



RPL - ITESA

MODUL **ALGORITMA** **PEMROGRAMAN** **Python**





REKAYASA PERANGKAT LUNAK ITESA MUHAMMADIYAH SEMARANG

**PROGRAM STUDI REKAYASA PERANGKAT LUNAK
ITESA MUHAMMADIYAH SEMARANG**

**MODUL CODING DASAR
PYTHON**

TIM PENYUSUN

Pengarah: Nurul Huda, S.Kom, M.Kom

Ketua: Maulany Citra Pandini

Sekretaris: M. Yanfaunnas

Anggota: 1. Latania Zhafira
2. Adi Fajar Saputra
3. Davit Setyawan

Tahun 2023

KATA PENGANTAR

Puji Syukur kepada Allah, Tuhan Yang Maha Esa, karena limpahan rahmat, taufik, dan ridha-Nya sehingga kami dapat menyelesaikan penyusunan modul Ujian Akhir Semester yang kami berikan judul "Algoritma Pemrograman Python".

Modul ini dirancang untuk memberikan penjelasan yang jelas tentang algoritma dasar dan konsep pemrograman yang terkait dengan bahasa pemrograman Python. Algoritma dan program adalah produk dari industri teknologi informasi.

Dalam modul ini, kami akan memperkenalkan berbagai topik penting yang terkait dengan algoritma dan pemrograman Python. serta akan melanjutkan dengan menyebutkan konsep pemrograman dasar seperti variabel, jenis data, dan struktur kontrol.

Modul ini dibuat untuk semua tingkat baik dari pemula yang baru memulai perjalanan mereka melalui pemrograman atau mereka yang ingin memperluas pemahaman mereka tentang algoritma dan program Python. Dalam setiap bagian, kami mencakup contoh kode, pelajaran, dan analog untuk membantu Anda memahami materi dan memajukan pemahaman Anda

Kami berharap modul ini dapat membantu memberikan materi yang dapat dipelajari untuk mahasiswa maupun pelajar yang mempelajari algoritma bahasa pemrograman python. Kami sangat mengharapkan kritik dan saran serta menerima dengan senang hati demi perbaikan dan kesempurnaan modul ini.

Kami mengucapkan terima kasih kepada berbagai pihak yang telah membantu proses penyelesaian modul ini, terutama dosen pengampu mata kuliah Etika Profesi dan Komunikasi yang telah membimbing penyusun dalam pembuatan modul ini. Semoga modul ini dapat bermanfaat bagi kita semua.

Semarang, 7 Juni 2023

Penyusun

DAFTAR ISI

KATA PENGANTAR.....	5
DAFTAR ISI.....	6
MODUL 1 PENGENALAN ALGORITMA.....	7
1.1 Tujuan Pembelajaran.....	7
1.2 Dasar Teori.....	7
1.3 Pengertian Algoritma.....	7
MODUL 2 PENGENALAN DASAR BAHASA PEMROGRAMAN PYTHON.....	8
2.1 Pendahuluan.....	8
2.2 Pengenalan Dasar Python.....	8
2.3 Proses Instalasi Python.....	8
2.4 Cara Menjalankan Python.....	10
2.5 Syntax Dasar pada Python.....	11
MODUL 3 STRUKTUR DATA PEMROGRAMAN PYTHON.....	12
3.1 Variabel.....	12
3.2 Tipe Data.....	13
3.3 Operator.....	14
MODUL 4 PERCABANGAN.....	18
4.1 Pengertian.....	18
4.2 Struktur Kondisi if.....	18
4.3 Struktur Kondisi if/else.....	18
4.3 Struktur Kondisi switch case.....	19
MODUL 5 PERULANGAN.....	20
5.1 Perulangan for.....	20
5.2 Perulangan while.....	20
5.3 Perulangan Bersarang.....	21
MODUL 6 FUNGSI.....	22
6.1 Pendahuluan.....	22
6.3 Cara Membuat Fungsi pada Python.....	22
DAFTAR PUSTAKA.....	23

MODUL 1 PENGENALAN ALGORITMA

1.1 Tujuan Pembelajaran

Bagi para pemula yang ingin mempelajari algoritma dan dasar-dasar pemrograman menggunakan bahasa Python, buku modul “Algoritma Pemrograman Python” ini menjadi salah satu sumber belajar yang penting. Saat ini, python menjadi salah satu bahasa pemrograman yang populer digunakan dan mudah dipelajari oleh pemula.

Kemajuan teknologi dan era digital saat ini menjadi salah satu alasan pentingnya untuk mempelajari tentang algoritma dan dasar-dasar pemrograman. Menjadi seorang programmer menjadi salah satu impian banyak orang pada era ini, dan buku modul “Algoritma Pemrograman Python” ini menjadi solusi dalam membantu memahami dan mempelajari dasar-dasar pemrograman seperti algoritma.

1.2 Dasar Teori

Sebuah algoritma dianggap efektif ketika dioperasikan oleh sebuah pemroses atau *processor*. Pemroses dapat berupa manusia, robot, mesin, atau komputer, dan yang lainnya. *Processor* akan membaca dan melaksanakan instruksi yang ada pada algoritma. Sebuah pemroses tersebut harus memahami setiap langkah algoritma dan mengikuti prosedur operasi yang bersesuaian dengan prosedur tersebut.

