



**MELATIH LOGIKA  
DENGAN ALGORITMA**  
di **Python**

**Heribertus Ary Setyadi  
Yusuf Sutanto**

#### 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 digurakan 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).

---

# MELATIH LOGIKA

Dengan Algoritma di Python

Heribertus Ary Setyadi  
Yusuf Sutanto



---

**MELATIH LOGIKA DENGAN ALGORITMA DI PYTHON**

**Heribertus Ary Setyadi dan Yusuf Sutanto**

Desain Cover :  
**Rulie Gunadi**

Sumber :  
www.shutterstock.com (Lano4ka)

Tata Letak :  
**Zulita A.**

Proofreader :  
**A. Timor Eldian**

Ukuran :  
**x, 136 hlm, Uk: 15.5x23 cm**

ISBN :  
**978-623-02-7485-5**

Cetakan Pertama :  
**November 2023**

Hak Cipta 2023, Pada Penulis

---

Isi diluar tanggung jawab percetakan

---

**Copyright © 2023 by Deepublish Publisher**  
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 DEEPUBLISH**  
**(Grup Penerbitan CV BUDI UTAMA)**

Anggota IKAPI (076/DIY/2012)

Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoharjo, Ngaglik, Sleman.

Jl.Kaliurang Km.9,3 – Yogyakarta 55581

Telp/Faks: (0274) 4533427

Website: [www.deepublish.co.id](http://www.deepublish.co.id)

[www.penerbitdeepublish.com](http://www.penerbitdeepublish.com)

E-mail: [cs@deepublish.co.id](mailto:cs@deepublish.co.id)

## KATA PENGANTAR PENERBIT

Segala puji kami haturkan kepada Tuhan Yang Maha Esa, atas limpahan segala anugerah dan karunia-Nya. Dalam rangka mencerdaskan dan memuliakan umat manusia dengan penyediaan serta pemanfaatan ilmu pengetahuan dan teknologi untuk menciptakan industri *processing* berbasis sumber daya alam (SDA) Indonesia, Penerbit Deepublish dengan bangga menerbitkan buku dengan judul *Melatih Logika dengan Algoritma di Python*.

Terima kasih dan penghargaan terbesar kami sampaikan kepada penulis yang telah memberikan kepercayaan, perhatian, dan kontribusi penuh demi kesempurnaan buku ini. Semoga buku ini bermanfaat bagi semua pembaca, mampu berkontribusi dalam mencerdaskan dan memuliakan umat manusia, serta mengoptimalkan pemanfaatan ilmu pengetahuan dan teknologi di tanah air.

Hormat Kami,

**Penerbit Deepublish**

## KATA PENGANTAR

Puji syukur kepada Dzat semesta alam atas selesainya buku *Melatih Logika dengan Algoritma di Python*. Modal dasar kuliah di bidang komputer adalah logika dan bahasa algris. Tidak ada orang yang logikanya rendah, yang ada hanya orang yang tidak mau melatih logika. Ingatlah bahwa kemampuan mengikuti kemauan. Dengan melatih logika berpikir secara sistematis maka akan dengan mudah memahami bahasa pemrograman apapun.

Buku ini membahas sekaligus melatih logika dengan menggunakan algoritma pemrograman Python. Pembahasan dimulai dari konsep logika dan algoritma dengan berbagai contoh kasus dalam kehidupan sehari-hari. Melatih logika dimulai dari penggunaan percabangan dengan membahas berbagai bentuk kondisi disertai penjelasan logika program juga gambar diagram alur. Hal yang sama juga terdapat pada bab dalam membahas perulangan, larik, fungsi dan membuat modul sendiri. Yang paling menarik dalam buku ini adalah penjelasan dalam membuat modul sehingga dapat melakukan kustomisasi program Python dengan perintah atau pernyataan buatan sendiri. Pada bab terakhir dijelaskan logika pengurutan dan pencarian data dengan menggunakan beberapa metode. Semua pembahasan disertai penjelasan logika dalam menyelesaikan suatu kasus sebelum memulai membuat program.

Demikian semoga buku ini dapat bermanfaat bagi pembaca semua. Diharapkan adanya masukan yang membangun untuk perbaikan buku ini dalam edisi mendatang.

