



# BAHASA PEMROGRAMAN PYTHON

Tim Penulis:

Siti Maesaroh | Afiyati | Lukman Hakim  
Yunita Sartika Sari | Mohamad Yusuf | Eza Budi Perkasa  
Wiranti Sri Utami | Norbertus Tri Suswanto Saptadi  
Siti Mutmainah | Khairunnas | Eka Purnama Harahap  
Zumhur Alamin | Inna Sabily Karima  
Andi Saputra | Roy Mubarak



# **BAHASA PEMROGRAMAN PYTHON**

**Siti Maesaroh  
Afiyati  
Lukman Hakim  
Yunita Sartika Sari  
Mohamad Yusuf  
Eza Budi Perkasa  
Wiranti Sri Utami  
Norbertus Tri Suswanto Saptadi  
Siti Mutmainah  
Khairunnas  
Eka Purnama Harahap  
Zumhur Alamin  
Inna Sabily Karima  
Andi Saputra  
Roy Mubarak**

# BAHASA PEMROGRAMAN PYTHON

## **Tim Penulis:**

Siti Maesaroh

Afiyati

Lukman Hakim

Yunita Sartika Sari

Mohamad Yusuf

Eza Budi Perkasa

Wiranti Sri Utami

Norbertus Tri Suswanto Saptadi

Siti Mutmainah

Khairunnas

Eka Purnama Harahap

Zumhur Alamin

Inna Sabily Karima

Andi Saputra

Roy Mubarak

**Editor** : Muhamad Rizal Kurnia, M.E., C.Ed.

**Tata Letak** : Asep Nugraha, S.Hum.

**Desain Cover** : Septimike Yourintan Mutiara, S.Gz.

**Ukuran** : UNESCO 15,5 x 23 cm

**Halaman** : ix, 201

**ISBN** : 978-623-8385-69-0

**Terbit Pada** : Mei 2024

**Anggota IKAPI** : No. 073/BANTEN/2023

## **Hak Cipta 2024 @ Sada Kurnia Pustaka dan Penulis**

*Hak cipta dilindungi undang-undang dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara apapun tanpa izin tertulis dari penerbit dan penulis.*

## **PENERBIT PT SADA KURNIA PUSTAKA**

Jl. Warung Selikur Km.6 Sukajaya – Carenang, Kab. Serang-Banten

Email : sadapenerbit@gmail.com

Website : sadapenerbit.com & repository.sadapenerbit.com

Telpon/WA : +62 838 1281 8431

# KATA PENGANTAR

Segala puji dan syukur kami panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya yang melimpah tanpa henti, sehingga kami dapat menyelesaikan buku berjudul **Bahasa Pemrograman Python**. Buku ini merupakan kolaborasi dari berbagai Penulis di bidang IT dari berbagai perguruan tinggi ternama di Indonesia. Penulis, seorang ahli dalam industri IT dan akademisi dibidang IT dengan pengalaman panjang, membimbing pembaca melalui berbagai tahapan siklus hidup pengembangan Bahasa pemrograman dengan penekanan pada praktik terbaik dan metodologi mutakhir.

Python adalah bahasa pemrograman tingkat tinggi yang dirancang dengan filosofi yang menekankan keterbacaan kode dan sintaks yang bersih. Dibuat oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991, Python telah berkembang menjadi salah satu bahasa pemrograman yang paling populer di dunia. Python adalah bahasa pemrograman yang sering digunakan dalam pengembangan aplikasi web, perangkat lunak, ilmu data, dan *machine learning* (ML). Para pengembang memilih Python karena efisiensinya, kemudahan belajar, dan kemampuannya berjalan di berbagai platform. Python dapat diunduh secara gratis, terintegrasi dengan baik ke semua jenis sistem, dan mempercepat proses pengembangan.

Buku ini disusun secara sistematis untuk memudahkan pembaca memahami konsep dasar hingga teknik-teknik lanjutan dalam pemrograman Python yang terdiri dari: pengenalan python, variabel dan tipe data, operator, fungsi, input dan output, struktur data lanjutan, *file handling*, paket standar python, pemrograman fungsional, *concurrency*, *GUI programming*, *web development*, *data science* dan *machine learning*, *testing* dan *debugging* serta penggunaan python untuk pengembangan aplikasi.

Kami sebagai Penulis merasa sangat bangga dan berbahagia dengan penerbitan buku ini, kami berharap buku ini dapat

memberikan manfaat dan inspirasi bagi banyak orang. Buku ini perlu untuk dimiliki, dijadikan referensi dan sebagai media pembelajaran dan penambah ilmu pengetahuan bagi para pembacanya. Namun demikian kami tetap berharap mendapatkan Masukan, Saran dan Kritik Membangun untuk Kesempurnaan buku ini. Akhir kata, kami semua mengucapkan terimakasih kepada semua pihak yang telah membantu dan mengapresiasi buku kami.

Tim Penulis

# DAFTAR ISI

<b>KATA PENGANTAR .....</b>	<b>iii</b>
<b>DAFTAR ISI .....</b>	<b>v</b>
<b>BAB 1 PENGENALAN PYTHON.....</b>	<b>1</b>
Definisi Bahasa Pemrograman Python.....	1
Instalasi Python .....	2
Baris Perintah Python.....	3
Cara Mengeksekusi Python.....	4
Identasi Python.....	8
Hal Penting yang Perlu Diingat dalam Penggunaan Python .....	8
Daftar Pustaka.....	10
Profil Penulis .....	11
<b>BAB 2 VARIABEL DAN TIPE DATA PYTHON.....</b>	<b>12</b>
Definisi Variabel dan Tipe Data Pada Python .....	12
Macam-macam Variabel pada Python .....	13
Daftar Pustaka.....	22
Profil Penulis .....	23
<b>BAB 3 OPERATOR .....</b>	<b>24</b>
Operator .....	24
Operator Aritmatika.....	24
Operator Perbandingan.....	25
Operator Penugasan.....	26
Operator Logika.....	27
Operator Bitwise.....	28
Operator Khusus .....	29
Daftar Pustaka.....	30
Profil Penulis.....	31
<b>BAB 4 FUNGSI.....</b>	<b>32</b>
Pengertian Fungsi.....	32
Fungsi dengan Argument dan Parameter .....	33
Fungsi dengan Mengembalikan Nilai .....	35
Fungsi yang Memanggil Dirinya Sendiri.....	38
Daftar Pustaka.....	41
Profil Penulis.....	42

<b>BAB 5 INPUT OUTPUT .....</b>	<b>43</b>
Pendahuluan .....	43
Jendela Interaktif ( <i>Interactive Window</i> ) .....	43
Dasar-dasar Input Output.....	44
Pengambilan Data ( <i>input</i> ).....	46
Format dan Output.....	52
Daftar Pustaka.....	60
Profil Penulis.....	61
<b>BAB 6 STRUKTUR DATA LANJUTAN.....</b>	<b>62</b>
Definisi Struktur Data.....	62
Jenis Struktur Data.....	62
<i>Stack</i> (Tumpukan).....	64
<i>Queue</i> (Antrian).....	66
<i>Tree</i> (Pohon).....	70
Daftar Pustaka.....	74
Profil Penulis.....	75
<b>BAB 7 FILE HANDLING.....</b>	<b>76</b>
<i>File Handling</i> .....	76
<i>File Handling</i> dengan Python .....	76
Kelebihan <i>File Handling</i> pada Python .....	77
Kekurangan <i>File Handling</i> Pada Python.....	77
Mode Akses Pada Python .....	78
<i>Text File</i> Pada Python.....	79
<i>File Open</i> Pada Python .....	80
<i>Read File</i> Pada Python .....	81
<i>Create dan Write File</i> Baru Pada Python.....	82
<i>Append File</i> pada Python .....	82
<i>File</i> Menjadi Input/Output Pada Python .....	83
<i>Read dan Write File</i> Pada Python .....	83
<i>Close File</i> pada Python .....	84
<i>Rename atau Delete File</i> pada Python .....	84
Daftar Pustaka.....	86
Profil Penulis.....	87
<b>BAB 8 PAKET STANDAR PYTHON .....</b>	<b>88</b>
Pendahuluan .....	88
Module .....	90

<i>Package</i> .....	91
<i>Library</i> .....	93
Penutup .....	100
Daftar Pustaka.....	102
Profil Penulis.....	103
<b>BAB 9 PEMROGRAMAN FUNSIONAL .....</b>	<b>104</b>
Pendahuluan .....	104
Dasar-dasar Pemrograman Fungsional .....	105
Pemrograman Fungsional dengan Python .....	107
Daftar Pustaka.....	112
Profil Penulis.....	113
<b>BAB 10 CONCURRENCY .....</b>	<b>114</b>
Apa Itu <i>Concurrency</i> ?.....	114
<i>Concurrent vs Sequential</i> .....	117
<i>Threads</i> dan <i>Multithreading</i> .....	122
<i>Concurrency</i> dengan <i>Asyncio</i> pada Python.....	123
Daftar Pustaka.....	125
Profil Penulis.....	126
<b>BAB 11 GUI PROGRAMMING.....</b>	<b>127</b>
Definisi GUI.....	127
Definisi Pemrograman atau <i>Programming</i> .....	128
Bahasa Pemrograman untuk <i>Programming</i> .....	128
Definisi <i>GUI Programming</i> .....	129
Evolusi <i>GUI Programming</i> .....	130
Jenis <i>GUI Programming</i> .....	130
Contoh <i>GUI Programming</i> .....	131
Manfaat <i>GUI Programming</i> .....	131
Aspek Lain dalam <i>GUI Programming</i> .....	132
Tantangan dalam <i>GUI Programming</i> .....	132
<i>GUI Programming</i> pada Phyton .....	133
Daftar Pustaka.....	138
Profil Penulis.....	139
<b>BAB 12 WEB DEVELOPMENT.....</b>	<b>140</b>
Pendahuluan .....	140
Pengenalan HTML dan CSS.....	141
Integrasi HTML dan CSS dengan Python .....	143

Flask: Framework Web Python.....	145
<i>Routing dan Template</i> dalam Flask .....	149
Interaksi Web dengan Database.....	150
Pengembangan Aplikasi Web dengan Django.....	152
MVC atau MTV dalam Django.....	158
<i>User Authentication dan User Authorization</i> .....	161
Daftar Pustaka.....	164
Profil Penulis.....	166
<b>BAB 13 DATA SCIENCE DAN MACHINE LEARNING .....</b>	<b>167</b>
Pengenalan <i>Data Science</i> .....	167
Mengapa Kita Memerlukan <i>Data Science</i> ?.....	168
Manfaat <i>Data Science</i> .....	170
Tahapan <i>Data Science</i> .....	172
Perbedaan antara <i>Data Science</i> dan <i>Business Intelligence</i> .....	174
Pengantar ke <i>Machine Learning</i> .....	175
Jenis <i>Machine Learning</i> .....	175
Aplikasi <i>Machine Learning</i> .....	176
Tantangan dalam <i>Machine Learning</i> .....	176
Daftar Pustaka.....	177
Profil Penulis.....	178
<b>BAB 14 TESTING DAN DEBUGGING .....</b>	<b>179</b>
Pendahuluan .....	179
<i>Testing dan Debugging</i> .....	181
Diagram Alur .....	184
<i>Debugging</i> .....	186
Daftar Pustaka.....	188
Profil Penulis.....	189
<b>BAB 15 PENGGUNAAN PYTHON UNTUK PENGEMBANGAN APLIKASI .....</b>	<b>190</b>
Pendahuluan .....	190
<i>Web Development</i> .....	190
<i>Data Science</i> .....	191
<i>Artificial Intelligence (AI) dan Machine Learning (ML)</i> .....	193
Desktop GUI Applications.....	194
<i>Game Development</i> .....	195

<i>Network Programming</i> .....	196
<i>Scripting</i> .....	197
<i>Internet of Things (IoT)</i> .....	198
Daftar Pustaka.....	200
Profil Penulis.....	201

# BAB 1

## PENGENALAN PYTHON

Siti Maesaroh, S.Kom., M.T.I.  
Universitas Mercu Buana

### Definisi Bahasa Pemrograman Python

Python adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh Guido van Rossum pada tahun 1991. Python didesain dengan filosofi yang menekankan keterbacaan kode dan sintaksis yang sederhana, sehingga membuatnya menjadi pilihan yang populer untuk pemula maupun pengembang berpengalaman.

Beberapa fitur utama Python adalah:

#### 1. Sintaksis yang Sederhana

Python menggunakan indentasi untuk menandai blok kode, membuatnya mudah dibaca dan dipahami. Berikut adalah contoh sintaks sederhana dalam Python untuk mencetak teks ke layar:

A screenshot of a code editor window with a dark background. The title bar at the top left says 'python' and at the top right there is a 'Copy code' button. The main area of the editor contains a single line of Python code: `print("Hello, World!")`. The code is color-coded: 'print' is in orange, the string is in green, and the parentheses and spaces are in white.

