

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
KE - 10
PYTHON FUNCTIONS



DISUSUN OLEH :

Restu Wibisono

2340506061

JURUSAN TEKNOLOGI INORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR

2023

LAPORAN
SESUAI NAMA MATA KULIAH PRAKTIKUM



Diisi Mahasiswa Praktikan										
Nama Praktikan	Restu Wibisono									
NPM	2340506061									
Rombel	03									
Judul Praktikum	Python Functions									
Tanggal Praktikum	01 November 2023									
Diisi Asisten Praktikum										
Tanggal Pengumpulan	<table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>									
Catatan										

PENGESAHAN		NILAI
Diperiksa oleh :	Disahkan oleh :	
Asisten Praktikum	Dosen Pengampu	
Ludfi Arfiani	Imam Adi Nata	

PROGRAM STUDI S1 TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS TIDAR
2023

I. Tujuan Praktikum

Adapun tujuan praktikum ini sebagai berikut:

1. Memahami pengenalan konsep dasar fungsi dalam python yang mencakup sintaks pembuatan fungsi.
2. Belajar cara untuk mendefinisikan fungsi dalam python menggunakan kata "def".
3. Mempelajari penggunaan parameter serta argumen dalam fungsi, belajar untuk mendefinisikan parameter serta bagaimana memasukkan argumen disaat memanggil fungsi.
4. Belajar penggunaan parameter dalam fungsi untuk menerima input dari user dan menghasilkan output sesuai keinginan.

II. Dasar Teori

Python Functions

Dalam praktikum pemrograman, Python functions digunakan untuk mengorganisir dan memisahkan bagian-bagian program yang berbeda dalam blok kode yang dapat digunakan kembali. Fungsi-fungsi ini dapat menerima argumen (input), melakukan operasi tertentu, dan mengembalikan hasil (output) menggunakan pernyataan return.

Untuk mendefinisikan sebuah fungsi dalam Python, saat menggunakan kata kunci def diikuti dengan nama fungsi yang diinginkan. Setelah itu, juga untuk menentukan argumen (jika ada) di dalam tanda kurung setelah nama fungsi. Blok kode yang terkait dengan fungsi tersebut diletakkan di dalam blok yang diindentasi, yang akan dieksekusi saat fungsi dipanggil.

Argumen yang diberikan pada fungsi merupakan nilai yang diperlukan untuk melakukan operasi yang diinginkan. Dalam praktikum ini, dapat menggunakan argumen tersebut untuk melakukan perhitungan faktorial atau menghitung luas dan keliling bangun datar.

Setelah menjalankan operasi yang diinginkan di dalam fungsi, dapat menggunakan pernyataan `return` untuk mengembalikan hasil atau nilai output dari fungsi tersebut. Ketika pernyataan `return` dieksekusi, fungsi akan segera keluar dan mengembalikan nilai yang ditentukan. Ini memungkinkan untuk menggunakan hasil fungsi di bagian lain program atau menyimpannya dalam variabel.

Dalam praktikum, dapat menggunakan fungsi yang telah didefinisikan untuk menghitung faktorial dari suatu bilangan atau menghitung luas dan keliling dari berbagai bangun datar. Kemudian, dapat memanfaatkan nilai yang dikembalikan oleh fungsi tersebut untuk menampilkan hasil atau melakukan operasi lebih lanjut di dalam program.

Dengan menggunakan Python functions dan konsep `def` serta `return`, dapat membuat program yang lebih terstruktur, mudah dibaca, dan dapat digunakan kembali. Juga dapat memisahkan tugas-tugas yang berbeda menjadi fungsi-fungsi yang terpisah, sehingga memudahkan pemeliharaan dan pengembangan program di masa mendatang.

Dalam laporan praktikum, dapat menjelaskan dengan lebih rinci mengenai dasar teori di atas, meliputi langkah-langkah untuk mendefinisikan fungsi, penggunaan argumen, operasi yang dilakukan di dalam fungsi, penggunaan pernyataan `return` untuk mengembalikan hasil, serta contoh penggunaan fungsi-fungsi tersebut dalam program dan hasil yang dihasilkan.

III. Metode Praktikum

A. Alat dan bahan

1. Komputer: Pastikan memiliki komputer yang dapat menjalankan Python. Komputer harus memiliki sistem operasi yang kompatibel dengan Python dan spesifikasi yang memadai untuk menjalankan aplikasi Python dengan lancar.
2. Python: Pastikan Python terinstal di komputer. Bisa didapatkan dengan mengunduh Python versi terbaru dari situs resmi Python dan mengikuti petunjuk instalasi terlampir.
3. Editor teks atau IDE: Pilih editor teks atau IDE (lingkungan pengembangan terintegrasi) yang sesuai dengan preferensi. Beberapa opsi populer termasuk PyCharm, Visual Studio Code, Atom, Sublime Text, atau IDLE (bawaan Python). Pastikan memiliki editor teks atau IDE yang mendukung Python dan diinstal di komputer.
4. Modul Latihan: Jika ada modul yang berkaitan dengan praktik pemrograman Python, baca dan pahami materi tersebut sebelum memulai praktik.
5. Koneksi internet (opsional): Jika ingin mencari lebih banyak materi referensi, mengunduh lebih banyak paket atau modul Python, atau mengakses dokumentasi online, pastikan memiliki koneksi Internet yang stabil.