# DAFTAR ISI

KATA PENGANTAR PENERBIT.....	v
KATA PENGANTAR .....	vi
DAFTAR ISI.....	vii
<b>BAB 1 Logika Algoritma.....</b>	<b>1</b>
1.1 Pengertian Logika .....	1
1.2 Pengertian Algoritma .....	2
<b>BAB 2 Bahasa Pemrograman Python.....</b>	<b>12</b>
2.1 Pengantar Python.....	12
2.2 Instalasi Python.....	13
2.3 Instalasi Editor Visual Studio Code.....	14
2.4 Tipe Data di Python .....	16
2.5 Operator di Python.....	23
2.6 Input di Python .....	27
2.7 Ouput di Python.....	30
<b>BAB 3 Percabangan .....</b>	<b>32</b>
3.1 Konsep Percabangan.....	32
3.2 Kondisi Tunggal ( <i>if</i> ).....	32
3.3 Kondisi <i>if-else</i> .....	34
3.4 Kondisi <i>if-elif</i> .....	36
3.5 Kondisi <i>if</i> Majemuk.....	40
3.6 Kondisi <i>if</i> Tersarang.....	44
3.7 Kondisi Komposit.....	46
3.8 Kondisi Menggunakan Proposisi.....	49
3.9 Contoh Kasus Percabangan .....	51
<b>BAB 4 Perulangan .....</b>	<b>55</b>
4.1 Konsep Perulangan .....	55
4.2 Perulangan Menggunakan <i>For</i> .....	56

4.3	Perulangan Menggunakan <i>While</i> .....	66
4.4	Perulangan Bersarang .....	75
4.5	Contoh Kasus Perulangan Bersarang .....	76
4.6	Perintah <i>Break</i> .....	82
4.7	Perintah <i>Continue</i> dalam Perulangan Bersarang .....	83
<b>BAB 5</b>	<b>Array (Larik).....</b>	<b>84</b>
5.1	Pengertian <i>Array</i> di Python.....	84
5.2	Larik Pengganti <i>If</i> .....	84
5.3	Larik Dua dimensi.....	86
5.4	Jenis Matriks .....	87
5.5	Operasi pada Matriks.....	90
5.6	Modul <i>Array</i> .....	92
5.7	Modul <i>Numpy</i> .....	94
<b>BAB 6</b>	<b>Fungsi (Function).....</b>	<b>95</b>
6.1	Fungsi Bawaan ( <i>Built In</i> ) .....	95
6.2	Pengertian Fungsi Bawaan .....	97
6.3	Fungsi dengan Parameter.....	98
6.4	Fungsi dengan Beberapa <i>Return</i> .....	100
6.5	Contoh Kasus Menggunakan Function.....	101
6.6	Fungsi yang Memanggil dirinya Sendiri (Rekursi) .....	102
<b>BAB 7</b>	<b>Modul.....</b>	<b>107</b>
7.1	Pengertian <i>Module</i> .....	107
7.2	Membuat dan Memanggil <i>Module</i> .....	108
7.3	Membuat Alias.....	109
<b>BAB 8</b>	<b>Pengurutan Menggunakan <i>Array</i>.....</b>	<b>112</b>
8.1	Pengantar Pengurutan.....	112
8.2	<i>Bubble Sort</i> .....	113
8.3	<i>Selection Sort</i> .....	116
8.4	<i>Insertion Sort</i> .....	118
8.5	<i>Merge Sort</i> .....	120
8.6	<i>Quick Sort</i> .....	123
8.7	Fungsi <i>Sort()</i> .....	125

<b>BAB 9 Pencarian Menggunakan <i>Array</i></b> .....	127
9.1 Pengantar Pencarian.....	127
9.2 <i>Sequential Search</i> .....	127
9.3 <i>Binary Search</i> .....	130
9.4 Pencarian Data Tanpa Metode.....	132
<b>DAFTAR PUSTAKA</b> .....	134
<b>BIODATA PENULIS</b> .....	136

deeppublish / publisher

Melatih Logika dengan Algoritma di Python

# Logika Algoritma

## 1.1 Pengertian Logika

Secara etimologis, logika berasal dari bahasa Yunani yaitu *logos* yang memiliki beberapa arti antara lain pikiran, akal budi, pengertian, ucapan, bahasa, kata dan ilmu. Dari kata *logos* lalu diturunkan menjadi kata sifat yaitu *logis* yang sudah tidak asing dalam percakapan sehari-hari. Pembicaraan tentang perilaku yang logis sebagai lawan terhadap perilaku yang tidak logis yaitu tentang tata cara yang logis, tentang penjelasan yang logis, tentang jalan pikiran yang logis dan sejenisnya. Dalam semua contoh tersebut, kata logis kurang lebih memiliki arti yang sama dengan „masuk akal“. Secara singkat, logis adalah segala sesuatu yang sesuai dengan juga dapat diterima oleh akal sehat.

