

# **CERDAS MENGUASAI PYTHON**



---

# CERDAS MENGUASAI PYTHON

## Dalam 24 Jam

---

**Rolly M. Awangga**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.’  
Imam Syafi’i*

# CONTRIBUTORS

---

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1</b>	<b>Mengenal Python dan Anaconda</b>	<b>1</b>
<b>2</b>	<b>Pemrograman Dasar</b>	<b>5</b>





# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1</b>	<b>Mengenai Python dan Anaconda</b>
<b>1</b>	<b>1</b>
1.1	Resume
1.1.1	Resume Sejarah Python
1.1.2	Perbedaan Python 2 dan Python 3
1.1.3	Implementasi dan penggunaan Python pada Perusahaan
	ix

1.2	Instalasi	3
1.2.1	Cara Pemakaian Script dan interpreter python	3
1.2.2	Cara Pemakaian spyder termasuk variable explorer	3
1.3	Mencoba Python	3
1.4	Identasi	4
<b>2</b>	<b>Pemrograman Dasar</b>	<b>5</b>
2.1	Teori	5
2.1.1	Variable, pemakaian Variable dan Jenis-Jenis Type data	5
2.1.2	Ketrampilan Pemrograman	8
2.1.3	Ketrampilan Penanganan Error	10

# DAFTAR GAMBAR

---



# DAFTAR TABEL

---



# Listings

---

src/1174083.py	9
src/1174083.py	9
src/1174083.py	9
src/1174083.py	9
src/1174083.py	9
src/1174083.py	9
src/1174083.py	9
src/1174083.py	10
src/1174083.py	10
src/1174083.py	10
src/1174083.py	10
src/1174083_2err.py	11





# FOREWORD

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat  
Februari, 2019*



# ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.



# ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association





# GLOSSARY

---

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald



# SYMBOLS

---

- $A$  Amplitude
- $\&$  Propositional logic symbol
- $a$  Filter Coefficient
  
- $\mathcal{B}$  Number of Beats



# INTRODUCTION

---

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[?].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$



# BAB 1

---

## MENGENAL PYTHON DAN ANACONDA

---

pembelajaran pada pertemuan pertama antara lain:

1. membaca sejarah python dan meresmikan dengan bahasa yang mudah dipahami dan dimengerti
2. melakukan instalasi anaconda
3. menggunakan spyder

dikumpulkan dengan pull request ke github

### 1.1 Resume

#### 1.1.1 Resume Sejarah Python

Bahasa pemrograman Python dirilis pertama kali oleh Guido van Rossum di tahun 1991, yang sudah dikembangkan sejak tahun 1989. Awal pemilihan nama Python tidak secara langsung berasal dari nama ular piton, tapi sebuah acara humor di BBC pada era 1980an dengan judul “Monty Python’s Flying Circus“. Monty



Python adalah kelompok lawak yang membawakan acara tersebut. Kebetulan Guido van Rossum adalah penggemar dari acara ini. Pada tahun 1994, Python 1.0 dirilis, yang diikuti dengan Python 2.0 pada tahun 2000. Python 3.0 keluar pada tahun 2008.

### **1.1.2 Perbedaan Python 2 dan Python 3**

#### **1.1.2.1 Python 2**

Dipublikasikan pada akhir tahun 2000, Python 2 dinilai lebih transparan dan inklusif untuk pengembangan software ketimbang versi sebelumnya. Hal ini didukung dengan adanya PEP – Python Enhancement Proposal, sebuah spesifikasi teknis yang menjadi tuntunan informasi untuk penggunaanya dan menggambarkan fitur baru pada Python itu sendiri. Sebagai tambahan, Python 2 dilengkapi dengan berbagai fitur programatikal seperti cycle-detecting garbage collector untuk mengotomasi manajemen memori, peningkatan dukungan untuk Unicode, list comprehension untuk membuat sebuah list berdasarkan list yang sudah ada. Unifikasi pada tipe data Python dan class ke satu hirarki terjadi pada rilis Python 2.2