**Gambar 1.1: Print "Hello Word!"**

Sumber : Diolah Oleh Penulis

#### 2. Dukungan untuk Berbagai Platform

Python dapat dijalankan di berbagai sistem operasi seperti Windows, macOS, dan Linux.

### 3. Interpretasi

Python adalah bahasa pemrograman yang diinterpretasi, artinya kode Python dieksekusi baris per baris oleh interpreter Python.

### 4. Dukungan untuk Berbagai Paradigma Pemrograman

Python mendukung pemrograman berorientasi objek, pemrograman fungsional, dan pemrograman berbasis aturan.

### 5. Dinamis dan Tipe Data yang Kuat

Python adalah bahasa pemrograman dinamis, yang berarti Anda tidak perlu mendeklarasikan tipe data secara eksplisit saat membuat variabel. Python juga memiliki tipe data yang kuat, yang memungkinkan Anda bekerja dengan data secara efisien.

### 6. Memiliki Berbagai *Library* dan *Framework*

Python memiliki berbagai *library* dan *framework* yang luas, seperti Django untuk pengembangan web, NumPy untuk komputasi numerik, dan TensorFlow untuk kecerdasan buatan.

### 7. Mudah Diperluas

Python dapat diperluas dengan menulis ekstensi dalam bahasa pemrograman lain, seperti C atau C++. Ini memungkinkan pengembang untuk mengoptimalkan kinerja aplikasi Python jika diperlukan.

### 8. Komunitas yang Besar

Python memiliki komunitas pengembang yang besar dan aktif, yang berarti Anda dapat dengan mudah menemukan dukungan, tutorial, dan solusi untuk masalah yang mungkin Anda hadapi.

Python digunakan dalam berbagai bidang, seperti pengembangan web, analisis data, kecerdasan buatan, pengembangan perangkat lunak, dan lain-lain. Keunggulan Python dalam hal keterbacaan dan kegunaan membuatnya menjadi bahasa pemrograman yang sangat populer di kalangan pengembang.

## Instalasi Python

Instalasi Python dapat dilakukan dengan langkah-langkah berikut:

### 1. Unduh Python

Kunjungi situs web resmi Python di [python.org](https://python.org) dan unduh installer Python terbaru untuk sistem operasi Anda. Pilih versi Python yang sesuai dengan sistem operasi yang Anda gunakan (Windows, macOS, atau Linux).

## 8. Praktek dan Konsistensi

Seperti bahasa pemrograman lainnya, praktek yang konsisten akan membantu Anda menjadi lebih baik dalam menguasai Python. Cobalah untuk menulis kode yang bersih, mudah dibaca, dan mengikuti konvensi penamaan yang disarankan (PEP 8).

\*\*\*\*\*

## Daftar Pustaka

- Beazley, David M. "Python Essential Reference." Addison-Wesley Professional, 2009.
- Beazley, David, and Brian K. Jones. "Python Cookbook." O'Reilly Media, 2013.
- Dawson, Michael. "Python Programming for the Absolute Beginner." Course Technology PTR, 2010.
- Lutz, Mark. "Learning Python." O'Reilly Media, 2013.
- Matthes, Eric. "Python Crash Course: A Hands-On, Project-Based Introduction to Programming." No Starch Press, 2019.
- McKinney, Wes. "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython." O'Reilly Media, 2017.
- Pilgrim, Mark. "Dive Into Python 3." Apress, 2009.
- Sweigart, Al. "Automate the Boring Stuff with Python: Practical Programming for Total Beginners." No Starch Press, 2015.
- VanderPlas, Jake. "Python Data Science Handbook: Essential Tools for Working with Data." O'Reilly Media, 2016.
- Zelle, John. "Python Programming: An Introduction to Computer Science." Franklin, Beedle & Associates Inc, 2016.

## PROFIL PENULIS



### **Siti Maesaroh, S.Kom., M.T.I.**

Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2010 silam. Hal tersebut membuat penulis memilih untuk masuk ke Sekolah Perguruan Tinggi di bidang Sistem Informasi. Setelah lulus S1, 2 tahun kemudian, penulis menyelesaikan studi S2 di prodi Teknik Informatika Program Pasca Sarjana Universitas Raharja. Penulis memiliki kepakaran dibidang *Web Technology dan Data Science*. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti di bidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI. Selain peneliti, penulis juga aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini. Dengan tujuan mengembangkan solusi perangkat lunak inovatif yang memecahkan tantangan bisnis dengan menggunakan keterampilan pemrograman dan pengetahuan teknologi terkini. Terus meningkatkan kemampuan teknis dan kepemimpinan melalui pengalaman praktis dan pembelajaran berkelanjutan.

Email Penulis: [mays41946@gmail.com](mailto:mays41946@gmail.com).

# BAB 2

# VARIABEL DAN TIPE DATA

# PYTHON

**Dr. Afiyati, S.Si., M.T.**  
Universitas Mercu Buana

## **Definisi Variabel dan Tipe Data Pada Python**

Variabel dalam Python adalah nama yang digunakan untuk menetapkan sebuah nilai atau objek. Secara teknis, variabel adalah referensi atau label yang menunjuk ke suatu nilai di dalam memori komputer. Anda dapat memberikan nama apa pun pada variabel, asalkan mengikuti aturan penamaan variabel dalam Python. Variabel digunakan untuk menyimpan data yang dapat diakses dan diubah selama program berjalan.

Di Python, Anda tidak perlu mendeklarasikan tipe data variabel secara eksplisit seperti pada beberapa bahasa pemrograman lainnya. Python menggunakan konsep "*dynamic typing*", yang berarti tipe data variabel ditentukan secara otomatis berdasarkan nilai yang Anda berikan ke variabel tersebut.

Berikut adalah contoh penggunaan variabel dalam Python:

```
python Copy code

# Menetapkan nilai ke variabel
nama = "Alice"
umur = 30
berat_badan = 60.5

# Mengakses nilai variabel
print("Nama:", nama)
print("Umur:", umur)
print("Berat Badan:", berat_badan)

# Mengubah nilai variabel
umur = 31
print("Umur baru:", umur)
```

**Gambar 2.1: Variable pada Python**

Sumber : Dibuat oleh penulis

## Macam-macam Variabel pada Python

Di Python, terdapat beberapa jenis variabel yang dapat digunakan, tergantung pada jenis data yang ingin disimpan. Beberapa jenis variabel umum dalam Python antara lain:

### 1. Variabel String

Variabel string yang digunakan untuk menyimpan teks. Berikut Contohnya:

```
python Copy code

nama = "Alice"
```

**Gambar 2.2: Variabel String**

Sumber : Dibuat Oleh Penulis

String dalam Python dapat ditulis di dalam tanda kutip tunggal (' '), tanda kutip ganda (" "), atau tanda kutip tiga (""" atau '''). Berikut adalah contoh penggunaan variabel string dalam Python:

## Daftar Pustaka

- Beazley, David M. "Python Essential Reference." Addison-Wesley Professional, 2009.
- Beazley, David, and Brian K. Jones. "Python Cookbook." O'Reilly Media, 2013.
- Dawson, Michael. "Python Programming for the Absolute Beginner." Course Technology PTR, 2010.
- Downey, A. (2015). Think Python: How to Think Like a Computer Scientist. O'Reilly Media.
- Matthes, Eric. "Python Crash Course: A Hands-On, Project-Based Introduction to Programming." No Starch Press, 2019.
- McKinney, W. (2018). Python for Data Analysis. O'Reilly Media.
- McKinney, Wes. "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython." O'Reilly Media, 2017.
- Pilgrim, Mark. "Dive Into Python 3." Apress, 2009.
- Sweigart, A. (2015). Automate the Boring Stuff with Python. No Starch Press.
- Sweigart, Al. "Automate the Boring Stuff with Python: Practical Programming for Total Beginners." No Starch Press, 2015.
- VanderPlas, Jake. "Python Data Science Handbook: Essential Tools for Working with Data." O'Reilly Media, 2016.
- Zelle, John. "Python Programming: An Introduction to Computer Science." Franklin, Beedle & Associates Inc, 2016.
- Ziadé, T. (2008). Expert Python Programming. Packt Publishing.

## PROFIL PENULIS



### **Dr. Afiyati, S.Si., M.T.**

Lahir di Pekanbaru, 16 Oktober 1969 penulis menyelesaikan studi S1 di Universitas Gadjah Mada prodi Ilmu Komputer pada tahun 1988 – 1994 dan S2 di prodi Magister Teknik Elektro Universitas Mercu Buana, kemudian melanjutkan S3 di Universitas Gadjah Mada prodi Doktoral Ilmu Komputer Sebagai penulis yang berpengalaman dalam bidang teknologi dan pendidikan, saya percaya bahwa Python adalah bahasa yang sangat menarik untuk dieksplorasi dan dipelajari oleh pemula maupun profesional dalam dunia pemrograman.

Buku saya akan fokus pada pengantar yang komprehensif tentang Python, dimulai dari konsep dasar seperti variabel, tipe data, dan struktur kontrol, hingga topik yang lebih lanjut seperti pemrograman berorientasi objek, pemrosesan file, dan pengembangan web. Saya akan menggunakan pendekatan yang ramah dan mudah dipahami, dengan penjelasan yang jelas dan contoh kode yang relevan untuk membantu pembaca memahami konsep-konsep tersebut.

Selain itu, saya juga akan menyediakan latihan-latihan praktis dan proyek-proyek mini untuk melatih pembaca dalam menerapkan pengetahuan yang telah mereka pelajari. Saya ingin buku saya menjadi sumber belajar yang berguna bagi siapa saja yang tertarik untuk memahami dan menguasai Python, baik untuk tujuan akademis maupun profesional. Dengan menulis buku ini, saya berharap dapat menyebarkan pengetahuan tentang Python dan memberikan kontribusi positif bagi komunitas pemrograman secara luas. Saya berkomitmen untuk menyajikan materi yang akurat, relevan, dan terkini sesuai dengan perkembangan terbaru dalam dunia pemrograman Python.

Email Penulis: [afiyati.reno@mercubuana.ac.id](mailto:afiyati.reno@mercubuana.ac.id)

# BAB 3

## OPERATOR

Lukman Hakim, ST., M.Kom.  
Universitas Mercu Buana

### Operator

Operator merupakan simbol-simbol yang sudah diketahui kompil器和 digunakan untuk operasi perhitungan aritmatika dan logika, serta nilai yang dikenal untuk operasi. Ada 6 jenis operator yang digunakan dalam pemrograman Python yaitu Operator Aritmatika, Operator Perbandingan/ Relasi, Operator Penugasan, Operator Logika, Operator Bitwise dan Operator Khusus. (Kadir, 2019)

### Operator Aritmatika

Operator aritmatika merupakan operator untuk operasi perhitungan (Kalb, 2022), (Kadir, 2019). Operator aritmatika terdiri dari penjumlahan, perkalian, pembagian, sisa bagi, pengurangan, pemangkatan. Untuk operator aritmatika dapat dilihat pada tabel 3.1 Operator Aritmatika sebagai berikut (Suryantara, 2023):

Tabel 3.1 : Operator Aritmatika

Operator	Simbol	Contoh
Penjumlahan	+	$X + Y$ or $+2$
Perkalian	*	$X * Y$
Pembagian	/	$X / Y$
Sisa bagi	%	$X \% Y$
Pengurangan	-	$X - Y$ or $-2$
Pemangkatan	**	$X ** Y$

Sumber : (Kadir, 2019)

## Contoh koding :

```
print('Penjumlahan: ', 1 + 4)
print('Pengurangan: ', 4 - 1)
print('Perkalian: ', 2 * 4)
print ('Pembagian: ', 4 / 2)
print('Pembagian: ', 6 / 2)
print('Pembagian: ', 7 / 2)
print(' Pembagian : ', 7 // 2)
print('Modulus: ', 3 % 2)
print (' Pembagian : ', 7 // 3)
print('Pangkat: ', 3 ** 2)
```

## Hasil :

```
Penjumlahan: 5
Pengurangan: 3
Perkalian: 8
Pembagian: 2.0
Pembagian: 3.0
Pembagian: 3.5
Pembagian: 3
Modulus: 1
Pembagian: 2
Pangkat: 9
```

## Operator Pembandingan

Operator pembandingan digunakan untuk membandingkan 2 nilai. Operator pembandingan juga digunakan untuk kondisi operasi logika dengan menghasilkan nilai *true* atau *false* (Kadir, 2019), (Parmar, 2024). Untuk operator pembandingan dapat dilihat pada tabel 3.2. Operator Pembandingan/Relasi sebagai berikut :

**Tabel 3.2 : Operator Pembandingan atau Relasi**

Operator	Simbol	Contoh
lebih Besar	>	A > B
lebih Kecil	<	A < B
sama dengan	==	A == B
tidak sama dengan	!=	A != B
lebih besar sama dengan	>=	A >= B
lebih kecil sama dengan	<=	A <= B

Sumber : (kadir, 2019)

## Daftar Pustaka

- Kadir, A. (2019). *Logika Pemrograman Python*. Jakarta: Elex Media Komputindo.
- Kalb, I. (2022). *Object-Oriented Python*. San Francisco: William Pollock, Library of Congress Cataloging-in-Publication Data.
- Parmar, D. M. (2024, April 19). *Python Introduction*. Retrieved from github:  
[https://github.com/milaan9/01\\_Python\\_Introduction/blob/main/012\\_Python\\_Operators.ipynb](https://github.com/milaan9/01_Python_Introduction/blob/main/012_Python_Operators.ipynb)
- Suryantara, I. G. (2023). *Python :Bahasa Pemrograman Era Digital*. Jakarta: Elex Media Komputindo.