Ketika kita berbicara tentang algoritma dan pemrograman, komputer adalah yang bertanggung jawab untuk memproses algoritma tersebut. Sedangkan bahasa yang kita gunakan adalah bahasa yang dimengerti oleh orang lain ketika kita berbicara dengan mereka. Dalam hal ini, terdapat kaitannya dengan bahasa daerah masing-masing orang tentunya memiliki perbedaan yang mempengaruhi tingkat pemahaman manusia terhadap suatu bahasa tersebut. Begitupun berlaku juga pada komputer yang memiliki bahasa yang harus kita pahami, yaitu bahasa komputer. Kita perlu memahami bahasa komputer untuk berinteraksi dengan komputer tersebut, sehingga komputer mengerti apa yang kita perintahkan untuk segera diproses. Program adalah istilah yang digunakan untuk menggambarkan algoritma yang ditulis dalam bahasa komputer.

1.3 Pengertian Algoritma

Algoritma merupakan kumpulan langkah yang disusun secara sistematis dan logis untuk menyelesaikan suatu masalah. Algoritma ditulis sebelum penulisan program dalam dunia pemrograman sederhana. Masalah perhitungan matematik adalah salah satu yang dapat diselesaikan dengan pemrograman komputer.

Algoritma adalah inti ilmu komputer atau ilmu informatika. Terdapat banyak bidang ilmu komputer yang menghasilkan istilah algoritma itu sendiri. Namun, kita harus memperhatikan bahwa algoritma tidak selalu berarti ilmu komputer saja. Selain itu, kita sebagai manusia memiliki proses algoritma dalam kehidupan sehari-hari kita. Seperti contohnya, terdapat algoritma dalam menggambarkan langkah-langkah atau prosedur yang dimasukkan ke dalam resep untuk membuat kue atau masakan. Pada setiap resep pasti ada urutan langkah-langkah yang diperlukan untuk membuat masakan. Masakan yang diinginkan tidak dapat dibuat jika langkah-langkah pada proses pembuatan tidak logis atau masuk akal.

MODUL 2 PENGENALAN DASAR BAHASA PEMROGRAMAN PYTHON

2.1 Pendahuluan

Python adalah bahasa pemrograman yang banyak digunakan yang sering digunakan di banyak bidang yang berbeda, termasuk pengembangan web, analisis data, kecerdasan buatan, dan perangkat lunak konstruksi. Bahasa ini diciptakan pada awal 1990-an oleh Guido van Rossum, dan sejak saat itu telah tumbuh menjadi bahasa yang paling banyak digunakan di dunia.

Salah satu alasan utama mengapa Python begitu populer adalah karena sintaks yang sederhana dan mudah dipahami. Python didasarkan pada filosofi "bacaan manusia", yang membuat kode mudah dibaca dan ditulis. Karena itu, belajar bahasa adalah pilihan yang sangat bijak untuk pemula yang ingin memahami tata bahasa Python.

Selain itu, Python juga memiliki banyak fitur lainnya. Bahasa ini mendukung pemrograman fungsional dan berorientasi objek, dan mengandung koleksi besar modul dan pustaka yang dikenal sebagai Perpustakaan Standar Python. Toko ini menawarkan berbagai alat dan fungsi yang mudah digunakan yang memungkinkan pengembang untuk membangun aplikasi dengan cepat dan efektif.

Memahami dasar-dasar Python akan memberi Anda dasar yang Anda butuhkan untuk mempelajari topik yang lebih canggih dalam bahasa dan memungkinkan Anda untuk terus mengembangkan berbagai aplikasi berbasis Python.

2.2 Pengenalan Dasar Python

Struktur bahasa yang digunakan Python tidak terlalu rumit dibandingkan bahasa pemrograman lainnya. Python dikenal sebagai bahasa pemrograman yang ramah pengguna karena mudah dipahami, mudah dipelajari, dan menggunakan perintah yang lebih ringkas. Jadi, programmer lebih fokus pada pengembangan program daripada menulis kode.

Python diciptakan oleh Guido Van Rossum dan menjadi terkenal pada tahun 1991. Python adalah bahasa pemrograman yang pada dasarnya open source. Secara umum, Python digunakan untuk mengembangkan perangkat lunak, situs web, skrip sistem, dan matematika. Python juga sering digunakan untuk membuat berbagai program, termasuk CLI, GUI (desktop), mobile, web, IoT, game, dan program hacking.

Python memiliki berbagai macam library yang dapat digunakan dalam ilmu data, beberapa di antaranya termasuk Pandas untuk analisis data, Numpy untuk komputasi matematika, dan Matplotlib untuk visualisasi data.

2.3 Proses Instalasi Python

File instalasi Python tersedia dapat diunduh di situs web Python.org. Situs web ini menawarkan kedua versi Python 2 dan Python 3.

1. Buka python.org di browser
2. Pilih tombol Download
3. Pilih sesuai sistem operasi yang kamu gunakan seperti Windows, Linux atau Mac
4. Klik link pada versi Python yang diinginkan
5. Secara otomatis file akan terdownload

Cara menginstal pada sistem operasi Windows, MacOS, dan Linux

1. Windows

Windows merupakan sistem operasi yang dikeluarkan oleh Microsoft, berikut cara install Python yang sudah di download sebelumnya:

- 1) Buka file install yang sudah di download dari website resmi Python dengan double click
- 2) Pilih "Install for all users" agar bisa digunakan oleh semua user di komputer atau laptop
- 3) Konfigurasi Python agar dikenali oleh sistem operasi dengan checklist Add Python 3.7 to PATH atau versi yang kamu install
- 4) Klik Install Now
- 5) Jika instalasi berhasil akan muncul tulisan Setup was Successful

Selanjutnya, mulailah Python dengan memulai perintah command prompt (CMD) dan ketik "python" untuk beralih ke mode Python. Selain itu, Anda dapat memulai Python menggunakan Python Shell dengan menggunakan fungsi Pencarian Windows. Jalankan Python Shell sebagai administrator dengan mengklik kanan.