#### **1.1.2.2 Python 3**

Python 3 diharapkan sebagai masa depan Python dan merupakan versi yang saat tulisan ini dibuat masih aktif dikembangkan. Python 3 sendiri adalah versi dengan banyak perubahan yang dirilis akhir tahun 2008. Fokus dari Python 3 itu sendiri adalah untuk melakukan perapian pada codebase dan menghapuskan duplikasi (redundancy). Perubahan terbesar pada Python 3 termasuk memasukkan statemen print ke dalam built-in function. Awalnya, Python 3 mengalami hambatan pada pengadopsiannya. Itu akibat dari tidak adanya backwards compatibility dengan Python 2. Hal ini membuat pengguna Python sangat berat hati untuk pindah ke versi 3 ini. Tambahnya, banyak sekali library yang hanya tersedia untuk Python 2., tapi setelah tim pengembangan di balik Python 3 telah berulang kali menjelaskan bahwa dukungan terhadap Python 2 akan segera dihentikan, dan semakin banyak library disalin ke Python 3, maka penerapan Python 3 semakin lama semakin meningkat.

### **1.1.3 Implementasi dan penggunaan Python pada Perusahaan**

daftar berikut adalah beberapa perusahaan yang menggunakan Python, diantaranya:

1. Google adalah perusahaan besar yang menggunakan banyak kode Python di dalam mesin pencariannya. Dan mesin pencari google adalah yang paling terkenal di dunia.
2. Youtube, situs video terbesar dan terpopuler di dunia, sebagian besar kodenya ditulis dalam bahasa Python.
3. Facebook, media sosial terbesar di dunia, menggunakan Tornado, sebuah framework Python untuk menampilkan timeline.

4. Instagram, siapa yang tidak kenal. Instagram menggunakan Django, framework python sebagai mesin pengolah sisi server dari aplikasinya.
5. Pinterest, banyak menggunakan python untuk membangun aplikasinya.
6. Dropbox, barangkali Anda adalah salah seorang pengguna layanan ini. Drop-box menggunakan python baik di sisi server maupun di sisi pengguna layanannya.
7. Quora, salah satu situs tanya jawab terbesar di dunia, dibangun menggunakan Python.
8. NASA, badan antariksa Amerika ini menggunakan Python untuk bidang sainsnya.
9. NSA, badan mata – mata Amerika banyak menggunakan Python untuk analisa kriptografi dan intelijen.
10. Blender, Maya, software pembuat animasi 3D terkenal, menggunakan Python sebagai salah satu bahasa skrip pemrogramannya.
11. Raspberry Pi, komputer mini yang banyak digunakan sebagai mikrokontroller, menggunakan Python sebagai bahasa utamanya.

## 1.2 Instalasi

### 1.2.1 Cara Pemakaian Script dan interpreter python

### 1.2.2 Cara Pemakaian spyder termasuk variable explorer

## 1.3 Mencoba Python

Untuk memulai suatu pemrograman, kita akan awali dengan membuat sebuah hello world. Di Python, cukup mudah untuk membuat sebuah hello world. Silahkan buat sebuah file dengan nama helloworld.py kemudian buat kode berikut di dalam file tersebut:

```
print "Hello world..."
```

Sekarang mari kita eksekusi file tersebut di konsol dengan perintah berikut: python helloworld.py

```
Hello world...
```

## 1.4 Identasi

Ketika menulis kode program Python perlu memperhatikan indentasi, karena kode program Python distrukturkan berdasarkan indentasi. Kode program yang berada pada sisi kiri yang sama maka dibaca sebagai satu blok, untuk membuat sub blok maka cukup dengan memberikan jarak spasi atau tab ke kanan. Soal indentasi ini akan lebih jelas ketika pembahasan tentang pencabangan, perulangan, fungsi, class, dan materi yang lain yang membutuhkan penulisan kode program bersarang. Contohnya adalah sebagai berikut: `import sys`

```
if len(sys.argv) > 2:  
  
    print("Harap memasukkan argumen.")  
  
sys.exit(1)
```

## BAB 2

---

# PEMROGRAMAN DASAR

---

## 2.1 Teori

Praktek teori penunjang yang dikerjakan :

### 2.1.1 Variable, pemakaian Variable dan Jenis-Jenis Type data

Variabel merupakan tempat menyimpan data, sedangkan tipe data adalah jenis data yang terseimpan dalam variabel. Variabel bersifat mutable, artinya nilainya bisa berubah-ubah.

