

**BUKU GRATIS UNTUK SEMUA**

# **AI & ML**

**Untuk Sains dan Teknik**

**ZOHAN SYAH FATOMI**

Copyright © 2019 Sholihun

DEPARTEMEN FISIKA FAKULTAS MIPA, UNIVERSITAS GADJAH MADA

[GITHUB.COM/LAURETHTEX/CLUSTERING](https://github.com/LAURETHTEX/CLUSTERING)

Tulisan ini digunakan untuk mencoba halaman copyright apakah bisa atau tidak,  
terima kasih.

*18 Agustus 2020*



## KATA PENGANTAR

Segala puji dan syukur penulis sampaikan ke hadirat Alloh SWT, Tuhan semesta alam yang telah melimpahkan segala kenikmatan dan kesempatan sehingga penulis dapat menyelesaikan penyusunan buku Pemrograman dan Komputasi Numerik : Menggunakan Python. Buku ini menyajikan pembahasan terkait dengan dasar pemrograman dan penerapannya pada komputasi numerik. Buku ini ditulis sebagai panduan bagi pembaca secara umum khususnya bagi pembaca dengan background sains dan teknik. Bagi pembaca yang belum mengerti terkait bahasa pemrograman bisa memulai belajar pemrograman dari awal melalui buku ini. Bagi pembaca yang mempunyai pengetahuan pemrograman, buku ini dapat digunakan untuk memperdalam kemampuan pemrograman dan juga sekaligus untuk memahami terkait komputasi numerik. Dalam buku berikut ini, penulis menggunakan Python karena selain cukup powerfull untuk komputasi numerik, Python merupakan software pemrograman mudah untuk dipelajari ditunjukkan dengan deklarasi program (code) yang tidak panjang. Selain karena motivasi kemudahan dalam pembelajaran, penulis menggunakan Python karena software pemrograman ini sedang paling banyak (ingin) dikuasai oleh pengembang pemrograman di seluruh dunia. Harapan penulis, selain pembaca mendapatkan ilmu terkait komputasi numerik, dengan pemrograman Python pembaca dapat mengembangkan ilmunya dalam bidang lain seperti Data Science atau pengembangan aplikasi baik berbentuk mobile-based, dekstop-based maupun web-based. Sebagai suatu karya, buku ini tentunya belum sempurna, sehingga masukan dari berbagai pihak sangat diperlukan. Terakhir, semoga buku ini bermanfaat bagi bagi pembaca dan berkontribusi dalam penyebaran khazanah keilmuan dan pendidikan.

Yogyakarta, 20 Maret 2020

Penulis



# DAFTAR ISI

KATA PENGANTAR .....	iii
DAFTAR GAMBAR .....	vii
DAFTAR TABEL .....	ix

<b>I</b>	<b>PEMROGRAMAN PYTHON</b>	<b>xi</b>
----------	---------------------------	-----------

<b>1</b>	<b>PENDAHULUAN .....</b>	<b>1</b>
1.1	Apa itu Program?	1
1.2	Bahasa Pemrograman	1
1.2.1	Bahasa Mesin .....	3
1.2.2	Bahasa Assembly .....	3
1.2.3	Bahasa Tingkat Tinggi .....	4
1.3	Mengapa harus Python?	5
1.4	Soal Latihan	5
<b>2</b>	<b>VISUALISASI DATA .....</b>	<b>7</b>
2.1	Apa itu Data?	7
2.2	Fungsi Data	7
2.3	Jenis Data	7
2.3.1	Data Kualitatif .....	7
2.3.2	Data Kuantitatif .....	8

---

<b>2.4</b>	<b>Data Diskrit vs Data Kontinyu</b>	<b>9</b>
2.4.1	Data Diskrit . . . . .	9
2.4.2	Data Kontinyu . . . . .	9
<b>A</b>	<b>GLOSARIUM</b> . . . . .	<b>11</b>
	<b>DAFTAR PUSTAKA</b> . . . . .	<b>15</b>

## DAFTAR GAMBAR

- 1.1 (Atas) Komunikasi 2 orang berbeda warga negara (orang A dan orang B) sebagai abstraksi komunikasi antara manusia dengan komputer. (Bawah) Komunikasi 2 orang berbeda warga negara (orang A dan orang B) kemudian ditengai oleh seorang translator (Orang C) *(Gambar ini dirancang dan digambar oleh penulis)* . . . . . 2