2. MacOS

Dikembangkan dan didistribusikan oleh Apple Inc., MacOS adalah sistem operasi anti-graffiti. Berikut adalah cara menginstal pabrik di macOS:

- 1) Download versi Python yang diinginkan atau yang paling baru di Mac
- 2) Pada file install Python klik kanan dan pilih Open
- 3) Klik Continue pada tahapan selanjutnya dimulai dari Introduction hingga Installation Type
- 4) Masukkan password user komputer jika diminta pada tahap Installation
- 5) Jika instalasi berhasil akan muncul "Instalasi telah berhasil"
- 6) Uji coba Python menggunakan command prompt (CMD) sama seperti Windows

3. Linux

Linux adalah sistem operasi yang menggantikan sistem operasi UNIX dan sepenuhnya open source. Linux mungkin tidak begitu populer seperti Windows, di mana hanya 1,5 persen pengguna yang benar-benar menggunakannya. Linux adalah sistem operasi yang dapat diandalkan dan stabil dan yang jarang memiliki masalah seperti virus, malware, proses terhenti, dan hal-hal lain dari sifat itu. Python dapat diinstal pada Linux menggunakan prosedur berikut:

- 1) Download development package yang dibutuhkan untuk membangun Python
- 2) Download file instalasi Python terbaru di website resmi dan akan ada file .tar.xz (tarball) berisi kode sumber Python
- 3) Jika muncul dialog box, pilih Simpan
- 4) Double click pada file yang sudah berhasil di download
- 5) Extract file
- 6) Konfigurasi script dan jalankan di terminal yaitu:

```
$ cd Python-3.*  
./configure
```

- 7) Tunggu hingga instalasi selesai
- 8) Lakukan pengecekan dengan menuliskan python3-version di terminal Linux

2.4. Cara Menjalankan Python

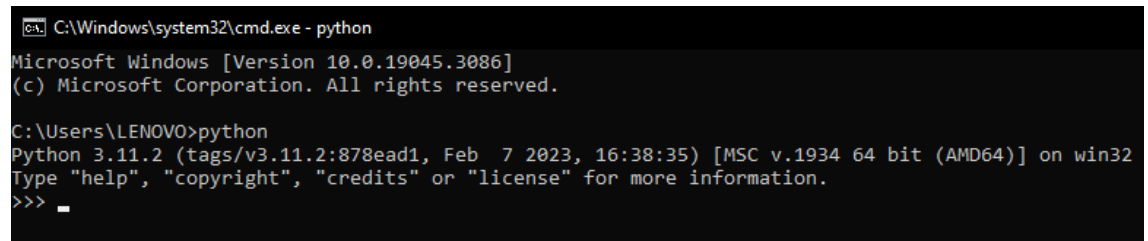
- CMD

Ada dua model CMD: interaktif dan scrip. Mode interaktif, tidak cocok untuk program panjang, tetapi mode Scrip, di sisi lain, cocok bagi program panjang.

Berikut cara menjalankannya:

1. Mode interactive

Mode interactive ini hanya bisa menyelesaikan masalah yang simpel saja. Caranya, buka CMD & ketikan python maka akan tampil seperti berikut:



```
C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>python
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

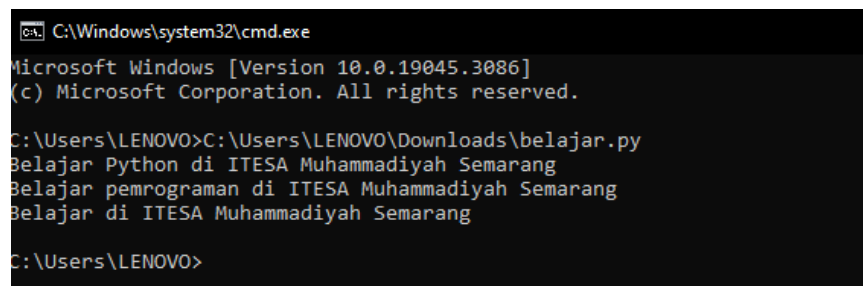
2. Mode script

Dalam mode Script kita bisa membuat beberapa baris code dimana file tersebut berekstensi .py yang artinya bahwa file tersebut adalah file python.

Contoh file yang diberi nama belajar.py

