

DASAR DASAR PYTHON 3.8/3.9

**SEBUAH PENGANTAR PRAKTIS
PEMROGRAMAN PYTHON
UNTUK PEMROGRAMAN ILMIAH**



MAULANA MALIK NASHRULLOH, S.Si.

Dr. BRIAN RAHARDI, S.Si., M.Sc

Editor:

Dr. Brian Rahardi, S.Si., M.Sc

Cover Desainer:

Moh. Ali Anwar

Layouter:

Maulana Malik Nashrulloh, S.Si.

Siti Imama Khoiriyah

(INBIO Indonesia)

ISBN:

978-623-98383-2-4

Publisher: Future Science

Redaksi:

Perum Sarimadu II B3 No.09

Pakisaji, Kab.Malang, Jawa Timur, Indonesia 65162

Email: admin@futuresciencepress.com

DAFTAR ISI

KATA PENGANTAR	I
DISCLAIMER	IV
DAFTAR ISI	VI
PROFIL PENULIS	XVIII
KONVENSI PENULISAN DAN PENYAJIAN INFORMASI BUKU	XIX
Konvensi Teks	xix
Konvensi Spasi	xx
Konvensi Penulisan	xx
Konvensi Penyajian Informasi Khusus	xxi
BAB 1 PENGENALAN PYTHON	1
1.1. Apa itu Python?	1
1.2. Sejarah Python	2
1.3. Kenapa Harus Mempelajari Python?	6
BAB 2 INSTALASI ANACONDA3 DAN PYTHON 3.8/3.9 PADA WINDOWS	11
2.1. Instalasi Python 3.8	11
2.2. Instalasi Python 3.9 Melalui <i>Tarball</i>	17
BAB 3 PRIMER PYTHON 3.8/3.9	21
3.1. Interpreter Python	21
3.2. Pengenalan Spyder	23
3.3. Mengenal Penulisan Kode Pemrograman Python	24
3.3.1. Penulisan Baris Perintah	25
3.3.2. Indentasi	27
3.3.3. Memberi Komentar ke Dalam Kode	29

3.3.4.	Penulisan Variabel	30
3.4.	Tipe Data	32
3.4.1.	Tipe None	32
3.4.2.	Tipe String	33
3.4.3.	Tipe Numerik	33
3.4.4.	Tipe Boolean	34
3.4.5.	Tipe Data Terstruktur	34
3.4.6.	Tipe File	34
3.5.	Escape Sequences	35
3.6.	%-Formatting	36
3.7.	Operator	36
3.7.1.	Operator Penugasan (<i>Assignment Operators</i>)	37
3.7.2.	Operator Aritmatika (<i>Arithmetic Operators</i>)	38
3.7.3.	Operator Relasional (<i>Relational Operators</i>)	39
3.7.4.	Operator Boolean (<i>Boolean Operators</i>)	39
3.7.5.	Operator Bitwise (<i>Bitwise Operators</i>)	40
3.7.6.	Operator Penugasan Bitwise (<i>Bitwise Assignment Operations</i>)	41
3.7.7.	Operator Kondisional (<i>Conditional Operator</i>)	41
3.7.8.	Operator Objek (<i>Object Operator</i>)	42
3.7.9.	Operator String (<i>String Operators</i>)	43
3.7.10.	Operator Penugasan Ekspresi (<i>Expression Assignment Operator</i>)	43
3.8.	Input dan Output	44
3.9.	Fungsi	46
3.10.	Import	47
BAB 4 SELEKSI KONDISI		50
4.1.	Seleksi Kondisi Satu Kondisi Melalui Statemen <code>if...</code>	50
4.2.	Seleksi Kondisi Dua Kondisi Melalui Statemen <code>if... else...</code>	51
4.3.	Seleksi Kondisi Tiga atau Lebih Kondisi Melalui Statemen <code>if... elif... else...</code>	53