## DAFTAR TABEL

2.1	Data kualitatif pada sapi . . . . .	8
2.2	Data kuantitatif pada sapi . . . . .	8



**BAGIAN I**

**PEMROGRAMAN PYTHON**



# 1. PENDAHULUAN

## 1.1 Apa itu Program?

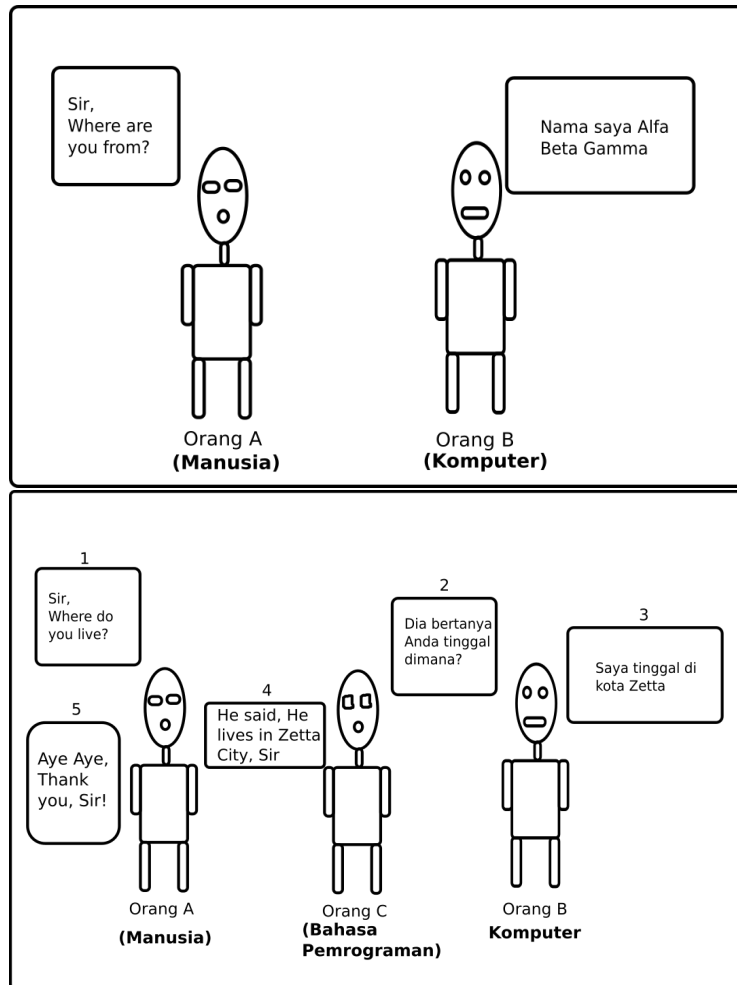
Program merupakan kumpulan perintah dari manusia kepada komputer untuk melakukan suatu tugas tertentu. Biasanya suatu program tidak berdiri tunggal melainkan tersusun dalam desain tertentu untuk membentuk suatu perangkat lunak (*software*). Menurut fungsinya perangkat lunak dibedakan menjadi 3 kategori :

1. *Perangkat lunak sistem*, perangkat lunak yang bertanggung jawab dalam manajemen komponen perangkat keras (*hardware*). Contohnya adalah sistem operasi, kernel, driver dlsb.
2. *Perangkat lunak programming*, perangkat lunak yang digunakan programmer untuk menulis program. Contohnya adalah *text-editor*, compiler, interpreter, linker, debugger (biasanya telah menjadi sebuah satu-kesatuan perangkat lunak yang disebut dengan *Integrated Development Environment (IDE)*). Contohnya Netbeans, Pycharm, Anaconda, Eclipse dlsb.
3. *Perangkat lunak aplikasi*, perangkat lunak yang digunakan untuk melakukan tugas tertentu. Contohnya MS. Office yang digunakan untuk membantu proses bisnis kantor, Google Chrome yang digunakan untuk browsing internet, dlsb.

