

INBIO
research made easy



DASAR - DASAR PYTHON 3.10

Penulis

Maulana Malik Nashrulloh, S.Si., Ph.D.
Dr. Brian Rahardi, S.Si., M.Sc.

Authors:

Maulana Malik Nashrulloh, S.Si., Ph.D.

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

Editor:

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

Cover Desainer:

Erik Febriyanto

Layouter:

Maulana Malik Nashrulloh, S.Si., Ph.D.

ISBN:

978-623-98383-6-2

Publisher:

Future Science

Redaksi:

Jl. Terusan Surabaya Gang 1A No. 71 RT.002/RW.005, Kel.
Sumbersari, Kec. Lowokwaru, Kota Malang Provinsi Jawa Timur

Email:

admin@futuresciencepress.com

DAFTAR ISI

KATA PENGANTAR	I
<i>DISCLAIMER</i>	III
PROFIL PENULIS	IV
KONVENSI PENULISAN DAN PENYAJIAN INFORMASI BUKU	V
Konvensi Teks	v
Konvensi Penulisan	v
Konvensi Penyajian Informasi Khusus	vi
DAFTAR ISI	VIII
BAB 1 PENGENALAN PYTHON	1
1.1. Apa itu Python?	1
1.2. Sejarah Python	2
1.3. Kenapa Harus Mempelajari Python?	5
BAB 2 INSTALASI PYTHON 3.10 PADA WINDOWS	9
2.1. Instalasi Python 3.10	9
2.2. Instalasi Spyder	13
2.3. Instalasi Jupyter Notebook/JupyterLab	16
BAB 3 PRIMER PYTHON 3.10	20
3.1. Interpreter Python	20
3.2. Pengenalan Singkat Lingkungan IDE	21
3.2.1. Spyder	21
3.2.2. Jupyter Notebook	22
3.2.3. JupyterLab	24
3.2.4. Google Colab	26
3.3. Mengenal Penulisan Kode Pemrograman Python	27

3.3.1.	Penulisan Baris Perintah	27
3.3.2.	Indentasi	28
3.3.3.	Memberi Komentar ke Dalam Kode	30
3.3.4.	Penulisan Variabel	31
3.4.	Tipe Data	32
3.4.1.	Tipe None	32
3.4.2.	Tipe <i>String</i>	32
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.	Operator	34
3.5.1.	Operator Penugasan (<i>Assignment Operators</i>)	35
3.5.2.	Operator Aritmatika (<i>Arithmetic Operators</i>)	36

3.5.3.	Operator Relasional (<i>Relational Operators</i>)	36
3.5.4.	Operator Boolean (<i>Boolean Operators</i>)	37
3.5.5.	Operator <i>Bitwise</i> (<i>Bitwise Operators</i>)	37
3.5.6.	Operator Penugasan Bitwise (<i>Bitwise Assignment Operations</i>)	38
3.5.7.	Operator Kondisional (<i>Conditional Operators</i>)	39
3.5.8.	Operator Objek (<i>Object Operator</i>)	39
3.5.9.	Operator <i>String</i> (<i>String Operators</i>)	39
3.5.10.	Operator Penugasan Ekspresi (<i>Expression Assignment Operators</i>)	40
3.6.	Input dan Output	41
3.7.	Fungsi	42
3.8.	Import	42
BAB 4	SELEKSI KONDISI	44
4.1.	Seleksi Kondisi Satu Kondisi Melalui Statemen <code>if...</code>	44
4.2.	Seleksi Kondisi Dua Kondisi Melalui Statemen <code>if... else...</code>	45
4.3.	Seleksi Kondisi Tiga atau Lebih Kondisi Melalui Statemen <code>if... elif... else...</code>	46
4.4.	Seleksi Kondisi dengan Kondisi Bersarang Melalui Nested <code>if</code>	48
4.5.	Statemen <code>pass</code> dalam Seleksi Kondisi	51
4.6.	Statemen <code>match... case...</code>	51
BAB 5	PERULANGAN	56
5.1.	Perulangan dengan Statemen <code>for...</code>	56
5.1.1.	Penggunaan <code>for...</code> untuk Melakukan Iterasi	56
5.1.2.	Penggunaan <code>for...</code> dengan Tipe Data Terstruktur	60
5.1.3.	Mengakses Nilai Indeks dengan <code>enumerate</code>	62
5.2.	Perulangan dengan Statemen <code>for... else...</code>	62
5.3.	Perulangan dengan Statemen <code>while...</code>	63
5.4.	Perulangan dengan Statemen <code>while... else...</code>	64
5.5.	Statemen <code>break</code> dalam Perulangan	65
5.5.1.	Penggunaan <code>break</code> pada Perulangan <code>for...</code>	65
5.5.2.	Penggunaan <code>break</code> pada Perulangan <code>while...</code>	67
5.6.	Statemen <code>continue</code> dalam Perulangan	69