4.4.	Seleksi Kondisi dengan Kondisi Bersarang Melalui Nested <code>if</code>	56
4.5.	Statemen <code>pass</code> dalam Seleksi Kondisi	60
BAB 5	PERULANGAN	61
5.1.	Perulangan dengan Statemen <code>for...</code>	61
5.1.1.	Penggunaan <code>for...</code> untuk Melakukan Iterasi	62
5.1.2.	Penggunaan <code>for...</code> dengan Tipe Data String dan Tipe Data Terstruktur	67
5.1.3.	Mengakses Nilai Indeks dengan Metode <code>enumerate</code>	69
5.2.	Perulangan dengan Statemen <code>for... else...</code>	70
5.3.	Perulangan dengan Statemen <code>while...</code>	71
5.4.	Perulangan dengan Statemen <code>while... else...</code>	73
5.5.	Statemen <code>break</code> dalam Perulangan	74
5.5.1.	Penggunaan <code>break</code> pada Perulangan <code>for...</code>	74
5.5.2.	Penggunaan <code>break</code> pada Perulangan <code>while...</code>	77
5.6.	Statemen <code>continue</code> dalam Perulangan	80
5.6.1.	Penggunaan <code>continue</code> Pada Perulangan <code>for...</code>	80
5.6.2.	Penggunaan <code>continue</code> Pada Perulangan <code>while...</code>	84
5.7.	Tidak Lazim Ada <code>pass</code> dalam Perulangan	89
BAB 6	PENANGANAN KESALAHAN	91
6.1.	Memahami Konsep Kesalahan dalam Pemrograman Python	91
6.2.	Membangkitkan Kesalahan dengan <code>raise</code>	102
6.3.	Asersi dengan <code>assert</code>	103
6.4.	Mengenal Blok Penanganan Kesalahan <code>try...except... else... finally...</code> dan turunannya	105
6.5.	Penanganan Kesalahan dengan <code>try... except... else... finally...</code>	107
6.6.	Penanganan Kesalahan dengan <code>try... except...</code>	111
6.7.	Penanganan Kesalahan dengan <code>try... except...else...</code>	112
6.8.	Penanganan Kesalahan dengan <code>try... except...finally...</code>	114
6.9.	Tetap Selalu Konstan dengan <code>try...finally...</code>	115
6.10.	Menggunakan <code>except... as... :</code>	118

6.11. Bisakah Kita Menangkap Semua Kesalahan?	119
6.12. Penanganan Banyak Kesalahan dengan Blok <code>try... except...</code> Multi Eksepsi	121
6.13. Beberapa Catatan Tentang Blok Penanganan Kesalahan	124
6.13.1. Berhati – hatilah dalam menggunakan <code>except ... as...</code> :	124
6.13.2. Menggunakan Pustaka <code>traceback</code>	126
6.13.3. Mendayagunakan <code>raise</code> dan <code>assert</code>	127
BAB 7 MANIPULASI STRING	129
7.1. Mengetahui String dalam Python	129
7.2. Operasi String	129
7.3. Fungsi Bawaan (<i>Built-in</i>) Objek String	131
7.3.1. Mengetahui Nilai Angka dari Karakter dengan <code>ord()</code>	131
7.3.2. Mengetahui Nilai Karakter dari Angka dengan <code>chr()</code>	132
7.3.3. Mengkonversi Semua Tipe Data menjadi string dengan <code>str()</code>	133
7.3.4. Menentukan Panjang String dengan <code>len()</code>	134
7.4. Pengindeksan String (<i>String Indexing</i>)	135
7.5. Mengiris String (<i>String Slicing</i>)	138
7.6. Memodifikasi String	141
7.7. Memformat String dan Menginterpolasi Variabel ke dalam String	143
7.7.1. Menggunakan pola cetak	144
7.7.2. Konkatenasi string/penggabungan string (<i>String concatenation</i>).	145
7.7.3. Menggunakan %-formatting	147
7.7.3.1. %parameter di %-formatting	148
7.7.3.2. Satu Variabel untuk Satu Parameter %-formatting	150
7.7.3.3. Bekerja dengan Banyak Parameter di %-formatting	151
7.7.3.4. Bekerja dengan Variabel Substitusi di %-formatting	152
7.7.4. Menggunakan metode <code>str.format()</code>	154
7.7.4.1. Bekerja dengan Variabel Substitusi di <code>str.format()</code>	157
7.7.4.2. :parameter di <code>str.format()</code>	158
7.7.5. Menggunakan f-string	174