## PROFIL PENULIS



### **Lukman Hakim, S.T., M.Kom.**

Lahir di Desa Dadap, Kab Tangerang, 27 Oktober 1977. Jenjang Pendidikan S1 Teknik Komputer ditempuh di Universitas Yarsi, Kota Jakarta lulus tahun 2001. Pendidikan S2 Ilmu Komputer, lulus tahun 2004 di Universitas Bunda Mulia. Saat ini menjabat sebagai Kepala Laboratorium Fakultas Ilmu Komputer di Universitas Mercu Buana.

Tahun 2010-2021 pernah menjabat sebagai Ketua Program Studi Teknik Informatika Universitas Bunda Mulia, pernah sebagai tenaga ahli untuk proyek pengembangan Sistem Informasi pengolahan usaha hasil hutan (SIPUHH) tahun 2008, Kalimantan Timur Kementerian Kehutanan. Penulis merupakan aktif sebagai reviewer, editor jurnal, penulis dan peneliti. Ketertarikan penulis pada bidang rekayasa perangkat lunak dan perangkat keras seperti sensor. Pengalaman mengajar atau dosen di berbagai perguruan tinggi seperti Universitas Mercu Buana Jakarta, Universitas Bundamulia, Universitas Esa Unggul, Universitas Bina Nusantara, Kalbis.

Email: lhakim2710@gmail.com atau [lukman\\_hakim@mercubuana.ac.id](mailto:lukman_hakim@mercubuana.ac.id)  
WA : 081314410170.

# BAB 4

# FUNGSI

**Yunita Sartika Sari**  
Universitas Mercu Buana

## **Pengertian Fungsi**

Fungsi di Python adalah sekumpulan perintah atau baris kode yang dikelompokkan bersama untuk dapat dipanggil atau digunakan secara berulang. Sebuah fungsi dapat menerima parameter, mengembalikan nilai, dan dapat dipanggil secara independen. Dengan menggunakan fungsi, program besar dapat dibagi menjadi bagian-bagian kecil dengan tugas yang terpisah. Ini membuat kode program lebih dapat digunakan kembali (*reusable*) dan lebih terstruktur. Berikut adalah contoh sintaksis fungsi:

Fungsi di Python memiliki sintaksis umum yang terbagi menjadi empat bagian:

1. Kata kunci "def" yang menandakan bahwa blok kode adalah sebuah fungsi.
2. Nama fungsi yang dibuat.
3. Parameter yang akan diterima oleh fungsi (opsional).
4. Blok kode fungsi yang berisi perintah-perintah yang akan dilakukan oleh fungsi.

Sebagai contoh, sebuah fungsi sederhana bernama `"halo_people()"` akan mencetak "Halo python! Halo people!":

```
def hi_people():  
    print('Hi python! Hi people!')
```

Untuk memanggil fungsi di atas, kita dapat menggunakan perintah "`print()`" seperti berikut:

```
hi_people()
```

Output yang dihasilkan adalah:

```
Hi python! Hi people
```

Fungsi tersebut dapat dipanggil berkali-kali seperti ini:

```
hi_people()
```

```
hi_people()
```

```
hi_people()
```

Outputnya akan menjadi:

```
Hi python! Hi people
```

```
Hi python! Hi people
```

```
Hi python! Hi people
```

## Fungsi dengan Argument dan Parameter

Selain itu, sebuah fungsi juga dapat menerima parameter atau argumen. Ini adalah nilai atau variabel yang diberikan ke fungsi untuk diproses lebih lanjut. Sebagai contoh:

```
def selamat_datang(nama):
    print(f'Halo {nama}, selamat datang!')
```

Kemudian kita bisa memanggil fungsi tersebut dengan parameter yang berbeda:

```
selamat_datang('Syila')
```

```
selamat_datang('Lela')
```

Dengan menggunakan fungsi dan parameter ini, kita dapat menghasilkan output yang sama tanpa perlu menggunakan perintah "`print()`" secara berulang.

### 1. Parameter Wajib

Parameter dalam Python dapat berjumlah lebih dari satu, dan bisa menjadi wajib atau opsional. Misalkan, dalam fungsi berikut:

```
def perkenalan(nama, asal):
    print(f"Perkenalkan saya {nama} dari {asal}")
```

Jika dipanggil dengan dua parameter:

```
perkenalan("Yunita Sartika", "Jakarta")
```

## Fungsi

```
# Panggil beberapa kali untuk mensimulasikan
# cara kerja
tampilkanAngka(3)
tampilkanAngka(3, 2)
tampilkanAngka(3, 3)
```

Jika dijalankan, kita akan mendapatkan output:

```
Perulangan ke 1
Perulangan ke 2
Perulangan ke 3
```

### c. Langkah 3: Buat fungsi rekursif

Berikut adalah implementasi dalam bentuk kode program:

```
def tampilkanAngka(batas, i=1):
    print(f'Perulangan ke {i}')
    if i < batas:
        # ini adalah bagian yang membuatnya
        rekursif
        tampilkanAngka(batas, i + 1)
# Memanggil fungsi tampilkanAngka untuk pertama
kali
tampilkanAngka(10)
```

Output :

```
Perulangan ke 1
Perulangan ke 2
Perulangan ke 3
...
Perulangan ke 8
Perulangan ke 9
Perulangan ke 10
```

\*\*\*\*\*

## Daftar Pustaka

Irsyad, R. (2018). Penggunaan Python Web Framework Flask Untuk Pemula. Laboratorium Telematika, Sekolah Teknik Elektro & Informatika, Institut Teknologi Bandung.

<https://jagongoding.com/python/dasar/fungsi-rekursif/>

<https://jagongoding.com/python/dasar/fungsi/>

<https://github.com/jagongoding-com/python-dasar>

## PROFIL PENULIS



### **Yunita Sartika Sari**

Lahir di Tangerang, 9 Juni 1989. Saat ini, bekerja sebagai dosen program studi Sistem Informasi di Universitas Mercu Buana Jakarta. Penulis Menyelesaikan S1 Sistem Informasi pada tahun 2013 dan S2 Sistem Informasi pada tahun 2016 di Universitas Budi Luhur. Selain menjadi dosen, penulis juga sebagai instruktur di beberapa perusahaan dan di Kominfo. Penulis juga merupakan anggota IASPRO Wilayah Banten sebagai Asesor dari LSP P1 Universitas Mercu Buana. Beberapa karya ilmiah telah diterbitkan dalam jurnal nasional, prosiding dan jurnal internasional dengan tema Sistem Informasi yang terindeks Google Scholar dan Scopus. Beberapa buku juga sudah diterbitkan di penerbit berskala nasional. Penulis dapat dihubungi melalui email: [yunita.sartika@mercubuana.ac.id](mailto:yunita.sartika@mercubuana.ac.id)

# BAB 5

## INPUT OUTPUT

**Mohamad Yusuf, S.Kom., M.C.S.**

Universitas Mercu Buana

### **Pendahuluan**

Input dan output (I/O) merupakan aspek penting dalam pengembangan program, terutama karena interaksi dengan user sering kali diperlukan. Program yang baik harus secara efektif mengkomunikasikan masukan apa pun dalam Python dari user dan menampilkan hasilnya. User dapat memberikan input program secara manual dari keyboard atau menggunakan data dari sumber eksternal, seperti file atau database. Python menawarkan kepada kita beberapa fungsi bawaan yang memudahkan pembuatan program dengan cepat.

### **Jendela Interaktif (*Interactive Window*)**

Jendela interaktif IDLE berisi shell Python, yang merupakan antarmuka pengguna tekstual yang digunakan untuk berinteraksi dengan bahasa Python. Anda dapat mengetik sedikit kode Python ke dalam jendela interaktif dan tekan `Enter` untuk segera melihat hasilnya. Oleh karena itu namanya jendela interaktif. Jendela interaktif terbuka secara otomatis saat Anda memulai IDLE (David Amos, 2020).

Anda akan melihat teks berikut dan ditampilkan di bagian atas jendela:

## 6. Implementasi Input dan Print

### a. Contoh 1

Menggunakan metode `split()` bersamaan dengan fungsi `input()`. Metode `split()` akan memisahkan *string* berdasarkan spasi dan menghasilkan *list*.

```
a, b = input("Masukkan dua angka: ").split()
```

### b. Contoh 2

```
while True:
    Name = input("Masukan Nama Mahasiswa : ")
    Name = Name.title()
    if Name.replace(' ', '').replace('.', '').replace("'", '').isalpha():
        break
    else:
        print("Nama hanya boleh mengandung huruf, spasi, titik, dan apostrof..")
while True:
    Major = input("Jurusan Mahasiswa: ")
    Major = Major.capitalize()
    if Major in ["Manajemen", "Accounting", "Infomatika"]:
        break
    else:
        print("Masukan jurusan yang valid")
while True:
    Ttl_GPA = input("Masukan Nilai IPK: ")
    if Ttl_GPA.isdigit() and int(Ttl_GPA) > 0 and int(Ttl_GPA) <= 564:
        break
    else:
        print("IPK total harus berupa bilangan bulat antara 1 dan 564.")
while True:
    Ttl_Credit = input("Masukkan Total Kredit: ")
    if Ttl_Credit.isdigit() and int(Ttl_Credit) > 0 and int(Ttl_Credit) <= 141:
        break
    else:
        print("Total Kredit harus berupa bilangan bulat antara 1 dan 141.")
```

## Daftar Pustaka

- Bernd Klein. (2023, November 8). *Formatted Output*. <https://Python-Course.Eu/>.
- David Amos, D. B. J. J. F. H. (2020). *Python Basics: Vol. ISBN: 9781775093329* (Dan Bader;Joanna Jablonski, Ed.; 4th ed.).
- Julien Harbulot. (2018, December 27). *How and when to use stdout and stderr?* <https://Julienharbulot.Com/>.
- Nick Parlante. (2020a). *input()*. <https://Cs.Stanford.Edu/>.
- Nick Parlante. (2020b). *print() and Standard Out*. <https://Cs.Stanford.Edu/>.
- Pavan Vadapalli. (2024, February 28). *Print In Python (Introduction)*. [Upgrad.Com](https://Upgrad.Com/).
- Refsnes Data. (2024). *Python String format() Method*. <https://Www.W3schools.Com/>.

## PROFIL PENULIS



### **Mohamad Yusuf, S.Kom., M.C.S.**

Lahir di Jakarta, tahun 1976. Pendidikan S1 didapatkan dari Universitas Budi Luhur Jakarta, jurusan Teknik Informatika. Pendidikan S2 didapatkan dari Preston University, Islamabad Pakistan Jurusan Informatika. Penulis memiliki keahlian yang mendalam dalam Web Technology dan Data Science. Sebagai bagian dari upaya membangun karir sebagai dosen profesional, penulis aktif terlibat dalam riset di bidang spesialisasinya. Beberapa penelitian yang telah dilakukan didanai oleh lembaga perguruan tinggi tempat penulis mengajar. Selain itu, penulis juga rajin menulis buku dengan tujuan memberikan kontribusi positif bagi kemajuan bangsa dan negara tercinta.

Email Penulis: [mhd.yusuf@hotmail.com](mailto:mhd.yusuf@hotmail.com)

# BAB 6

## STRUKTUR DATA LANJUTAN

**Eza Budi Perkasa, M.Kom.**  
Institut Sains dan Bisnis Atma Luhur

### Definisi Struktur Data

Struktur data merupakan suatu abstraksi model penyimpanan, pengaturan, atau susunan data di dalam memori komputer. Struktur data dapat juga diartikan sebagai kumpulan dari suatu variabel yang dapat dinyatakan dengan sebuah nama. Setiap variabel dapat memiliki tipe yang bervariasi. Struktur data umumnya digunakan untuk mengelompokkan sejumlah informasi yang berhubungan menjadi suatu kesatuan. Struktur data merupakan salah satu dari tiga komponen penyusun program komputer. Komponen lainnya adalah algoritma dan bahasa pemrograman (Setyaningsih, 2012). Struktur data terbagi menjadi dua kelompok: Struktur data sederhana dan struktur data majemuk.

### Jenis Struktur Data

Struktur data memiliki banyak jenis. Struktur tersebut umumnya dibuat dari penggabungan tipe data elementer (Seymour, 2014). Beberapa contoh jenis struktur data adalah sebagai berikut.