Sebagai ilmu, logika disebut dengan *logike episteme* (Latin: *logica scientia*) atau ilmu logika (ilmu pengetahuan) yang mempelajari kecakapan dalam berpikir secara lurus, tepat dan teratur. Logika merupakan studi tentang penggunaan prinsip dan metode untuk menganalisa dan membedakan penalaran yang benar dari yang salah. Saat menganalisa dan bernalar tentang suatu masalah maka akan menghasilkan argumen yang membenarkan keyakinan dalam mendukung kesimpulan. Akan tetapi tidak semua alasan merupakan alasan yang tepat. Penalaran bukan satu-satunya cara untuk mendukung argumen yang dibuat atau diterima. Ketika seseorang ingin membuat pernyataan yang dapat diandalkan maka penalaran yang benar merupakan satu-satunya dasar yang kuat. Penggunaan metode dan teknik *logic one* dapat membedakan antara penalaran yang sehat dan yang salah.

Logika merupakan sebuah ilmu pengetahuan dimana objek materialnya adalah berpikir (khususnya penalaran/proses penalaran) dan objek formal logika adalah berpikir/penalaran yang ditinjau dari segi ketepatannya. Logika bersifat a priori. Kebenaran logika tidak dapat ditemukan dan diuji secara empiris, tetapi kebenaran diuji secara akal. Objek Logika menurut Muhammad Zainuddin terdapat dari:

1. Objek materiil: penalaran / cara berpikir
2. Objek formal: hukum, prinsip, asas
3. Produk: produk berfikir (konsep, proposisi yang diekspresikan dalam bentuk ungkapan lisan atau tulisan).

Ilmu mantik bertujuan membimbing manusia ke arah berfikir yang benar, logis juga sistematis memiliki banyak manfaat antara lain:

1. Menjadikan daya pikir lebih tajam dan berkembang dengan cara banyak latihan berfikir, dengan demikian akan mampu melakukan analisis dan memecahkan permasalahan secara ilmiah dan runtut.
2. Membantu untuk berpikir tepat sehingga mampu menempatkan sesuatu pada posisinya dan mengerjakan sesuatu tepat waktu dengan kata lain dapat berpikir efektif dan efisien.
3. Mampu membedakan alur berpikir yang benar dan yang keliru agar dapat menghasilkan kesimpulan yang benar juga.
4. Membantu seseorang yang melatih logika untuk berpikir secara rasional, kritis, lurus, tetap, tertib, metodis dan koheren.
5. Menambah kemampuan dalam berpikir secara abstrak, cermat juga objektif.
6. Meningkatkan kecerdasan dan kemampuan berpikir tajam secara mandiri.
7. Meningkatkan rasa cinta akan kebenaran ilmiah dengan menghindari kesalahan, kekeliruan serta kesesatan berpikir.

## 1.2 Pengertian Algoritma

Algoritma adalah urutan langkah dalam menyelesaikan masalah yang telah disusun secara logis dan sistematis. Pada beberapa konteks, algoritma dapat didefinisikan sebagai urutan langkah-langkah yang spesifik dalam menyelesaikan suatu pekerjaan. Pengertian

## 2.1 Pengantar Python

Python adalah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido van Rossum pada tahun 1991 dan dikembangkan lebih lanjut oleh *Python Software Foundation*. Python adalah bahasa pemrograman yang memungkinkan untuk bekerja dengan cepat dan mengintegrasikan sistem dengan lebih efisien. Bahasa pemrograman python sangat cocok untuk pemula juga untuk orang yang berpengalaman dengan bahasa pemrograman lain seperti C++ dan Java. Saat ini sebagian besar perusahaan mencari karyawan yang paham pemrograman python. Orang yang memahami pemrograman python lebih banyak memiliki peluang dalam bekerja sebagai pembuat atau desain program. Jadi mahasiswa atau pemrogram pemula harus mulai belajar python dan unggul di dalamnya. Beberapa alasan Python lebih populer antara lain :

1. Kode yang lebih pendek dan kemudahan penulisan
2. Pemrogram dapat mengekspresikan konsep logis dalam baris kode yang lebih sedikit dibandingkan dengan bahasa seperti C++ atau Java.
3. Python mendukung beberapa paradigma pemrograman, seperti pemrograman atau prosedural yang berorientasi objek, imperatif dan fungsional.
4. Ada fungsi bawaan untuk hampir semua konsep yang sering digunakan.
5. Filosofi Python adalah "kesederhanaan adalah yang terbaik".