## 1.2 Bahasa Pemrograman

Komputer tidak bisa menulis program. Komputer juga tidak mengerti bahasa manusia. Disinilah letak peranan dari bahasa pemrograman, yaitu sebagai jembatan komunikasi antara manusia dengan komputer. Untuk abstraksi yang lebih jelas terkait dengan peranan bahasa pemrograman dapat Anda amati pada gambar (1.1).

Pada gambar tersebut (gambar atas) terdapat 2 orang yang berbeda warga negara (Orang A dan orang B) yang sedang mencoba berkomunikasi. Namun, karena menggunakan bahasa yang berbeda dapat dipastikan akan terjadi miskomunikasi. Permasalahan ini dapat diatasi apabila Anda bisa mengundang orang ketiga (orang C) sebagai *translator* yang menghubungkan komunikasi antara orang A dan orang



Gambar 1.1: (Atas) Komunikasi 2 orang berbeda warga negara (orang A dan orang B) sebagai abstraksi komunikasi antara manusia dengan komputer. (Bawah) Komunikasi 2 orang berbeda warga negara (orang A dan orang B) kemudian ditengai oleh seorang translator (Orang C) (*Gambar ini dirancang dan digambar oleh penulis*)

B.

Kedua orang yang ingin berkomunikasi dapat dibayangkan sebagai manusia yang ingin memberikan instruksi pada komputer. Tentu saja, manusia tidak bisa secara langsung memberikan instruksi pada komputer. Itulah sebabnya bahasa pemrograman diciptakan yaitu untuk memberikan jembatan komunikasi antara manusia dan komputer.

Dari sini bisa didefinisikan bahwa bahasa pemrograman adalah bahasa yang digunakan untuk berkomunikasi dengan komputer. Menurut tingkatannya, bahasa pemrograman dapat dibedakan menjadi 3 kategori yaitu :

- Bahasa mesin
- Bahasa assembly
- Bahasa tingkat tinggi

### 1.2.1 Bahasa Mesin

Pada gambar (1.1) disebutkan bahwa mesin dibayangkan dapat berkomunikasi dengan sesama mesin dengan menggunakan suatu bahasa tertentu. Bahasa itulah yang dimaksud dengan bahasa mesin. Bahasa mesin (kode mesin) merupakan bahasa tingkat rendah yang secara langsung dikonstruksi dan hanya diperuntukan untuk mesin. Disebut "tingkat rendah" karena konfigurasi kodenya bisa berbeda-beda untuk setiap mesin/platform. Bahasa mesin terdiri atas 2 buah nilai saja : 0 dan 1 (sehingga membentuk sistem binary), yang mana biasanya membentuk kelompok byte (8 bit). Berikut adalah contoh dari pada struktur bahasa mesin :

```
01010101 11011011 11110010
```

Mempelajari bahasa mesin bisa diibaratkan seperti mempelajari bahasa manusia purba<sup>1</sup>. Karena kebudayaan manusia purba masih rendah tentu sistem bahasa manusia purba tidak memiliki struktur bahasa yang lazim. Sehingga bahasa tersebut sangat susah untuk diartikan dan dipelajari oleh orang manusia modern<sup>2</sup>. Hanya orang-orang tertentu yang bisa mengerti (mempelajari) sistem bahasa manusia purba<sup>3</sup>.

Mirip dengan itu, bahasa mesin sangat susah untuk dipelajari. Mengapa? Karena untuk pengaplikasiannya, Anda harus mempelajari terlebih dahulu bagaimana struktur dari konstruksi mesin yang dibangun (mengingat bahasa mesin merupakan bahasa yang tergantung oleh arsitektur mesin). Selain itu, penulisan format binary tentu tidak lazim bagi orang modern seperti kita. Dibandingkan dengan jenis bahasa pemrograman yang lain, bahasa mesin memiliki beberapa kelebihan dan kekurangan.

Kelebihan dari bahasa mesin :

- Cepat, karena bahasa mesin dieksekusi secara langsung oleh mesin
- Penggunaan memori yang efisien

Kekurangan dari bahasa mesin :

- Tidak bisa dimengerti oleh manusia (hanya terdiri dari digit 0 dan 1).
- Jika terjadi kesalahan (*error*) program akan susah untuk memecahkannya.
- Tidak ada fungsi matematika yang tersedia.
- Manipulasi lokasi memori dilakukan secara langsung, sehingga membutuhkan programmer yang bisa melakukan pelacakan untuk setiap lokasi memori.