5.6.1.	Penggunaan <code>continue</code> Pada Perulangan <code>for...</code>	69
5.6.2.	Penggunaan <code>continue</code> Pada Perulangan <code>while...</code>	72
5.7.	Tidak Lazim Ada <code>pass</code> dalam Perulangan	77
BAB 6 PENANGANAN KESALAHAN		78
6.1.	Memahami Konsep Kesalahan dalam Pemrograman Python	78
6.2.	Membangkitkan Kesalahan dengan <code>raise</code>	89
6.3.	Asersi dengan <code>assert</code>	90
6.4.	Mengenal Blok Penanganan Kesalahan <code>try... except... else... finally...</code> dan turunannya	91
6.5.	Penanganan Kesalahan dengan <code>try... except... else... finally...</code>	93
6.6.	Penanganan Kesalahan dengan <code>try... except...</code>	95
6.7.	Penanganan Kesalahan dengan <code>try... except... else...</code>	96
6.8.	Penanganan Kesalahan dengan <code>try... except... finally...</code>	97

6.9. Tetap Selalu Konstan dengan <code>try... finally...</code>	99
6.10. Menggunakan <code>except... as...:</code>	101
6.11. Bisakah Kita Menangkap Semua Kesalahan?	101
6.12. Penanganan Banyak Kesalahan dengan Blok <code>try... except... Multi Eksepsi</code>	103
6.13. Beberapa Catatan Tentang Blok Penanganan Kesalahan	105
6.13.1. Berhati – hatilah dalam Menggunakan <code>except ... as...:</code>	105
6.13.2. Menggunakan Pustaka <code>traceback</code>	106
6.13.3. Mendayagunakan <code>raise</code> dan <code>assert</code>	107
BAB 7 MANIPULASI <i>STRING</i>	110
7.1. Menenal <i>String</i> dalam Python	110
7.2. Operasi <i>String</i>	110
7.3. Fungsi Bawaan (<i>Built-in</i>) Objek <i>String</i>	111
7.3.1. Mengetahui Nilai Angka dari Karakter dengan <code>ord()</code>	112
7.3.2. Mengetahui Nilai Karakter dari Angka dengan <code>chr()</code>	112
7.3.3. Mengkonversi Semua Tipe Data menjadi <i>string</i> dengan <code>str()</code>	113
7.3.4. Menentukan Panjang <i>String</i> dengan <code>len()</code>	114
7.4. Pengindeksan <i>String</i> (<i>String Indexing</i>)	114
7.5. Mengiris <i>String</i> (<i>String Slicing</i>)	117
7.6. Memodifikasi <i>String</i>	119
7.7. Memformat <i>String</i> dan Menginterpolasi Variabel ke dalam <i>String</i>	121
7.7.1. Menggunakan pola cetak	121
7.7.2. Konkatenasi <i>string</i> /penggabungan <i>string</i> (<i>String concatenation</i>).	122
7.7.3. Menggunakan %-formatting	124
7.7.3.1. %parameter di %-formatting	124
7.7.3.2. Satu Variabel untuk Satu Parameter %-formatting	126
7.7.3.3. Bekerja dengan Banyak Parameter di %-formatting	127
7.7.3.4. Bekerja dengan Variabel Substitusi di %-formatting	128
7.7.4. Menggunakan metode <code>str.format()</code>	129
7.7.4.1. Bekerja dengan Variabel Substitusi di <code>str.format()</code>	131
7.7.4.2. :parameter di <code>str.format()</code>	131

