

Sayuti Rahman | Arnes Sembiring
Dodi Siregar | Husnul Khair
I Gusti Prahmana | Ratih Puspadini
Muhammad Zen



PYTHON: DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK

PYTHON : DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK

Sayuti Rahman
Arnes Sembiring
Dodi Siregar
Husnul Khair
I Gusti Prahmana
Ratih Puspadini
Muhammad Zen



TAHTA MEDIA GROUP

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

REPUBLIK INDONESIA
KEMENTERIAN HUKUM DAN HAK ASASI MANUSIA

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	EC00302357427, 20 Juli 2023
Pencipta	
Nama	Sayuti Rahman, Arnes Sembiring dkk
Alamat	Komplek Rumah Pondok 6 Blok K-27, Pecur Sei Tuan, Deli Serdang, Deli Serdang, Sumatera Utara, 20371
Kewarganegaraan	Indonesia
Pemegang Hak Cipta	
Nama	Sayuti Rahman, Arnes Sembiring dkk
Alamat	Komplek Rumah Pondok 6 Blok K-27, Pecur Sei Tuan, Deli Serdang, Deli Serdang, Sumatera Utara, 20371
Kewarganegaraan	Indonesia
Jenis Ciptaan	Buku
Judul Ciptaan	PYTHON : DASAR DAN PEMROGRAMAN BERORIENTASI OLEK
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	20 Juli 2023, di Sumatera (solo)
Jangka waktu perlindungan	Bertaku selama hidup Pencipta dan terus berlangsung selama 70 (tujuh puluh) tahun setelah Pencipta meninggal dunia, terhitung mulai tanggal 1 Januari tahun berikutnya.
Nomor pencatatan	000490362

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atas produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



di. MENTERI HUKUM DAN HAK ASASI MANUSIA
Direktur Hak Cipta dan Desain Industri

Anggoro Dasananta
NIP. 196412081991011002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Sayuti Rahman	Komplek Rumah Pondok 6 Blok K-27, Percut Sei Tuan, Deli Serdang
2	Arnes Sembiring	Jln Bunga Baldu No 20 C Medan
3	Dodi Siregar	Dusun XI Mulia No 4, Seintia, Percut Sei Tuan, Deli Serdang
4	Husni Khair	Jln Ganda Perum Villa Alia No 55 LK X, Cengkeh Turi, Binjai Utara
5	I Gusti Pradmaso	Jln Angrek D.2 T KW Damai, Kwala Begumit, Binjai, Langkat
6	Ratih Puspadini	Jln Gaperu Gang Famili No 110-A LK I, Medan Helvetia
7	Muhammad Zen	Dusun I Bukit Gantang, Wampu, Langkat

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Sayuti Rahman	Komplek Rumah Pondok 6 Blok K-27, Percut Sei Tuan, Deli Serdang
2	Arnes Sembiring	Jln Bunga Baldu No 20 C Medan
3	Dodi Siregar	Dusun XI Mulia No 4, Seintia, Percut Sei Tuan, Deli Serdang
4	Husni Khair	Jln Ganda Perum Villa Alia No 55 LK X, Cengkeh Turi, Binjai Utara
5	I Gusti Pradmaso	Jln Angrek D.2 T KW Damai, Kwala Begumit, Binjai, Langkat
6	Ratih Puspadini	Jln Gaperu Gang Famili No 110-A LK I, Medan Helvetia
7	Muhammad Zen	Dusun I Bukit Gantang, Wampu, Langkat



**PYTHON : DASAR DAN PEMROGRAMAN
BERORIENTASI OBJEK**

Penulis:

Sayuti Rahman
Arnes Sembiring
Dodi Siregar
Husnul Khair
I Gusti Prahmana
Ratih Puspadini
Muhammad Zen

Desain Cover:

Tahta Media

Editor:

Tahta Media

Proofreader:

Tahta Media

Ukuran:

x,217, Uk: 15,5 x 23 cm

ISBN: 978-623-147-100-0

Cetakan Pertama:

Juli 2023

Hak Cipta 2023, Pada Penulis

Isi diluar tanggung jawab percetakan

Copyright © 2023 by Tahta Media Group

All Right Reserved

Hak cipta dilindungi undang-undang

Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PENERBIT TAHTA MEDIA GROUP

(Grup Penerbitan CV TAHTA MEDIA GROUP)

Anggota IKAPI (216/JTE/2021)

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Segala puji dan syukur kami panjatkan kepada Allah SWT yang telah melimpahkan rahmat dan karunia-Nya yang tak terhingga. Dengan berkat-Nya, penulis berhasil menyelesaikan buku yang berjudul "PYTHON: DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK". Buku ini diharapkan dapat memberikan manfaat bagi mahasiswa dan praktisi sebagai referensi dalam mempelajari pemrograman Python.

Bahasa pemrograman Python telah menjadi sangat populer saat ini. Pengembangan Python melibatkan jutaan programmer, peneliti, dan pengguna dari berbagai latar belakang, tidak hanya di kalangan IT, karena Python bersifat open source. Banyak programmer dan peneliti yang beralih ke penggunaan bahasa ini. Python memiliki berbagai kegunaan, seperti pengembangan aplikasi web, aplikasi desktop, IoT, dan banyak lagi. Python juga memiliki integrasi dengan sistem database dan kemampuan untuk membaca dan mengubah file, sehingga sering digunakan dalam prototyping atau pengembangan perangkat lunak dengan cepat dan andal.

Perkembangan dan kegunaan Python sangat luar biasa sehingga bahasa ini menjadi bahasa pemrograman nomor satu di dunia. Namun, penggunaan Python masih belum begitu populer di Indonesia. Oleh karena itu, kami memperkenalkan bahasa pemrograman Python dalam Pemrograman Berorientasi Objek.

Buku ini terbagi menjadi 12 bab, dengan 6 bab membahas dasar-dasar pemrograman dan 6 bab berikutnya membahas pemrograman berorientasi objek. Kami memulai dengan pembahasan dasar-dasar pemrograman Python untuk memberikan pemahaman dasar kepada mahasiswa sebelum mempelajari pemrograman berorientasi objek. Dengan membaca dan mempraktikkan buku ini, diharapkan mahasiswa dan praktisi dapat belajar bahasa pemrograman Python dengan baik.

Kami ingin mengucapkan terima kasih kepada Ikatan Profesi Komputer Informatika Nusantara (IKAPKIN) sebagai wadah kolaborasi beberapa institusi di bidang Komputer dan Informatika. Terima kasih juga kepada institusi yang menjadi tempat para penulis bernaung. Penghargaan terbesar

juga kami sampaikan kepada rekan-rekan yang telah memberikan saran sepanjang penulisan buku ini.

Wassalamualaikum Wr. Wb.