### 1.2.2 Bahasa Assembly

Menggunakan bahasa mesin memang memiliki kelebihan dalam hal kecepatan, namun demikian menggunakan bahasa mesin tentu saja sangat tidak mudah bagi programmer karena skrip kode hanya terdiri dari 2 digit bilangan saja (0 dan 1). Maka dari itu, programmer yang ingin bekerja dalam ranah pemrograman tingkat rendah (bahasa mesin) biasanya lebih menyenangi untuk menggunakan bahasa Assembly.

Bahasa assembly pada dasarnya menggantikan bentuk digit binary (0 dan 1) ke dalam simbol alfanumerik<sup>4</sup> dengan tujuan mempermudah bentuk bahasa mesin. Contohnya adalah sebagai berikut :

---

<sup>1</sup>*Homo Erectus* misalnya

<sup>2</sup>baca "kita"

<sup>3</sup>Sejarahwan atau Arkeolog

<sup>4</sup>kombinasi antara huruf alfabet dan angka-angka



```
ADD 1,2, result
```

Program di atas bertujuan untuk melakukan penjumlahan bilangan 1 dan 2 kemudian menampilkan hasilnya.

Jika bahasa mesin bisa diibaratkan sebagai bahasa manusia zaman kuno, maka bahasa assembly bisa diibaratkan sebagai bahasa Sanskerta. Bahasa Sanskerta merupakan bahasa kuno namun lebih terbarukan dibandingkan dengan bahasa yang digunakan oleh manusia purba. Bahasa Sanskerta sudah memiliki struktur bahasa yang valid dan jelas sehingga masih "dimungkinkan" untuk dibaca dan dipelajari. Namun demikian, untuk mempelajarinya Anda harus menguasai sistem penulisan bahasa Sanskerta terlebih dahulu. Hal ini menjadi faktor ketidakpraktisan mengingat budaya kita sudah tidak menggunakan bahasa Sanskerta lagi.

Mirip dengan itu, mengembangkan suatu perangkat lunak dengan menggunakan bahasa assembly menjadi barang yang bersifat tidak mungkin untuk dilakukan. Dibandingkan dengan jenis bahasa pemrograman yang lain, bahasa Assembly memiliki beberapa kelebihan dan kekurangan.

Kelebihan dari bahasa assembly :

- Lebih mudah digunakan dibandingkan bahasa mesin.
- Masih tergolong bahasa tingkat rendah, sehingga kurang cocok digunakan seorang pemula.

Kekurangan dari bahasa Assembly :

- Tidak ada bentuk simbolik pada lokasi memori.
- Masih susah untuk digunakan (khususnya orang awam).
- Bentuk bahasa Assembly sangat bergantung pada arsitektur dari mesin (platform) yang digunakan.

### 1.2.3 Bahasa Tingkat Tinggi

Berbeda dengan kedua jenis bahasa pemrograman di atas yang termasuk golongan bahasa pemrograman tingkat rendah, sehingga untuk penggunaannya memerlukan keahlian khusus<sup>5</sup>, untuk kali ini akan dibahas bahasa pemrograman tingkat tinggi. Dibandingkan dengan bahasa tingkat rendah, dengan menggunakan bahasa pemrograman tingkat tinggi Anda lebih dimudahkan. Bahasa tingkat tinggi bisa diibaratkan sebagai bahasa Inggris yang biasa Anda pelajari di sekolah. Bahasa Inggris sudah terkonstruksi secara apik sehingga sangat mudah untuk dipelajari<sup>6</sup>.