#### 1. Pemakaian Variabel

Variabel di python dapat dibuat dengan format seperti ini:

NamaVariabel = (nilai)

Contoh:

VariabelKu = "ini isi variabel"

variabel2 = 20

Kemudian untuk melihat isi variabel, kita dapat menggunakan fungsi print.  
print VariabelKu

print variabel2

### (a) Aturan Penulisan Variabel

- Nama variabel boleh diawali menggunakan huruf atau garis bawah (`_`), contoh: `nama`, `_nama`, `namaKu`, `nama_variabel`.
- Karakter selanjutnya dapat berupa huruf, garis bawah (`_`) atau angka, contoh: `__nama`, `n2`, `nilai1`.
- Karakter pada nama variabel bersifat sensitif (case-sensitif). Artinya huruf besar dan kecil dibedakan. Misalnya, `variabel_Ku` dan `variabel_ku`, keduanya adalah variabel yang berbeda.
- Nama variabel tidak boleh menggunakan kata kunci yang sudah ada dalam python seperti `if`, `while`, `for`, dsb.

### (b) Tipe Data

Cara mengisi nilai variabel ditentukan dengan jenis datanya, misalkan untuk tipe data teks (string) maka harus diapit dengan tanda petik ("`...`"). Sedangkan untuk angka (integer) dan boolean tidak perlu diapit dengan tanda petik.

### (c) Jenis-Jenis Tipe Data

- Boolean, Contoh `true` atau `false`
- String, Contoh `"Belajar Python"`
- Integer, Contoh 15 atau 1234
- Float, Contoh 2.5 atau 0.55
- List, Contoh `['abcd', 123, 1.5]`

## 2. Meminta input dan melakukan output

```
x = input("masukan nama: ")
```

```
print('Hallo, ' + x) #dengan perintah ini, akan menampilkan output
```

## 3. Operator dan Konvert

- Tambah contoh `x + y`
- Kurang contoh `x - y`
- Bagi contoh `x / y`
- Kali contoh `x * y`
- Modulus contoh `x`
- Pangkat `x ** y`
- equal contoh `x == y`
- not equal contoh `x != y`

- lebih besar dari contoh  $x > y$
- kurang dari  $x < y$
- Konvert string ke integer, contoh `x = int("123")`
- Konvert integer ke string, contoh `x = str(456)`

#### 4. Perulangan di Python

- Perulangan for  
 contoh :  
 ulang = 2  
 for i in range(ulang):  
     print ("Perulangan ke-" +str(i))  
 Hasil :  
 Perulangan ke-0  
 Perulangan ke-1
- Perulangan While  
 contoh :  
 jawab = 'ya'  
 hitung = 0  
 while(jawab == 'ya'):  
     hitung += 1  
     jawab = input("Ulang lagi tidak? ")  
 print ("Total perulangan: " + str(hitung))

#### 5. Kodisi di Python

- (a) **Kondisi If**  
 Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar True. Jika kondisi bernilai salah False maka statement/kondisi if tidak akan di-eksekusi. Dibawah ini adalah contoh penggunaan kondisi if pada Python
- ```
a = 33
b = 200
if b > a:
    print("b lebih besar dari a")
```
- (b) **Kondisi If Else** Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else. Dibawah ini adalah contoh penggunaan kondisi if else pada Python
- ```
a = 200
b = 33
if b > a:
    print("b lebih besar dari a")
else:
```

```
print("b bukan lebih besar dari a")
```

- (c) Kondisi **Elif** Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari “kondisi if”. Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi “else”, bedanya kondisi “elif” bisa banyak dan tidak hanya satu. Dibawah ini adalah contoh penggunaan kondisi elif pada Python

```
a = 33
b = 33
if b > a:
    print("b lebih besar dari a")
elif a == b:
    print("a sama dengan b")
```

## 6. Error yang sering dialami

- (a) Syntax Error, Cara mengatasinya dengan cara melihat kode dan mengecek kesalahan dalam penulisan.
- (b) Run-time Error, Cara mengatasinya mengecek file pada directory nya, dan memastikan file nya tidak ada yang terhapus.
- (c) Logical Error, Cara mengatasinya mengecek kode secara manual karena error tidak akan ternotice, tetapi akan terasa karena keluaran berbeda dengan yang diharapkan.