```
print ("Belajar Python di ITESA Muhammadiyah Semarang")
print ("Belajar pemrograman di ITESA Muhammadiyah Semarang")
print ("Belajar di ITESA Muhammadiyah Semarang")
```

Untuk menjalankannya kita bisa menggunakan perintah belajar.py dengan menggunakan CMD.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>C:\Users\LENOVO\Downloads\belajar.py
Belajar Python di ITESA Muhammadiyah Semarang
Belajar pemrograman di ITESA Muhammadiyah Semarang
Belajar di ITESA Muhammadiyah Semarang

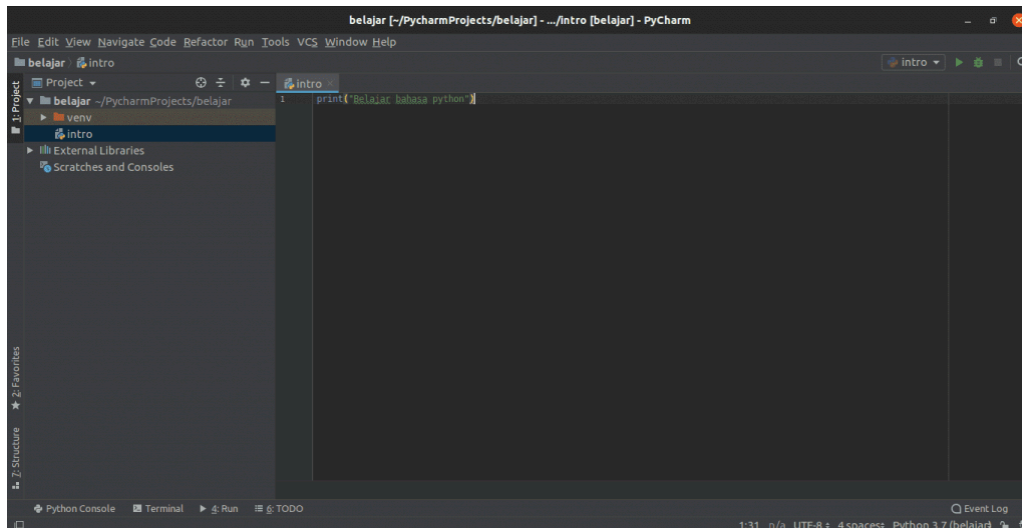
C:\Users\LENOVO>
```

- IDE (pycharm)

Pada pycharm sudah menyediakan running langsung. Berikut cara menggunakannya:

1. Buat projek baru dengan Klik File-New project,dan tulis nama project nya dan klikOK
2. Buat satu file yang berekstensi .py

cara menjalankan python dengan IDE(pycharm) dengan klik kanan pada file dan pilih run untuk menjalankan,maka program python yang kita buat akan berjalan.



2.5 Syntax Dasar pada Python

Bahasa pemrograman Python jadi koding yang menggabungkan kapabilitas dengan sintaks kode yang jelas. Selain JavaScript, Python juga jadi bahasa pemrograman yang hampir bisa digunakan untuk berbagai bidang seperti pengembangan website, aplikasi, game, pembuatan artificial intelligence, hingga pengembangan software.

Contoh yang paling sederhana adalah ketika Anda ingin membuat sebuah hello world sebagai awal dari pembuatan suatu pemrograman. Ketika menggunakan Python, kode yang dibuat untuk hello world sangat sederhana, hanya seperti ini:

```
print "Hello, World!"
```

Sintaks dasar yang berbeda digunakan dalam bahasa pemrograman. Dengan kata lain, "Hello world!" dalam kode python adalah contoh sederhana dari pemrograman sederhana. Setiap bahasa memiliki karakteristik dan kode yang unik. Python tidak memerlukan banyak kode karena itu adalah bahasa yang cukup sederhana. Ini sesuai dengan prinsip PEP 20.

Pada bahasa Python, ada beberapa hal yang perlu Anda ketahui terkait sintaks dasar seperti statement (pernyataan), baris dan indentasi dan tanda kutip.

1. Statement (pernyataan) beberapa statement yang dikenali oleh Python, statement penugasan, statement pengecekan (if), statement pengulangan (for). dalam sebuah statement yang terdiri dari beberapa baris dengan menggunakan backlash (\) tapi yang perlu diingat adalah, statement yang menggunakan tanda kurung seperti (), { }, hingga [] tidak lagi memerlukan backlash (\).
2. Baris dan indentasi Python tidak menggunakan tanda { } untuk memberikan tanda pada grup kode. Grup kode ditandai dengan menggunakan indentasi (spasi) yang harus sama pada setiap grup kode. Kesalahan jumlah spasi akan ditandai dengan peringatan seperti di bawah ini:
SyntaxError: unexpected indent
3. Tanda kutip Untuk menandai string, bahasa pemrograman Python bisa menggunakan tanda kutip satu (') atau tanda kutip ganda (") hingga tiga tanda kutip atau empat sekalipun. Selain itu, string juga diawali dan diakhiri oleh tanda kutip yang sama.

MODUL 3 STRUKTUR DATA PEMROGRAMAN PYTHON

3.1 Variabel

Variabel adalah lokasi memori yang disediakan untuk menyimpan nilai. Ini berarti saat Anda membuat variabel, Anda menyimpan sebagian memori. Variabel menyimpan data untuk diimplementasikan selama eksekusi program. Isi dari variabel tersebut nantinya dapat diubah menggunakan fungsi tertentu pada program yang menggunakan variabel.

Variabel dapat menyimpan berbagai jenis data. Dalam pemrograman Python, variabel memiliki sifat dinamis, artinya variabel Python tidak perlu dideklarasikan dengan tipe data tertentu dan variabel Python dapat dimodifikasi saat program sedang berjalan. Ada juga aturan tertentu untuk menulis sendiri variabel Python, yaitu: Karakter pertama harus berupa huruf atau garis bawah. Karakter berikutnya dapat berupa huruf, garis bawah, atau angka. Karakter nama variabel dipisahkan dengan huruf besar dan huruf kecil. Ini berarti bahwa perbedaan dibuat antara huruf kecil dan huruf besar. Misalnya, variabel nama depan dan nama depan adalah variabel yang berbeda.

Memulai membuat variabel di Python sangat sederhana: Anda cukup menulis variabel dan mengisinya dengan nilai dengan menambahkan tanda sama dengan = diikuti dengan nilai yang diinginkan.