Mirip dengan itu, bahasa tingkat tinggi dikonstruksi untuk mempermudah manusia. Penggunaan sistem pembahasaannya lebih mendekati bahasa manusia dibandingkan dengan bahasa mesin. Bahkan, dengan menggunakan bahasa tingkat tinggi, seseorang yang tidak mengerti tentang arsitektur mesin atau sistem binary bisa melakukan mengkonstruksi program (koding). Bahasa tingkat tinggi juga merupakan bahasa pemrograman yang tidak bergantung pada platform tertentu sehingga sebagai konsekuensinya program yang dihasilkan dapat dieksekusi pada mesin yang berbeda. Namun demikian, jenis bahasa pemrograman yang lebih berorientasi kepada bahasa manusia ini membutuhkan suatu penerjemah, agar instruksi yang diberikan oleh manusia bisa dieksekusi oleh komputer. Penerjemah tersebut dapat dilakukan oleh

<sup>5</sup>semisal terkait dengan ilmu konstruksi mesin

<sup>6</sup>mayoritas manusia modern menguasai bahasa inggris

compiler atau interpreter (tergantung dari jenis bahasa pemrogramannya). Dibandingkan dengan jenis bahasa pemrograman yang lain, bahasa tingkat tinggi memiliki beberapa kelebihan dan kekurangan.

Kekurangan dari bahasa tingkat tinggi :

- Harus tersedia penerjemah (compiler atau interpreter).
- Kecepatan eksekusi lebih rendah jika dibandingkan dengan bahasa mesin dan bahasa assembly.

Kelebihan dari bahasa tingkat tinggi adalah :

- Mudah digunakan (biasanya menggunakan Inggris).
- Dapat digunakan di berbagai mesin.
- Cocok untuk pemula yang memulai belajar bahasa pemrograman.

Contoh dari bahasa tingkat tinggi antara lain adalah C, C++, C, Java, Fortran, Ruby, Perl, Python dlsb. Bahasa pemrograman tersebut dikonstruksi berdasarkan filosofi dan tujuan komputasi tertentu.

### 1.3 Mengapa harus Python?

Dewasa ini, Python menjadi bahasa pemrograman yang paling ingin dikuasai oleh mayoritas semua orang (terutama dari kalangan programmer). Antusiasme tersebut didukung oleh pesatnya perkembangan teknologi Data Sains<sup>7</sup>. Pada bidang Data Sains, dibandingkan dengan bahasa pemrograman lainnya, Python memiliki keunggulan tersendiri dalam hal analitik karena tersedianya library NumPy, Pandas dan Matplotlib (sebagai visualisasi data).

Berbicara terkait library, kini telah tersedia berjuta-juta library di Python. Hal tersebut tentu saja akan mempermudah pekerjaan Anda karena Anda tidak diwajibkan lagi untuk membuat program dari awal. Untuk menuliskan program tertentu Anda hanya perlu memanggil fungsi dari suatu library kemudian menerapkan fungsi tersebut ke dalam program yang sedang Anda kembangkan.

Selain itu, Python sangat mudah untuk dipelajari. Jika Anda belum pernah menginjakkan kaki di dunia IT (pemrograman khususnya) maka Python merupakan langkah awal yang tepat. Python juga sangat cocok untuk digunakan sebagai media pembelajaran bahasa pemrograman di dunia pendidikan mengingat Python termasuk bahasa pemrograman yang bersifat *open source*. Meskipun Python merupakan bahasa pemrograman yang mudah dipelajari, Python tetap bisa untuk dijadikan basis untuk membangun sistem yang kompleks.

Satu hal lagi yang membuat Python disenangi oleh kalangan penggiat IT. Python memiliki komunitas yang luas. Jika saja Anda mengalami kesulitan belajar, selalu mengalami kesalahan (error) dalam pengembangan program Anda, Anda bisa mengunjungi komunitas Python di forum online yang tersebar luas di internet. Contohnya Anda bisa mengunjungi website : <https://www.python.org/community/> dan <https://www.stackoverflow.com>.

### 1.4 Soal Latihan

1. Buatlah Essay dengan tema "mengapa belajar pemrograman itu penting"!

---

<sup>7</sup>Data Sains adalah ilmu yang mempelajari perilaku data, baik itu data kuantitatif ataupun numerik.