## 7. Cara memakai Try Except

Python menyediakan metode penanganan eksepsi dengan menggunakan pernyataan try dan except. Di dalam blok try kita meletakkan baris program yang kemungkinan akan terjadi error. Bila terjadi error, maka penanganannya diserahkan kepada blok except.

contoh :

```
try:
    print(x)
except:
    print("terjadi error bre ")
```

## 2.1.2 Ketrampilan Pemrograman

Buat program di python dengan ketentuan :

1. Jawaban

```

1 print("  **  **  *****  *****  *****  *****  *****")
2 print("*****  *****  *****  *****  *****  *****  *****")
3 print("  **  **  **  **  **  **  **  **  **  **  **  **")
4 print("  **  **  **  **  **  **  **  **  **  *****  *****")
5 print("  **  **  **  *****  *****  *****  *****  *****")
6 print("  **  **  **  *****  *****  *****  **  **  **")
7 print("  **  **  **  **  **  **  **  *****  *****")
8 print("  **  **  **  **  *****  *****  *****  *****")

```

## 2. Jawaban

```

1 npm = input("Masukkan NPM : ")
2 ulang = 1
3 while (ulang <= 83):
4     print("Halo, " + str(npm) + " apa kabar?")
5     ulang += 1

```

## 3. Jawaban

```

1 npm = input("Masukkan NPM : ")
2 ulang = 1
3 while (ulang <= 83):
4     print("Halo, " + str(npm[4:7]) + " apa kabar?")
5     ulang += 1

```

## 4. Jawaban

```

1 npm = input("Masukkan NPM : ")
2 print("Halo, " + str(npm[-3]) + " apa kabar?")

```

## 5. Jawaban

```

1 a = 1
2 b = 1
3 c = 7
4 d = 4
5 e = 0
6 f = 8
7 g = 3
8
9 npm = [a,b,c,d,e,f,g]
10
11 for n in npm:
12     print(n, end = "")
13
14 print()

```

## 6. Jawaban

```

1 print(a+b+c+d+e+f+g)

```

## 7. Jawaban

```

1 print(a*b*c*d*e*f*g)

```



## 8. Jawaban

```
1 for n in npm:
2     print(n)
```

## 9. Jawaban

```
1 for n in npm:
2     if (n % 2 == 0):
3         if (n != 0):
4             print(n, end = "")
```

## 10. Jawaban

```
1 for n in npm:
2     if (n % 2 != 0):
3         print(n, end = "")
```

## 11. Jawaban

```
1 for n in npm:
2     if (n % 2 != 0):
3         print(n, end = "")
4
5 print()
```

### 2.1.3 Ketrampilan Penanganan Error

Bagian Penanganan error dari script python.

## 1. Jawaban

- (a) Syntax Errors, adalah suatu keadaan saat kode python mengalami kesalahan penulisan. Solusinya adalah memperbaiki penulisan kode yang salah.
- (b) Zero Division Error, adalah exceptions yang terjadi saat eksekusi program menghasilkan perhitungan matematika pembagian dengan angka nol (0). Solusinya adalah tidak membagi suatu yang hasilnya nol.
- (c) Name Error, adalah exception yang terjadi saat kode melakukan eksekusi terhadap local name atau global name yang tidak terdefinisi. Solusinya adalah memastikan variabel atau function yang dipanggil ada atau tidak salah ketik.
- (d) Type Error, adalah exception yang terjadi saat dilakukan eksekusi terhadap suatu operasi atau fungsi dengan type object yang tidak sesuai. Solusinya adalah mengkonversi variabelnya sesuai dengan tipe data yang akan digunakan.

## 2. Jawaban

```
1 a = "7"
2 b = 7
3
4 try:
5     print(a+b)
6 except TypeError:
7     print("Beda tipe data")
```