Hormat Kami,

Penulis

DAFTAR ISI

KATA PENGANTAR.....	vi
DAFTAR ISI	viii
BAB 1 PENGANTAR DAN INSTALASI PYTHON	1
1.1 Sejarah Python.....	1
1.2 Mengapa Belajar Python	1
1.3 Install Anaconda.....	2
1.4 Jupyter Notebook.....	5
1.5 Aturan Penulisan Pada Python	8
1.6 Soal dan Kunci Jawaban.....	17
BAB 2 TIPE DATA DASAR PYTHON.....	20
2.1 Tipe Number.....	20
2.2 Konversi Data Number.....	22
2.3 Tipe Data String	24
2.4 Tipe Data Boolean.....	35
2.5 Soal dan Kunci Jawaban.....	36
BAB 3 OPERATOR PADA PYTHON.....	39
3.1 Operator Aritmatika.....	39
3.2 Operator Perbandingan	41
3.3 Operator Logika.....	43
3.4 Operator Identitas	45
3.5 Operator Keanggotaan.....	47
3.6 Operator Bitwise.....	48
3.7 Operator Penugasan.....	51
3.8 Soal dan Kunci Jawaban.....	53
BAB 4 DATA COLLECTION DAN PENCABANGAN	56
4.1 Data List	56
4.2 Data Tuple	57
4.3 Data Set	59
4.4 Data Dictionaries	62
4.5 Pernyataan Bersyarat IF.....	64
4.6 Soal dan Kunci Jawaban.....	67
BAB 5 PERULANGAN.....	72
5.1 Perulangan While	72

5.2	Perulangan For.....	77
5.3	Perulangan pada Data Collection	80
5.4	Perulangan Bersarang.....	81
5.5	Soal dan Kunci Jawaban.....	86
BAB 6 FUNGSI PADA PYTHON		89
6.1	Membuat dan Memanggil Fungsi.....	89
6.2	Argumen dan Parameter Fungsi	91
6.3	Argumen Arbitrary	92
6.4	Argumen dengan Kata Kunci	93
6.5	Argumen Arbitrary dengan Kata Kunci	94
6.6	Nilai Parameter Default	95
6.7	List Sebagai Argumen	98
6.8	Mengembalikan Nilai Fungsi	99
6.9	Rekursi.....	100
6.10	Lambda	102
6.11	Soal dan Kunci Jawaban.....	104
BAB 7 KELAS DAN OBJEK PADA PYTHON		108
7.1	Pemrograman Berorientasi Objek	108
7.2	Pengertian Objek dan Kelas	109
7.3	Metode Init	110
7.4	Metode Str	111
7.5	Intansiasi Objek	113
7.6	Metode Pada Objek	114
7.7	Parameter Self	115
7.8	Implementasi Objek dengan Berbagai Metode.....	116
7.9	Soal dan Kunci Jawaban.....	118
BAB 8 KONSTRUKTOR DAN DESTRUKTOR PADA PYTHON		121
8.1	Konstruktor.....	121
8.2	Destructor	128
8.3	Soal dan Kunci Jawaban.....	134
BAB 9 INHERITANCE (PEWARISAN)		138
9.1	Inheritance pada Python	138
9.2	Kesalahan Tidak Memanggil Konstruktor Induk	140
9.3	Fungsi Super	141
9.4	Single Inheritance	143

9.5	Multiple Inheritances	144
9.6	Multilevel Inheritance.....	145
9.7	Hierarchical Inheritance	147
9.8	Hybrid inheritance	149
9.9	Member Private Super Class	150
9.10	Overriding Metode pada Inheritance	152
9.11	Konsep "is-a" dan "has-a"	153
9.12	Soal dan Kunci Jawaban.....	154
BAB 10	ENKAPSULASI.....	157
10.1	Pengertian dan Keuntungan Enkapsulasi.....	157
10.2	Member Protected	159
10.3	Member Private	160
10.4	Member Publik	162
10.5	Metode Setter dan Getter	163
10.6	Properti	165
10.7	Praktik Terbaik dalam Enkapsulasi	168
10.8	Studi Kasus.....	169
10.9	Soal dan Kunci Jawaban.....	173
BAB 11	POLIMORFISME PADA PYTHON	176
11.1	Pengenalan Polimorfisme	176
11.2	Poimorfisme Metode	177
11.3	Polimorfisme Pada Inheritance (Pewarisan).....	178
11.4	Polimorfisme Abstrak.....	180
11.5	Polimorfisme dengan <i>Magic Methode</i>	182
11.6	Konsep Duck Typing pada Python.....	192
11.7	Soal dan Kunci Jawaban.....	193
BAB 12	VARIABEL STATIS PADA PYTHON	197
12.1	Pengertian Variabel Statis	197
12.2	Metode Statis	202
12.3	Akses Metode dan Variabel Statis.....	205
12.4	Batasa dan Pertimbangan.....	209
12.5	Soal dan Kunci Jawaban.....	210
DAFTAR PUSTAKA.....		213
BIOGRAFI PENULIS.....		215

BAB 1

PENGANTAR DAN INSTALASI PYTHON

1.1 SEJARAH PYTHON

Bahasa pemrograman Python, yang sangat populer saat ini, pertama kali ditemukan oleh Guido van Rossum di Stichting Mathematisch Centrum (CWI), Amsterdam pada tahun 1991 (Awangga et al., 2019). Pengembangan Python terinspirasi oleh bahasa pemrograman ABC yang berkembang pada saat itu. Salah satu perbedaan utama Python dengan bahasa pemrograman lainnya adalah pengembangannya melibatkan jutaan programmer, peneliti, dan pengguna dari berbagai latar belakang, tidak hanya dari kalangan IT, karena Python bersifat open source.

Python adalah bahasa pemrograman yang menggunakan interpreter untuk menjalankan kode programnya. Interpreter tersebut dapat menerjemahkan kode secara langsung, dan Python dapat dijalankan di berbagai platform seperti Windows, Linux, dan lain-lain. Python mengadopsi paradigma pemrograman dari beberapa bahasa lain, termasuk paradigma pemrograman prosedural seperti bahasa C, pemrograman berorientasi objek seperti Java, dan bahasa fungsional seperti Lisp. Kombinasi paradigma ini memudahkan para programmer dalam mengembangkan berbagai proyek menggunakan Python.

1.2 MENGAPA BELAJAR PYTHON

Banyak programmer dan peneliti beralih ke penggunaan bahasa pemrograman Python. Python dapat digunakan untuk berbagai keperluan, seperti pengembangan aplikasi web, aplikasi desktop, IoT, dan berbagai aplikasi lainnya. Python juga memiliki integrasi dengan sistem database dan mampu membaca serta mengubah file, sehingga sering digunakan untuk prototyping atau pengembangan perangkat lunak dengan cepat dan reliabel. Selain itu, Python juga digunakan secara luas oleh para peneliti karena kemampuannya dalam menangani data besar dan perhitungan matematika yang kompleks (Rahman et al., 2021).