#### 1. *Array* (Larik)

*Array* merupakan rangkaian elemen dengan urutan yang telah ditetapkan. Setiap elemen pada *array* dapat bertipe sama ataupun berbeda, tergantung dari bahasa pemrograman yang digunakan. Indeks digunakan untuk menunjukkan elemen yang akan diakses.

```
        if vertex[i] != '-':
            print(vertex[i], end=" ")
        else:
            print("  ") # 2x spasi
else:
    print("Daftar vertex tak dapat membentuk tree.")
```

Pada program tersebut, penyebutan daftar simpul harus diawali dengan spasi karena menandakan elemen yang tidak digunakan pada indeks ke-0. Selain itu juga, terdapat tanda *dash* yang menyatakan *null link*. Terdapat juga variabel validasi untuk memeriksa apakah daftar simpul tersebut dapat membentuk *binary tree* atau tidak.

\*\*\*\*\*

### Daftar Pustaka

- Setyaningsih, E. (2021). *Struktur Data*. Yogyakarta: AKPRIND PRESS.
- Seymour, L. (2014). *Data Structures*. New Delhi: McGraw Hill Education.

## PROFIL PENULIS



### **Eza Budi Perkasa, M.Kom.**

Penulis merupakan salah satu dosen di Institut Sains dan Bisnis Atma Luhur program studi Teknik Informatika sejak tahun 2016. Penulis mulai tertarik di dunia komputer sejak SD dan mulai mempelajarinya lebih dalam sejak SMA melalui pelatihan Olimpiade Sains Nasional bidang Informatika pada tahun 2008. Ketertarikan tersebut membuat penulis melanjutkan pendidikannya ke Sekolah Tinggi Manajemen Informatika dan Komputer Atma Luhur untuk jenjang Strata 1 program studi Teknik Informatika pada tahun 2010. Saat itu juga, penulis berhak memperoleh beasiswa untuk melanjutkan ke Universitas Budi Luhur jenjang Strata 2 program studi Magister Ilmu Komputer pada tahun 2014.

Penulis memiliki kepakaran di bidang Kecerdasan Tiruan dan Algoritma. Penelitian yang dilakukan didanai oleh internal perguruan tinggi dan juga Kementerian Riset, Teknologi, dan Pendidikan Tinggi. Penulis juga memiliki prestasi lainnya berupa peraih medali emas di Olimpiade Nasional 2023 jenjang Umum bidang Matematika, medali perunggu di Olimpiade Rihand Creative Guru 2023 bidang TIK, dan finalis di Olimpiade Rihand Creative Guru 2023 bidang Matematika. Saat ini, penulis juga menduduki jabatan sebagai Sekretaris di Asosiasi Pendidikan Tinggi Informatika dan Komputer (APTIKOM) Provinsi Kepulauan Bangka Belitung periode 2021-2025.

Email Penulis: [ezabudiperkasa@atmaluhur.ac.id](mailto:ezabudiperkasa@atmaluhur.ac.id)

# BAB 7

## *FILE HANDLING*

**Wiranti Sri Utami, S.Kom., M.T.I.**  
Universitas Cendekia Abditama

### ***File Handling***

*File handling* merupakan suatu bagian satu konsep dalam pemrograman (Dancuk, 2022). *File handling* adalah suatu landasan dari pemrograman python yang dipergunakan untuk membaca *file* dari segmen yang sangat penting. Baik berupa *file* teks, CSV maupun dengan data JSON (Ramuglia, 2023).

### ***File Handling dengan Python***

*File handling* pada Python merupakan suatu cara yang tepat dalam melakukan bermacam penanganan operasi *file* dan membutuhkan suatu penanganan yang tepat dalam menulis berbagai program menggunakan bahasa python dengan menentukan kode program yang aman, terkendali dan bekerja dengan baik (GeeksforGeeks, 2023). *File handling* pada python dapat dipermudah dengan suatu cara meliputi tahapan pembuatan *file*, membuka *file* dan menutup *file* (Dancuk, 2022).

*File handling* di python adalah suatu kegiatan yang sangat penting dilakukan dalam menggunakan aplikasi berbasis web. Kegiatan yang dilakukan dapat berupa membuka *file* hingga mengontrol *file* menggunakan suatu mode akses. Hal ini dapat memberikan penjelasan terhadap *file handling* setelah *file* dibuka (Fagbuyiro, 2022).

## Kelebihan *File Handling* pada Python

Menurut GeeksforGeeks, 2023. Kelebihan *file handling* pada python terdiri dari sebagai berikut :

### 1. Keserbagunaan

*File handling* pada python dapat memberikan berbagai kemungkinan yang ada yaitu mendukung operasi seperti untuk membuat, menulis, membaca *file*, menambahkan, mengganti nama *file* hingga dapat menghapus *file*.

### 2. Fleksibilitas

*File handling* pada python bersifat sangat fleksibel, hal ini dikarenakan dapat bekerja melalui berbagai macam jenis dari *file* yang ada seperti *file* teks, *file* biner, *file* CSV.

### 3. Ramah Pengguna

Pada python mampu memfasilitasi *user interface* yang ramah bagi pengguna dalam melakukan *file handling*, dimana pengguna dapat dengan mudah melakukan berbagai manipulasi *file*.

### 4. Lintas Platform

*File handling* pada python memiliki fungsi yang dapat bekerja pada berbagai jenis dari platform seperti Windows, Mac, maupun Linux, dan dapat memperbolehkan integrasi dengan memiliki kompatibilitas yang baik.

## Kekurangan *File Handling* Pada Python

Menurut GeeksforGeeks, 2023. Kekurangan *file handling* pada python terdiri dari sebagai berikut :

### 1. Rawan Kesalahan

Operasi pada *file handling* pada python dapat memiliki kerentanan terhadap kesalahan, terutama pada kode yang tidak hati-hati ditulis, maupun apabila terdapat kendala dari sistem *file* seperti perizinan *file*, maupun saat kunci *file*, dan lain sebagainya.

### 2. Resiko Keamanan

*File handling* pada python memiliki kekurangan dalam memicu resiko keamanan, terutama apabila suatu program menerima *input* dari pengguna yang sedang melakukan pengaksesan atau sedang melakukan modifikasi suatu *file* yang sifatnya sensitif pada suatu sistem.

Berdasarkan sintaks di atas pengguna yang memiliki *file* yang sudah ada dengan nama *file* yaitu *cobatest.txt*, maka pengguna dapat mengganti nama *file* dengan *test2.txt*. Pengguna juga dapat mengganti nama *file* dengan diberikan suatu lokasi *file* yang lebih spesifik seperti contoh sintaks berikut ini (Maulid, 2022) :

```
Import os
Os.rename("C:/Documents/Python/cobatest.txt", "C:/Documents/Python/test2.txt")
```

Pada sintaks di atas pengguna dapat melihat perubahan nama *file* yang terjadi. Nama *file* sebelumnya bernama *cobatest.txt*, setelah dilakukan *rename file* menggunakan fungsi *rename*, maka nama *file* telah dirubah menjadi *test2.txt* yang berada di lokasi direktorat data C di dalam folder Documents dan dalam folder Python.

\*\*\*\*\*

## **Daftar Pustaka**

- Dancuk, M. (2022, 02 24). *File handling in Python: Create, Open, Append, Read, Write*. Retrieved from [www.phoenixnap.com](http://www.phoenixnap.com): <https://phoenixnap.com/kb/file-handling-in-python>.
- Fagbuyiro, D. (2022, 08 26). *File handling in Python How to Create, Read, and Write to a File*. Retrieved from [www.freecodecamp.org](http://www.freecodecamp.org): <https://www.freecodecamp.org/news/file-handling-in-python/>.
- Geeksforgeeks.org. (2023, 12 26). *File handling in Python*. Retrieved from [www.geeksforgeeks.org](http://www.geeksforgeeks.org): <https://www.geeksforgeeks.org/file-handling-python/>.
- Geeksforgeeks.org. (2024, 04 04). *Open a File in Python*. Retrieved from [www.geeksforgeeks.org](http://www.geeksforgeeks.org): <https://www.geeksforgeeks.org/open-a-file-in-python/>.
- Maulid, Reyvan. (2022, 12 27). *Alternatif Mode Operasi Python untuk Handling File*. Retrieved from [www.dqlab.id](http://www.dqlab.id): <https://dqlab.id/alternatif-mode-operasi-python-untuk-handling-file>.
- Pankaj. (2022, 08 04). *Python File Operations Read and Write to files with Python*. Retrieved from [www.digitalocean.com](http://www.digitalocean.com): <https://www.digitalocean.com/community/tutorials/python-read-file-open-write-delete-copy>.
- Ramuglia, G. (2023, 08 28). *Python Open File Quick Start Guide*. Retrieved from [www.ioflood.com](http://www.ioflood.com): <https://ioflood.com/blog/python-open-file/>.
- Rumanti, Emmanuella. (2019). *Membaca dan Menulis File dengan Python*. Retrieved from [www.belajarpython.com](http://www.belajarpython.com): <https://belajarpython.com/2019/10/membaca-menulis-file-python/>

## PROFIL PENULIS



### **Wiranti Sri Utami, S.Kom., M.T.I.**

Penulis sangat tertarik dengan bidang ilmu komputer bermula dari tahun 2005, penulis masuk ke SMA Negeri 6 Kota Tangerang mengambil Jurusan Ilmu Pengetahuan Alam (IPA) dan berhasil lulus pada tahun 2008. Kemudian penulis melanjutkan pendidikan pada program Profesional IT di CEP-CCIT FTUI, Kota Depok dengan Jurusan Software Engineering hingga menyelesaikan program studi pada tahun 2010, selanjutnya pada tahun 2016 penulis berhasil menyelesaikan pendidikan Sarjana (S1) di STMIK Raharja, Tangerang dengan Program Studi Teknik Informatika. Dalam tiga tahun berikutnya penulis berhasil menyelesaikan pendidikan Pascasarjana Strata 2 (S2) Magister Teknik Informatika di Universitas Raharja, Tangerang dengan konsentrasi Business Intelligence. Penulis mengampu mata kuliah Algoritma dan Pemrograman, Sistem Berbasis Pengetahuan, Pemrograman Visual, Rekayasa Perangkat Lunak, Mobile Computing. Penulis mempunyai kepakaran dalam bidang Teknologi Web Data Scientist. Dalam mewujudkan karir sebagai dosen profesional, penulis aktif menulis artikel yang dipublikasikan pada jurnal ilmiah dan penulis pun aktif menulis buku, dengan tujuan ilmu yang dimiliki penulis sesuai dengan bidang kepakarannya mampu memberikan kontribusi yang positif untuk bangsa dan negara yang sangat tercinta ini.

Email Penulis : whiranty68@gmail.com

# BAB 8

## PAKET STANDAR PYTHON

**Dr. Ir. Norbertus Tri Suswanto Saptadi, S.Kom., M.T., M.M., IPM.**  
Universitas Atma Jaya Makassar

### **Pendahuluan**

Python *package* atau paket python merupakan suatu himpunan dari berbagai modul yang teratur, terstruktur, dan sistematis melalui direktori sehingga tim *analyst* mampu mengakses berbagai alat dan metode dengan praktis (Kencana Putri and Ichsanuddin Nur, 2023). Teknologi *digital* merupakan teknologi yang mengoperasikan dengan sedikit penggunaan tenaga manusia dan bertujuan menggunakan sistem otomatis melalui sistem komputer (Wibowo *et al.*, 2023).

Python *library* merupakan himpunan kode yang menyediakan fungsionalitas tertentu. Pustaka yang ditulis akan memainkan peran penting dalam pembelajaran mesin dan ilmu data. Python diidentifikasi sebagai sebuah *high level language* yang dirilis oleh Guido Van Rossum dan diperkenalkan awal 1991 (Alfarizi *et al.*, 2023).

Python merupakan bagian dari usaha pengembangan bahasa pemrograman populer yang dimanfaatkan dalam pengelola *data scientist* dan *developer software* untuk penggunaan data dalam jumlah dan memiliki ukuran yang besar (*big data*) (Haldibekova, 2022). Bahasa pemrograman python digunakan untuk melakukan prediksi (*prediction*) berdasarkan informasi dari suatu data yang bersifat historis, otomatisasi berbagai tugas penting, memudahkan suatu proses, dan memformulasi data secara teliti dalam memperoleh *insight* melalui aktivitas *business intelligence*. Python mempunyai

## 11. NLTK (<https://www.nltk.org/>)

NLTK adalah singkatan dari "*Natural Language Toolkit*". Merupakan perpustakaan (*library*) *open-source* dalam bahasa pemrograman python yang akan digunakan untuk melakukan proses bahasa alami (*Natural Language Processing* atau NLP). NLTK menyediakan alat dan sumber daya untuk membangun program yang efektif dalam pemrosesan dan analisis teks.