B. Langkah kerja

1. Persiapan:
 - Pastikan telah menginstal Jupyter Notebook di komputer. Notebook Jupyter dapat diinstal menggunakan Anaconda atau pip.
 - Pastikan juga Python diinstal di komputer.
 - Buka Jupyter Notebook dengan menjalankan "jupyter notebook" di terminal atau baris perintah.

2. Buat buku catatan baru:
 - Saat buku catatan Jupyter terbuka di browser, klik tombol "Baru" untuk membuat buku catatan baru.
 - Pilih kernel yang sesuai yaitu. Python 2 atau Python 3.
3. Tulis kodenya:
 - Di buku catatan baru, akan melihat sel kosong yang bisa diisi dengan kode Python.
 - Tulis kodenya sesuai petunjuk pada modul latihan.
4. Jalankan kode:
 - Untuk menjalankan kode di dalam sel, klik tombol "Run" atau tekan "Shift+Enter" pada keyboard.
 - Hasil kode ditampilkan di bawah sel.
5. Periksa hasilnya dan perbaiki:
 - Perhatikan hasil yang ditampilkan setelah menjalankan kode.
 - Jika ada error atau kekeliruan, perbaiki kode yang salah tersebut.
6. Penyusunan Laporan:
 - Di akhir latihan, buat laporan yang memuat tujuan latihan, langkah-langkah yang dilakukan, hasil observasi, analisis dan kesimpulan yang diambil

IV. Hasil dan Analisis

A. Weekly Exercise 1

```
# Write your code below and press Shift+Enter to execute
def faktorial(n):
    if(n == 0 or n == 1):
        return 1
    else:
        return n * faktorial(n - 1)

nilai = 4

if nilai < 0:
    print("Bilangan harus non-negatif")
else:
    hasil = faktorial(nilai)
    print("Nilai faktorial dari", nilai, "adalah", hasil)

Nilai faktorial dari 4 adalah 24
```

(Gambar 1)

1. Program diawali dengan mendefinisikan fungsi faktorial(n). Fungsi ini nantinya akan menerima argumen n yang termasuk bilangan bulat non-negatif.
2. Di dalam penggunaan fungsi faktorial(n), terdapat sebuah if-else statement yang digunakan sebagai penanganan kasus dasar saat n adalah 0 atau 1. Jika n tersebut adalah 0 atau 1, maka fungsi akan mengembalikan nilai 1, karena jika faktorial dari 0 dan 1 didefinisikan sebagai 1.
3. Jika n bukan 0 atau 1, maka program akan menjalankan blok else. Di dalam blok else ini, fungsi faktorial(n) akan memanggil dirinya sendiri secara rekursif dengan argumen n-1 dan mengalikan hasilnya dengan n. Jadi dengan demikian, fungsi ini akan secara berulang memanggil terus dirinya sendiri dengan nilai n yang akan semakin kecil hingga sampai kasus dasar (0 atau 1). Hal ini memungkinkan perhitungan faktorial n dengan memecahnya menjadi submasalah yang lebih kecil.
4. Setelah mendefinisikan bagaimana fungsi faktorial(n), program kemudian menginisialisasi variabel nilai dengan nilai 4.

5. Selanjutnya, terdapat if-else statement untuk memeriksa apakah nilai itu bilangan negatif. Jika nilai merupakan bilangan negatif, program akan mencetak pesan "Bilangan harus non-negatif".
6. Jika nilai merupakan non-negatif, maka program akan menjalankan blok else. Dalam blok else, program memanggil fungsi-fungsi faktorial dengan argumen nilai dan hasilnya akan disimpan dalam variabel hasil.
7. Akhirnya, program akan mencetak pesan "Nilai faktorial dari 4 adalah 24" dengan menggunakan nilai-nilai input dan nilai faktorial yang telah dihitung, yaitu hasil.

B. Weekly Esecise 2

```
# Write your code below and press Shift+Enter to execute
def luasLingkaran():
    jariJari = int(input("Masukkan jari-jari:"))
    luas = 3.14 * jariJari * 2
    return luas

def luasPersegi():
    sisi = int(input("Masukkan panjang sisi:"))
    luas = sisi * sisi
    return luas

def luasSegitiga():
    alas = int(input("Masukkan panjang alas:"))
    tinggi = int(input())
    luas = alas * tinggi / 2
    return luas

while True:
    print("Kalkulator Jagat")
    print("Pilih Bangun Datar")
    print("1. Luas Lingkaran")
    print("2. Luas Persegi")
    print("3. Luas Segitiga")
    print("Keluar")

    pilih = int(input("1/2/3/4"))

    if pilih == 1:
        luas = luasLingkaran()
        print("Luas lingkaran adalah", luas)
    elif pilih == 2:
        luas = luasPersegi()
        print("Luas Persegi adalah", luas)
    elif pilih == 3:
        luas = luasSegitiga()
        print("Luas segitiga adalah", luas)
    elif pilih == 4:
        break
    else:
        print("apaan dah")

✓ 10.0s
```

