**LAPORAN PRAKTIKUM**

*Cambria, 20 bold*

**ALGORITMA DAN STRUKTUR DATA**



*4.5 cm x 4.5 cm*



 IK SULASTRI NINGSIH

*Nama Lengkap*

*NIM*

240306030

Github.com IKA SULASTRI NINGSIH 

*Akun Github*

Program Studi Teknologi Informasi

Fakultas Dakwah dan Ilmu Komunikasi

Universitas Islam Negeri Mataram

*Cambria, 16*

2024

*Footnote, Cambria, 10*

|  |  |
| --- | --- |
| Pertemuan | 8 |
| Topik | fungsi |
| Repository |  |
| Tanggal | Rabu,13 November 2024 |

A. Tujuan

Tujuan praktikum fungsi

1. Modularisasi, fungsi memungkinkan pemrograman untuk membagi program menjadi bagian-bagian kecil yang lebih terstruktur. ini membuat program mudah dipahami, di debug, dan diperbaiki karena setiap fungsi bertangung jawab hanya untuk satu tugas.
2. Keteraturan dan Kejelasan Kode: Dengan menggunakan fungsi, kode menjadi lebih rapi dan teratur. Kode yang terpecah menjadi fungsi-fungsi yang berbeda menjadi lebih mudah dibaca dan dipahami.
3. Penggunaan Kembali Kode (Reusability): Fungsi memungkinkan pemrogram menggunakan kembali kode yang sama di berbagai bagian program tanpa perlu menulis ulang. Ini menghemat waktu dan mengurangi potensi kesalahan.
4. Mengurangi Duplikasi: Jika ada kode yang harus dieksekusi berulang kali, menempatkannya dalam fungsi menghindari duplikasi kode, sehingga lebih mudah diubah dan dipelihara.
5. Mengurangi Kompleksitas: Fungsi membantu menyederhanakan masalah yang kompleks dengan memecahnya menjadi langkah-langkah yang lebih kecil. Ini memungkinkan pemrogram fokus pada satu tugas kecil saat menulis fungsi tertentu.
6. Parameterisasi: Fungsi dapat menerima parameter, yang memungkinkan fleksibilitas lebih besar, karena fungsi dapat dipanggil dengan berbagai nilai input untuk mendapatkan hasil yang berbeda.
7. Pengujian yang Lebih Mudah: Dengan memisahkan logika program ke dalam fungsi-fungsi, setiap fungsi bisa diuji secara terpisah, sehingga lebih mudah mendeteksi dan memperbaiki kesalahan.
8. Fungsi sangat penting dalam pemrograman, terutama dalam proyek-proyek besar yang memerlukan organisasi kode yang baik dan mudah dipelihara.

B. Requirement

1. Sistem Operasi yang digunakan : windows. 10 vers

2. Browser : Google Chrome Version 129.0.6668.100 (Official Build) (x86\_64)

3. Tools yang digunakan : Visual Studio Code

C. Dasar Teori

1. Dasar teori fungsi

Fungsi dalam pemrograman adalah blok kode yang dirancang untuk menjalankan tugas tertentu dan dapat dipanggil sesuai kebutuhan. Fungsi memungkinkan pengembang untuk menyusun program menjadi bagian-bagian yang lebih kecil, sehingga lebih mudah dikelola dan dipahami. Fungsi juga mengoptimalkan proses pemrograman melalui modularitas dan penggunaan ulang kode.

Karakteristik Utama Fungsi

Deklarasi Fungsi: Proses mendefinisikan fungsi dilakukan dengan nama fungsi yang unik, diikuti oleh parameter (jika ada), dan blok perintah yang akan dijalankan. Nama fungsi sebaiknya sesuai dengan tugas yang dilakukan agar mudah dikenali.

Parameter dan Argumen: Parameter adalah variabel yang ditentukan saat mendeklarasikan fungsi untuk menerima input dari argumen yang diberikan saat fungsi dipanggil. Argumen adalah nilai aktual yang diberikan pada parameter ketika fungsi dijalankan.

Isi atau Blok Kode Fungsi: Berisi perintah atau proses yang akan dijalankan ketika fungsi dipanggil.

Return Value (Nilai Kembali): Fungsi dapat mengembalikan nilai kepada pemanggil menggunakan pernyataan return. Nilai ini dapat digunakan lebih lanjut dalam proses lain di program utama.

Jenis Fungsi

Fungsi Built-in: Fungsi bawaan yang sudah disediakan oleh bahasa pemrograman. Misalnya, dalam Python ada fungsi print() untuk menampilkan output.

Fungsi Buatan Pengguna (User-defined Function): Fungsi yang didefinisikan oleh pengguna untuk kebutuhan spesifik dalam program.

Contoh Penggunaan Fungsi (Dalam Python)

python

Salin kode

def kali(a, b):

hasil = a \* b

return hasil

Pada contoh ini:

kali adalah nama fungsi.

a dan b adalah parameter yang menerima nilai input.

Fungsi mengembalikan hasil perkalian a dan b.

Manfaat Menggunakan Fungsi

Modularitas: Membantu memecah program menjadi bagian yang lebih kecil, sehingga mudah dikelola.

Penggunaan Ulang Kode: Fungsi yang sudah didefinisikan dapat digunakan berkali-kali tanpa menulis ulang kode.

Mengurangi Duplikasi Kode: Menghindari pengulangan kode yang sama di beberapa tempat dalam program.

Pemeliharaan Lebih Mudah: Dengan struktur fungsi, perbaikan atau pembaruan pada kode lebih mudah dilakukan dan lebih terstruktur.