## Perbandingan pemrograman java dan python

Pemrograman java:

```
public class HelloWorld
{ public static void main
(String[] args)
{ System.out.println("Hello,
world!"); } }
```

Pemrograman Python:

```
print("Hello, world!")
```

## 2.2 Instalasi Python

Unduh python IDLE dari situs resmi Python dengan alamat <https://www.python.org/downloads/>

Gambar 2.1. Tampilan Situs Resmi Python

Pilih versi Python yang diinginkan lalu klik *download*. Dalam buku ini masih menggunakan Python versi 3.5.

Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64
Windows x86 embeddable zip file	Windows	
Windows x86 executable installer	Windows	
Windows x86 web-based installer	Windows	

Gambar 2.2. Tampilan Pilihan Versi Python

### 3.1 Konsep Percabangan

Tidak seluruh baris perintah dalam program selalu dieksekusi. Sering suatu perintah akan diproses jika memenuhi suatu syarat atau kondisi yang telah ditentukan. Pada umumnya kondisi berbentuk *boolean* yang merupakan suatu ekspresi relasional bernilai *true* atau *false* tergantung dari nilai setiap operasi yang digunakan. Untuk menentukan kondisi dan aksi yang akan diberikan tergantung dari jumlah kasus atau pilihan yang dari permasalahan yang ada, apakah hanya ada satu pilihan, dua pilihan atautkah terdiri atas banyak pilihan. Dengan demikian percabangan adalah struktur program yang berfungsi untuk melakukan proses pengujian terhadap satu atau dua atau banyak kondisi. Pernyataan *if* adalah bentuk pernyataan untuk mengambil keputusan yang paling sederhana dan digunakan untuk memutuskan apakah suatu pernyataan atau blok pernyataan akan dieksekusi atau tidak yaitu jika suatu kondisi bernilai benar maka blok pernyataan dieksekusi jika salah maka tidak dieksekusi.

### 3.2 Kondisi Tunggal (*if*)

Percabangan jenis ini, hanya ada satu kondisi yang menjadi syarat untuk melakukan satu buah atau satu blok instruksi. Pernyataan *if* menentukan sebuah kondisi (atau blok kode) yang akan dieksekusi jika dan hanya jika kondisi bernilai benar. Sintaksis :

```
if (kondisi):  
    blok pernyataan yang dieksekusi
```

jika kondisi benar

Kondisi bisa ditulis dengan atau tanpa menggunakan sepasang tanda kurung. Suatu blok pernyataan ditulis menjorok (tidak rata kiri) atau identasi. Jadi blok di bawah pernyataan *if* akan ditulis seperti yang ditunjukkan pada contoh berikut:

```
if kondisi :  
    pernyataan benar1  
    pernyataan benar2  
pernyataan3
```

Jika kondisi bernilai benar maka pernyataan benar1 dan benar2 akan dieksekusi namun jika kondisi bernilai salah maka pernyataan3 yang akan dieksekusi. Diagram alir dapat digambarkan sebagai berikut:



Gambar 3.1. Diagram Alir Kondisi Tunggal

#### 4.1 Konsep Perulangan

Dalam pemrograman, perulangan adalah sekumpulan perintah atau instruksi yang dijalankan komputer berulang kali hingga suatu kondisi tertentu bernilai benar atau kondisi tersebut telah terpenuhi. Perulangan (*looping*) merupakan bagian yang tugasnya melaksanakan tindakan untuk mengulang proses sesuai keinginan. Banyak program atau perangkat lunak yang melakukan pekerjaan diulang-ulang sampai tercapai suatu kondisi yang telah ditentukan. Inilah sebabnya perulangan adalah bagian penting dari pemrograman, karena dengan perulangan programmer tidak perlu repot untuk menulis kode program sebanyak yang dibutuhkan.

Setiap bahasa pemrograman memiliki sintaks struktur perulangan tersendiri, namun secara umum ada aturan struktur perulangan yang harus diikuti, yaitu:

1. Inisialisasi (penetapan nilai) adalah langkah awal untuk proses mengulang kondisi suatu baris perintah, termasuk memberi nilai awal ke dalam suatu variabel. Langkah ini dilakukan sebelum masuk ke bagian utama perulangan.
2. Proses: sebuah proses dieksekusi dalam perulangan yang berisi perintah yang harus dilakukan berulang-ulang.
3. Perulangan: proses pengulangan terjadi di dalam blok perulangan yang merupakan kondisi tambahan untuk melanjutkan proses perulangan.
4. Perhentian: merupakan suatu kondisi untuk menghentikan proses perulangan. Kondisi untuk menghentikan perulangan

sangat penting untuk menghindari perulangan tak terbatas (terus menerus melakukan perulangan).

Terdapat dua bentuk perulangan dalam Python yaitu bentuk `for` dan `while`.