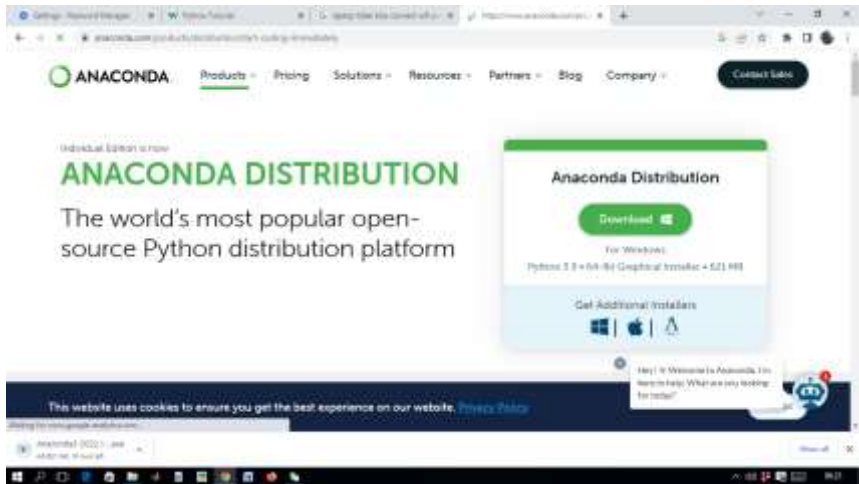
Ada beberapa alasan mengapa Python menjadi pilihan utama, yaitu:

1. Python dapat berjalan di berbagai platform seperti Windows, Linux, macOS, Android, Raspberry Pi, dan lain-lain.
2. Python memiliki sintaks yang sederhana dan mirip dengan bahasa Inggris.
3. Sintaks Python memungkinkan penulisan kode yang lebih ringkas dibandingkan dengan bahasa pemrograman lain.
4. Python menggunakan interpreter, sehingga program dapat dieksekusi dengan cepat setelah selesai dibuat.
5. Python mendukung paradigma pemrograman prosedural, berorientasi objek, dan fungsional.

Berdasarkan penjelasan di atas, mari kita mulai menggunakan Python. Ada banyak perangkat lunak pendukung yang tersedia, namun dalam buku ini kita akan menggunakan Anaconda.

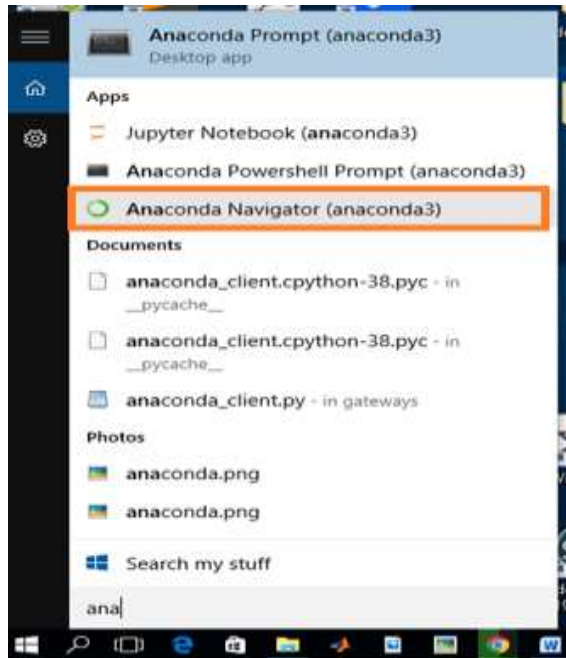
1.3 INSTALL ANACONDA

Anaconda adalah perangkat lunak yang memungkinkan pengelolaan paket, dependensi, dan lingkungan data science dengan menggunakan Python. Dengan Anaconda, Anda dapat mengembangkan proyek data science menggunakan berbagai IDE seperti Jupyter Lab, Jupyter Notebook, Spider, dan lainnya. Anda dapat mengunduh perangkat lunak Anaconda dari situs resmi mereka di <https://www.anaconda.com/products/distribution>, seperti yang terlihat dalam Gambar 1 berikut ini.



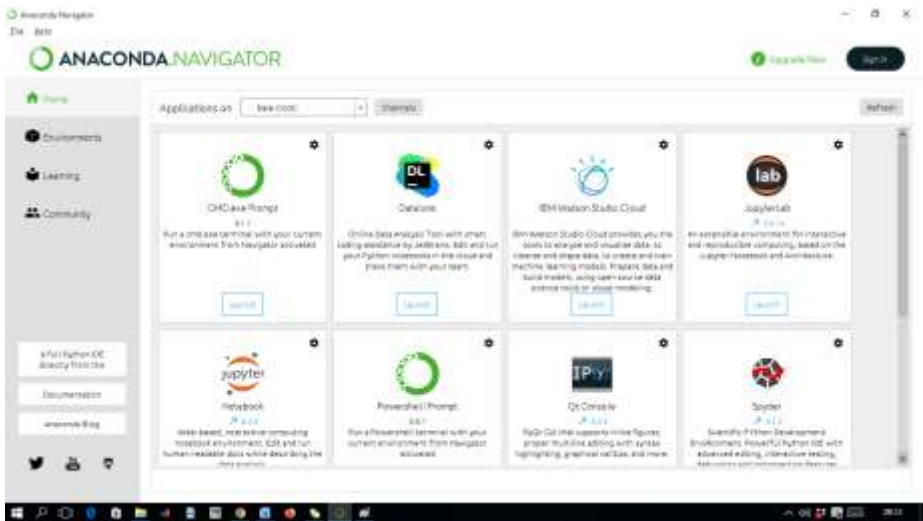
Gambar 1 Halaman Download Anaconda

Seperti yang ditunjukkan dalam Gambar 1, Anda dapat mengunduh paket Anaconda dari halaman tersebut. Jika Anda menggunakan Windows 64 bit, Anda dapat langsung mengunduh perangkat lunak yang tercantum di halaman utama. Namun, jika Anda menggunakan versi lain dari Windows, Anda dapat memilih "get additional installers" untuk mendapatkan versi yang sesuai dengan kebutuhan komputer Anda. Setelah Anda selesai mengunduh perangkat lunak, lanjutkan dengan proses instalasi sampai selesai. Setelah itu, Anda dapat mencari Anaconda di kotak pencarian Windows dan mengkliknya untuk menjalankannya, seperti yang terlihat pada Gambar 2 berikut ini.



Gambar 2 Klik Anaconda Navigator

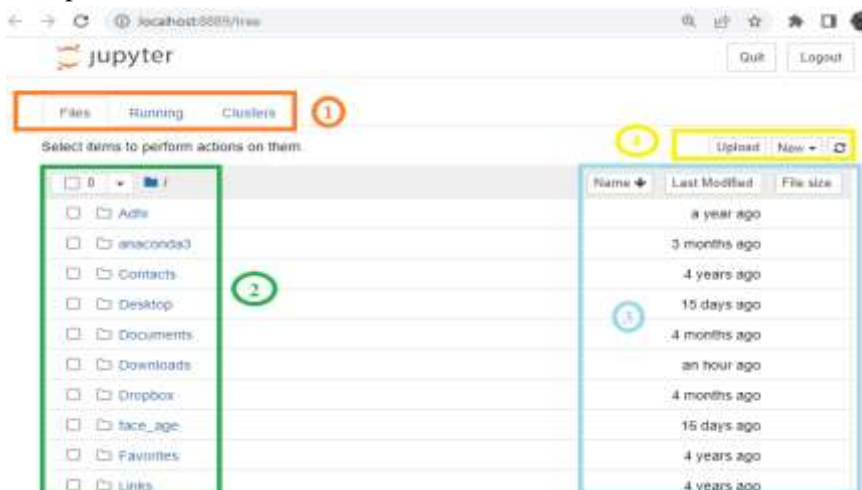
Gambar 2 menunjukkan bahwa ketika kita mencari "Anaconda" di kotak pencarian Windows, akan muncul beberapa pilihan seperti "Anaconda Prompt", "Jupyter Notebook", "Powershell", dan "Anaconda Navigator". "Anaconda Prompt" biasanya digunakan untuk menginstal paket atau menjalankan Python dengan versi yang spesifik. "Jupyter Notebook" biasanya digunakan sebagai editor bahasa pemrograman Python. "Powershell" memiliki fungsi yang sama dengan "Anaconda Prompt", namun dijalankan melalui "powershell.exe", yang merupakan versi terbaru dari "cmd.exe" pada Windows. "Anaconda Navigator" digunakan untuk menampilkan semua framework yang dapat digunakan dalam pemrograman Python. "Anaconda Navigator" ditampilkan dalam Gambar 3 berikut ini.