Kesimpulan

Fungsi adalah elemen dasar dalam pemrograman yang memberikan fleksibilitas, efisiensi, dan keteraturan dalam menulis kode. Dengan memanfaatkan fungsi, pengembang dapat membangun program yang lebih modular dan mudah dikelola.

D. Implementasi

1. # Fungsi Mencari Nilai Maksimal

2. Baris 2-8: Mendefinisikan fungsi max yang menerima 3 parameter (angka1, angka2, angka3)

python

def max(angka1, angka2, angka3):

if ((angka1 > angka2)) and ((angka1 > angka3)): # Jika angka1 lebih besar dari angka2 DAN angka3

return angka1 # Mengembalikan angka1

elif((angka2 > angka1)) and ((angka2 > angka3)): # Jika angka2 lebih besar dari angka1 DAN angka3

return angka2 # Mengembalikan angka2

else: # Jika kondisi di atas tidak terpenuhi

return angka3 # Mengembalikan angka3

3. Baris 10-12: Meminta input dari pengguna untuk 3 angka

python

a = int(input("masukan angka1: ")) # Input angka pertama

b = int(input("masukan angka2: ")) # Input angka kedua

c = int(input("masukan angka3: ")) # Input angka ketiga

4. Baris 14: Memanggil fungsi max dengan parameter a, b, c dan menyimpan hasilnya di variabel cek\_max

python

cek\_max = max(a,b,c)

5. Baris 15-16: Mencetak hasil dengan format yang menampilkan nilai maksimal dari ketiga angka

python

print("nilai maksimal dari",a," dan ",b," dan ",c, "adalah",cek\_max)

Program ini berfungsi untuk:

1. Menerima 3 angka dari pengguna

2. Membandingkan ketiga angka tersebut

3. Menentukan angka terbesar

4. Menampilkan hasilnya ke layar

Output akan menampilkan kalimat yang memberitahu nilai maksimal dari ketiga angka yang dimasukkan.

1. # fungsi mencari luas persegi panjang

2. Baris 3-4: Mendefinisikan fungsi untuk menghitung luas persegi panjang

python

def luaspersegipanjang (panjang, lebar):

return panjang\*lebar # Menghitung luas dengan rumus panjang x lebar

3. Baris 6-7: Mendefinisikan fungsi untuk menghitung keliling persegi panjang

python

def kelilingpersegipanjang (panjang, lebar):

return 2\*(panjang+lebar) # Menghitung keliling dengan rumus 2 x (panjang + lebar)

4. Baris 9-10: Meminta input dari pengguna untuk nilai panjang dan lebar

python

x=float(input("masukan panjang: ")) # Input # Input panjang (menggunakan float untuk angka desimal)

y=float(input("masukan lebar: ")) # Input lebar (menggunakan float untuk angka desimal)

5. Baris 12: Komentar untuk contoh nilai (panjang=10, lebar=5)

python

# nilai luas jika panjang=10, dan lebar=5

6. Baris 13-14: Memanggil fungsi dan menyimpan hasilnya

python

hasil\_luas = luaspersegipanjang (x,y) # Menghitung luas

hasil\_keliling = kelilingpersegipanjang (x,y) # Menghitung keliling

7. Baris 15-16: Mencetak hasil perhitungan

python

print("luasnya adalah ", hasil\_luas) # Menampilkan hasil luas

print("kelilingnya adalah ", hasil\_keliling) # Menampilkan hasil keliling

Program ini berfungsi untuk:

1. Menerima input panjang dan lebar dari pengguna

2. Menghitung luas persegi panjang (panjang × lebar)

3. Menghitung keliling persegi panjang (2 × (panjang + lebar))

4. Menampilkan hasil perhitungan luas dan keliling

Output akan menampilkan dua baris yang memberitahu luas dan keliling dari persegi panjang berdasarkan nilai panjang dan lebar yang dimasukkan.

1. Baris 1-5: Mendefinisikan fungsi faktorial yang menggunakan rekursi

python

def faktorial(n):

if n==0 or n==1: # Basis rekursi: jika n adalah 0 atau 1

return 1 # Mengembalikan nilai 1

else:

return n\*(faktorial(n-1)) # Rekursi: n dikali faktorial dari (n-1)

2. Baris 7: Komentar untuk mencari nilai faktorial

python

# mencari nilai faktorial

3. Baris 8-10: Input dan output program

python

a = int(input("masukan nilai yang akan dicari: ")) # Meminta input angka

cari\_faktorial = faktorial(a) # Memanggil fungsi faktorial

print("nilai dari", a, "! adalah", cari\_faktorial) # Menampilkan hasil

Program ini berfungsi untuk menghitung faktorialProgram ini berfungsi untuk menghitung faktorial dari sebuah angka menggunakan metode rekursi. Faktorial (dilambangkan dengan !) adalah hasil perkalian semua bilangan bulat positif dari 1 sampai n.

Contoh cara kerja:

- Jika input a = 5

- Maka 5! = 5 × 4 × 3 × 2 × 1 = 120

- Fungsi akan memanggil dirinya sendiri dengan nilai yang lebih kecil hingga mencapai 1 atau 0

Output akan menampilkan hasil perhitungan faktorial dari angka yang dimasukkan.

E. Daftar Pustaka

Downey, A. B. (2015). Think Python: How to Think Like a Computer Scientist. O'Reilly Media.

Lutz, M. (2013). Learning Python. O'Reilly Media.

Zelle, J. (2004). Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates.

Guttag, J. V. (2016). Introduction to Computation and Programming Using Python. The MIT Press.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. The MIT Press.