7.7.5.1.	Keterbatasan f-string	177
7.7.5.2.	:#parameter di f-string	183
7.7.5.3.	Parameter padding di f-string	186
7.7.6.	Menggunakan template string standar dari pustaka string	187
7.7.6.1.	Substitute() vs safe_substitute()	191
7.8.	Menggunakan Escape Sequence	193
BAB 8	TIPE DATA TERSTRUKTUR	196
8.1.	Perbedaan Array, List, Tuple, Dictionary, Set, dan Frozenset	196
8.2.	Array Dasar dengan Pustaka array	197
8.2.1.	Membuat Objek array dengan Pustaka array	198
8.2.2.	Metode Objek array dengan Pustaka array	199
8.2.3.	Mengekstrak Elemen dari Objek array dengan Pustaka array	203
8.3.	Array Dasar dengan List	204
8.3.1.	Membuat Array 1-Dimensi dengan List	204
8.3.2.	Membuat Array Multi-Dimensi dengan List	205
8.3.3.	Mengekstrak Elemen dari Objek array dengan List	211
8.3.4.	Metode Array dengan List	212
8.4.	Bekerja dengan List	212
8.4.1.	Membuat List	212
8.4.2.	Mengekstrak Elemen dari List	213
8.4.3.	Membuat List Komposit	218
8.4.4.	Operasi Penugasan, Relasional, dan Logika Elemen List	219
8.4.5.	Metode List	221
8.5.	Bekerja dengan Tuple	225
8.5.1.	Membuat Tuple	225
8.5.2.	Mengekstrak Elemen dari Tuple	227
8.5.3.	Membuat Tuple Komposit	231
8.5.4.	Operasi Penugasan, Relasional, dan Logika Elemen Tuple	232
8.5.5.	Metode Tuple	234

8.6.	Bekerja dengan Dictionary	236
8.6.1.	Membuat Dictionary	237
8.6.2.	Mengekstrak Elemen dari Dictionary	238
8.6.3.	Operasi Penugasan, Relational, dan Logika Elemen Dictionary	238
8.6.4.	Menggabung dan Mengupdate Dictionary	240
8.6.4.1.	Menggunakan Operator Union untuk Menggabung dan <i>In-union</i> untuk Mengupdate Dictionary (Hanya untuk Python >= 3.9!)	240
8.6.4.2.	Bagaimana Menggabung dan Mengupdate Dictionary untuk Python <= 3.9?	242
8.6.4.3.	Perbandingan Hasil dari Metode – Metode Penggabungan Dictionary	248
8.6.5.	Metode Dictionary	248
8.7.	Bekerja dengan Data Himpunan (Set dan Frozenset)	252
8.7.1.	Membuat Set	253
8.7.2.	Membuat Frozenset	254
8.7.3.	Mengekstrak Elemen Set dan Frozenset	255
8.7.4.	Keunikan (Uniqueness) Set dan Frozenset	257
8.7.5.	Metode Set dan Frozenset	258
8.7.5.1.	Operasi Dasar Himpunan pada Set dan Frozenset	260
8.7.5.2.	Operasi Manipulasi Pada Set	262
8.7.6.	Tidak ada Operasi Penugasan, Relasional, dan Logika Elemen Set dan Frozenset	263
BAB 9	FILE DAN DIREKTORI	264
9.1.	Memahami File	264
9.2.	Aman Dalam Bekerja dengan File	266
9.3.	Operasi File Dasar pada Python	268
9.3.1.	Fungsi <code>open()</code> untuk membuat ,menulis, dan membaca file	269
9.3.2.	Membaca file dengan <code>file.read()</code>	276
9.3.3.	Membaca file dengan <code>file.readline()</code>	278
9.3.4.	Menulis file dengan <code>file.write()</code>	280
9.3.4.1.	Menambah isi file dengan <code>append</code>	280
9.3.4.2.	Menulis isi file dengan <code>write</code>	282