NLTK menjadi perpustakaan yang digunakan dalam pemrosesan bahasa alami di python dan fondasi proyek NLP. Penggunaan alat dan dukungan komunitas membuat NLTK menjadi pilihan peneliti dan praktisi yang tertarik dalam proses bahasa alami. Fitur meliputi alat pemrosesan teks, model statistik, sumber daya dan korpus, dokumentasi yang kaya, komunitas yang aktif, dan fleksibilitas dan penggunaan yang luas.

## Penutup

Python *package* atau paket python merupakan suatu himpunan dari berbagai modul yang teratur, terstruktur, dan sistematis melalui direktori sehingga tim *analyst* mampu mengakses berbagai alat dan metode dengan praktis. Teknologi *digital* merupakan teknologi yang mengoperasikan dengan sedikit penggunaan tenaga manusia dan bertujuan menggunakan sistem otomatis melalui sistem komputer.

*Module* dalam python adalah *file* yang berisi definisi variabel, fungsi, dan kelas yang dapat digunakan di dalam program dengan *file* ekstensi dengan format *.py* yang dieksekusi *interpreter* python. Nama dari module *.py* adalah nama dari *file* dengan contoh *file* bernama "*latihan.py*". Setiap *module* python mempunyai tabel simbol yang dimanfaatkan sebagai tabel simbol global oleh berbagai fungsi yang diidentifikasi dalam *module*.

Paket (*package*) dalam python adalah struktur hierarkis yang berisi modul dan sub paket yang digunakan untuk mengorganisir kode Python menjadi unit yang lebih besar dan terstruktur. Paket memungkinkan pengelompokan dan pengaturan yang logis dari kode Python, memfasilitasi pemeliharaan, pengembangan, dan penggunaan kembali kode serta mempunyai satu *module constructor* (`__init__.py`).

Sebuah modul dalam *file* python berisi definisi variabel, fungsi, dan kelas. Modul digunakan untuk mengorganisir dan membagi kode ke dalam unit kecil. Modul digunakan untuk menyediakan fungsionalitas tertentu yang dipanggil kembali dalam program.

*Library* (perpustakaan) dalam bahasa pemrograman python adalah kumpulan modul yang menyediakan fungsi dan alat yang digunakan untuk memenuhi kebutuhan dalam pengembangan perangkat lunak. *Library* dibangun untuk menyelesaikan tugas yang umum dalam suatu pemrograman seperti melakukan pemrosesan data, interaksi jaringan, pengolahan gambar, dan analisis matematika. Python menyediakan berbagai paket dan *library* yang populer dalam pengembangan proyek *data science*. *Library* memiliki karakter *reusable* yang berfungsi untuk dapat digunakan berulang kali, dimana dan kapan saja. Program perangkat lunak dibangun secara profesional mencakup dokumentasi, *library*, struktur sistem, konfigurasi data.

\*\*\*\*\*

## Daftar Pustaka

- Abdul muthalib, M. *et al.* (2023) 'Pengiraan Pose Model Manusia Pada Repetisi Kebugaran AI Pemrograman Python Berbasis Komputerisasi', *INFOTECH journal*, 9(1), pp. 11–19. Available at: <https://doi.org/10.31949/infotech.v9i1.4233>.
- Alfarizi, M.R.S. *et al.* (2023) 'Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning', *Karya Ilmiah Mahasiswa Bertauhid (KARIMAH TAUHID)*, 2(1), pp. 1–6.
- Haldibekova, A. (2022) *Penerapan & Implementasi Big Data di Berbagai Sektor (Pembangunan Berkelanjutan Era Industri 4.0 dan Society 5.0), Ilmu Pengetahuan dan Potensi Keilmuan: Landasan Pembangunan Masyarakat yang Inovatif Berkelanjutan*.
- Kencana Putri, A. and Ichsanuddin Nur, D. (2023) 'Penggunaan Bahasa Python Untuk Analisis Dan Visualisasi Data Penduduk Di Desa Sumberjo, Nganjuk', *Jurnal Pengabdian Kepada Masyarakat*, 3(3), pp. 206–217. Available at: [https://jurnalfkip.samawa-university.ac.id/karya\\_jpm/index](https://jurnalfkip.samawa-university.ac.id/karya_jpm/index).
- Margareth, H. (2020) *Pengantar Data Science dan Aplikasinya bagi Pemula*, Unpar Press. Bandung: Unpar Press.
- Rahman, S. *et al.* (2023) *Python : Dasar Dan Pemrograman Berorientasi Objek*, Penerbit Tahta Media.
- Santoso, J.T. (2020) *Analisis Big Data*. Semarang: Yayasan Prima Agus Teknik dan Universitas STEKOM.
- Septian, R.F. (2013) 'Belajar Pemrograman Python Dasar', *Poss-Upi*, pp. 1–108.
- Wibowo, S.H. *et al.* (2023) *Teknologi Digital Di Era Modern*. Edited by Diana Purnama Sari. Padang: PT. Global Eksekutif Teknologi.
- Zatin Niqotaini *et al.* (2023) *Rekayasa Perangkat Lunak*. Edited by E. Mardiani. Jakarta: PT Penamuda Media.

## PROFIL PENULIS



**Dr. Ir. Norbertus Tri Suswanto Saptadi, S.Kom., M.T., M.M., IPM.**

Penulis lahir di Cirebon Jawa Barat, tanggal 7 Juni 1975. Memiliki Jabatan Fungsional Lektor Kepala, Pembina Tingkat I (IV/b). Berpendidikan Sarjana Komputer (S.Kom.) di Universitas Teknologi Digital Indonesia (UTDI) tahun 1998, Magister Manajemen (M.M.) di Universitas Hasanuddin (UNHAS) tahun 2004, Magister Teknologi Informasi (M.T.) di Universitas Gadjah Mada (UGM) tahun 2007, Insinyur (Ir.) di Pendidikan Profesi Insinyur UNHAS tahun 2020, Insinyur Profesional Madya (IPM.) di Persatuan Insinyur Indonesia (PII) tahun 2021, Doktor (Dr.) di Fakultas Teknik UNHAS tahun 2023, dan Program Pendidikan Reguler Angkatan (PPRA) LX Lemhannas RI tahun 2020. Menjadi tenaga pengajar (Dosen) pada Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Atma Jaya Makassar (UAJM). Peraih Poster terbaik DPRM Dikti tahun 2016. Dosen berprestasi IKDKI tahun 2020 dan 2021. Pernah menjabat Kepala UPT Komputer, Kepala BAPSI, Wakil Dekan FT, Dekan FT dan FTI, Wakil Rektor III, Ketua Penjaminan Mutu. Tim PAK Dosen dan Asesor BKD UAJM. Reviewer International Conference dan Jurnal SINTA. Pemenang Hibah Kemdikbud Penelitian Dosen Pemula, Bersaing, Fundamental, dan Strategi Nasional. Penulis artikel media massa Tribun Timur, Koinonia, Bisnis Sulawesi, Sesawi.net, Mirifica.net, HidupKatolikCom, OMKNet, KatolikanaTV, Jalan Hidup Katolik, dll. Aktifis organisasi IKA Lemhannas RI LX, IARMI, ISKA, APTIKOM, BAPOMI Sulsel, IKDKI SulSelTraBar, LP3KD Sulsel, Komkep KAMS, Komsos KAMS, PUKAT KAMS, TPP KAMS, FMKI KAMS, UPS KAMS, Pengurus Kebun Sawit Laimbo, FDI, PII Makassar, INAPR, Dewan Keuangan Paroki dan Program Ayo Sekolah Mariso, dll.

Email Penulis: [ntsaptadi@gmail.com](mailto:ntsaptadi@gmail.com)

# BAB 9

## PEMROGRAMAN FUNGSIONAL

**Siti Mutmainah, S.Kom., M.Kom.**  
Universitas Muhammadiyah Bima

### **Pendahuluan**

Pemrograman fungsional merupakan konsep pemrograman yang mengutamakan penggunaan fungsi sebagai objek utama. Konsep ini membangun fungsi yang menerima input dan menghasilkan output tanpa adanya efek samping yang tidak diinginkan. Pemrograman fungsional menjadi fokus utama dalam pengembangan perangkat lunak modern karena beberapa alasan seperti, kemampuan dalam menuliskan kode yang lebih ekspresif dan dapat digunakan kembali, potensi peningkatan kinerja dan efisiensi dalam konteks aplikasi yang kompleks dan berskala besar. Melalui pemahaman konsep-konsep ini dan penggunaan *tools* dan teknik yang mendukungnya, diharapkan dapat membangun perangkat lunak yang lebih handal, *maintainable*, dan mudah dikembangkan di era perangkat lunak yang semakin kompleks dan dinamis (Lott, 2018). Bab ini membahas konsep dasar pemrograman fungsional dan mengintegrasikannya ke dalam bahasa pemrograman Python. Tujuan dari bab ini yaitu memberikan pemahaman mengenai konsep pemrograman fungsional dalam meningkatkan kualitas, kejelasan, dan efisiensi syntax Python. Beberapa topik yang akan dibahas meliputi:

#### 4. Rekursi

Suatu teknik pemanggilan fungsi berulang kali untuk menyelesaikan tugas yang lebih besar. Hal ini merupakan salah satu dari beberapa *tool* yang sangat penting dalam pemrograman fungsional. *Tool* ini memungkinkan pemecahan masalah yang elegan dan efisien sesuai dengan konsep (Geeksforgeeks, 2022). Konsep rekursi dalam pemrograman fungsional melibatkan dua komponen utama:

- a. Kasus Dasar (*Base Case*): Kondisi ini merupakan kondisi ketika rekursi berhenti dan fungsi tidak memanggil dirinya sendiri lagi. Tanpa *case base* yang benar, fungsi akan tetap memanggil dirinya sendiri tanpa batas dan menyebabkan *stack overflow*.
- b. Langkah Rekursif: Hal ini merupakan langkah-langkah yang diambil oleh fungsi untuk mendekati *case base*. Pada langkah ini fungsi akan memanggil dirinya sendiri dengan argumen yang berbeda guna menyelesaikan tugas-tugas yang lebih kecil. Contoh-contoh penggunaan rekursi dalam pemrograman

Python:

```
## Faktorial:
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

print(factorial(5))
```

120

```
## Fibonacci Sequence:
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

print(fibonacci(7))
```

13

```
## Pencarian Binary
def binary_search(arr, target, low, high):
    if low > high:
        return -1
    mid = (low + high) // 2
    if arr[mid] == target:
        return mid
    elif arr[mid] > target:
        return binary_search(arr, target, low, mid - 1)
    else:
        return binary_search(arr, target, mid + 1, high)

arr = [2, 4, 7, 10, 13, 17, 19, 22]
target = 13
print(binary_search(arr, target, 0, len(arr)-1))

4
```

\*\*\*\*\*

### Daftar Pustaka

- Geeksforgeeks. (2022). *Functional Programming Paradigm*. Geeksforgeeks. <https://www.geeksforgeeks.org/functional-programming-paradigm/>
- Liem, I. (2008). *Dasar Pemrograman - Pemrograman Fungsional*. [https://repository.unikom.ac.id/45769/1/Pemrograman Fungsional.pdf](https://repository.unikom.ac.id/45769/1/Pemrograman-Fungsional.pdf)
- Lott, S. F. (2018). *Functional Python programming*. Packt Publishing Ltd.
- Mertz, D. (2001). Charming Python: Functional programming in Python, Part 1. *DeveloperWorks*. <https://www.eecg.utoronto.ca/~jzhu/csc326/readings/functional-charming1.pdf>
- Mertz, D. (2015). *Functional Programming in Python* (Meghan Blanchette (ed.)). O'Reilly Media.
- Priambodo, B. (2017). *Perkenalan Paradigma Pemrograman Fungsional Praktis*. Medium.Com. <https://medium.com/paradigma-fungsional/perkenalan-paradigma-pemrograman-fungsional-praktis-7bc5dce026b9>
- Python Software Foundation. (n.d.). *Modul Pemrograman Fungsional*. Python. <https://docs.python.org/id/3.8/library/functional.html>
- Van Hattem, R. (2016). *Mastering Python*. Packt Publishing Ltd.

## PROFIL PENULIS



### **Siti Mutmainah, S.Kom., M.Kom.**

Penulis memiliki keahlian yang mendalam dalam ilmu komputer. Penulis menempuh pendidikan sarjana (S1) di jurusan Teknik Informatika di Universitas AMIKOM Yogyakarta. Setelah menyelesaikan gelar sarjana, penulis melanjutkan pendidikan di tingkat magister (S2) dengan konsentrasi Data Sains Informatika pada Universitas Islam Indonesia (UII) dan berhasil menyelesaikan program tersebut pada tahun 2023. Selama masa studi di tingkat magister, penulis mengkhususkan diri dalam berbagai aspek data sains, termasuk pengolahan data, analisis statistik, dan pemodelan prediksi.