Gambar 3 Anaconda Navigator

1.4 JUPYTER NOTEBOOK

Jupyter Notebook adalah sebuah editor Python berbasis web yang dapat dijalankan melalui peramban (browser). Untuk membuka Jupyter Notebook, Anda dapat mencarinya melalui kotak pencarian Windows atau mengklik opsi "Launch" pada Anaconda Navigator. Tampilan awal Jupyter Notebook dapat dilihat pada Gambar 4 berikut ini.

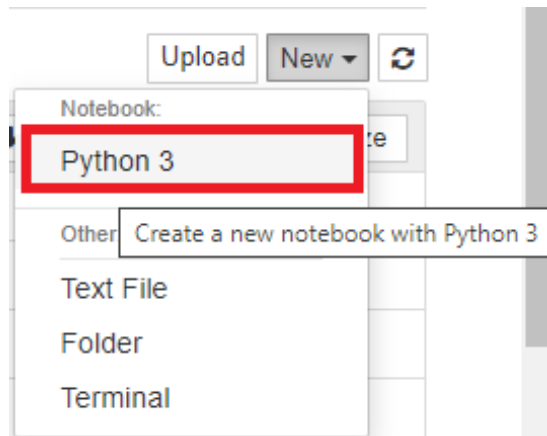


Gambar 4 Tampilan Awal Jupyter Notebook

Pada Gambar 4, Jupyter Notebook memiliki beberapa bagian utama sebagai berikut:

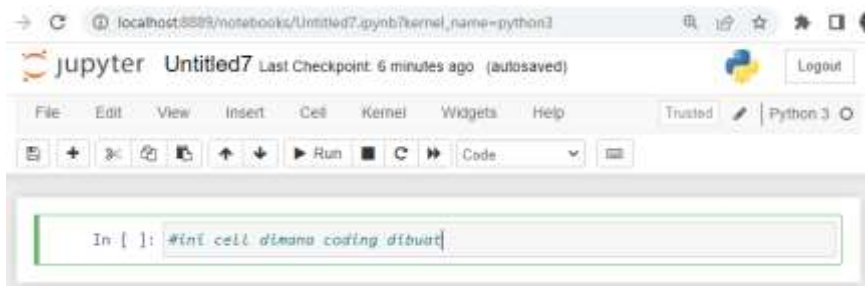
1. Menu "Files", "Running", dan "Cluster":
 - a. Menu "Files" digunakan untuk menampilkan semua file dan folder yang ada dalam path Jupyter.
 - b. Menu "Running" digunakan untuk melihat file yang sedang dibuka, sedangkan menu "Cluster" digunakan untuk melihat file yang menggunakan komputasi paralel interaktif dengan IPython.
2. Tampilan File:

Tampilan file akan ditampilkan sesuai dengan menu yang dipilih pada bagian pertama, yaitu "Files", "Running", atau "Cluster".
3. Menu "Name", "Last Modified", dan "File Size":
 - a. Menu-menu ini digunakan untuk mengurutkan file dalam tampilan. Jika Anda memilih menu "Name", file akan diurutkan berdasarkan nama. Klik sekali untuk mengurutkan dari A-Z dan klik dua kali untuk mengurutkan dari Z-A.
 - b. Menu "Last Modified" digunakan untuk mengurutkan file berdasarkan waktu terakhir kali file diubah. Klik sekali untuk mengurutkan dari yang terbaru hingga yang paling lama diubah, dan klik dua kali untuk mengurutkan sebaliknya.
 - c. Menu "File Size" digunakan untuk mengurutkan file berdasarkan ukuran file.
4. Menu "Upload" dan "New":
 - a. Menu "Upload" digunakan untuk mengunggah file ke dalam host Jupyter.
 - b. Menu "New" digunakan untuk membuat file atau folder baru. Jika Anda ingin membuat file baru dan memulai pemrograman Python dengan Jupyter Notebook, pilih menu "Python 3". Hal ini ditampilkan pada Gambar 5 berikut ini.



Gambar 5 Membuat File baru Python

Setelah Anda memilih "Python 3" seperti yang ditunjukkan pada Gambar 5, akan muncul jendela editor Jupyter Notebook. Tampilan editor Jupyter Notebook ini dapat dilihat pada Gambar 6 berikut.



Gambar 6 Editor Jupyter Notebook












Terlihat pada Gambar 6, editor Jupyter Notebook terdiri dari empat bagian utama, yaitu:

1. Header: Bagian ini menampilkan nama file yang saat ini sedang diedit, dalam contoh ini adalah "Untitled7".
2. Menu: Terdapat menu-file, edit, view, insert, cell, kernel, widgets, dan help. Menu ini memberikan opsi untuk mengatur dan mengedit notebook.
3. Toolbar: Terletak di bawah menu, toolbar menyediakan ikon-ikon yang memiliki fungsi berbeda. Setiap ikon memiliki tugas khusus. Penjelasan lebih lanjut dapat ditemukan pada Tabel 1 berikut.

4. Sel (Cell): Digunakan untuk membuat baris program atau teks. Anda dapat menulis kode Python atau menambahkan teks di dalam sel ini.

Toolbar dilengkapi dengan beberapa ikon yang memiliki fungsi yang berbeda-beda. Penjelasan mengenai ikon-ikon tersebut disajikan dalam Tabel 1 sebagai berikut.

Tabel 1 Toolbar Jupyter notebook

Ikon	Nama	Fungsi
	Save	Menyimpan file yang sedang dikerjakan
	Insert Cell	menambah Cell coding
	Cut Selected cell	Meng-cut cell yang aktif
	Copy selected cell	Meng-copy cell yang aktif
	Paste cell below	Mem-paste cell yang telah dicopy sebelumnya
	Move selected cell up	Memindahkan cell aktif ke atas, jika cell lebih dari satu dan berada dibawah
	Move selected cell down	Memindahkan cell ke bawah, jika cell lebih dari satu dan berada di atas
	Run cell	Menjalankan baris program yang berada pada cell aktif
	Interrupt the carnel	Menghentikan program yang sedang proses atau berjalan
	Restart the carnel	Melakukan restart kernel pada cell aktif
	Restart the carnel	Restart kernel pada seluruh cell

1.5 ATURAN PENULISAN PADA PYTHON

Aturan penulisan dalam Python sangat penting untuk memastikan kode yang konsisten, mudah dibaca, dan mudah dimengerti oleh orang lain. Beberapa aturan termasuk dalam aturan penulisan Python, seperti menggunakan indentasi yang konsisten, memisahkan pernyataan dalam baris yang berbeda, mengikuti konvensi penamaan variabel dengan huruf kecil dan

garis bawah, menggunakan komentar untuk menjelaskan kode, memilih tanda kutip yang konsisten untuk string, menempatkan pernyataan import di awal file, membatasi panjang baris hingga 79 karakter, dan mengikuti pedoman gaya penulisan. Mengikuti aturan penulisan ini membantu menjaga kualitas kode Python dan meningkatkan kerja sama dalam pengembangan perangkat lunak.