9.3.5.	Mengoperasikan dua file dengan operasi berbeda (satu baca, satu tulis)	283
9.4.	Merekam Data Masukan ke Dalam File	284
9.5.	Operasi File Lanjutan pada Python	288
9.5.1.	Modul – Modul Esensial	289
9.5.2.	Mendapatkan Atribut File	289
9.5.2.1.	Mengakses Tipe File dan Hak Akses melalui Makna Rahasia <code>st_mode</code>	294
9.5.2.2.	Mengakses Data Waktu File	312
9.5.3.	Mengkopi File	314
9.5.4.	Memindah File	318
9.5.5.	Mengganti Nama File	320
9.5.6.	Menghapus File	321
9.5.7.	Menghapus File dengan Aman menggunakan <code>secure_delete</code>	324
9.5.8.	Terlalu Paranoid untuk Menghapus File? Gunakan <code>send2trash</code> !	326
9.6.	Bekerja dengan Direktori/Folder	329
9.6.1.	Membuat Direktori/Folder	329
9.6.1.1.	Membuat Direktori/Folder Tunggal	330
9.6.1.2.	Membuat Direktori/Folder Majemuk	332
9.6.2.	Mengkopi Folder	335
9.6.3.	Memindah Folder	338
9.6.4.	Mengganti Nama Folder	339
9.6.5.	Menghapus Folder	341
9.6.6.	Menghapus Folder dengan Aman menggunakan <code>secure_delete</code>	346
9.6.7.	Menggunakan <code>send2trash</code> untuk Membuang Folder ke Recycle Bin/Trash	349
9.6.8.	Mendapatkan Daftar Direktori	350
9.6.9.	Memfilter Daftar Direktori	354
9.6.10.	Menampilkan Struktur Pohon Direktori	357
9.7.	Mencocokkan Nama File dengan Pola (<i>Globbing</i>)	362
9.7.1.	<i>Globbing</i> dengan Metode <code>endswith()</code> dan <code>startswith()</code>	363
9.7.2.	<i>Globbing</i> dengan <code>fnmatch.fnmatch()</code>	366

9.7.3.	<i>Globbing</i> dengan Modul <code>glob</code>	369
9.7.4.	<i>Globbing</i> dengan Fungsi <code>pathlib.Path.glob()</code> dan <code>pathlib.Path.rglob()</code>	372
9.8.	Menelusuri Direktori dengan <code>os.walk()</code>	375
9.8.1.	Penelusuran Normal	376
9.8.2.	Menelusur Link Simbolik	379
9.8.3.	Telusur Balik	382
9.9.	Dasar – Dasar Pengarsipan dan Kompresi File di Python	384
9.9.1.	Pengarsipan dengan <code>shutil.make_archive()</code>	385
9.9.2.	Memekarkan arsip dengan <code>shutil.unpack_archive()</code>	387
BAB 10	FUNGSI	389
10.1.	Mengenal Terminologi Dasar Tentang Fungsi	390
10.2.	Membuat Fungsi dengan <code>def</code>	391
10.3.	Membuat Fungsi Sederhana Dengan Pengembalian	392
10.3.1.	Membuat Fungsi Sederhana Tanpa Pengembalian	395
10.3.2.	Beda Fungsi Dengan Pengembalian dengan Fungsi Tanpa Pengembalian	395
10.4.	Statemen <code>lambda</code>	397
10.4.1.	<code>lambda</code> Sebagai Pembentuk Fungsi	397
10.4.2.	<code>lambda</code> Sebagai Argumen Suatu <i>Higher-order Function</i>	399
10.5.	Ruang Lingkup Variabel/Cakupan Variabel (<i>Variable Scope</i>)	402
10.5.1.	Kata Kunci <code>global</code> untuk Mengakses Variabel Global	404
10.6.	Parameter dalam Fungsi	405
10.7.	Melewatkan Argumen dengan Kata Kunci	407
10.8.	Pass-by Value? Pass-by Reference? Bukan, itu Pass-by Object Reference!	411
10.8.1.	Pembuktian <i>Pass-by Object Reference</i> di Python	415
10.8.2.	Manipulasi? OK! Penugasan kembali (<i>Reassignment</i>)? No!	426
10.8.3.	Mungkinkah <i>Pass-by Reference</i> dan <i>Pass-by Value</i> dilakukan di Python?	429
10.9.	Argumen Default	432
10.10.	Jumlah Argumen yang Berubah	437
10.11.	Fungsi Bersarang (<i>Nested Function</i>): Fungsi dalam Fungsi	442