(Gambar 2)

1. Program akan dimulai dengan cara mendefinisikan tiga fungsi: `luasLingkaran()`, `luasPersegi()`, dan `luasSegitiga()`. Masing-masing fungsi ini akan digunakan untuk menghitung luas lingkaran, persegi, dan segitiga.
2. Setelah program mendefinisikan fungsi-fungsi tersebut, program lalu masuk ke dalam sebuah while loop yang akan berjalan terus menerus. Loop ini berguna untuk menampilkan menu kepada user agar user dapat memilih bentuk geometri apa dan menghitung luasnya.
3. Di dalam loop, menu ditampilkan menggunakan perintah `print()`.
4. Pilihan user akan disimpan di dalam variabel `pilih` menggunakan fungsi `input()`.
5. Jika user memilih opsi 1, maka fungsi `luasLingkaran()` dipanggil dan luas yang dikembalikan akan disimpan dalam variabel `luas`. Lalu luas tersebut kemudian ditampilkan menggunakan perintah `print()`.
6. Jika user memilih opsi 2, maka fungsi `luaspersegi()` dipanggil dan luas yang dikembalikan akan disimpan dalam variabel `luas`. Lalu luas tersebut kemudian ditampilkan menggunakan perintah `print()`.
7. Jika user memilih opsi 3, maka fungsi `luassegitiga()` dipanggil dan luas yang dikembalikan akan disimpan dalam variabel `luas`. Lalu luas tersebut kemudian ditampilkan menggunakan perintah `print()`.
8. Jika user memilih opsi 4, loop akan keluar menggunakan perintah `break`.
9. Jika user memilih opsi lainnya, akan adapesan kesalahan ditampilkan menggunakan perintah `print()`.

V. Kesimpulan

Adapun kesimpulan yang dapat diambil pada praktikum ini sebagai berikut

1. Program faktorial ini menggunakan pendekatan rekursif yang digunakan untuk menghitung faktorial suatu bilangan. Faktorial adalah hasil perkalian dari semua bilangan bulat positif dari 1 hingga nilai bilangan yang diberikan. Dalam rekursi, fungsi faktorial memanggil dirinya sendiri dengan nilai argumen yang lebih kecil hingga mencapai kasus dasar (faktorial dari 0 atau 1). Kemudian, nilai-nilai yang akan dikembalikan dari pemanggilan rekursif dan digunakan untuk menghitung faktorial yang lebih besar. Proses ini berlanjut hingga mencapai nilai faktorial yang diminta oleh pengguna.
2. Fungsi-fungsi dalam program menggunakan kata kunci “def” untuk mendefinisikan fungsi-fungsi tersebut. Pada saat menggunakan pemanggilan fungsi, argumen yang diperlukan diberikan. Di dalam fungsi, pernyataan-pernyataan dijalankan, juga termasuk pemanggilan fungsi itu sendiri jika masih ada pendekatan rekursif. Pada akhirnya, nilai hasil perhitungan atau nilai yang diinginkan dikembalikan menggunakan pernyataan return. Penggunaan return ini memungkinkan nilai yang dihasilkan di dalam fungsi dapat untuk digunakan di luar fungsi tersebut, seperti untuk ditampilkan kepada user atau digunakan dalam perhitungan yang lebih kompleks.
3. Dengan menggunakan fungsi-fungsi terpisah, program-program tersebut berjalan memanfaatkan konsep modularitas. Modularitas memungkinkan memisahkan tugas dan tanggung jawab antara fungsi-fungsi yang berbeda. Ini dapat memudahkan pemeliharaan dan pengembangan kode, serta bisa memungkinkan penggunaan kembali kode dalam konteks yang berbeda. Selain itu, pemisahan tugas ini juga akan meningkatkan keterbacaan dan kejelasan program.
4. Program-program tersebut menggunakan input dari user dengan menggunakan fungsi input() kemudian menampilkan output menggunakan fungsi print(). Interaksi ini bisa memungkinkan user memasukkan nilai-nilai yang diperlukan dan melihat hasil perhitungan dengan jelas. Penggunaan

input dan output yang baik memainkan peran penting dalam membuat program yang interaktif dan mudah digunakan.

VI. Referensi

- Pratama, E. K. 2022. Optimalisasi Query Database E Learning Jti Politeknik Negeri Jember Menggunakan Index (Doctoral dissertation, Politeknik Negeri Jember).
- Sari, M. P., Setiawansyah, S., & Budiman, A. 2021. Perancangan Sistem Informasi Manajemen Perpustakaan Menggunakan Metode Fast (Framework for the Application System Thinking)(Studi Kasus: Sman 1 Negeri Katon). *Jurnal Teknologi Dan Sistem Informasi*, 2(2), 69-77.
- Wiliam Gozali, W. G., & Alham Fikri Aji, A. F. A. 2014. PEMROGRAMAN KOMPETITIF DASAR.