1.5.1 Baris Program

Baris program pada python tidak memerlukan semicolon (“titik koma”), cukup menekan enter untuk membedakan baris satu dengan baris berikutnya. Contoh program dapat dilihat pada Program 1 Berikut.

#Program 1.1
print("Hallo world") print("Saya sedang belajar python")
Output: Hallo world Saya sedang belajar python

Seperti terlihat pada Program 1.1, baris program pertama *print("Hello World")* dianggap satu baris program. *print("Saya sedang belajar python")* merupakan baris kedua. Kedua baris ini dipisahkan dengan enter. Python juga dapat menganggap semicolon sebagai pemisah baris. Akan tetapi menyatukan beberapa baris program dengan semicolon merupakan penulisan program yang buruk. Berikut penulisan program dengan menggunakan smicolon sebagai pemisah baris seperti terlihat pada Program 1.2.

#Program 1.2
print("Hallo world");print("Saya sedang belajar python")
Output: Hallo world Saya sedang belajar python

Seperti terlihat pada program 1.2, Kedua pernyataan print ditulis dalam satu baris dengan dipisahkan oleh tanda semikolon. Kedua pernyataan print akan dieksekusi secara berurutan dalam satu baris, dan hasilnya akan dicetak dalam satu baris juga. Keuntungan dari menggunakan tanda semikolon (;) untuk memisahkan pernyataan print adalah:

1. Kode lebih ringkas: Dengan menggabungkan beberapa pernyataan dalam satu baris, kode menjadi lebih singkat dan lebih padat.
2. Meningkatkan kecepatan eksekusi: Dalam beberapa kasus, penggunaan tanda semikolon dapat meningkatkan kecepatan eksekusi program. Meskipun perbedaannya mungkin kecil, penggunaan semikolon dapat mengurangi jumlah baris kode yang dieksekusi.

Namun, perlu diperhatikan bahwa penggunaan tanda semikolon (;) untuk memisahkan pernyataan bukanlah praktik yang umum dalam gaya penulisan Python yang disarankan. Gaya penulisan yang lebih umum adalah dengan menggunakan satu pernyataan per baris, kecuali jika ada alasan khusus untuk menggabungkan pernyataan dalam satu baris.

1.5.2 Komentar

Komentar memiliki peran penting dalam bahasa pemrograman sebagai sarana untuk mendokumentasikan program. Komentar digunakan untuk memisahkan bagian-bagian program dan memberikan penjelasan yang dianggap penting. Dalam Python, tanda "#" digunakan untuk menandai komentar. Baris program yang dijadikan komentar tidak akan dieksekusi oleh interpreter. Fitur ini dapat digunakan untuk menonaktifkan baris program yang mungkin penting dan ingin disimpan untuk digunakan di masa yang akan datang. Berikut adalah contoh penggunaan komentar seperti yang terlihat pada Program 1.3 berikut ini.

#Program 1.3
#cetak print("Hallo world") #isi variable a=1 b=2 print(b)
Output: Hallo world 2

Seperti terlihat pada Program 1.3, Komentar "#cetak" menjelaskan bahwa baris berikutnya mencetak string "Hallo world" ke konsol. Komentar tersebut tidak berdampak pada eksekusi program dan hanya berfungsi sebagai penjelasan bagi pembaca atau pengembang

Komentar "# isi variable" memberikan penjelasan bahwa baris-baris berikutnya digunakan untuk menginisialisasi variabel 'a' dan 'b'. Variabel 'a' diatur dengan nilai 1, variabel 'b' diatur dengan nilai 2, dan kemudian nilai variabel 'b' dicetak ke konsol. Komentar ini membantu pembaca atau pengembang untuk memahami tujuan dari baris kode yang terkait dengan inisialisasi variabel.

Komentar sangat berguna dalam pemrograman, terutama saat bekerja dalam tim atau saat kembali ke kode yang telah lama ditulis. Mereka membantu orang lain atau bahkan diri sendiri untuk lebih memahami maksud dan alur kode, serta memudahkan proses debugging dan pemeliharaan kode.

1.5.3 Blok dan Indentasi

Blok program adalah bagian penting dari sebuah program. Dalam bahasa pemrograman C, blok program diawali dengan tanda "{" dan diakhiri dengan tanda "}". Namun, dalam Python, blok program ditandai oleh indentasi. Indentasi dalam Python menggunakan 4 spasi, yang menunjukkan bagian dari blok program. Pada contoh program yang diberikan, terdapat penggunaan indentasi untuk mengatur tingkat blok kode dalam Python. Indentasi digunakan untuk membedakan antara blok kode yang terkait dengan

kondisi if dan blok kode di luar kondisi if. Anda dapat melihat contohnya pada Program 1.4 berikut ini.

#Program 1.4
if (5 > 2): # ini bagian if print("5 lebih besar dari 2") # ini bukan bagian if print("Ini bukan bagian if")
Output: 5 lebih besar dari 2 Ini bukan bagian if

Di dalam blok if, pernyataan `print("5 lebih besar dari 2")` akan dieksekusi karena kondisi `5 > 2` bernilai True. Indentasi (empat spasi dalam contoh ini) menunjukkan bahwa pernyataan `print` tersebut adalah bagian dari blok if. Di luar blok if, pernyataan `print("Ini bukan bagian if")` juga akan dieksekusi. Indentasi pada pernyataan ini menunjukkan bahwa pernyataan tersebut berada di luar blok if dan akan selalu dieksekusi, tidak tergantung pada kondisi if.

Penting untuk menggunakan indentasi yang konsisten dan benar dalam Python, karena interpretasi kode Python bergantung pada indentasi. Jika indentasi tidak konsisten atau tidak sesuai, maka program tidak akan berjalan dengan benar atau dapat menyebabkan kesalahan sintaks.

1.5.4 Variable

Variabel adalah sebuah tempat untuk menyimpan nilai. Dalam Python, Anda tidak perlu mendeklarasikan variabel secara eksplisit. Variabel akan dibuat secara otomatis ketika Anda pertama kali memberikan nilai kepadanya (Pane and Saputra, 2020). Cara mendeklarasikan variabel dalam Python dapat dilihat pada contoh Program 1.5 berikut ini.