#### 4.2 Perulangan Menggunakan *For*

Pernyataan `for` dalam Python memiliki karakteristik sendiri jika dibandingkan dengan bentuk `for` dalam bahasa pemrograman C atau Pascal. Tidak sekedar melakukan perulangan dalam bentuk angka atau ekspresi aritmatika. Dengan `for` di Python dapat melakukan perulangan melalui berbagai tipe data sekuensial seperti `list`, `string` dan `tuple`. Sintaksis umum dari `for` adalah:

```
for variabel in sekuensial:  
    blok perintah
```

Tiga bagian utama dari perulangan `for` adalah:

1. Sekuensial: suatu nilai yang dapat diulang-ulang (*iterable*).  
Tipe data yang *iterable* antara lain: `list`, `string`, `tuple` dan lainnya.
2. Variabel: setiap elemen diekstraksi dari sekuensial.
3. Blok perintah: sekumpulan perintah yang akan dieksekusi secara berulang.

Diagram alir dari perulangan `for` secara garis besar dapat dilihat pada gambar 4.1.

## Array (Larik)

### 5.1 Pengertian *Array* di Python

Larik adalah kumpulan beberapa variabel yang memiliki tipe data sama dan dapat menyimpan banyak nilai sekaligus dalam satu variabel. Larik merupakan kumpulan item yang disimpan di lokasi memori (RAM) yang berdekatan. Larik termasuk tipe data terstruktur dalam pemrograman. Dengan menggunakan larik memungkinkan untuk menyimpan data yang banyak dan diindeks serta referensi objek. Dalam larik sekelompok variabel dapat memiliki nama yang sama. Larik di Python sedikit berbeda daripada di bahasa pemrograman lain (biasa dikelola dalam modul khusus *array*). Python memiliki tipe data yang dapat digambarkan mirip dengan larik, namun dengan karakteristik dan keunikannya sendiri. Tipe data yang termasuk dalam larik di Python antara lain *list*, *tupel*, *dictionary* dan *set*. Gambar 5.1 menyajikan ilustrasi alokasi penyimpanan larik di memori.



### 5.2 Larik Pengganti *If*

Larik dimensi satu dapat juga difungsikan untuk menyelesaikan suatu kasus yang biasanya menggunakan percabangan. Dengan

menggunakan larik maka suatu kasus percabangan dapat diselesaikan dengan program yang sangat singkat. Tidak semua kasus percabangan dapat diselesaikan dengan memanfaatkan larik. Terdapat kriteria yang harus dipenuhi dan harusnya bisa diketahui dengan beberapa contoh kasus di bawah ini.

Contoh kasus untuk menentukan nama hari saat dimasukkan angka 1 sampai 7. Biasanya kasus tersebut dikerjakan menggunakan percabangan. Hari dimulai dari senin yang bernilai 1 sampai minggu bernilai 7. Jika menggunakan percabangan maka membutuhkan `if` dan `else` untuk menyelesaikan kasus tersebut. Perhatikan penggunaan larik pada program di bawah ini:

```
hari = ["Senin", "Selasa", "Rabu", "Kamis",  
"Jumat", "Sabtu", "Minggu"]  
n = int(input('Masukkan nilai hari [1-7]: '))  
print('Nama harinya : ',hari[n-1])
```

Hasil eksekusi :

```
Masukkan nilai hari [1-7]: 3  
Nama harinya : Rabu
```

Dari program di atas, hanya dengan 3 baris perintah sudah dapat menyelesaikan kasus yang seharusnya terdiri dari lebih 7 baris perintah jika menggunakan percabangan. Bisa dibayangkan jika kasusnya menentukan nama bulan, membutuhkan 11 `if` jika diselesaikan dengan percabangan, namun cukup 3 baris perintah juga jika diselesaikan dengan larik.

Kasus konversi nilai angka skala 5 menjadi nilai huruf juga dapat diselesaikan dengan singkat menggunakan larik, seperti program di bawah ini:

```
nh = ["E", "D", "C", "B", "A"]  
n = int(input('Masukkan nilai [0-4]: '))  
print('nilai hurufnya : ',nh[n])
```

Dalam kasus menentukan nama kartu yang dijelaskan dan dibuat programnya pada bab 3 membutuhkan percabangan yang

## Fungsi (Function)

### 6.1 Fungsi Bawaan (*Built In*)

Fungsi bawaan Python adalah kumpulan perintah atau fungsi yang sudah disediakan oleh bahasa pemrograman Python. Fungsi atau perintah tersebut sudah dapat langsung digunakan sehingga tidak perlu lagi mengimpor suatu modul tertentu. Karena banyak fungsi bawaan yang telah disediakan, maka fungsi-fungsi tersebut dapat dikategorikan menjadi beberapa kelompok sebagai berikut:

#### 1. Fungsi Dasar

Nama	Fungsi
<code>print()</code>	Menampilkan teks dan/atau nilai variabel ke layar.
<code>int()</code>	Mengkonversi nilai menjadi tipe data integer.
<code>float()</code>	Mengkonversi nilai menjadi tipe data float.
<code>str()</code>	Mengkonversi nilai menjadi tipe data string.
<code>input()</code>	Mengambil masukan dari pengguna melalui keyboard.