10.12. Decorator	450
10.13. <i>Overloading</i> Fungsi pada Python dengan <code>multipledispatch</code>	454
10.14. Fungsi Rekursif	459
10.15. Generator	468
BAB 11 OBJECT ORIENTED PROGRAMMING (OOP)	475
11.1. Memahami Konsep OOP Secara Umum	477
11.1.1. Empat Pilar OOP	477
11.1.1.1. Abstraksi	477
11.1.1.2. Enkapsulasi	478
11.1.1.3. Inheritance	479
11.1.1.4. Polimorfisme	479
11.1.2. Fitur bersama OOP dan Non-OOP	480
11.1.2.1. Variabel	480
11.1.2.2. Prosedur	480
11.1.3. Fitur bersama OOP dan Non-OOP	481
11.1.3.1. Kelas dan Objek (Class and Object)	481
11.1.3.2. <i>Class-based</i> dan <i>Prototype-based Style Orientation</i>	481
11.1.1.1. <i>Dynamic Dispatch</i>	482
11.1.1.2. Delegasi	483
11.1.1.3. Rekursi Terbuka (<i>Open Recursion</i>)	484
11.2. Bekerja dengan Kelas dan Objek	484
11.2.1. Mendefinisikan Kelas dan Metode di Dalam Kelas	484
11.2.2. Membuat Objek dari Kelas	488
11.2.3. Membuat dan Menggunakan Atribut Objek	489
11.2.4. Menggunakan Metode Objek	490
11.2.5. Membuat Kelas dan Objek	490
11.3. Variabel Statis dan <i>Instance</i>	494
11.4. Lebih Lanjut dengan Konstruktor	495
11.4.1. <code>__init__()</code>	496

11.4.2.	<i>Default</i> Konstruktor untuk Kelas <i>Default</i>	500
11.4.3.	<code>__new__()</code>	504
11.4.4.	<code>__init__()</code> dan <code>__new__()</code> dalam Satu Kelas	511
11.5.	Destruktor	515
11.6.	Lebih Lanjut dengan Metode (<i>Method</i>)	519
11.6.1.	Fungsi Internal	519
11.6.2.	Metode Statis dan Metode Kelas	522
11.7.	<i>Dunder</i>	525
11.8.	Melindungi Data dan Proses dari Intervensi dengan Enkapsulasi Melalui Pemberian Pemodifikasi Akses (<i>Access Modifier</i>)	530
11.8.1.	Public vs Protected vs Private	531
11.8.2.	Bekerja dengan <i>Getter</i> , <i>Setter</i> , dan <i>Deleter</i>	542
11.8.3.	Bekerja dengan <i>Property</i>	546
11.9.	Pewarisan (<i>Inheritance</i>)	552
11.9.1.	Pewarisan Tunggal (<i>Single Inheritance</i>)	553
11.9.2.	Fungsi <code>super()</code> untuk Pewarisan Tunggal	557
11.9.3.	Pewarisan Jamak (<i>Multiple Inheritance</i>)	563
11.9.4.	Melacak Instance dan Subclass	564
11.9.5.	Mencegah Pewarisan Kelas	569
11.9.6.	Bekerja dengan Abstrak dan <i>Metaclass</i>	572
11.10.	Komposisi	579
11.11.	Polimorfisme	581
11.12.1.	Single Dispatch	586
11.12.2.	Multiple Dispatch	590
11.12.2.1.	Menggunakan <code>multipledispatch</code>	590
11.12.2.2.	Menggunakan <code>plum-dispatch</code>	593
11.12.2.3.	Menggunakan <code>multimethod</code>	596
BAB 12	PEMROGRAMAN MODULER DAN MANAJEMEN PAKET	600
12.1.	Konsep Modul, Paket, dan Pustaka dalam Python	600