#Program 1.5
<pre>x = 10 # secara otomatis diinisialisasi sebagai integer (int) y = "belajar python" # secara otomatis diinisialisasi sebagai string (str) print(x) # menampilkan isi variabel x print(y) # menampilkan isi variabel y</pre>
<p>Output:</p> <pre>10 belajar python</pre>

Pada Program 1.5 tersebut, kita mendefinisikan dua variabel:

1. Variabel `x` diinisialisasi dengan nilai 10. Python secara otomatis menentukan jenis variabel berdasarkan nilai yang diberikan, dalam hal ini, `x` akan dianggap sebagai integer (int).
2. Variabel `y` diinisialisasi dengan string "belajar python". Kembali, Python secara otomatis menentukan bahwa variabel `y` adalah tipe data string (str) berdasarkan nilai yang diberikan.

Kemudian, kita menggunakan pernyataan print untuk menampilkan isi dari variabel `x` dan variabel `y` ke konsol. Output yang dihasilkan dari program ini akan mencetak nilai variabel `x` (10) dan nilai variabel `y` ("belajar python") secara berurutan. Variabel dalam Python dapat digunakan untuk menyimpan dan memanipulasi data. Mereka memungkinkan kita untuk merujuk ke nilai yang tersimpan dengan nama variabel yang kita tentukan sendiri.

1.5.4.1 Membuat Nama Variable

Berikut adalah beberapa aturan yang perlu diikuti saat menamai variabel dalam Python:

1. Karakter yang diperbolehkan: Nama variabel dalam Python dapat terdiri dari huruf (baik huruf besar maupun huruf kecil), angka, dan garis bawah (`_`). Namun, nama variabel harus dimulai dengan huruf atau garis bawah. Python membedakan huruf besar dan huruf kecil, sehingga `nama` dan `Nama` dianggap sebagai dua variabel yang berbeda.
2. Karakter khusus yang tidak diperbolehkan: Beberapa karakter khusus seperti spasi, tanda titik (`.`), tanda dolar (`$`), dan simbol matematika

tidak diperbolehkan dalam penamaan variabel. Selain itu, Python juga memiliki kata kunci yang tidak dapat digunakan sebagai nama variabel karena mereka memiliki arti khusus dalam bahasa Python, seperti ``if``, ``for``, ``while``, dll.

3. Pilihan nama yang deskriptif: Pilihlah nama variabel yang deskriptif dan mencerminkan makna nilai yang diwakili oleh variabel tersebut. Nama variabel yang baik dapat membantu memahami tujuan dan fungsi variabel ketika membaca atau memelihara kode di kemudian hari.
4. Case-sensitive: Python memperlakukan huruf besar dan huruf kecil sebagai karakter yang berbeda. Misalnya, ``nilai`` dan ``Nilai`` dianggap sebagai dua variabel yang berbeda. Oleh karena itu, perhatikan penulisan huruf besar dan kecil saat mengacu pada variabel.
5. Menghindari konflik dengan nama bawaan: Python memiliki sejumlah nama bawaan (built-in) yang memiliki fungsi dan arti tertentu dalam bahasa. Sebaiknya hindari menggunakan nama yang sama dengan nama bawaan ini untuk menghindari konflik dan kebingungan.
6. Menggunakan gaya penulisan yang konsisten: Pilihlah satu gaya penulisan dan tetap gunakan dengan konsisten dalam penamaan variabel. Contoh gaya penulisan yang umum digunakan adalah gaya `snake_case`, di mana kata-kata dipisahkan oleh garis bawah (misalnya: ``nama_depan``, ``jumlah_barang``).

Mematuhi aturan-aturan ini akan membantu menjaga konsistensi dan kejelasan dalam penamaan variabel dalam kode Python.

1.5.4.2 Memberi Nilai Banyak Variabel

Anda dapat memberikan nilai pada banyak variabel dengan tipe data yang sama dalam satu baris dengan menggunakan pemisah koma. Berikut adalah contoh cara memberikan nilai pada beberapa variabel dengan tipe data yang sama:

#Program 1.6
<pre>x, y, z = "santi", "citra", "Rozy" print(x, y, z) # cara menampilkan semua variabel tanpa turun baris print(y) print(z)</pre>
Output: santi citra Rozy citra Rozy

Berikut penjelasan dari Program 1.6:

1. Baris pertama menginisialisasi variabel `x`, `y`, dan `z` dengan nilai masing-masing "santi", "citra", dan "Rozy". Dalam satu baris, nilai-nilai ini diberikan kepada variabel-variabel tersebut secara berurutan.
2. Baris kedua menggunakan pernyataan print untuk menampilkan nilai semua variabel (`x`, `y`, dan `z`) pada satu baris, tanpa turun ke baris baru antara variabel-variabelnya. Outputnya akan mencetak "santi citra Rozy" ke layar.
3. Baris ketiga mencetak nilai variabel `y` ke layar pada baris baru. Outputnya akan mencetak "citra".
4. Baris keempat mencetak nilai variabel `z` ke layar pada baris baru. Outputnya akan mencetak "Rozy".

Semua variabel dapat diberikan nilai dalam satu baris dengan menggunakan pemisah koma antara nilai-nilai tersebut. Kemudian, nilai-nilai variabel tersebut dapat dicetak sesuai kebutuhan.

1.5.4.3 Variabel Global

Variabel dapat memiliki cakupan (scope) global atau lokal. Perbedaannya adalah variabel global dapat diakses dan digunakan oleh semua fungsi dan kelas dalam program, sedangkan variabel lokal hanya dapat digunakan di dalam fungsi tertentu di mana variabel tersebut dideklarasikan. Berikut adalah contoh yang menggambarkan perbedaan antara variabel global dan lokal, seperti terlihat pada Program 1.7.

#Program 1.7
<pre> x = "awesome" # didefinisikan di bagian awal program, variabel x dapat digunakan pada semua fungsi def myfunc(): y = "fantastic" # variabel y hanya digunakan pada fungsi print("Python is " + x + " and " + y) def myfunc_2(): y = "oke" # variabel y hanya digunakan pada fungsi print("Python is " + x + " and " + y) myfunc() myfunc_2() </pre>
<p>Output:</p> <pre> Python is awesome and fantastic Python is awesome and oke </pre>

Dalam Program 1.7 tersebut, variabel `x` didefinisikan di bagian awal program dan menjadi variabel global yang dapat diakses oleh semua fungsi. Variabel `y` didefinisikan di dalam setiap fungsi sehingga hanya dapat digunakan di dalam fungsi tersebut (variabel lokal). Pada saat memanggil fungsi `myfunc()`, variabel `x` digunakan bersama dengan variabel lokal `y` yang nilainya adalah "fantastic". Hasilnya mencetak "Python is awesome and fantastic".