7.7.5.	Menggunakan <i>f-string</i>	145
7.7.5.1.	Keterbatasan <i>f-string</i>	147
7.7.5.2.	:#parameter di <i>f-string</i>	152
7.7.5.3.	Parameter <i>padding</i> di <i>f-string</i>	154
7.7.6.	Menggunakan template <i>string</i> standar dari pustaka <i>string</i>	155
7.7.6.1.	<code>substitute()</code> vs <code>safe_substitute()</code>	158
7.8.	Menggunakan <i>Escape Sequence</i>	160
BAB 8 TIPE DATA TERSTRUKTUR		163
8.1.	Perbedaan Array, List, Tuple, Dictionary, Set, dan Frozenset	163
8.2.	Array Dasar dengan Pustaka <code>array</code>	164
8.2.1.	Membuat Objek Array dengan Pustaka <code>array</code>	164
8.2.2.	Metode Objek array dengan Pustaka <code>array</code>	165
8.2.3.	Mengekstrak Elemen dari Objek array dengan Pustaka <code>array</code>	169

8.3. Array Dasar dengan List	169
8.3.1. Membuat Array 1-Dimensi dengan List	169
8.3.2. Membuat Array Multi-Dimensi dengan List	170
8.3.3. Mengekstrak Elemen dari Objek array dengan List	174
8.3.4. Metode Array dengan List	175
8.4. Bekerja dengan List	175
8.4.1. Membuat List	175
8.4.2. Mengekstrak Elemen dari List	176
8.4.3. Membuat List Komposit	179
8.4.4. Operasi Penugasan, Relasional, dan Logika Elemen List	180
8.4.5. Metode List	181
8.5. Bekerja dengan Tuple	184
8.5.1. Membuat Tuple	184
8.5.2. Mengekstrak Elemen dari Tuple	186
8.5.3. Membuat Tuple Komposit	189
8.5.4. Operasi Penugasan, Relasional, dan Logika Elemen Tuple	190
8.5.5. Metode Tuple	191
8.6. Bekerja dengan Dictionary	192
8.6.1. Membuat Dictionary	193
8.6.2. Mengekstrak Elemen dari Dictionary	194
8.6.3. Operasi Penugasan, Relational, dan Logika Elemen Dictionary	195
8.6.4. Menggabung dan Mengupdate Dictionary	196
8.6.4.1. Operator <code> </code> (<i>union</i>)	197
8.6.4.2. Operator <code> =</code> (<i>in-union</i>)	197
8.6.4.3. Metode <code>dict.update()</code>	198
8.6.5. Metode Dictionary	199
8.7. Bekerja dengan Data Himpunan (Set dan Frozenset)	202
8.7.1. Membuat Set	202
8.7.2. Membuat Frozenset	203
8.7.3. Mengekstrak Elemen Set dan Frozenset	203

8.7.4.	Keunikan (Uniqueness) Set dan Frozenset	205
8.7.5.	Metode Set dan Frozenset	206
8.7.5.1.	Operasi Dasar Himpunan pada Set dan Frozenset	207
8.7.5.2.	Operasi Manipulasi Pada Set	209
8.7.6.	Tidak ada Operasi Penugasan, Relasional, dan Logika Elemen Set dan Frozenset	210
8.8.	<i>Sequence Zipping</i> dengan <code>zip()</code>	210
BAB 9	FILE DAN DIREKTORI	213
9.1.	Memahami File	213
9.2.	Aman Dalam Bekerja dengan File	214
9.3.	Operasi File Dasar pada Python	216
9.3.1.	Fungsi <code>open()</code> untuk membuat ,menulis, dan membaca file	216
9.3.2.	Membaca file dengan <code>file.read()</code>	222
9.3.3.	Membaca file dengan <code>file.readline()</code>	223
9.3.4.	Menulis file dengan <code>file.write()</code>	225

