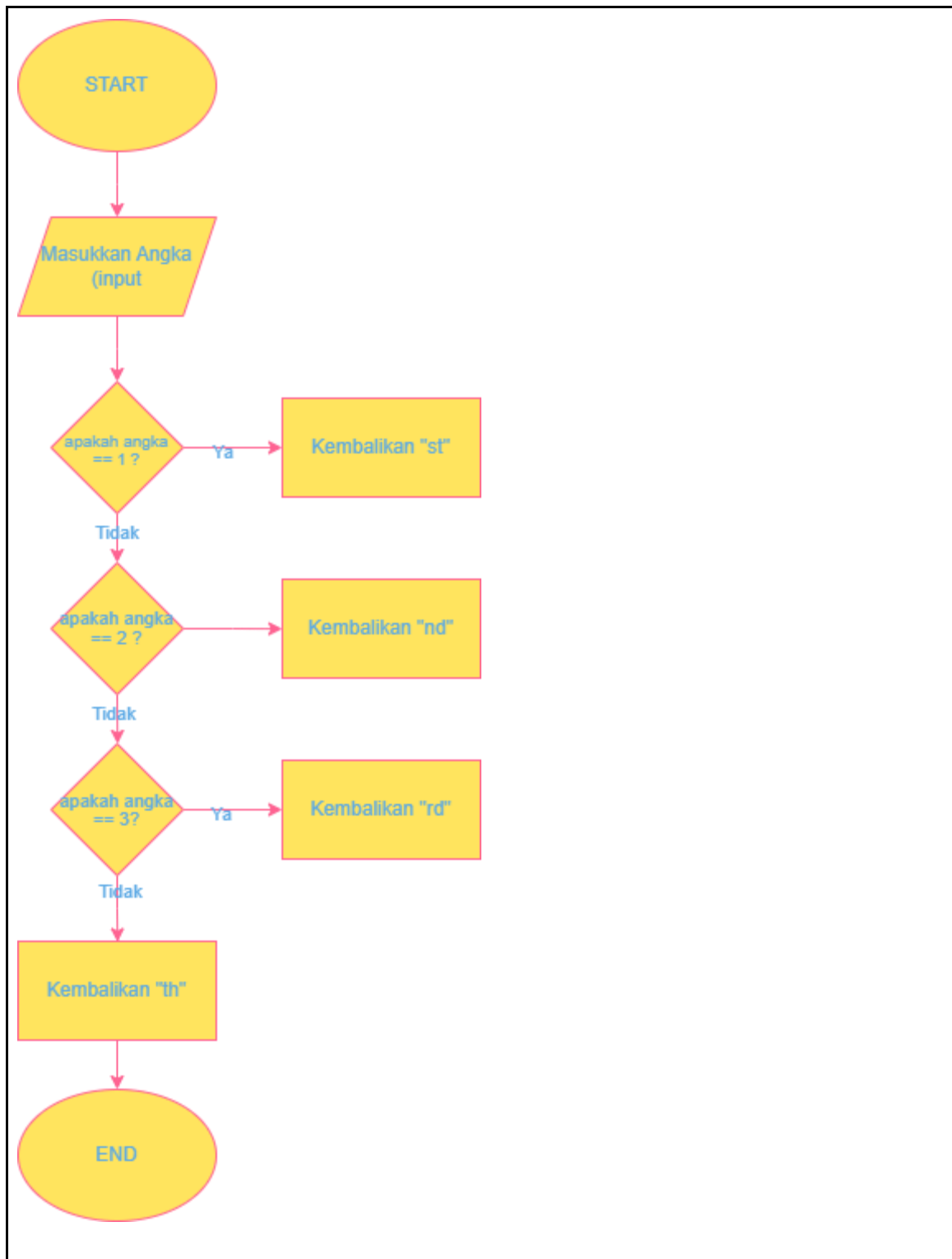
- **Fungsi ordinal(number):**
 - Cek apakah angka adalah 1, 2, atau 3 dan beri akhiran yang sesuai (st, nd, rd).
 - Untuk angka selain 1, 2, atau 3, tambahkan akhiran th.

Output:

- Menampilkan angka yang diikuti dengan akhiran ordinal yang sesuai. Dengan tipe Data String
- **Contoh Output:**
 - Jika input 1, output: "1st".
 - Jika input 2, output: "2nd".
 - Jika input 13, output: "13th".

Flowchart





Source Code



```

def ordinal(number):
    if number == 1:
        return f"{number}st"
    elif number == 2:
        return f"{number}nd"
    elif number == 3:
        return f"{number}rd"
    else:
        return f"{number}th"

# Contoh penggunaan
inputs = [1, 2, 13] # Angka yang ingin diproses
for number in inputs:
    print(f">>{number} //input")
    print(f">>{ordinal(number)} //output.")

```

Output

```

>>1 //input
>>1st //output.
>>2 //input
>>2nd //output.
>>13 //input
>>13th //output.

```

4. File Praktikum

Github Repository:



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 akan tetap berjalan dengan sesuai karena Python menggunakan sistem passing by assignment, yang artinya Python hanya menyalin referensi nilai dari b ke a. Jadi, nama variabel yang digunakan untuk memanggil fungsi (b dalam contoh ini) tidak harus sama dengan nama parameter dalam deklarasi fungsi (a).
2. Program yang dibikin bertujuan untuk mengubah angka menjadi bentuk ordinal yang sesuai dengan aturan penulisan angka dalam bahasa Inggris. Ketika program dijalankan, pertama-tama pengguna diminta untuk memasukkan sebuah angka, yang kemudian diproses oleh fungsi `ordinal(number)`. Fungsi `ordinal(number)` memeriksa nilai dari angka yang diberikan. Jika angka tersebut adalah 1, maka program menambahkan akhiran "st" ke angka tersebut. Begitu juga jika angka yang dimasukkan adalah 2, maka akan ditambahkan akhiran "nd", dan jika angka adalah 3, akhiran yang ditambahkan adalah "rd". Untuk angka selain 1, 2, atau 3, program akan menambahkan akhiran "th" sebagai tanda ordinal. Setelah angka diproses dan akhiran ordinal ditambahkan, program akan menampilkan hasilnya di layar. Dalam contoh penggunaan, jika pengguna memasukkan angka 1, program akan menampilkan "1st", jika angka 2, maka akan tampil "2nd", dan untuk angka 13 akan muncul "13th". Program ini menggunakan loop untuk memproses beberapa angka sekaligus dari list yang telah ditentukan dan menampilkan hasilnya satu per satu.

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, penggunaan fungsi dalam pemrograman mempermudah pengorganisasian kode dan membuatnya lebih efisien. Dengan parameter dalam fungsi, kita bisa menangani berbagai nilai input tanpa perlu menulis ulang kode. Dalam latihan kedua, program berhasil mengubah angka menjadi bentuk ordinal yang sesuai, memanfaatkan fungsi untuk memproses dan menampilkan hasilnya secara dinamis.



7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	✓	
2.	Latihan Kedua	✓	

8. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	30 Menit	Menarik
2.	Latihan Kedua	30 Menit	Menarikk

Keterangan:

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