Pada saat memanggil fungsi `myfunc_2()`, variabel `x` juga digunakan bersama dengan variabel lokal `y` yang nilainya adalah "oke". Hasilnya mencetak "Python is awesome and oke". Jika Anda ingin mengubah nilai variabel global `x` dari dalam fungsi, Anda harus mendeklarasikan variabel tersebut sebagai global menggunakan pernyataan `global`. Contohnya seperti di bawah ini:

#Program 1.8
<pre> x = "awesome" def myfunc(): global x x = "awesome_" # nilai x berubah y = "fantastic" print("Python is " + x + " and " + y) def myfunc_2(): y = "oke" print("Python is " + x + " and " + y) myfunc() myfunc_2() </pre>
<p>Output:</p> <pre> Python is awesome_ and fantastic Python is awesome_ and oke </pre>

Dalam contoh Program 1.8 ini, dengan mendeklarasikan `global x` di dalam fungsi `myfunc()`, kita dapat mengubah nilai variabel global `x` menjadi "awesome_". Hasilnya mencetak "Python is awesome_ and fantastic" saat memanggil `myfunc()`. Saat memanggil `myfunc_2()`, variabel global `x` yang telah diubah juga digunakan bersama dengan variabel lokal `y`, dan mencetak "Python is awesome_ and oke".

1.6 SOAL DAN KUNCI JAWABAN

Untuk meningkatkan pemahaman pembaca, penting untuk menyediakan soal dan kunci jawaban yang dapat membantu mereka. Berikut adalah daftar sepuluh pertanyaan yang perlu diperhatikan:

1. Apa yang dimaksud dengan Python dan kapan pertama kali ditemukan?
2. Apa keunggulan Python dibandingkan dengan bahasa pemrograman lain?

3. Apa itu Anaconda dan apa manfaatnya dalam pengembangan proyek data science?
4. Bagaimana cara menginstal Anaconda?
5. Bagaimana cara menjalankan Jupyter Notebook?
6. Apa saja bagian utama yang ada pada tampilan awal Jupyter Notebook?
7. Apa saja aturan penulisan dalam Python yang penting untuk diikuti?
8. Apa perbedaan antara memisahkan baris program dengan menggunakan enter dan menggunakan tanda semikolon (;)?
9. Mengapa komentar penting dalam pemrograman Python? Bagaimana cara menulis komentar dalam Python?
10. Apa itu blok dalam Python dan bagaimana indentasi digunakan untuk menandai blok kode?

Kunci Jawaban :

1. Python adalah bahasa pemrograman tingkat tinggi yang diciptakan oleh Guido van Rossum. Python pertama kali diperkenalkan pada tahun 1991.
2. Keunggulan Python dibandingkan dengan bahasa pemrograman lain antara lain:
 - a. Mudah dipahami dan mudah dibaca karena sintaksnya yang bersifat eksplisit.
 - b. Memiliki banyak library dan modul yang dapat digunakan untuk berbagai keperluan.
 - c. Mendukung pemrograman berorientasi objek.
 - d. Memiliki pengelolaan memori otomatis (garbage collection).
 - e. Bersifat portable dan dapat dijalankan di berbagai sistem operasi.
3. Anaconda adalah platform distribusi Python yang menyediakan berbagai paket dan library yang populer dalam pengembangan proyek data science. Anaconda menyediakan lingkungan virtual yang dapat memudahkan manajemen dan isolasi proyek-proyek Python yang berbeda.
4. Untuk menginstal Anaconda, Anda dapat mengikuti langkah-langkah berikut:
 - a. Unduh paket instalasi Anaconda dari situs web resminya.
 - b. Ikuti panduan instalasi yang disediakan.
 - c. Pilih opsi instalasi yang sesuai dengan sistem operasi Anda.

- d. Setelah proses instalasi selesai, Anaconda siap digunakan.
5. Untuk menjalankan Jupyter Notebook, ikuti langkah-langkah berikut:
 - a. Buka terminal atau command prompt.
 - b. Ketik perintah "jupyter notebook" dan tekan Enter.
 - c. Ini akan membuka Jupyter Notebook dalam browser web default Anda.
6. Pada tampilan awal Jupyter Notebook, Anda akan melihat beberapa komponen utama seperti:
 - a. Daftar file dan folder di direktori saat ini.
 - b. Tampilan daftar notebook yang ada.
 - c. Tombol untuk membuat notebook baru atau mengelola notebook yang ada.
7. Beberapa aturan penulisan penting dalam Python antara lain:
 - a. Menggunakan indentasi untuk menandai blok kode.
 - b. Menggunakan huruf besar dan huruf kecil secara berbeda (case-sensitive).
 - c. Menggunakan tanda kutip tunggal atau ganda untuk string.
 - d. Menggunakan kata kunci yang telah ditentukan dengan benar.
8. Perbedaan antara memisahkan baris program dengan menggunakan enter dan menggunakan tanda semikolon (;) adalah sebagai berikut:
 - a. Memisahkan baris program dengan enter mengikuti konvensi penulisan Python yang baik, yang mengutamakan keterbacaan kode.
 - b. Menggunakan tanda semikolon (;) memungkinkan Anda untuk menulis beberapa pernyataan dalam satu baris, tetapi mengurangi keterbacaan kode.
9. Komentar penting dalam pemrograman Python karena membantu menjelaskan maksud dan tujuan kode, serta memberikan informasi penting kepada pembaca atau pengembang kode. Cara menulis komentar dalam Python adalah dengan menggunakan tanda pagar (#) diikuti oleh teks komentar.
10. Blok dalam Python adalah sekumpulan pernyataan yang terdiri dari beberapa baris yang dieksekusi bersama sebagai satu unit. Indentasi digunakan untuk menandai blok kode dalam Python. Setiap pernyataan yang memiliki indentasi yang sama dianggap menjadi bagian dari blok yang sama.

DAFTAR PUSTAKA

- Awangga, R.M., Prastya, R., Simamora, T.A.D., Majesty, D., Fadilah, I., 2019. Dasar-dasar Python. Kreatif.
- Baihaqi, W.M., Kom, S., 2021. Pemrograman Berbasis Event dengan PyQt5. Zahira Media Publisher.
- Enterprise, J., 2021. Belajar Sendiri Mengolah Data dengan Python dan Pandas. Elex Media Komputindo.
- Hillar, G.C., 2015. Learning Object-oriented programming. Packt Publishing Ltd.
- Hughes, J.M., 2010. Real World Instrumentation with Python: Automated Data Acquisition and Control Systems. “O’Reilly Media, Inc.”
- Kuhlman, D., 2009. A python book: Beginning python, advanced python, and python exercises. Dave Kuhlman Lutz.
- Lutz, M., 2013. Learning python: Powerful object-oriented programming. “O’Reilly Media, Inc.”
- Marvin, R., Omondi, A., 2018. Python Fundamentals: A practical guide for learning Python, complete with real-world projects for you to explore. Packt Publishing Ltd.
- Navlani, A., Fandango, A., Idris, I., 2021. Python Data Analysis: Perform data collection, data processing, wrangling, visualization, and model building using Python. Packt Publishing Ltd.
- Pane, S.F., Saputra, Y.A., 2020. Big Data: Classification Behavior Menggunakan Python. Kreatif.
- Phillips, D., 2018. Python 3 object-oriented programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8. Packt Publishing Ltd.
- Phillips, D., 2015. Python 3 object-oriented programming. Packt Publishing Ltd.
- Rahman, S., Ramli, M., Arnia, F., Muharar, R., Zen, M., Ikhwan, M., 2021. Convolutional Neural Networks Untuk Visi Komputer Jaringan Saraf Konvolusional untuk Visi Komputer (Arsitektur Baru, Transfer Learning, Fine Tuning, dan Pruning). Deepublish.
- Romzi, M., Kurniawan, B., 2020. Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma. JTIM: Jurnal Teknik Informatika

- Mahakarya 3, 37–44.
- Smith, K.W., 2015. Cython: A Guide for Python Programmers. “ O’Reilly Media, Inc.”
- Vini Indriasari, S.T., 2021. Scripting untuk ArcGIS dengan Python dan ArcPy. Deepublish.