Di bawah ini adalah contoh penggunaan variabel dalam bahasa pemrograman Python

```
In [1]: #proses memasukan data ke dalam variabel
nama = "Adi Fajar"
#proses mencetak variabel
print(nama)

#nilai dan tipe data dalam variabel dapat diubah
umur = 22 #nilai awal
print(umur) #mencetak nilai umur
type(umur) #mengecek tipe data umur
umur = "dua puluh dua" #nilai setelah diubah
print(umur) #mencetak nilai umur
type(umur) #mengecek tipe data umur

namaDepan = "Anfa"
namaBelakang = "Yaunas"
nama = namaDepan + " " + namaBelakang
umur = 19
hobi = "Baca buku"
print("Biodata\n", nama, "\n", umur, "\n", hobi)

#contoh variabel lainya
inivariabel = "Halo"
ini_juga_variabel = "Hai"
inivariabeljuga = "Hi"
inivariabel222 = "Bye"

panjang = 8
lebar = 3
luas = panjang*lebar
print(luas)
```

```
Adi Fajar
22
dua puluh dua
Biodata
Anfa Yaunas
19
Baca buku
24
```

3.2 Tipe Data

Tipe data adalah disk atau memori di komputer yang digunakan untuk menyimpan data. Python sendiri memiliki tipe data yang cukup unik dibandingkan dengan bahasa pemrograman lainnya.

Tipe data dari bahasa pemrograman Python adalah sebagai berikut:

Tipe Data	Contoh	Penjelasan
Boolean	True atau False	Menyatakan benar True yang bernilai 1, atau salah False yang bernilai 0
String	"Yuk belajar python"	Menyatakan karakter/kalimat bisa berupa huruf angka, dll (diapit tanda " atau ')
Integer	22 atau 1266	Menyatakan bilangan bulat
Float	3.14 atau 0.99	Menyatakan bilangan yang mempunyai koma
Hexadecimal	9j atau 1d3	Menyatakan bilangan dalam format heksa (bilangan berbasis 16)
Complex	1+5j	Menyatakan pasangan angka real dan imajiner
List	['xyz', 786, 2.23]	Data untaian yang menyimpan berbagai tipe data dan isinya bisa diubah-ubah
Tuple	('xyz', 768, 2.23)	Data untaian yang menyimpan berbagai tipe data tapi isinya tidak bisa diubah
Dictionary	{'nama': 'adi', 'id': 2}	Data untaian yang menyimpan berbagai tipe data berupa pasangan penunjuk dan nilai

Untuk mencoba berbagai macam tipe data, silahkan coba script Python dibawah ini.

```
In [2]: #tipe data Boolean
print(True)

#tipe data String
print("Ayo belajar Python")
print('Belajar Python Sangat Mudah')

#tipe data Integer
print(20)

#tipe data Float
print(3.14)

#tipe data Hexadecimal
print(9j)

#tipe data Complex
print(5j)

#tipe data List
print([1,2,3,4,5])
print(["satu", "dua", "tiga"])

#tipe data Tuple
print((1,2,3,4,5))
print(("satu", "dua", "tiga"))

#tipe data Dictionary
print({"nama": "Adi", 'umur': 20})
#tipe data Dictionary dimasukan ke dalam variabel biodata
biodata = {"nama": "Fajar", 'umur': 21} #proses inisialisasi variabel biodata
print(biodata) #proses pencetakan variabel biodata yang berisi tipe data Dictionary
print(type(biodata))
#fungsi untuk mengecek jenis tipe data. akan tampil <class 'dict'> yang berarti dict adalah tipe data dictionary
```

Hasil:

```
True
Ayo belajar Python
Belajar Python Sangat Mudah
20
3.14
9j
5j
[1, 2, 3, 4, 5]
['satu', 'dua', 'tiga']
(1, 2, 3, 4, 5)
('satu', 'dua', 'tiga')
{'nama': 'Adi', 'umur': 20}
{'nama': 'Fajar', 'umur': 21}
<class 'dict'>
```

3.3 Operator

Operator adalah konstruksi yang dapat memanipulasi nilai dari operan. Sebagai contoh operasi $3 + 2 = 5$. Disini 3 dan 2 adalah operan dan + adalah operator.

Bahasa pemrograman Python mendukung berbagai macam operator, diantaranya :

1. Operator Aritmatika (Arithmetic Operators)
2. Operator Perbandingan (Comparison (Relational) Operators)
3. Operator Penugasan (Assignment Operators)
4. Operator Logika (Logical Operators)
5. Operator Bitwise (Bitwise Operators)
6. Operator Keanggotaan (Membership Operators)
7. Operator Identitas (Identity Operators)

Operator Aritmatika

Operator	Contoh	Penjelasan
Penjumlahan +	$1 + 3 = 4$	Menjumlahkan nilai dari masing-masing operan atau bilangan
Pengurangan -	$4 - 1 = 3$	Mengurangi nilai operan di sebelah kiri menggunakan operan di sebelah kanan
Perkalian *	$2 * 4 = 8$	Mengalikan operan/bilangan
Pembagian /	$10 / 5 = 2$	Untuk membagi operan di sebelah kiri menggunakan operan di sebelah kanan
Sisa Bagi %	$11 \% 2 = 1$	Mendapatkan sisa pembagian dari operan di sebelah kiri operator ketika dibagi oleh operan di sebelah kanan
Pangkat **	$8 ** 2 = 64$	Memangkatkan operan disebelah kiri operator dengan operan di sebelah kanan operator
Pembagian Bulat //	$10 // 3 = 3$	Sama seperti pembagian. Hanya saja angka dibelakang koma dihilangkan