#### 2. Fungsi Manipulasi Data

Nama	Fungsi
<code>len()</code>	Menghasilkan panjang (jumlah karakter) dari string, jumlah elemen dari list dan tuple.
<code>upper()</code>	Merubah tipe data string menjadi huruf kapital.
<code>lower()</code>	Merubah tipe data string menjadi huruf kecil.
<code>split()</code>	Memisahkan suatu string menjadi bagian-bagian yang dipisahkan berdasarkan karakter tertentu.
<code>append()</code>	Menambahkan elemen ke dalam list.
<code>sort()</code>	Mengurutkan elemen dalam daftar list secara ascending.

Nama	Fungsi
reverse()	Mengurutkan elemen dalam daftar list secara descending.
count()	Menghitung jumlah elemen yang muncul dari suatu nilai dalam tuple.
match()	Mencocokkan pola pada awal string.
search()	Melakukan pencarian pola di dalam string.
findall()	Menemukan semua kemunculan pola dalam string.
format()	Memformat string dengan nilai yang ditentukan.
strip()	Menghapus karakter spasi atau karakter khusus.
replace()	Menggantikan kemunculan suatu substring dengan substring lain dalam string.
chr()	Mengkonversi kode ascii menjadi karakter.
ord()	Mengkonversi karakter menjadi kode ascii.
bin()	Mengkonversi bilangan decimal menjadi biner.
hex()	Mengkonversi bilangan decimal menjadi hexadecimal.
oct()	Mengkonversi bilangan decimal menjadi oktal.
type()	Menghasilkan tipe data dari suatu variabel.

### 3. Fungsi Matematika dan Statistik

Nama	Fungsi
abs()	Menghasilkan nilai absolut (nilai positif) suatu angka.
max()	Menghasilkan nilai terbesar dari suatu objek.
min()	Menghasilkan nilai terkecil dari suatu objek.
pow()	Menghitung pangkat suatu angka.
sqrt()	Menghitung akar kuadrat suatu angka.
round()	Pembulatan angka
divmod()	Menghasilkan nilai hasil bagi dan sisanya.
sum()	Menghasilkan jumlah dari elemen-elemen suatu objek bertipe angka.
mean()	Menghitung rata-rata dari sekumpulan nilai angka.
median()	Menghitung nilai tengah dari sekumpulan angka.
var()	Menghitung simpangan baku dari sekumpulan nilai angka.
std()	Menghitung simpangan baku dari sekumpulan nilai angka.

### 4. Fungsi Tanggal dan Waktu

Nama	Fungsi
date()	Membuat objek tanggal dengan tahun, bulan, dan hari tertentu.

### 7.1 Pengertian *Module*

Dasar untuk membuat modul adalah fungsi atau di dalam modul banyak berisi fungsi buatan sendiri. Sebenarnya modul dalam Python sama dengan jika membuat program Python sebelumnya sebab sama-sama memiliki ekstension .py. Membuat modul memungkinkan untuk mengelola kode program secara logis. Dengan modul maka kode program dapat dikelompokkan secara terpisah dari program utama yang memanggil modul tersebut. Penggunaan modul dapat menyembunyikan kode program tertentu yang dianggap penting sehingga saat program Python dilihat orang lain tanpa menyalin modulnya maka program tidak dapat dieksekusi. Dalam program utama, modul dipanggil dengan perintah "import". Saat programmer menulis perintah import, artinya memanggil modul yang tersimpan di tempat tertentu. Dengan membuat modul juga dapat memasukkan nilai tertentu ke dalam suatu variabel yang nantinya langsung dapat digunakan dalam program utama.

Dengan membuat modul sering disebut dengan istilah pemrograman secara modular. Pendekatan modular maksudnya program yang sangat besar dipecah menjadi beberapa bagian yang lebih kecil secara terpisah dan dikelola dengan lebih rapi sesuai dengan fungsinya masing-masing. Terdapat beberapa keuntungan menggunakan metode modular, yaitu kode program menjadi lebih sederhana, memudahkan saat *maintenance* program dan modul yang dibuat dapat digunakan atau dipanggil saat membuat program lain.