Keahlian penulis terutama terfokus pada pengembangan teknik *machine learning* dan *deep learning* menggunakan bahasa pemrograman Python. Penulis telah mendalami berbagai konsep dan teknik dalam *machine learning* dan *deep learning*, dan mampu mengimplementasikannya secara efektif dalam penyelesaian masalah praktis. Saat ini, penulis menjalani karier sebagai seorang dosen yang berdedikasi. Selain kegiatan mengajar di perguruan tinggi, penulis juga terlibat dalam kegiatan penelitian dan pengabdian kepada masyarakat. Terlibat dalam pengembangan di bidang teknologi informasi, penulis berusaha untuk memberikan kontribusi positif kepada masyarakat melalui penelitian yang inovatif dan pengaplikasian teknologi informasi dalam berbagai bidang kehidupan. Dengan kombinasi antara keahlian akademis yang kuat dan pengalaman praktis yang luas, penulis bertekad untuk terus berkontribusi dalam pengembangan ilmu komputer dan teknologi informasi secara lebih luas lagi.

# BAB 10

## *CONCURRENCY*

**Mgr inz. Khairunnas, S.Kom.**  
Universitas Muhammadiyah Bima

### ***Apa Itu Concurrency?***

Diperkirakan volume data yang harus diolah oleh sistem komputer akan meningkat dua kali lipat setiap dua tahun. Sebagai contoh, menurut *International Data Corporation* (IDC), pada tahun 2020 diperkirakan setiap orang di planet ini akan menghasilkan sekitar 5.200 GB data. Dengan jumlah data yang begitu masif, kebutuhan akan kekuatan pengolahan data menjadi sangat kritis. Meskipun berbagai teknik pengolahan data terus berkembang dan digunakan setiap hari, pemrograman yang melibatkan *concurrency* masih menjadi salah satu metode terdepan untuk mengolah data dengan efektif dan akurat.

Meski istilah *concurrency* mungkin terdengar menakutkan bagi sebagian orang, prinsip di baliknya sebenarnya cukup sederhana dan seringkali ditemui, bahkan di luar dunia pemrograman. Namun, ini tidak berarti membuat program yang berjalan secara konkuren itu mudah; sebaliknya, jenis program ini memang lebih kompleks untuk ditulis dan dipahami. Akan tetapi, dengan menerapkan struktur *concurrency* yang tepat dan efisien, kita dapat mencapai peningkatan waktu eksekusi yang signifikan (Nguyen, 2018).

Secara bahasa *concurrency* dapat didefinisikan sebagai dua atau lebih peristiwa atau keadaan yang terjadi atau ada pada waktu yang sama. Dalam dunia Ilmu Komputer, *concurrency* merujuk pada kemampuan sebuah program, algoritma, atau tugas untuk dijalankan

Banyak kasus dimana aplikasi terkunci ini terjadi saat menunggu operasi Input/Output (I/O), yang melibatkan interaksi dengan perangkat keras komputer seperti keyboard, *hard drive*, atau yang paling sering, kartu jaringan. Selama aplikasi *synchronous* menunggu data dari sebuah input pengguna atau sumber eksternal seperti API web, tidak ada yang bisa dilakukan selain menunggu operasi itu selesai. Ini sering menyebabkan masalah kinerja dan responsivitas, karena aplikasi tidak bisa melakukan tugas lain selama operasi panjang tersebut berlangsung.

Untuk mengatasi masalah ini, kita bisa memperkenalkan prinsip *concurrency*. Secara sederhana, *concurrency* memungkinkan lebih dari satu tugas untuk diproses secara bersamaan. Contohnya, dalam konteks I/O, ini bisa berarti mengizinkan beberapa permintaan web berlangsung sekaligus, atau memungkinkan koneksi simultan ke server. Salah satu cara untuk mencapai ini di Python adalah dengan menggunakan *library asyncio*, singkatan dari I/O *synchronous*. Librari ini mengadopsi model pemrograman asinkron yang memungkinkan pengelolaan beberapa operasi I/O sekaligus tanpa mengorbankan responsivitas aplikasi.

```
import asyncio

async def main():
    print('Hello ...')
    await asyncio.sleep(1)
    print('... World!')

asyncio.run(main())
```

(Python Documentation, 2024)

\*\*\*\*\*

## Daftar Pustaka

Forbes, E. (2017). *Learning Concurrency in Python*. Packt Publishing.

Fowler, M. (2022). *Python Concurrency with asyncio*. Manning Publications Co.

Hunt, J. (2019). *Advanced Guide to Python 3 Programming*. Springer Nature Switzerland AG.

Nguyen, Q. (2018). *Mastering Concurrency in Python*. Packt Publishing.

Python Documentation. (2024). *asyncio — Asynchronous I/O*. <https://docs.python.org/3/library/asyncio.html>

## PROFIL PENULIS



### **Mgr inz. Khairunnas, S.Kom.**

Penulis saat ini menjabat sebagai dosen ilmu komputer di Universitas Muhammadiyah, sebuah peran yang diemban sejak belum lama ini. Meski belum memiliki pengalaman luas dalam menulis buku, penulis memiliki ketertarikan mendalam untuk terus belajar dan memperluas wawasan akademik. Penulis menyelesaikan pendidikan sarjana di Universitas Mataram dalam jurusan Teknik

Informatika, kemudian melanjutkan studi pascasarjana di Vistula University, Polandia dengan beasiswa dari Pemerintah Provinsi NTB, dengan spesialisasi dan fokus pada *Applied Data Science*. Pengetahuan yang diperoleh selama periode studi kini diaplikasikan dalam pembelajaran dan pembimbingan mahasiswa. Penulis juga terus mendalami bidang *data science* dan teknologi terbaru, yang relevan dengan spesialisasi akademis. Melalui pengembangan karier ini, penulis bertujuan untuk memperkaya pengalaman menulis dan berkontribusi lebih aktif dalam komunitas ilmiah, terutama dalam upaya peningkatan mutu pendidikan dan penelitian.

Email Penulis: [kkhairunnas.khairunnas@gmail.com](mailto:kkhairunnas.khairunnas@gmail.com)

# BAB 11

## *GUI PROGRAMMING*

**Eka Purnama Harahap, S.Kom., M.TI.**

Universitas Raharja

### **Definisi GUI**

Sejarah GUI: Dikembangkan oleh Xerox Palo Alto pada akhir tahun 1970. Sistem ini digunakan oleh Apple untuk Macintosh dan Windows yang dimiliki Microsoft setelah selesai berhasil. Mengaktifkan GUI menyelesaikan kesulitan pengguna dengan respons yang tidak konsisten.

Grafik antarmuka pengguna (GUI) adalah cara interaksi antara komputer dan manusia di mana komputer menawarkan tampilan grafis yang mudah dipahami oleh pengguna (user) selama pengoperasian suatu aplikasi (Harianto et al., 2019).

GUI menampilkan objek yang dapat menyebabkan masalah dan menunjukkan tindakan awal pengguna. Adanya GUI memungkinkan kita melihat input yang diterima dan menunjukkan tanggapan. Saat terjadi hubungan, GUI dapat melihat perubahan rona, ukuran, dan visibilitas (Hope, 2023).

Berdasarkan pemahaman di atas, dapat dikatakan bahwa antarmuka pengguna grafis (GUI) atau antarmuka pengguna grafis adalah tampilan visual sebuah program komputer yang memungkinkan pengguna berinteraksi dengan perangkat lunak menggunakan elemen grafis seperti ikon, tombol, jendela, dan menu. GUI membuat antarmuka pengguna lebih mudah dipahami dan mudah dipahami daripada antarmuka berbasis teks.

```

v = IntVar()

Radiobutton(window, text="Pilihan 1",
variable=v, value=1).pack()

Radiobutton(window, text="Pilihan 2",
variable=v, value=2).pack()

window.mainloop()

```

### b. Membuat *scrollbar*

*Scrollbar* digunakan untuk menavigasi konten yang panjang atau lebar. Elemen ini berfungsi jika konten tidak muat dalam jendela aplikasi.

Syntax:

```

from tkinter import Tk, Scrollbar, Text

window = Tk()

scrollbar = Scrollbar(window)
scrollbar.pack(side="right", fill="y")

text = Text(window, wrap="none",
yscrollcommand=scrollbar.set)
text.pack(side="left", fill="both",
expand=True)

scrollbar.config(command=text.yview)

window.mainloop()

```

### c. Menampilkan gambar

*PhotoImage* digunakan untuk menampilkan gambar dari file. Cocok untuk aplikasi yang perlu menampilkan media visual.

Syntax:

```
from tkinter import Tk, Label, PhotoImage

window = Tk()

gambar = PhotoImage(file="gambar.png")

label = Label(window, image=gambar)
label.pack()

window.mainloop()
```

\*\*\*\*\*

## Daftar Pustaka

- Dian, D. (2019, January 30). *12 Modul Python untuk Membuat Aplikasi GUI (Desktop)*. <https://www.petanikode.com/python-gui/>.
- Fakhruzzaman, M. N., & Utami, I. Q. (2023). *Buku Ajar Algoritma Pemrograman*. Airlangga University Press.
- Juliarto, R. (2021, April 14). *Apa Itu Programming? Berikut Penjelasan Lengkapnya*. <https://www.dicoding.com/blog/apa-itu-programming-berikut-penjelasan-lengkapnya/>.
- Karim, A., Bangun, B., Purnama, I., Harahap, S. Z., Irmayani, D., Nasution, M., Haris, M., & Munthe, I. R. (2020). *Pengantar teknologi informasi*. Yayasan Labuhanbatu Berbagai Gemilang.
- Kurniastuti, I., & Andini, A. (2018). Perancangan Program Penentuan Histogram Citra dengan Graphical User Interface (GUI). *Applied Technology and Computing Science Journal*, 1(1), 11–17.
- Lutz, M. (2010). *Programming Python: powerful object-oriented programming*. “O’Reilly Media, Inc.”
- Nosrati, M. (2011). Python: An appropriate language for real world programming. *World Applied Programming*, 1(2), 110–117.
- Putro, H. P., Widyaningsih, T. W., Englishtina, I., Nursanty, E., & Dema, H. (2023). *Development Of Artificial Intelligence Applications (Studi Kasus & Implementasi AI Menggunakan Berbagai Bahasa Pemrograman)*. PT. Sonpedia Publishing Indonesia.
- RevoU Staff. (2023, August 18). *10 Contoh Program Python Tkinter untuk Membangun GUI*. <https://revou.co/panduan-teknis/python-tkinter>.

## PROFIL PENULIS



### **Eka Purnama Harahap, S.Kom., M.TI.**

Kelahiran Baringin Jaya, 18 September 1993.

Ia merupakan Sarjana lulusan STMIK Raharja pada tahun 2016. Dua tahun kemudian lulus sebagai Magister Teknologi Informasi di STMIK Raharja yang sejak 2019 telah berubah menjadi Universitas Raharja. Saat ini sedang menempuh pendidikan Doktorat Ilmu Komputer di Universitas Kristen Satya Wacana. Sejak 2016 sampai hingga saat ini, ia telah memiliki **10+** Mata kuliah yang diampu, dimana 2 mata kuliah diantaranya yaitu Teknologi Platform dan Pengantar Kecerdasan Buatan. Selama 7 tahun mengabdikan, banyak pengalaman yang didapatkan sehingga mengajarkannya lebih mengenal beraneka ragam karakter.

Pada Tri Dharma Penelitian, ia aktif menulis karya ilmiah yang terpublikasi di kancah nasional maupun internasional. Diantaranya terdapat 50+ Jurnal Nasional Terakreditasi yang terindex Google Scholar dan SINTA serta 10+ Jurnal Internasional Terindex SCOPUS. Ia juga merupakan bagian dari Web of Science Academy, yakni sebagai graduate dan juga mentor. Selain penelitian, ia memiliki 3 hak kekayaan intelektual di bidang teknologi informasi. Bidang minatnya meliputi; *Machine Learning, Artificial Intelligence, Smart Tourism, dan Data Analytics.*

**Motto:** Al Jazaa Min Jinsil 'Amal

Email Penulis: [ekapurnamaharahap@raharja.info](mailto:ekapurnamaharahap@raharja.info)

# BAB 12

## *WEB DEVELOPMENT*

**Zumhur Alamin, M. Kom.**  
Universitas Muhammadiyah Bima

### **Pendahuluan**

Pengembangan web telah menjadi bagian tak terpisahkan dari dunia teknologi informasi. Dalam konteks ini, Python telah dikenal sebagai salah satu bahasa pemrograman populer dan handal untuk pengembangan web. Keunggulan Python dalam pengembangan web tidak hanya didasarkan pada kemudahan penggunaannya, tetapi juga pada fleksibilitasnya yang luar biasa serta ekosistem yang kaya akan *library* dan *framework*.

Python menjadi salah satu pilihan dalam pengembangan web karena kemudahan penggunaannya dan keunggulan sintaks yang bersih dan mudah dipahami, memungkinkan para pengembang fokus pada logika aplikasi tanpa terbebani oleh kompleksitas sintaks. Fleksibilitas dan kekuatan Python tercermin dalam beragamnya *library* dan *framework* yang mendukung pengembangan berbagai jenis aplikasi web, dari yang sederhana hingga kompleks. Dengan *framework* seperti Django, Flask, dan Pyramid, pengembang dapat mempercepat proses pengembangan dan membangun aplikasi web dengan lebih efisien (Cahyono, 2019).