Dibawah ini adalah contoh penggunaan Operator Aritmatika dalam bahasa pemrograman Python

```
In [3]: #OPERATOR ARITMATIKA

#Penjumlahan
print(13 + 2)
apel = 7
jeruk = 9
buah = apel + jeruk #
print(buah)

#Pengurangan
hutang = 10000
bayar = 5000
sisahutang = hutang - bayar
print("Sisa hutang Anda adalah ", sisahutang)

#Perkalian
panjang = 15
lebar = 8
luas = panjang * lebar
print(luas)

#Pembagian
kue = 16
anak = 4
kuePerAnak = kue / anak
print("Setiap anak akan mendapatkan bagian kue sebanyak ", kuePerAnak)

#Sisa Bagi / Modulus
bilangan1 = 14
bilangan2 = 5
hasil = bilangan1 % bilangan2
print("Sisa bagi dari bilangan ", bilangan1, " dan ", bilangan2, " adalah ", hasil)

#Pangkat
bilangan3 = 8
bilangan4 = 2
hasilPangkat = bilangan3 ** bilangan4
print(hasilPangkat)

#Pembagian Bulat
print(10//3)
#10 dibagi 3 adalah 3.3333. Karena dibulatkan maka akan menghasilkan nilai 3

15
16
Sisa hutang Anda adalah 5000
120
Setiap anak akan mendapatkan bagian kue sebanyak 4.0
Sisa bagi dari bilangan 14 dan 5 adalah 4
64
3
```

Operator Penugasan

Operator penugasan digunakan untuk memberikan atau memodifikasi nilai ke dalam sebuah variabel.

Operator	Contoh	Penjelasan
Sama dengan =	<code>a = 1</code>	Memberikan nilai di kanan ke dalam variabel yang berada di sebelah kiri.
Tambah sama dengan +=	<code>a += 2</code>	Memberikan nilai variabel dengan nilai variabel itu sendiri ditambah dengan nilai di sebelah kanan.
Kurang sama dengan -=	<code>a -= 2</code>	Memberikan nilai variabel dengan nilai variabel itu sendiri dikurangi dengan nilai di sebelah kanan.
Kali sama dengan *=	<code>a *= 2</code>	Memberikan nilai variabel dengan nilai variabel itu sendiri dikali dengan nilai di sebelah kanan.
Bagi sama dengan /=	<code>a /= 4</code>	Memberikan nilai variabel dengan nilai variabel itu sendiri dibagi dengan nilai di sebelah kanan.
Sisa bagi sama dengan %=	<code>a %= 3</code>	Memberikan nilai variabel dengan nilai variabel itu sendiri dibagi dengan nilai di sebelah kanan. Yang diambil nantinya adalah sisa baginya.
Pangkat sama dengan **=	<code>a **= 3</code>	Memberikan nilai variabel dengan nilai variabel itu sendiri dipangkatkan dengan nilai di sebelah kanan.
Pembagian bulat sama dengan //=	<code>a //= 3</code>	Membagi bulat operan sebelah kiri operator dengan operan sebelah kanan operator kemudian hasilnya diisikan ke operan sebelah kiri.

Prioritas Eksekusi Operator di Python

Dari semua operator diatas, masing-masing mempunyai urutan prioritas yang nantinya prioritas pertama akan dilakukan paling pertama, begitu seterusnya hingga prioritas terakhir.

Operator	Keterangan
<code>**</code>	Aritmatika
<code>~, +, -</code>	Bitwise
<code>*, /, %, //</code>	Aritmatika

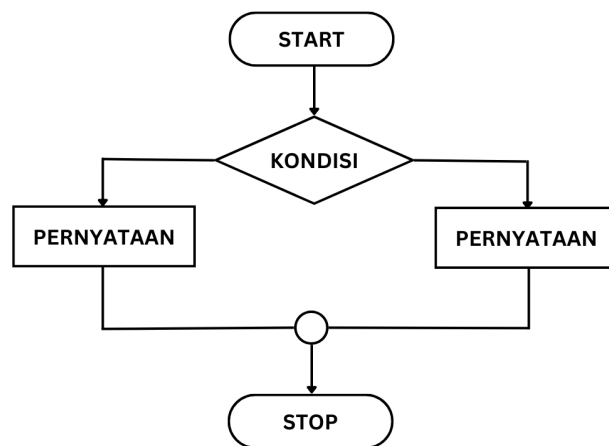
<code>+, -</code>	Aritmatika
<code>>>, <<</code>	Bitwise
<code>&</code>	Bitwise
<code>^</code>	Bitwise
<code><=, <, >, >=</code>	Perbandingan
<code><>, ==, !=</code>	Perbandingan
<code>=, %=, /=, //=, -=, +=, *=, **=</code>	Penugasan
<code>is, is not</code>	Identitas
<code>in, not in</code>	Membership (Keanggotaan)
<code>not, or, and</code>	Logika

MODUL 4 PERCABANGAN

4.1 Pengertian

Sebelum mempelajari jenis kondisi percabangan, Penulis akan menjelaskan definisi percabangan. Dalam pemrograman, percabangan berarti serangkaian proses yang digunakan untuk memilih dari berbagai opsi yang tersedia.