2. Di dalam bahasa pemrograman tingkat tinggi terdapat beberapa paradigma (sudut pandang) dalam penyajian pemecahan masalah. Diantaranya adalah pemrograman *Imperative*, *Logical*, *Functional* dan *Object-Oriented*. Jelaskan perbedaan dari keempat paradigma pemrograman berikut!
3. Di dalam bahasa tingkat tinggi terdapat 2 jenis penerjemah, Compiler dan Interpreter. Jelaskan bagaimana 2 penerjemah itu bekerja kemudian bandingkan juga kekurangan dan kelebihanannya!
4. Sebutkan kelebihan dan kekurangan bahasa pemrograman Python dibandingkan bahasa pemrograman yang lainnya!
5. Jelaskan bagaimana pembuatan alur program sehingga program tersebut dapat digunakan untuk memecahkan suatu masalah?

## 2. VISUALISASI DATA

### 2.1 Apa itu Data?

Data adalah representasi nyata dari suatu objek. Data dapat berbentuk teks, numerik, gambar (warna), suara, dlsb.

### 2.2 Fungsi Data

1. Sebagai acuan untuk melakukan pengkajian.
2. Sumber kajian dan evaluasi dari suatu aktivitas.
3. Acuan dalam menentukan kebijakan/keputusan untuk kedepannya.

### 2.3 Jenis Data

Pada dasarnya data dapat dibedakan menjadi 2 :

1. Data Kualitatif
2. Data Kuantitatif

#### 2.3.1 Data Kualitatif

Data kualitatif merupakan representasi obyek yang berbentuk pernyataan kualitas. Biasanya data kualitatif diformasi dari rekam pendapat yang sifatnya **subyektif**. Artinya, bisa saja antara 2 orang yang berbeda dapat memberikan pemberian data yang berbeda dalam 1 obyek yang sama.

Semisal terdefinisi seekor sapi. Sapi tersebut memiliki data kualitatif sebagai berikut.

Adapun metode pengumpulan data adalah sebagai berikut.

1. Pertanyaan (wawancara)
2. Diskusi
3. Observasi

Tabel 2.1: Data kualitatif pada sapi

Data	Deskripsi
Keadaan	Sehat
Warna	Hitam-putih
Tekstur	Lembut

Analisa data kualitatif. Tidak ada aturan yang pasti terkait dengan metode analisa pada data kualitatif. Mengapa? Hal tersebut dikarenakan data dianggap sesuatu yang telah benar (subyektif). Adapun metode yang sering kali digunakan adalah sebagai berikut :

1. Pendekatan Deduktif Struktur penelitian ditentukan oleh peneliti. Cepat dan Mudah.
2. Pendekatan Induktif Pendekatan ini tidak secara langsung ditentukan oleh peneliti. Lama dan susah.

### 2.3.2 Data Kuantitatif

Data kuantitatif adalah data yang bisa direpresentasikan dalam bentuk angka-angka (nilai numerik). Data kuantitatif dapat dihasilkan dari suatu eksperimen (pengukuran maupun observasi). Data kuantitatif relatif lebih bersifat obyektif karena didapatkan dari hasil pengukuran bukan dari pendapat seseorang.

Semisal terdefinisi seekor sapi. Sapi tersebut memiliki data kualitatif sebagai berikut.

Tabel 2.2: Data kuantitatif pada sapi

Data	Deskripsi
Berat	300 kg
Panjang	3 meter
Harga	16.000.000

Adapun metode pengumpulan data adalah sebagai berikut.

1. Pertanyaan (wawancara)
2. Sampling probabilitas
3. Survei atau kuesioner
4. Pengamatan

Analisa data kuantitatif :

1. Analisa Maxdiff
2. Tabulasi Silang
3. Analisa SWOT

Langkah analisa data kuantitatif:

1. Hubungkan skala pengukuran dengan variabel
2. Hubungkan statistik deskriptif dengan data
3. Tentukan skala pengukuran
4. Pemilihan tabel

## 2.4 Data Diskrit vs Data Kontinyu

Dari perilaku datanya data dapat dibedakan menjadi 2 :

1. Data diskrit
2. Data kontinyu

### 2.4.1 Data Diskrit

Data diskrit merupakan data yang terkumpul hasil dari pencacahan. Contoh data jumlah kehadiran orang dalam suatu acara atau jumlah peliharaan ayam Anda di kandang. Pada data diskrit Anda tidak mungkin mendapatkan nilai desimal, Anda tidak mungkin menghitung kehadiran pada suatu kelompok dengan jumlah 11.5 orang.