Kemampuan Python untuk menangani tantangan modern dalam pengembangan web seperti skalabilitas, keamanan, dan kinerja, juga membuatnya diminati oleh pengembang (Satria, 2023). Selain itu, dukungan dari komunitas pengembang yang besar turut memperkuat

### 3. Pengaturan Hak Akses Pengguna dalam Django

Dalam pengembangan aplikasi web dengan Django, pengaturan hak akses pengguna menjadi lebih mudah berkat fitur *built-in* yang disediakan oleh *framework* ini. Django memperkenalkan konsep peran dan grup pengguna yang memungkinkan pengembang untuk mengelola hak akses pengguna dengan lebih terstruktur. Beberapa langkah yang umum dalam pengaturan hak akses pengguna dalam Django meliputi:

#### a. Definisi Peran dan Grup

Langkah pertama adalah menentukan peran dan grup pengguna yang sesuai dengan kebutuhan aplikasi. Pengembang dapat menentukan peran seperti admin, pengguna biasa, atau pengguna dengan hak akses khusus, serta menempatkan pengguna dalam grup yang sesuai dengan peran mereka.

#### b. *Decorator* @login\_required

Django menyediakan *decorator* bawaan seperti @login\_required yang dapat digunakan untuk melindungi view atau rute tertentu agar hanya dapat diakses oleh pengguna yang sudah login. *Decorators* ini mempermudah pengembang dalam mengimplementasikan logika akses pengguna pada berbagai bagian aplikasi.

#### c. Otorisasi Berbasis *Permission*

Pengaturan *permission* merupakan bagian penting dalam mengatur hak akses pengguna. Django menyediakan sistem *permission built-in* yang memungkinkan pengembang untuk menetapkan *permission* pada level grup atau peran pengguna. *Permission* dapat ditetapkan pada level *view*, model, atau bahkan *field* dalam model, memberikan kontrol yang sangat detail terhadap akses pengguna.

#### d. *Custom Middleware*

Jika diperlukan, pengembang dapat membuat *middleware* kustom untuk mengontrol akses pengguna pada tingkat yang lebih rendah. *Middleware* ini dapat digunakan untuk melaksanakan logika akses khusus atau untuk menangani kasus-kasus yang tidak dapat ditangani dengan *decorators* atau *permission* bawaan.

Dengan menggunakan fitur otentikasi dan otorisasi yang disediakan oleh Django, pengembang dapat membangun aplikasi web yang aman dan terstruktur dengan mudah. Dengan mengatur akses pengguna dengan tepat, pengembang dapat memastikan bahwa hanya pengguna yang berwenang yang memiliki akses ke bagian-bagian sensitif dari aplikasi web.

\*\*\*\*\*

### **Daftar Pustaka**

- Andri Budiman, E., & Pringgo Digdo, G. (2023). Perancangan Fitur Audit Security Configuration Compliance Pada Aplikasi Helium Security. *Indonesian Journal Computer Science*, 2(2), 67–76. <https://doi.org/10.31294/ijcs.v2i2.2481>
- Bariyah, S. H., & Imania, K. A. N. (2022). Pengembangan Virtual Assistant Chatbot Berbasis Whatsapp Pada Pusat Layanan Informasi Mahasiswa Institut Pendidikan Indonesia-Garut. *J. Petik*, 8(1), 66–79.
- Cahyono, H. (2019). *Membangun Aplikasi Web Data Analysis dengan Framework Django: Mengenal berbagai macam pustaka di Python*. Hary Cahyono.
- Gaspar, D., & Stouffer, J. (2018). *Mastering Flask Web Development: Build Enterprise-grade, Scalable Python Web Applications*. Packt Publishing Ltd.
- Mukhlis, I. R., Irmawati, I., Sabur, F., Farkhan, M., Gunawan, P. W., Adhicandra, I., & Eldo, H. (2023). *BUKU AJAR PEMROGRAMAN WEB 1*. PT. Sonpedia Publishing Indonesia.
- Ngantung, R. K., & Pakereng, M. A. I. (2021). Model Pengembangan

Sistem Informasi Akademik Berbasis User Centered Design Menerapkan Framework Flask Python. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 5(3), 1052. <https://doi.org/10.30865/mib.v5i3.3054>

Rachmadi, T., & Kom, S. (2020). *Pengantar Teknologi Informasi* (Vol. 1). Tiga Ebook.

Rahman, S., Sembiring, A., Siregar, D., Prahmana, I. G., Puspadini, R., & Zen, M. (2023). *Python: Dasar dan Pemrograman Berorientasi Objek*. Penerbit Tahta Media.

Satria, D. (2023). *Pengantar Teknik Komputer: Konsep dan Prinsip Dasar*. PT. Sonpedia Publishing Indonesia.

## PROFIL PENULIS



### **Zumhur Alamin, M. Kom.**

Minat penulis terhadap ilmu komputer dimulai pada tahun 2005. Penulis memulai perjalanan pendidikan di Universitas Muhammadiyah Malang, mengambil jurusan Sarjana Teknik Informatika, dan berhasil meraih gelar sarjana pada tahun 2009. Pada tahun 2014 awal, penulis melanjutkan studi pada program magister Ilmu Komputer (M. Kom.) di Universitas Budi Luhur Jakarta dan berhasil menyelesaikan studi tahun 2015. Sebagai anak kedua dari enam bersaudara dari pasangan Ir. Zainuddin H. Hamzah dan Nurjanah, S. Sos, S. Pd., penulis telah menunjukkan komitmennya dalam dunia akademis. Kini, sebagai seorang dosen di Program Studi Ilmu Komputer di Universitas Muhammadiyah Bima sejak tahun 2022, penulis mengampu berbagai mata kuliah, termasuk Algoritma dan Pemrograman, serta memiliki ketertarikan di bidang Software Engineering dan Data Science. Untuk mewujudkan karier sebagai dosen profesional, penulis aktif sebagai peneliti dibidang kepakarannya tersebut. Selain peneliti, penulis juga aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini.

Email Penulis: [zumhur.alamin@gmail.com](mailto:zumhur.alamin@gmail.com)

# BAB 13

## *DATA SCIENCE DAN MACHINE LEARNING*

**Inna Sabily Karima, S.Kom., M.Kom.**  
Universitas Mercu Buana

### **Pengenalan Data Science**

Data *Science* adalah bidang interdisipliner yang memadukan ilmu statistik, matematika, dan ilmu komputer untuk menganalisis, memahami, dan mengekstrak wawasan dari data. Dalam era digital ini, data telah menjadi aset penting bagi perusahaan dan organisasi di semua sektor, dari bisnis hingga pemerintahan, yang memicu permintaan tinggi untuk ilmuwan data yang mampu mengolah dan menginterpretasi data besar.

Salah satu aspek fundamental dari Data *Science* adalah kemampuannya untuk mengambil data mentah dan mengubahnya menjadi informasi yang bermanfaat. Ini melibatkan berbagai teknik dan proses, termasuk pembersihan data, analisis eksploratif, visualisasi data, dan model statistik atau *machine learning*. Pembersihan data adalah langkah penting untuk memastikan bahwa dataset bebas dari kesalahan atau inkonsistensi, yang bisa berpengaruh pada keakuratan analisis. Analisis eksploratif membantu dalam memahami distribusi dan hubungan antar variabel dalam data, sedangkan visualisasi data adalah cara efektif untuk menyampaikan temuan kepada pemangku kepentingan.

Model statistik dan *machine learning* memungkinkan ilmuwan data untuk membuat prediksi atau mengklasifikasikan data berdasarkan informasi historis. Pendekatan ini tidak hanya bermanfaat dalam menilai tren saat ini, tetapi juga dalam meramalkan kejadian di masa depan. Misalnya, dalam sektor keuangan, model prediktif digunakan untuk menilai risiko kredit atau dalam perawatan kesehatan untuk meramalkan hasil pasien.

Transformasi digital dan ledakan data yang tersedia telah mengarah pada perkembangan teknik dan alat yang lebih canggih dalam Data Science. Teknologi seperti pembelajaran mendalam (*deep learning*) dan kecerdasan buatan (*artificial intelligence*) telah memperluas kemampuan dan aplikasi dari Data Science, mendorong inovasi dan efisiensi di berbagai bidang.

Oleh karena itu, Data Science tidak hanya penting untuk meningkatkan keputusan bisnis dan operasional, tetapi juga memiliki dampak sosial yang signifikan. Misalnya, analisis data dapat membantu dalam mengidentifikasi pola penyebaran penyakit, mengoptimalkan pengiriman sumber daya di daerah bencana, atau bahkan dalam merancang strategi pemberantasan kemiskinan. Sebagai hasilnya, Data Science merupakan salah satu bidang yang paling dinamis dan berpengaruh di abad ke-21, yang terus berkembang seiring dengan kemajuan teknologi.

## **Mengapa Kita Memerlukan Data Science?**

Dalam dunia yang semakin didominasi oleh data, kebutuhan akan Data Science semakin kritis. Data Science mengintegrasikan matematika, statistik, dan teknologi komputer untuk menganalisis dan memanfaatkan data dalam jumlah besar, memberikan keunggulan kompetitif dan membuka peluang inovasi baru di berbagai industri.

### **1. Manfaat Data Science dalam Bisnis**

Data Science memberikan berbagai manfaat penting dalam lingkup bisnis. Melalui analisis data yang efektif, organisasi dapat memahami pola dan tren yang kompleks, memungkinkan mereka membuat keputusan strategis yang lebih tepat dan efisien. Dari meningkatkan pengalaman pengguna hingga mengoptimalkan

## Aplikasi *Machine Learning*

*Machine Learning* telah diterapkan dalam berbagai industri dan bidang, termasuk:

### 1. Pengenalan Wajah

Di mana model *Machine Learning* digunakan untuk mengidentifikasi wajah dalam gambar atau video.

### 2. Pengolahan Bahasa Alami

Untuk memahami dan memproses teks manusia dalam bentuk pengenalan ucapan, penerjemahan bahasa, dan lainnya.

### 3. Analisis Keuangan

Untuk memprediksi harga saham, deteksi penipuan, atau melakukan analisis risiko kredit.

### 4. Kesehatan

Untuk diagnosis medis, pemetaan genom, atau memahami pola kesehatan populasi.

## Tantangan dalam *Machine Learning*

Meskipun potensi besar *Machine Learning*, ada beberapa tantangan yang harus diatasi, termasuk:

### 1. Kekurangan Data Berkualitas

Model *Machine Learning* membutuhkan data yang berkualitas untuk pelatihan yang efektif. Kekurangan data atau data yang tidak lengkap dapat menyebabkan kinerja model yang buruk.

### 2. Pengolahan dan Skalabilitas

Mengelola volume data yang besar dan kompleks dapat menjadi tantangan, terutama dalam konteks waktu pemrosesan dan pengoptimalan algoritma.

### 3. Interpretasi Model

Beberapa model *Machine Learning*, khususnya yang kompleks seperti neural networks, cenderung menjadi "hitam box", sulit untuk diinterpretasikan oleh manusia.

*Machine Learning* telah menjadi alat yang sangat penting dalam era data-driven ini. Dengan kemampuannya untuk belajar dari data dan membuat prediksi yang akurat, *Machine Learning* telah memberikan kontribusi besar terhadap berbagai bidang, mulai dari bisnis hingga

ilmu kesehatan. Dengan pemahaman yang baik tentang konsep dasar dan aplikasi *Machine Learning*, kita dapat memanfaatkan potensi besar teknologi ini untuk mendorong inovasi dan kemajuan di masa depan.

\*\*\*\*\*

## Daftar Pustaka

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Cady, F. (Ed.). (2015). *The Data Science Handbook*. Wiley.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media.
- Grus, J. (2015). *Data Science from Scratch: First Principles with Python*. O'Reilly Media.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Knaflic, C. N. (2015). *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley.
- McKinney, W. (2017). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Müller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media.
- Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media

## PROFIL PENULIS



### **Inna Sabily Karima, S.Kom., M.Kom.**

Ketertarikan penulis terhadap ilmu komputer khususnya data *science* mulai tahun 2006 silam. Hal tersebut membuat penulis memilih untuk melanjutkan studi pada jurusan Teknik Informatika dan berhasil lulus pada tahun 2010. Penulis kemudian melanjutkan Pendidikan S2 ke IPB university jurusan Ilmu Komputer pada tahun 2012 dan berhasil lulus dalam waktu dua tahun di 2014. Pada tahun 2023 sampai sekarang penulis melanjutkan studi program Doktor program studi Ilmu Komputer di IPB University. Penulis memiliki kepakaran di bidang Data *Science*. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti di bidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI. Selain peneliti, penulis juga aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini.