4.2 Struktur Kondisi if



Struktur if adalah struktur percabangan di mana data akan dikerjakan jika kondisi terpenuhi atau benar. Jika kondisi tidak terpenuhi atau salah, data di dalam struktur if tidak akan berjalan, blok if akan terlompati, dan program akan melakukan tindakan selanjutnya.

4.3 Struktur Kondisi if/else

Kode program akan melanjutkan ke kondisi If di bawah kondisi pertama jika tidak terpenuhi atau bernilai False. Jika hal ini juga terjadi, kode program akan melanjutkan ke kondisi If berikutnya, dan demikian seterusnya hingga ada kondisi If yang bernilai True di blok Else terakhir.

Contoh untuk teks:

```
In [1]: if True:
        print('Kode program ini akan dieksekusi')

        if False:
            print('Kode program ini tidak akan dieksekusi')

        print('Kode program ini akan selalu dieksekusi karena tidak termasuk pada percabangan')

Kode program ini akan dieksekusi
Kode program ini akan selalu dieksekusi karena tidak termasuk pada percabangan
```

Contoh untuk angka

```
In [2]: nilai = 50

print('Nilai anda adalah:', nilai, '\n')

if nilai >= 70:
    print('Selamat, anda lulus!')
else:
    print('Maaf, anda tidak lulus.')

Nilai anda adalah: 50

Maaf, anda tidak lulus.
```

4.4 Struktur Kondisi switch case

Dalam switch case, kita membandingkan isi sebuah variabel dengan sebuah nilai, dan jika proses perbandingan menghasilkan nilai, maka blok kode program akan dijalankan. Kondisi switch case terdiri dari dua bagian, perintah switch yang menunjukkan nama variabel yang akan diperiksa dan 1 atau lebih perintah case untuk setiap nilai yang ingin diperiksa.

Contoh:

```
In [3]: def switch(lang):
        if lang == "JavaScript":
            return "You can become a web developer."
        elif lang == "PHP":
            return "You can become a backend developer."
        elif lang == "Python":
            return "You can become a Data Scientist"
        elif lang == "Solidity":
            return "You can become a Blockchain developer."
        elif lang == "Java":
            return "You can become a mobile app developer"

print(switch("JavaScript"))
print(switch("PHP"))
print(switch("Java"))

You can become a web developer.
You can become a backend developer.
You can become a mobile app developer
```

MODUL 5 PERULANGAN

5.1 Perulangan for

Perulangan tersebut adalah perintah yang digunakan dalam Python untuk melakukan loop dari setiap elemen pada rangkaian yang ditandai dengan angka yang merupakan posisi atau indeks dari data atau data koleksi seperti List, Tuple, String, dan sebagainya.

Contoh:

```
In [4]: ulang = 10

for i in range(ulang):

    print(f"Perulangan ke-{i}")

Perulangan ke-0
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
Perulangan ke-6
Perulangan ke-7
Perulangan ke-8
Perulangan ke-9
```

Pertama, kita harus menghitung jumlah perulangan yang kita ingin ulangi, jika kita ingin mengulangi 10 kali maka tulis ulang = 10.

Variabel i menampung indeks, dan range() membuat list dengan range dari 0-10. Selain itu, fungsi str() mengubah tipe data integer ke string.

5.2 Perulangan while

Perulangan while digunakan ketika kita tidak tahu berapa banyak proses looping atau tidak dapat menghitungnya karena data yang digunakan tidak dapat dihitung.

Jenis perulangan while adalah while dan do...while. Perbedaan antara while dan do...while adalah bahwa do...while mengecek kondisi atau statement di belakang sesudah mengulang statement pertama, sesuai dengan situasinya. Jika syarat perulangannya tidak berhubungan dengan hasil perhitungan di blok kode yang diulang, maka gunakan while. Jika begitu, gunakan do/while.

Contoh:

```
In [5]: jawab = 'ya'
        hitung = 0

        while(jawab == 'ya'):
            hitung += 1
            jawab = input("Ulang lagi tidak? ")

        print(f"Total perulangan: {hitung}")

Ulang lagi tidak? ya
Ulang lagi tidak? ya
Ulang lagi tidak? ya
Ulang lagi tidak? tidak
Total perulangan: 4
```

5.3 Perulangan Bersarang

for loop, while loop, dan do while loop adalah contoh perulangan bersarang dalam pemrograman di mana suatu perulangan terletak di dalam perulangan lain.

Contoh:

```
In [6]: listKota = [  
        'Semarang', 'Surabaya', 'Jakarta'  
    ]  
  
    for kota in listKota:  
        print(kota)  
  
        listTempat = [  
            'Lapangan', 'Masjid', 'Mall'  
        ]  
  
        while listTempat:  
            print('-->', listTempat.pop(0))
```

```
Semarang  
--> Lapangan  
--> Masjid  
--> Mall  
Surabaya  
--> Lapangan  
--> Masjid  
--> Mall  
Jakarta  
--> Lapangan  
--> Masjid  
--> Mall
```

MODUL 6 FUNGSI

6.1 Pendahuluan

Penyelesaian masalah kompleks pada penggunaan perangkat lunak, dapat dilakukan dengan memecah kasus utama menjadi kasus yang lebih kecil lagi. Lalu berkonsentrasi mencari pemecahan yang terbaik dari masing-masing bagian yang kecil tersebut. Masing-masing bagian diselesaikan dengan menggunakan algoritma sebaik mungkin. Bagian kecil ini pada akhirnya bisa dapat digabungkan untuk memberikan jawaban yang optimal terhadap masalah yang ada.