9.3.4.1. Menambah isi file dengan append	225
9.3.4.2. Menulis isi file dengan write	226
9.3.5. Mengoperasikan dua file dengan operasi berbeda (satu baca, satu tulis)	227
9.4. Merekam Data Masukan ke Dalam File	229
9.5. Operasi File Lanjutan pada Python	231
9.5.1. Modul – Modul Esensial	231
9.5.2. Mendapatkan Atribut File	232
9.5.2.1. Mengakses Tipe File dan Hak Akses melalui Makna Rahasia <code>st_mode</code>	235
9.5.2.2. Mengakses Data Waktu File	253
9.5.3. Mengkopi File	254
9.5.4. Memindah File	258
9.5.5. Mengganti Nama File	259
9.5.6. Menghapus File	261
9.5.7. Menghapus File dengan Aman menggunakan <code>secure_delete</code>	263
9.5.8. Terlalu Paranoid untuk Menghapus File? Gunakan <code>send2trash</code> !	265
9.6. Bekerja dengan Direktori/Folder	266
9.6.1. Membuat Direktori/Folder	267
9.6.1.1. Membuat Direktori/Folder Tunggal	268
9.6.1.2. Membuat Direktori/Folder Majemuk	269
9.6.2. Mengkopi Folder	271
9.6.3. Memindah Folder	273
9.6.4. Mengganti Nama Folder	274
9.6.5. Menghapus Folder	276
9.6.6. Menghapus Folder dengan Aman menggunakan <code>secure_delete</code>	280
9.6.7. Menggunakan <code>send2trash</code> untuk Membuang Folder ke Recycle Bin/Trash	282
9.6.8. Mendapatkan Daftar Direktori	283
9.6.9. Memfilter Daftar Direktori	286
9.6.10. Menampilkan Struktur Pohon Direktori	289
9.7. Mencocokkan Nama File dengan Pola (<i>Globbing</i>)	294
9.7.1. <i>Globbing</i> dengan Metode <code>endswith()</code> dan <code>startswith()</code>	295

9.7.2.	<i>Globbing</i> dengan <code>fnmatch.fnmatch()</code>	297
9.7.3.	<i>Globbing</i> dengan Modul <code>glob</code>	299
9.7.4.	<i>Globbing</i> dengan Fungsi <code>pathlib.Path.glob()</code> dan <code>pathlib.Path.rglob()</code>	302
9.8.	Menelusuri Direktori dengan <code>os.walk()</code>	304
9.8.1.	Penelusuran Normal	305
9.8.2.	Menelusur Link Simbolik	307
9.8.3.	Telusur Balik	309
9.9.	Direktori Kerja (<i>Working Directory</i>)	311
9.10.	Dasar – Dasar Pengarsipan dan Kompresi File di Python	312
9.10.1.	Pengarsipan dengan <code>shutil.make_archive()</code>	313
9.10.2.	Memekarkan arsip dengan <code>shutil.unpack_archive()</code>	314
BAB 10 FUNGSI		316
10.1.	Mengenal Terminologi Dasar Tentang Fungsi	316
10.2.	Membuat Fungsi dengan <code>def</code>	317

10.3. Membuat Fungsi Sederhana Dengan Pengembalian	318
10.4. Membuat Fungsi Sederhana Tanpa Pengembalian	319
10.5. Beda Fungsi Dengan Pengembalian dengan Fungsi Tanpa Pengembalian	320
10.6. Statemen <code>lambda</code>	321
10.6.1. <code>lambda</code> Sebagai Pembentuk Fungsi	322
10.6.2. <code>lambda</code> Sebagai Argumen Suatu <i>Higher-order Function</i>	323
10.7. Ruang Lingkup Variabel/Cakupan Variabel (<i>Variable Scope</i>)	325
10.8. Kata Kunci <code>global</code> untuk Mengakses Variabel Global	327
10.9. Parameter dalam Fungsi	327
10.10. Melewatkan Argumen dengan Kata Kunci	328
10.11. <i>Pass-by Value? Pass-by Reference? Bukan, itu Pass-by Object Reference!</i>	332
10.11.1. Pembuktian <i>Pass-by Object Reference</i> di Python	335
10.11.2. Manipulasi? OK! Penugasan kembali (<i>Reassignment</i>)? No!	343
10.11.3. Mungkinkah <i>Pass-by Reference</i> dan <i>Pass-by Value</i> dilakukan di Python?	345
10.12. Argumen Default	348
10.13. Jumlah Argumen yang Berubah	352
10.14. Fungsi Bersarang (<i>Nested Function</i>)	356
10.15. Decorator	361
10.16. <i>Overloading</i> Fungsi pada Python dengan <code>multipledispatch</code>	364
10.17. Fungsi Rekursif	368
10.18. Generator	374
10.19. <i>Type Hinting</i> Masukan dan Kembalian Fungsi	379

BAB 11 OBJECT ORIENTED PROGRAMMING (OOP) 382

11.1. Memahami Konsep OOP Secara Umum	383
11.1.1. Empat Pilar OOP	383
11.1.1.1. Abstraksi	383
11.1.1.2. Enkapsulasi	384
11.1.1.3. Inheritance	384
11.1.1.4. Polimorfisme	385
11.1.2. Fitur bersama OOP dan Non-OOP	385