Adapun karakteristik dari data diskrit adalah sebagai berikut :

1. Dapat dihitung
2. Nilai tidak mungkin menjadi desimal (pecahan)
3. Tidak dapat diukur (menggunakan alat)
4. Biasanya direpresentasikan dalam bentuk grafik batang

### 2.4.2 Data Kontinyu

Data kontinyu merupakan data yang terkumpul dari hasil perhitungan. Semisal Anda ingin mengetahui panjang dari suatu meja. Anda akan mengukurnya menggunakan mistar bukan mencacahnya bukan? Hasil yang didapatkan pun bisa berbentuk desimal (pecahan), mungkin panjang dari meja Anda dapat bernilai 100.5 cm,

Adapun karakteristik dari data kontinyu adalah sebagai berikut :

1. Dapat diukur (menggunakan alat)
2. Nilai mungkin menjadi desimal (pecahan)
3. Tidak dapat dicacah
4. Biasanya direpresentasikan dalam bentuk grafik histogram

## 2.5 Data Numerik vs Data Kategori

Data juga dapat dibedakan berdasarkan nilai variabelnya, yaitu:

1. Data Numerik
2. Data Kategori

### 2.5.1 Data Numerik

Data yang identik dengan nilai numerik. Data numerik dapat dibedakan menjadi data diskrit dan kontinyu.

### 2.5.2 Data Kategori

Data yang bisa dikategorikan berdasarkan karakteristik masing-masing individu data. Contohnya status perkawinan data di KTP, agama, kota asal, hobi, jenis kelamin dlsb.

**Data Nominal**

**Data Ordinal**



## A. GLOSARIUM

**Algoritma** Tata cara untuk menyelesaikan masalah secara sistematis.

**Assembly** Bahasa pemrograman tingkat rendah (satu tingkat di atas bahasa mesin) yang sangat bergantung pada arsitektur mesin tertentu.

**Array** Variabel yang dapat menampung nilai lebih dari satu namun harus memiliki tipe data yang sama.

**Compiler** Suatu program khusus yang digunakan untuk mengeksekusi instruksi pada suatu skrip program dengan melakukan kompilasi (penerjemahan pada bahasa tertentu, contohnya bahasa mesin).

**Dict** Salah satu struktur data pada Python yang merupakan variabel multivalue yang komponen nilainya berupa kunci dan nilai (seperti kamus).

**Fungsi** Kumpulan instruksi program yang memiliki sebuah tujuan tertentu.

**IDE** Kepanjangan dari Integrated Development Environment yaitu gabungan (kesatuan) dari pada interpreter/compiler, library, text-editor, debugging, dsb yang dikonstruksi untuk memudahkan pengembangan suatu program.

**Interpreter** Suatu program khusus yang digunakan untuk mengeksekusi instruksi pada suatu skrip program tanpa melakukan kompilasi.

**Indentasi** Blok pada program Python yang dicirikan dengan tulisan yang lebih menjorok (hasil eksekusi tab).

**Kelas** Cetak biru dari objek.

**Konstanta** Wadah dari suatu nilai namun nilainya tidak berubah.



**Konstruktor** Method khusus yang digunakan untuk memberikan keadaan awal dari suatu objek sejak objek tersebut diinstansiasi (diwujudkan).

**Konvergen** Keadaan dimana hasil iteratif dari suatu perhitungan menuju pada suatu nilai tunggal.

**List** Salah satu struktur data pada Python yang merupakan variabel multinilai yang komponen nilainya bisa berubah.

**Numpy** Salah satu library pada Python yang digunakan untuk komputasi ilmiah berbasis array.

**Matplotlib** Salah satu library pada Python yang digunakan untuk visualisasi data.

**Matriks** Sebuah entitas matematika dimana datanya diatur sedemikian rupa menjadi deret dan kolom.

**Objek** Representasi atau pemodelan "sesuatu" yang eksis di dunia nyata untuk mewakili suatu atribut (data) dan perilaku (method) dari suatu sistem tertentu.

**Operator** Simbolisme dari suatu perintah manusia kepada komputer (melalui interpreter atau compiler) untuk melakukan suatu operasi khusus seperti operasi aritmatika, operasi logika, operasi relasi, dlsb.