Terdapat tiga komponen untuk membuat modularitas program di Python yaitu: fungsi, modul dan paket.

## 7.2 Membuat dan Memanggil *Module*

```
pi=22/7
def sigma(a,b):
    h=0
    for i in range(a,b+1):
        if i==b:
            print(i,'=',h)
        else:
            print(i,'+',end='')
            h=h+i
    print()
    return h

def pangkat(a,b):
    h=1
    for i in range(b):
        h=h*a
    return h
```

Program di atas disimpan dengan nama modulku, untuk menggunakan script membuat program maka akan seperti di bawah ini :

```
import modulku
r = int(input('Masukan jari-jari lingkaran : '))
l = modulku.pi*r*r
print('luas lingkaran = ',l)
x = int(input('Masukan nilai pertama : '))
y = int(input('Masukan nilai kedua : '))
modulku.sigma(x,y)
print(' pangkat ',y,' = ',modulku.pangkat(x,y))
```

Apabila yang diperlukan hanya beberapa komponen atau fungsi saja (misalnya hanya untuk menghitung sigma) sehingga dapat meringankan memori maka pemanggilan dari program utama menjadi sebagai berikut:

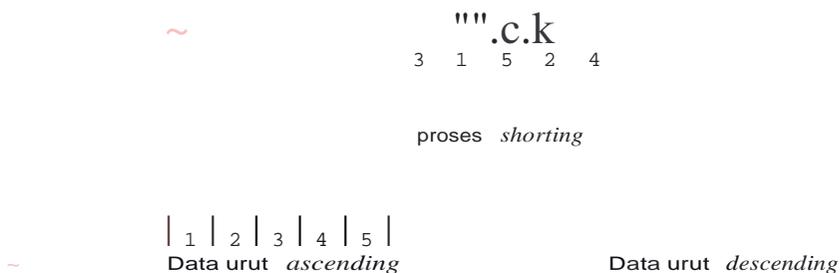
# Pengurutan Menggunakan Array

## 8.1 Pengantar Pengurutan

Dengan semakin banyak data yang disimpan maka pengelolaan dan pemeliharaan data juga semakin kompleks. Sudah banyak algoritma yang secara efisien dapat membantu dalam memproses data untuk menghasilkan informasi dengan cepat. Pengurutan atau *sorting*

~  
cara kerja algoritma dalam python. *Sorting*  
logika pengurutan dan

dan pencarian (*searching*) merupakan teknik dasar dalam mengelola data yang sangat banyak. Dalam bab 8 akan dibahas secara spesifik merupakan proses pengurutan dari data yang acak menjadi data yangurut berdasar syarat atau aturan tertentu. Secara garis besar proses pengurutan dibagi menjadi dua yaitu pengurutan dari data terkecil sampai terbesar (*ascending*) dan kebalikannya (*descending*) yaitu pengurutan dari data terbesar hingga terkecil. Pengurutan data secara *ascending* dan *descending* dapat digambarkan seperti pada gambar 8.1.



Gambar 8.1. Pengurutan Secara *Ascending* Dan *Descending*

## Pencarian Menggunakan Array

### 9.1 Pengantar Pencarian

*Searching* (pencarian) merupakan proses algoritma yang bertujuan untuk memeriksa, menemukan dan menentukan posisi suatu elemen tertentu dari sekumpulan elemen yang dapat berupa larik, pohon dan bentuk lainnya. Algoritma pencarian adalah algoritma dengan beberapa tahapan tertentu yang mencari elemen dari suatu argumen kunci yang telah diketahui sebelumnya. Setelah proses pencarian dilakukan maka akan didapatkan dua kemungkinan hasil, yaitu elemen yang dicari berhasil ditemukan atau tidak ditemukan.

Biasanya proses pencarian dilakukan pada elemen yang memiliki tipe data yang sama. Dalam bab ini akan dibahas pencarian dengan menggunakan operasi pada larik (*array*) sehingga tipe datanya sama. Hanya ada dua metode yang akan dipelajari dalam bab ini, yaitu sekuensial (*sequential search*) dan *binary search*.

### 9.2 *Sequential Search*

Dikenal juga sebagai pencarian linier, yang merupakan algoritma pencarian yang dianggap paling sederhana dan paling mudah dipahami. Proses algoritma ini menggunakan pendekatan *brute force*, yang akan menggunakan segala cara sampai elemen yang dicari ditemukan. Pada dasarnya proses metode ini adalah membandingkan satu per satu semua elemen yang terdapat dalam larik secara beruntun atau sekuensial. Proses perbandingan dimulai dari elemen pertama hingga elemen terakhir atau sudah menemukan elemen yang dicari.