BIOGRAFI PENULIS



Sayuti Rahman merupakan Dosen di Universitas Medan Area. Telah menyelesaikan studi S3 di Universitas Syiah Kuala, Banda Aceh. Sejak karirnya sebagai dosen, Sayuti tertarik dengan mendalami bidang citra digital. Sehingga saat ini Sayuti fokus dalam penelitian yang berkaitan dengan mengekstraksi informasi didalam citra digital yaitu Visi komputer. Bahasa pemrograman yang sering digunakan dalam penelitian adalah Python. Penelitian yang sedang dikerjakan saat ini adalah membangun sistem parkir cerdas dengan mengoptimalkan CNN dalam mengklasifikasi ruang parkir. Sayuti Rahman lahir di Bukit gantung, kec Wampu kab. Langkat pada tanggal 18 juni 1987.



Arnes Sembiring merupakan dosen di Universitas Medan Area. Studi S3 diselesaikan pada awal tahun 2023 di Universitas Syiah Kuala, Banda Aceh. Topik penelitian yang diminati berkaitan dengan kecerdasan buatan, *smart system*, pengolahan citra, pemodelan dan simulasi. Penelitian yang dilakukan saat ini adalah pengembangan arsitektur CNN yang efisien dan implementasinya dalam sistem pertanian pintar, baik yang dikembangkan pada lingkungan Matlab maupun Python. Penulis dapat dihubungi melalui email di arnessembiring@gmail.com.



Dodi Siregar. Alumnus Universitas Putra Indonesia "YPTK" Padang. Ia adalah dosen tetap Program Studi Teknik Informatika, Fakultas Teknik dan Komputer, Universitas Harapan Medan. Aktif diberbagai organisasi, antara lain: Relawan Jurnal Indoensia (RJI) Pengurus Daerah Sumatera Utara, APTIKOM Sumatera Utara, APTISI Sumatera Utara, IAI Sumatera Utara, Forum Komunikasi Pendidikan Tinggi (FKPT).

Lahir di tanjung Maria pada tanggal 8 Januari 1971.

Ayah tiga orang anak ini mengampu mata kuliah Etika Profesi Teknologi Informasi, Proposal Tugas Akhir, Sistem Pendukung Keputusan, Riset Teknologi Informasi dan Kerja Praktek. Aktif membantu dan mendampingi para pengelola jurnal elektronik diberbagai kampus maupun institusi lainnya. Ia juga pengelola jurnal elektronik dibidang teknik informatika dan teknologi informasi. Menjadi reviewer dan editorial tim diberbagai jurnal elektronik. Telah menulis 4 Buku referensi dengan cara berkolaborasi dengan penulis lainnya dari segala penjuru tanah air.



Husnul khair, M.Kom. lahir di kuala simpang tepatnya desa bukit rata 7 april 1985, telah menyelesaikan Diploma III tahun 2010 dan strata 1 Tahun 2013 pada program studi Teknik Informatika di STMIK Kaputama serta Megister Teknik Informatika di Universitas Sumatera Utara tahun 2018 dan sebagai Dosen Teknik Informatika Pada STMIK Kaputama sejak tahun 2018 dan mulai memahami serta mendalami beberapa matakuliah pemrograman dan jaringan komputer.

Pernah menjadi dosen pembimbing lapangan pada program MBKM Kampus Mengajar Angkatan 1 dan 2, serta juga merupakan IT Support pada beberapa Kampus di sumatera utara dan juga Instansi pemerintah. Serta selalu berusaha menjadi Dosen yang ramah dan tidak sombong serta bermanfaat untuk orang lain.



I Gusti Prahmana merupakan seorang dosen ilmu komputer STMIK Kaputama, Kota Binjai. Alumni program studi megister Teknik Informatika Fakultas Ilmu Komputer Universitas Sumatera Utara. I Gusti Prahmana adalah peneliti yang tertarik pada *machine learning* citra digital Bahasa pemrograman *Python*. Lahir pada tanggal 27 Agustus Tahun 1992 , di Kota Binjai.



Ratih Puspadini merupakan Dosen di STMIK Kaputama di Binjai, dengan bidang keahlian Elektronika, Logika Digital/Informatika, Mikrokontroller, Embedded System, Pemrograman Mobile, Machine Learning, Kecerdasan Buatan. Lahir di Medan, menyelesaikan studi S2 Teknik Informatika dan S1 Teknik Elektro di Universitas Sumatera Utara, Medan.



Muhammad Zen adalah seorang dosen di Universitas Pembangunan Pancabudi, Medan. Menyelesaikan studi magister di Universitas Putra Indonesia “Yptk” Padang pada tahun 2017 dan sarjana di Universitas Harapan Medan pada tahun 2014. Muhammad Zen adalah seorang peneliti yang tertarik pada bidang Sistem Informasi, Keamanan Data dan Internet of Things. Saat ini Muhammad Zen diberikan tugas tambahan sebagai tenaga ahli pada Biro Pengembangan Sistem Informasi UNPAB. Muhammad Zen juga baru saja lulus sertifikasi dosen pada tahun 2023 ini.

Buku ini adalah panduan lengkap yang ditujukan untuk mahasiswa dan praktisi yang ingin mempelajari bahasa pemrograman Python. Penulis menjelaskan dengan jelas dan terstruktur tentang dasar-dasar Python dan pemrograman berorientasi objek. Buku ini dirancang khusus untuk memperkenalkan Python kepada mahasiswa yang mungkin belum terlalu akrab dengan bahasa pemrograman ini. Dengan menggunakan contoh-contoh dan latihan praktis, pembaca akan memperoleh pemahaman yang komprehensif tentang Python dan kemampuan untuk mengaplikasikan konsep pemrograman berorientasi objek dalam pengembangan perangkat lunak.

Buku "PYTHON: DASAR DAN PEMROGRAMAN BERORIENTASI OBJEK" terdiri dari bab-bab yang terstruktur dengan baik. Bab pertama hingga keenam membahas dasar-dasar pemrograman Python, mencakup pengenalan bahasa, penggunaan variabel, struktur kontrol, dan fungsi. Sementara itu, bab-bab berikutnya, yaitu bab ketujuh hingga bab kedua belas, membahas pemrograman berorientasi objek, termasuk konsep class, objek, pewarisan (inheritance), polimorfisme, dan enkapsulasi. Dalam buku ini, pembaca akan dibimbing melalui penjelasan yang jelas dan contoh pengkodean yang relevan, sehingga mereka dapat memahami konsep-konsep tersebut dan mengembangkan keterampilan pemrograman Python yang solid.



CV. Tahta Media Group
Surakarta, Jawa Tengah
Web : www.tahtamedia.com
Ig : tahtamedia group
Telp/WA : +62 896 5427 3996

ISBN 978-623-147-100-0