**Jupyter Notebook** Salah satu IDE untuk bahasa Python berbasis web.

**Polimorfisme** Fitur dari paradigma objek yang mengizinkan Anda untuk mengimplementasikan variabel dan method yang sama pada anak kelas dengan cara yang berbeda.

**Program** Kumpulan instruksi untuk melaksanakan suatu tugas tertentu.

**Programer** Seseorang yang memiliki kapabilitas untuk mengkonstruksi program untuk menyelesaikan suatu masalah tertentu.

**Pycharm** Salah satu IDE untuk bahasa Python berbasis dekstop.

**Python** Salah satu jenis pemrograman tingkat tinggi yang menggunakan interpreter dengan fitur dynamic-binding dan dynamic-typing.

**Tipe data** Klasifikasi data berdasarkan perilakunya (ukuran data, pengoperasiannya, dlsb)

**Rekursif** Jenis fungsi yang digunakan untuk melaksanakan prinsip pengulangan dengan jalan memanggil dirinya sendiri.

**RPEL** Kepanjangan dari Read Evaluate Print Loop yaitu interaktif interpreter (biasanya berupa tampilan Command Line Interface).

**Sistem Linear** Sistem persamaan yang suku variabel independennya terbatas memiliki derajat pangkat satu.

**Sistem Nonlinear** Sistem persamaan yang suku variabel independennya memiliki derajat pangkat lebih dari satu.

**Set** Salah satu struktur data pada Python yang merupakan variabel multivali yang komponen nilainya bersifat unik.

**Sistem Operasi** Program (perangkat lunak) yang bertanggung jawab melakukan manajemen kinerja antara perangkat keras dengan perangkat lunak lainnya.

**Tuple** Salah satu struktur data pada Python yang merupakan variabel multivali yang komponen nilainya tidak bisa berubah.

**Variabel** Wadah dari suatu nilai yang berada di dalam memori RAM.



## DAFTAR PUSTAKA

- Boas, M.L. 2005. *Mathematical Methods in The Physical Sciences*. Canada: John Wiley & Sons, Inc.
- Burden, R.L. dan Faires, J.D. 2010. *Numerical Analysis, Ninth Edition*. Canada: Cengage Learning.
- Cooper, J. 2001. *A MatLab Companion for Multivariable Calculus*. San Diego: Harcour Academic Press.
- Esfandiari, R.S. 2017. *Numerical Methods for Engineers and Scientists Using MATLAB, Second Edition*. Florida: CRC Press.
- Goodrich, M.T., Tamassia, R. dan Goldwasser, M.H. 2013. *Data Structures and Algorithms in Python*. Florida: John Wiley & Sons, Inc.
- Gowrishankar, S. dan Veena, A. 2019. *Introduction to Python Programming*. Florida: CRC Press.
- Kiusalaas, J. 2005. *Numerical Methods in Engineering with Python*. New York: Cambridge University Press.
- Landau, R.H., Paez, M.J. dan Bordeianu, C.C. 2015. *Computational Physics: Problem Solving with Python*. Weinheim: Wiley-VCH.
- Mueller, J.P. 2018. *Beginning Programming with Python for Dummies, Second Edition*. Canada: John Wiley & Sons, Inc.
- Nagar, S. 2018. *Introduction to Python for Engineers and Scientists: Open Source Solutions for Numerical Computation*. New York: Apress.
- Nelli, P. 2018. *Python Data Analytics: with Pandas, NumPy and Matplotlib, Second Edition*. New York: Apress.

- Pine, D.J. 2019. *Introduction to Python for Science and Engineering*. Boca Raton: CRC Press.
- Riley, K. F., Hobson, M.P. dan Bence, J.C. 2006. *Mathematical Methods for Physics and Engineering*. New York: Cambridge University Press.
- Solomon, J. 2015. *Numerical Algorithms: Methods for Computer Vision, Machine Learning and Graphics*. Florida: CRC Press.
- <https://www.jetbrains.com/pycharm/>. Diakses tanggal 24 Desember 2019, pukul 06.06 WIB.
- <https://www.python.org/community/>. Diakses tanggal 24 Desember 2019, pukul 11.06 WIB.
- <https://www.stackoverflow.com>. Diakses tanggal 25 Desember 2019, pukul 14.06 WIB.