11.1.2.1. Variabel	385
11.1.2.2. Prosedur	385
11.1.3. Fitur khusus OOP	386
11.1.3.1. Kelas dan Objek (Class and Object)	386
11.1.3.2. <i>Class-based</i> dan <i>Protoype-based</i> Style Orientation	386
11.1.3.3. <i>Dynamic Dispatch</i>	386
11.1.3.4. Delegasi	387
11.1.3.5. Rekursi Terbuka (<i>Open Recursion</i>)	387
11.2. Bekerja dengan Kelas dan Objek	388
11.2.1. Mendefinisikan Kelas dan Metode di Dalam Kelas	388
11.2.2. Membuat Objek dari Kelas	391
11.2.3. Membuat dan Menggunakan Atribut Objek	391
11.2.4. Menggunakan Metode Objek	392
11.2.5. Membuat Kelas dan Objek	392

11.3. Variabel Statis dan <i>Instance</i>	395
11.4. Lebih Lanjut dengan Konstruktor	396
11.4.1. <code>__init__()</code>	396
11.4.2. <i>Default</i> Konstruktor untuk Kelas <i>Default</i>	399
11.4.3. <code>__new__()</code>	403
11.4.4. <code>__init__()</code> dan <code>__new__()</code> dalam Satu Kelas	409
11.5. Destruktor	412
11.6. Lebih Lanjut dengan Metode (<i>Method</i>)	415
11.6.1. Fungsi Internal	415
11.6.2. Metode Statis dan Metode Kelas	417
11.7. <i>Dunder</i>	419
11.8. Melindungi Data dan Proses dari Intervensi dengan Enkapsulasi Melalui Pemberian Pemodifikasi Akses (<i>Access Modifier</i>)	423
11.8.1. <i>Public vs Protected vs Private</i>	423
11.8.2. Bekerja dengan <i>Getter, Setter, dan Deleter</i>	432
11.8.3. Bekerja dengan <i>Property</i>	437
11.9. Pewarisan (<i>Inheritance</i>)	442
11.9.1. Pewarisan Tunggal (<i>Single Inheritance</i>)	443
11.9.2. Fungsi <code>super()</code> untuk Pewarisan Tunggal	446
11.9.3. Pewarisan Jamak (<i>Multiple Inheritance</i>)	451
11.9.4. Melacak <i>Instance</i> dan <i>Subclass</i>	453
11.9.5. Mencegah Pewarisan Kelas	457
11.9.6. Bekerja dengan Abstrak dan <i>Metaclass</i>	459
11.10. Komposisi	465
11.11. Polimorfisme	467
11.12. Bekerja dengan <i>Dynamic Dispatch</i>	471
11.12.1. Single Dispatch	471
11.12.2. Multiple Dispatch	473
11.12.2.1. Menggunakan <code>multipledispatch</code>	473
11.12.2.2. Menggunakan <code>plum-dispatch</code>	475

11.12.2.3. Menggunakan <code>multimethod</code>	478
BAB 12 REGULAR EXPRESSION (REGEX)	481
12.1. Apa itu Pola REGEX?	481
12.2. Simbol Khusus pada REGEX	483
12.3. Modul <code>re</code>	483
12.3.1. Metode Modul <code>re</code>	483
12.3.2. <i>Metacharacter</i> dan <i>Anchor</i> Modul <code>re</code>	484
12.3.3. Kelas Karakter Modul <code>re</code>	485
12.3.4. Asersi Modul <code>re</code>	485
12.3.5. <i>Quantifier</i> Modul <code>re</code>	486
12.3.6. <i>Escape Sequence</i> Modul <code>re</code>	486
12.3.7. <i>Flags</i> Modul <code>re</code>	487
12.4. Mencari Seluruh Cocokan pada Teks menggunakan <code>re.findall()</code>	488
12.5. Mencari Karakter Kata/Bukan Kata, Kata, dan Pasangan Kata	489