## DAFTAR PUSTAKA

- Aziz, T., Asmiatun, S. (2022). Penerapan Algoritma *Bubble Sort* pada Sistem Perpustakaan Menggunakan *Framework Codeigniter*. *Information Science and Library*, (1), 24-33. <http://dx.doi.org/10.26623/jisl>.
- Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. 2009. *Introduction to Algorithms*. London : The MIT Press
- Goodrich, M.T., Tamassia, R., Goldwasser, M.H. (2013). *Data Structures and Algorithms in Python*. Hoboken : John Wiley & Sons.
- Heryanto, Y, Fauziah, Harjanti, T.W. (2022). Analisis Perbandingan Ruang dan Waktu pada Algoritma Sorting Menggunakan Bahasa Pemrograman Python. *Jurnal Kesatria*, 4(2), 342-347. <https://doi.org/10.30645/jesatria.v4i2.169>.
- Hidayat, A.R. (2018). *Filsafat Berpikir Teknik-Teknik Berpikir Logis Kontra Kesenatan Berpikir*. Pamekasan : Duta Media Publishing.
- Kautsar, I.A. (2017). *Algoritma & Pemrograman Python Series*. Sidoarjo : UMSIDA Press.
- Khoirudin. (2017). *Algoritma dan Struktur Data Dengan Python 3*. Semarang : Universitas Semarang Press.
- Purnomo, R., Putra, T.D. (2023). Theoretical Analysis of Standard Selection Sort Algorithm. *Sinkron*, 8(2), 666-673. <https://doi.org/10.33395/sinkron.v8i2.12153>.
- Nease, R.D. (2011). *Data Structures and Algorithms Using Python*. Hoboken : John Wiley & Sons Inc.
- Putri, M.P, et al. (2022). *Algoritma Dan Struktur Data*. Bandung : Widina Bhakti Persada.
- Santoso, J.T. (2021). *Struktur Data dan Algoritma (Bagian 1)*. Semarang : Yayasan Prima Agus Teknik

- Saputra, D.Y.R., Andryana, S., Sholihati, I.D. (2021). Analisis Perbandingan Algoritma Bubble Sort Dan Selection Sort Pada Sistem Pendukung Keputusan Pemilihan Tempat Kost Berbasis IOS (*Iphone Operating System*). *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, 6(2), 318-324. <https://doi.org/10.29100/jipi.v6i2.2015>.
- Schneider, J.B., Broschat, S.L., Dahmen, J. (2019). *Algorithmic Problem Solving with Python*. Washington : Swan.
- Widodo, Wibowo, K.A.S.A., Vendyansyah, Nurlaily (2021). Penerapan Sequential Search Untuk Pengelolaan Data Barang. *Jurnal Antivirus*, 15(1), 86-97. doi: <https://doi.org/10.35457/antivirus.v15i1.1385>

## BIODATA PENULIS



**Heribertus Ary Setyadi, S.T., M.Kom.**

Saat ini berstatus sebagai dosen di Universitas Bina Sarana Informatika program Studi Sistem Informasi (D3) Kampus Kota Surakarta. Merupakan alumni Institut Sains dan Teknologi AKPRIND Yogyakarta Program Studi Manajemen Informatika & Teknik Komputer juga Magister Sistem Informasi Universitas Diponegoro Semarang. Pengalaman bekerja di dunia industri dimulai sebagai EDP di BPR dan dilanjutkan di dua perusahaan tekstil. Pengalaman mengajar dimulai tahun 2000 dari LPK, SMA, Akademi, Politeknik, Sekolah Tinggi sampai Universitas di karesidenan Surakarta. Ini merupakan buku kedua di penerbit deepublish walau sebelumnya telah menulis beberapa buku elektronik yang telah banyak dijadikan referensi.



**Yusuf Sulentio, S.Kom., M.H., M.Kom.**

Alumni Teknik Informatika Institut Sains Dan Teknologi Terpadu Surabaya, Magister Hukum di Universitas Muhammadiyah Surakarta dan Magister Teknik Informatika di Universitas Amikom Yogyakarta. Saat ini sedang tahap akhir Doktoral di Universitas Bina Nusantara. Pengalaman bekerja dimulai sebagai EDP di Toko Sinar Baru dilanjutkan EDP di Pabrik Cat dan Toko Komputer. Sejak 2007 berstatus sebagai dosen di Universitas Dharma AUB Program Studi Manajemen Informatika (D3) Kota Surakarta. Juga pernah mengajar di Universitas Sahid Surakarta dan STMIK Wikrama Bogor. Ini merupakan buku kedua yang berhasil diterbitkan.