Email Penulis: [inna.sabily@mercubuana.ac.id](mailto:inna.sabily@mercubuana.ac.id)

# BAB 14

## *TESTING DAN DEBUGGING*

**Andi Saputra, S.Kom, M.Kom**  
Universitas Putra Indonesia “YPTK” Padang

### **Pendahuluan**

Kunci dalam memastikan bahwa perangkat lunak yang dihasilkan berkualitas tinggi dan dapat diandalkan. Dalam tulisan ini, kami akan menjelajahi kedua konsep tersebut secara mendalam, dari pengertian dasar hingga teknik-teknik canggih yang digunakan dalam praktik industri.

Pengembangan perangkat lunak adalah proses yang kompleks, di mana setiap langkah memerlukan perhatian yang seksama untuk memastikan bahwa produk akhir memenuhi standar kualitas yang tinggi. Dalam dunia yang terus berkembang dengan cepat ini, pengguna memiliki harapan yang tinggi terhadap perangkat lunak yang mereka gunakan, sehingga kualitas dan keandalan menjadi kunci untuk meraih keberhasilan. Dua aspek yang sangat penting dalam upaya mencapai tujuan ini adalah *testing* dan *debugging*.

Pengaruh perangkat lunak telah menjadi salah satu bidang yang paling dinamis dan inovatif dalam dunia teknologi informasi. Dari aplikasi sederhana hingga sistem yang kompleks, perangkat lunak mendefinisikan cara kita berinteraksi dengan teknologi modern. Namun, di balik kemudahan penggunaan yang kita nikmati, ada proses yang kompleks dan sering kali rumit yang melibatkan pengembangan, pengujian, dan pemeliharaan perangkat lunak tersebut.

menciptakan situasi yang dapat direproduksi di mana *bug* muncul secara konsisten.

## 2. Pencarian *Bug*

Selanjutnya, Anda harus menemukan lokasi atau sebab-sebab potensial dari *bug* tersebut. Ini bisa dilakukan dengan memeriksa kode secara manual atau dengan menggunakan alat bantu *debugging*.

## 3. Analisis Kode

Periksa kode Anda dengan teliti untuk menemukan kemungkinan kesalahan, seperti kesalahan sintaks, logika yang salah, atau penggunaan variabel yang tidak tepat.

## 4. Pemantauan Variabel

Selama proses eksekusi, perhatikan nilai variabel-variabel yang terlibat. Kadang-kadang, nilai yang tidak terduga atau tidak valid dari variabel dapat menjadi petunjuk penting untuk menemukan *bug*.

## 5. Pengujian

Setelah Anda mengidentifikasi potensi lokasi *bug*, uji hipotesis Anda dengan memperbaiki kode dan menjalankannya kembali. Terus iterasi proses ini sampai *bug* teratasi.

## 6. Pemecahan *Bug*

Ketika Anda menemukan dan memperbaiki *bug*, pastikan untuk menguji kembali program secara menyeluruh untuk memastikan bahwa perubahan yang Anda lakukan tidak mempengaruhi fungsionalitas lainnya.

## 7. Dokumentasi

Terakhir, jangan lupa untuk mendokumentasikan *bug* yang ditemukan, langkah-langkah yang diambil untuk memperbaikinya, dan pelajaran yang dipetik dari pengalaman *debugging* tersebut. Ini akan membantu Anda dan orang lain di masa depan jika masalah serupa muncul lagi.

*Debugging* bisa menjadi proses yang menantang dan memakan waktu, tetapi merupakan keterampilan yang sangat penting bagi seorang pengembang perangkat lunak. Dengan latihan dan kesabaran, Anda akan menjadi lebih terampil dalam menemukan dan memperbaiki *bug*.

## **Daftar Pustaka**

- Edy Supriyanto (2005). *Macam – Macam Testing Sistem Informasi*. Fakultas Teknologi Informasi, Universitas Stikubank Semarang .  
Jurnal Teknologi Informasi DINAMIK Volume X, No. 1, Januari 2005 : 45-49
- PAN, Jiantao. *Software testing. Dependable Embedded Systems*, 1999, 5.2006: 1.
- Luo, L. (2001). *Software testing techniques. Institute for software research international Carnegie mellon university Pittsburgh, PA, 15232(1-19)*, 19.
- Singh, S. K., & Singh, A. (2012). *Software testing*. Vandana Publications.

## PROFIL PENULIS



### **Andi Saputra, S.Kom., M.Kom.**

Memiliki pengetahuan yang dasar dan intermediate seputar teknologi adalah yang yang sangat menarik. Dengan berdasarkan pengetahuan penulis dan minat bakat sejak bersekolah di sekolah kejuruan dengan jurusan Teknik Informatika, saya kemudian melanjutkan kuliah S1 Teknik Informatika di Universitas lancang kuning, setelah itu bekerja di berbagai tempat dengan bidang Komputer/ Teknik Informatika, setelah memiliki pengalaman yang sudah lebih dari 10 Tahun saya kemudian melanjutkan Studi S2 di universitas YPTK padang. Pengalaman dalam mendalami dunia teknologi yang semakin cepat berkembang membuat saya terus melakukan update ilmu agar menjadi pioner dan berkontribusi dalam menyampaikan ilmu yang ada. Saya memiliki kepakaran di bidang Jaringan dan Data Science. Dan untuk mewujudkan karier sebagai dosen profesional, saya juga memiliki pengalaman di bidang keguruan dan juga di jurusan multimedia pada minat bakat videography. Selain itu dengan bekerja di anak perusahaan PT Telkom saya juga memiliki pengangalaman teknis di bidang Jaringan baik itu Fiber Optik dan Juga Radio telekomunikasi.

Email Penulis: [ndi.saputra@gmail.com](mailto:ndi.saputra@gmail.com)

# BAB 15

## PENGGUNAAN PYTHON

### UNTUK PENGEMBANGAN

#### APLIKASI

**Roy Mubarak, S.T., M.Kom.**  
Universitas Mercu Buana

#### **Pendahuluan**

Python adalah bahasa pemrograman yang serbaguna dan mudah dipelajari, sehingga banyak digunakan dalam pengembangan aplikasi. Berikut adalah beberapa kegunaan Python dalam pengembangan aplikasi.

#### ***Web Development***

Python digunakan dalam pembuatan aplikasi web dengan kerangka kerja seperti Django dan Flask. Dengan kelebihan mudah dipelajari dan memiliki banyak pustaka, Python menjadi pilihan populer untuk pengembangan web.

Python digunakan secara luas dalam pengembangan web karena kemudahan penggunaannya dan kerangka kerja yang kuat. Berikut ini beberapa alasan mengapa Python populer dalam pengembangan web:

#### **1. Kerangka Kerja yang Populer**

Python memiliki beberapa kerangka kerja web yang populer dan kuat seperti Django, Flask, dan Pyramid. Django, misalnya, adalah

1. Sensor dan Perangkat Pemantauan, perangkat IoT dilengkapi dengan sensor yang dapat mengukur berbagai parameter seperti suhu, kelembaban, tekanan, gerakan, dan banyak lagi. Data dari sensor ini digunakan untuk memantau kondisi lingkungan atau perilaku perangkat.
2. Koneksi Jaringan, perangkat IoT terhubung ke internet melalui berbagai teknologi jaringan, termasuk Wi-Fi, Bluetooth, Zigbee, atau seluler. Ini memungkinkan perangkat untuk mentransfer data secara nirkabel dan berkomunikasi dengan platform lain atau pengguna.
3. Platform Komputasi, data yang dikumpulkan oleh perangkat IoT dikirim ke platform komputasi seperti cloud atau server lokal untuk analisis dan pengolahan lebih lanjut. Di sini, data dapat diolah untuk menghasilkan wawasan yang berarti atau untuk mengambil keputusan otomatis.
4. Aplikasi dan Antarmuka Pengguna, pengguna berinteraksi dengan perangkat IoT melalui aplikasi seluler, perangkat lunak desktop, atau antarmuka web. Ini memungkinkan pengguna untuk memantau kondisi perangkat, mengontrolnya, atau menerima pemberitahuan tentang peristiwa penting.
5. Keamanan dan Privasi, karena IoT melibatkan pengumpulan dan pertukaran data yang sensitif, keamanan dan privasi menjadi perhatian utama. Perlindungan data, enkripsi, dan autentikasi yang kuat diperlukan untuk mencegah akses tidak sah atau penyalahgunaan informasi.
6. Skalabilitas dan Interoperabilitas, IoT harus dirancang untuk mendukung pertumbuhan skala yang besar dan integrasi dengan berbagai sistem dan platform lain. Standar terbuka dan protokol komunikasi yang disepakati secara luas memungkinkan interoperabilitas antarperangkat dan perangkat lunak.

*Internet of Things* memiliki aplikasi luas dalam berbagai industri, termasuk manufaktur, kesehatan, transportasi, energi, pertanian, dan rumah pintar. Ini memberikan peluang untuk meningkatkan efisiensi, produktivitas, dan kualitas hidup, sambil memungkinkan inovasi baru dan model bisnis yang lebih adaptif. Namun, tantangan seperti keamanan, privasi, dan kompleksitas sistem perlu diatasi untuk

memaksimalkan potensi IoT secara keseluruhan. Dengan beragam kegunaan ini, Python telah menjadi salah satu bahasa pemrograman yang paling populer dan banyak digunakan di dunia.

\*\*\*\*\*

## Daftar Pustaka

- Beazley, David, and Brian K. Jones. "Python Cookbook." O'Reilly Media, 2013.
- Dawson, Michael. "Python Programming for the Absolute Beginner." Course Technology PTR, 2010.
- Downey, A. (2015). Think Python: How to Think Like a Computer Scientist. O'Reilly Media.
- Lutz, M. (2013). Learning Python: Powerful Object-Oriented Programming. O'Reilly Media.
- Lutz, M. (2014). Python Pocket Reference. O'Reilly Media.
- Matthes, E. (2019). Python Crash Course: A Hands-On, Project-Based Introduction to Programming. No Starch Press.
- McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. O'Reilly Media.
- Pilgrim, Mark. "Dive Into Python 3." Apress, 2009.
- Slatkin, B. (2015). Effective Python: 90 Specific Ways to Write Better Python. Addison-Wesley Professional.
- Sweigart, A. (2015). Automate the Boring Stuff with Python: Practical Programming for Total Beginners. No Starch Press.
- VanderPlas, Jake. "Python Data Science Handbook: Essential Tools for Working with Data." O'Reilly Media, 2016.

## PROFIL PENULIS



### **Roy Mubarak, S.T., M.Kom.**

Penulis seorang pendidik dengan pengalaman luas dalam mengajar dan melatih guru. Ia memiliki minat khusus dalam metodologi pembelajaran dan telah menerbitkan artikel dan buku tentang inovasi dalam pembelajaran. Diana juga memiliki keahlian dalam desain pembelajaran berbasis proyek dan implementasi teknologi dalam pembelajaran, Ketertarikan penulis terhadap ilmu komputer dimulai pada tahun 2015 silam. Penulis memiliki kepakaran dibidang *Software Engineering*. Dan untuk mewujudkan karir sebagai dosen profesional, penulis pun aktif sebagai peneliti dibidang kepakarannya tersebut. Beberapa penelitian yang telah dilakukan didanai oleh internal perguruan tinggi dan juga Kemenristek DIKTI. Selain peneliti, penulis juga aktif menulis buku dengan harapan dapat memberikan kontribusi positif bagi bangsa dan negara yang sangat tercinta ini. Dengan tujuan mengembangkan solusi perangkat lunak inovatif yang memecahkan tantangan bisnis dengan menggunakan keterampilan pemrograman dan pengetahuan teknologi terkini. Terus meningkatkan kemampuan teknis dan kepemimpinan melalui pengalaman praktis dan pembelajaran berkelanjutan.

WA : 082818925852

Email Penulis: [roy.mubarak@mercubuana.ac.id](mailto:roy.mubarak@mercubuana.ac.id).

# BAHASA PEMROGRAMAN PYTHON



Python adalah bahasa pemrograman tingkat tinggi yang dirancang dengan filosofi yang menekankan keterbacaan kode dan sintaks yang bersih. Dibuat oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991, Python telah berkembang menjadi salah satu bahasa pemrograman yang paling populer di dunia. Python adalah bahasa pemrograman yang sering digunakan dalam pengembangan aplikasi web, perangkat lunak, ilmu data, dan *machine learning* (ML). Para pengembang memilih Python karena efisiensinya, kemudahan belajar, dan kemampuannya berjalan di berbagai platform. Python dapat diunduh secara gratis, terintegrasi dengan baik ke semua jenis sistem, dan mempercepat proses pengembangan. Buku ini disusun secara sistematis untuk memudahkan pembaca memahami konsep dasar hingga teknik-teknik lanjutan dalam pemrograman Python yang terdiri dari: pengenalan python, variabel dan tipe data, operator, fungsi, input dan output, struktur data lanjutan, *file handling*, paket standar python, pemrograman fungsional, *concurrency*, *GUI programming*, *web development*, *data science* dan *machine learning*, testing dan debugging serta penggunaan python untuk pengembangan aplikasi.