12.6. Menggunakan <i>Character Set</i> di <code>re.findall()</code>	494
12.7. <i>Dot</i> REGEX (<code>.</code>)	496
12.8. <i>Asterisk</i> REGEX (<code>*</code>)	499
12.9. <i>Quantifier</i> <code>+</code> pada REGEX	501
12.10. <i>Quantifier</i> <code>?</code> pada REGEX	503
12.11. Mencocokkan Pola Suatu Karakter Sebanyak x Kali	505
12.11.1. Mencocokkan Pola Suatu Karakter Sebanyak m Kali	505
12.11.2. Mencocokkan Pola Suatu Karakter Sebanyak Rentang $m-n$ Kali	505
12.11.3. Mencocokkan Pola Suatu Karakter Sebanyak Lebih dari m Kali	506
12.12. <i>Greedy Mode</i> vs <i>Lazy Mode</i>	506
12.13. Pembatas <i>String</i> dan Baris	509
12.14. REGEX Logika	511
12.14.1. REGEX AND	511
12.14.2. REGEX OR	511
12.14.3. REGEX NOT	512
12.15. Memecah <i>String</i> dengan Pola REGEX dengan <code>re.split()</code>	513
12.16. Mengganti Sebagian String dengan Pola Menggunakan <code>re.sub()</code>	513
12.17. Mencari dengan REGEX dengan <code>re.search()</code>	513
12.18. Kompilasi REGEX dengan <code>re.compile()</code>	514
BAB 13 PEMROGRAMAN MODULER DAN MANAJEMEN PAKET	516
13.1. Konsep Modul, Paket, dan Pustaka dalam Python	516
13.2. Membuat Modul	518
13.3. Beda Modul dengan Program di Python	521
13.4. Jangan Berlebihan dalam Melakukan Impor Melingkar!	524
13.5. Membuat Pustaka Sendiri (Studi Kasus Uji Kimia Tanah)	524
13.5.1. Merancang dan Mengkode modul	525
13.5.2. File <code>__init__.py</code>	528
13.5.3. File <code>__main__.py</code> (Opsional)	530
13.5.4. Mengorganisasi Modul untuk Pustaka	530
13.6. Menggunakan Paket	531

13.7. Membuat Distribusi Paket <i>Wheel</i> Sendiri	533
13.7.1. LICENSE	534
13.7.2. pyproject.toml	535
13.7.3. README.md	535
13.7.4. setup.py	536
13.7.5. Konfigurasi Folder Build	537
13.7.6. Persiapan Sebelum Membangun <i>wheel package</i>	538
13.7.7. Membangun <i>Wheel Package</i>	540
13.7.8. Instalasi <i>Wheel Package</i> Secara <i>Offline</i>	542
13.7.9. Penghapusan Instalasi <i>Wheel Package</i>	544
BIBLIOGRAFI	546
DAFTAR PUSTAKA	547

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 gemar eksperimen, tersedia pula *source code* interpreter yang siap dikompilasi dengan C compiler yang sesuai^A. Selain itu, bahasa pemrograman Python sudah merupakan bahasa yang komplit, sudah tersedia ribuan paket pustaka dengan beraneka ragam fungsi siap pakai. Pustaka

– pustaka ini ditulis dengan bahasa Python juga, sehingga tersedia dalam bentuk skrip yang siap pakai yang penggunaannya hanya cukup dengan memanggil pustaka itu. Ada pula pustaka – pustaka tertentu yang juga ditulis dengan bahasa pemrograman lain atau merupakan komponen untuk mengakses fitur bahasa pemrograman lain, namun tidak banyak jumlahnya. Untuk pustaka demikian, maka perlulah kita menginstal paket pendukung yang terkait.

Ya. Meskipun Python adalah bahasa pemrograman tersendiri, Python memerlukan interpreter yang ditulis dengan bahasa C. Interpreter bertugas untuk menerjemahkan kode Python menjadi bytecode, sehingga anda tak perlu berpusing menerjemahkan kode Python kepada mesin. Default Interpreter Python dinamakan CPython. Selain CPython, banyak pula interpreter lain yang bisa anda gunakan. Akan tetapi, banyak proyeknya sudah kuno atau tertinggal. Contoh interpreter alternatif yang masih aktif adalah PyPy, Stackless Python, Numba, dan Nuitka.

Informasi Selengkapnya

Silahkan order dan hubungi kami:

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

Whats App atau Handphone : +62822-2987-2305

DASAR - DASAR PYTHON 3.10



ISBN 978-623-98383-6-2 (PDF)



Publisher

Future Science

Redaksi

Jl. Terusan Surabaya Gang 1 A No. 71 RT 002 RW 005, Kel. Sumbersari,
Kec. Lowokwaru, Kota Malang, Provinsi Jawa Timur

Email

admin@futuresciencepress.com

INBIO
research made easy