12.2. Membuat Modul	603
12.3. Beda Modul dengan Program di Python	605
12.4. Jangan Berlebihan dalam Melakukan Impor Melingkar!	608
12.5. Membuat Pustaka Sendiri (Studi Kasus Uji Kimia Tanah)	609
12.5.1. Merancang dan Mengkode modul	610
12.5.2. File <code>__init__.py</code>	613
12.5.3. File <code>__main__.py</code> (Opsional)	615
12.5.4. Mengorganisasi Modul untuk Pustaka	616
12.6. Menggunakan Paket	617
12.7. Membuat Distribusi Paket Sendiri	620
12.7.1. Membuat <i>Wheel Package</i>	620
12.7.1.1. LICENSE	621
12.7.1.2. <code>pyproject.toml</code>	622
12.7.1.3. README.md	623
12.7.1.4. <code>setup.py</code>	623
12.7.1.5. Konfigurasi Folder Build	625
12.7.1.6. Persiapan Sebelum Membangun <i>wheel package</i>	626
12.7.1.7. Membangun <i>Wheel Package</i>	627
12.7.2. Instalasi <i>Wheel Package</i> Secara <i>Offline</i>	629
12.7.3. Penghapusan Instalasi <i>Wheel Package</i>	630
12.7.4. Membuat <i>Conda Package</i> dengan <code>wheel2conda</code>	631
12.7.5. Instalasi <i>Conda Package</i> Secara <i>Offline</i>	633
12.7.6. Penghapusan Instalasi <i>Conda Package</i>	634
BIBLIOGRAFI	635
DAFTAR PUSTAKA	637
APPENDIX A INSTALASI ANACONDA3 PYTHON 3.9.1 DI UBUNTU 20.04 LTS, LINUX MINT	
20, ATAUPUN DERIVATIFNYA	644
A.1. Instalasi Paket Esensial Anaconda3	645

A.2. Instalasi Anaconda3 di Linux	646
A.3. Instalasi Python 3.9.1 pada Anaconda3 di Linux	650
A.4. Instalasi Spyder pada Anaconda3 di Linux	654

INBIO Indonesia

BAB 1 PENGENALAN PYTHON

1.1. Apa itu Python?

Python adalah bahasa pemrograman yang diciptakan oleh Guido van Rossum, seorang ahli matematika dan ilmu komputer berkebangsaan Belanda. Python lahir pada 1991, setelah dikembangkan sejak tahun 1989. Python dinamakan demikian sejatinya tidak mengacu kepada jenis ular sanca (*Python spp.*), melainkan berdasar kepada acara sitkom *Monty Python Flying Circus* (Van Rossum & De Boer, 1991) yang dibawakan oleh grup lawak dari Inggris Monty Python yang pernah tenar pada 1969 – 1974.

Python adalah bahasa pemrograman yang tergolong ke dalam *interpreted*, *high-level*, dan *general-purpose programming language*. *Interpreted* karena tidak seperti bahasa pemrograman lain yang memerlukan kompilasi untuk menjalankan kode perintah, Python menggunakan perangkat interpreter untuk menerjemahkan kode perintah sehingga dipahami oleh mesin, sehingga mesin dapat menjalankan instruksi sebagaimana dikehendaki programmer. Karenanya, Python merupakan *scripted language* yang instruksinya tersimpan dalam bentuk skrip – skrip, bukan program executable yang siap jalan. *High level* karena merupakan bahasa pemrograman yang tinggi (bisa dipahami manusia kata – katanya), berlawanan dengan bahasa pemrograman *low level* (yang hanya dipahami oleh mesin kata – katanya). *General purpose* karena penggunaannya luas untuk membuat program yang beragam (umum) gunanya dan tidak terbatas oleh kepentingan atau tujuan tertentu.

Sebagai bahasa yang terinterpretasi, tentu Python memiliki kelebihan utamanya adalah dari segi portabilitas dan kemudahan untuk menjalankan. Skrip Python bisa jalan dimana saja selama ada interpreter dan pustaka yang digunakan oleh skrip tersebut. *Executable* Interpreter Python telah tersedia untuk Windows, Linux, MacOS siap tersedia untuk diinstal dan dijalankan. Untuk yang

Informasi Selengkapnya

Silahkan order dan hubungi kami:

Website INBIO: <https://inbio-indonesia.org/>

Whats App atau Handphone : +62822-2987-2305



ISBN 978-623-98383-2-4 (PDF)



9 786239 838324