Fungsi, juga dikenal sebagai function, adalah kumpulan program atau instruksi yang dapat digunakan untuk melakukan tindakan atau tindakan tertentu. Fungsi ini digunakan pada program yang kompleks dan memiliki banyak fitur. Karena jika kita tidak menggunakannya, kita akan kesulitan menulis kode programnya, banyak kode yang harus ditulis, dan kode akan menjadi sulit dibaca dan dirawat. Kita dapat memecah program besar menjadi subprogram yang lebih sederhana dengan fungsi.

Jenis-jenis Fungsi:

1. *Standard Library Function*, merupakan fungsi yang disediakan oleh Interpreter Python dalam library atau file nya.
2. *User Defined Function*, merupakan fungsi yang diciptakan sendiri. Fungsi ini berbeda dari program utama, memiliki nama yang unik, dan dapat digabungkan ke dalam library.

Mendefinisikan dan Memanggil Fungsi:

- `def`, untuk mendefinisikan bahwa program itu termasuk fungsi
- `return`, untuk mengembalikan suatu nilai pada program yang memanggil fungsi awal.
- Fungsi blok diawali dengan `def` tersebut, yang diikuti oleh nama fungsi dan dilanjutkan dengan `()` tanda kurung.
- Tanda kurung tersebut digunakan untuk memasukkan tiap argumen atau parameter. Parameter didalamnya dapat kita tentukan sendiri.
- Setiap blok kode pada fungsi tersebut dimulai dengan tanda `:` titik dua.

6.2 Contoh Membuat Fungsi pada Python

```
In [1]: #contoh 1
def printme(str):
    "This prints a passed string into this function"
    print (str)
    return
printme("hello world")

hello world
```

In [2]: *#contoh 2 penjumlahan sebuah barisan angka*

```
def jumlah(angka):  
    total = 0  
    for x in angka:  
        total += x  
    return total  
  
print("jumlah: ", jumlah((7,3,8,1,5)))
```

jumlah: 24

In [3]: *#contoh 3 pengecekan angka apakah ganjil atau genap?*

```
def cek_ganjil_genap(angka):  
    if angka%2==0:  
        print("genap")  
    else:  
        print("ganjil")  
    return  
  
cek_ganjil_genap(8) #input angka, misalkan 8
```

genap

In [4]: *#contoh 4 perhitungan rata - rata*

```
def rata_rata(a,b,c,d):  
    return (a+b+c+d)/4  
rata_rata(2,4,6,8)
```

Out[4]: 5.0

In [5]: *#contoh 5 perhitungan sederhana matematika*

```
def calc(angka1,angka2):  
    print(angka1+angka2)  
    print(angka1-angka2)  
    print(angka1*angka2)  
    print(angka1/angka2)  
  
calc(3,4) #inputan 2 angka, angka1=3, dan angka2=4
```

7
-1
12
0.75

DAFTAR PUSTAKA

Fahrudin, Tresna Maulana. 2023. Algoritma dan Pemrograman Dasar dalam Bahasa Pemrograman Python. Gresik: Thalibul Ilmi.

Khotibul Umam, “Algoritma dan Pemrograman Komputer dengan Python”, Duta Media Publishing, 2021, diakses 6 Juli 2023 dari <http://repository.iainmadura.ac.id/455/1/Buku%20Ajar%20Algoritma%20Pemrograman%20Komputer%20dengan%20Python%20%28Khotibul%20Umam%29%20B5.pdf>

Petanikode.com, “Belajar Pemrograman Python: Pengenalan Dasar Python dan Persiapan Awal”, 15 September 2018, <https://www.petanikode.com/python-linux>, [diakses pada 30 Juni 2023]

Malasngoding.com, “Tutorial Python # 3 : Cara Menjalankan File Python”, September 2022, <https://www.malasngoding.com/cara-menjalankan-file-python>, [diakses pada 30 Juni 2023]

Belajarpython.com, “Tipe Data Python”, <https://belajarpython.com/tutorial/tipe-data-python>, [diakses pada 1 Juli 2023]

Dqlab.id, “IF Python dengan Mengenal Jenis Operasinya”, 8 Januari 2022, <https://dqlab.id/cara-download-dan-install-python-pada-berbagai-sistem-operasi>, [diakses pada 2 Juli 2023]

Jagongoding.com, “Python Dasar: Percabangan (If-Else dan Elif) 🐍”, 17 Januari 2021, <https://jagongoding.com/python/dasar/percabangan>, [diakses pada 2 Juli 2023]

Freecodecamp.org, “Python Switch Statement – Switch Case Example”, 5 Agustus 2022, <https://jagongoding.com/python/dasar/percabangan>, [diakses pada 2 Juli 2023]

Dian Prawira, “Algoritma dan Pemrograman”, Modul HMSI Universitas Tanjungpura, 2017, diakses pada 27 Mei 2023 dari <https://hmsifmipauntan.com/wp-content/uploads/2019/09/modul-algoritma-pemrograman-python.pdf>

Fakultas Teknik, “Praktikum Dasar Pemrograman dengan Bahasa Python”, Modul FT Universitas Pancasila, 2022, diakses 28 Mei 2023 dari <https://teknik.univpancasila.ac.id/labkom/praktikum/Modul%20Praktikum/Pyhton/modul%20praktikum%20python.pdf>

Tri Wanda Septian, “Pemrograman Python”, Modul Praktikum Algoritma dan Pemrograman 1, 2021, diakses pada 28 Mei 2023 dari <https://github.com/twseptian/pemrograman-python>