

**Nama : satria nur saputro**

**Kelas : Kangae**

**QUIZ**

Durasi: 30 menit (open-book)

Durasi: 15 menit (closed-book)

**SESI PAGI**

Python OOP

1. Apa itu OOP dan cobalah buat 2 implementasi konsep OOP menggunakan python!
2. Sebutkan dan jelaskan 2 jenis modules yang ada di Python!
3. Hitunglah berapa hasil dari  $\sqrt{175} + 4\sqrt{7} + (1/3) + \sqrt[3]{\log 81}$  menggunakan module **math**!
4. Apa itu custom exception handling dan berikan 1 contoh implementasinya!

Artificial Intelligence

1. Apa perbedaan dari structured dan unstructured data dan sebutkan 3 sumber data!
2. Jelaskan perbedaan AI, Machine Learning dan Deep Learning!
3. Apa perbedaan Machine Learning dengan Tradisional software?
4. Sebutkan 3 contoh aplikasi Machine Learning di sektor smart city dan agriculture!

Maths for AI

1. Mengapa ilmu matematika penting ketika mempelajari AI?
2. Jelaskan perbedaan vektor dan skalar!
3. Apa perbedaan dari Jarak Euclidean dan Manhattan?
4. Apa kegunaan matriks dalam ilmu matematika ketika menerapkan AI?

Machine Learning

1. Jelaskan perbedaan supervised dan unsupervised learning dan sebutkan 2 contohnya!
2. Sebutkan setiap tahap yang ada di Machine Learning life cycle dan jelaskan!
3. Apa perbedaan regression dengan clustering?
4. Apa itu algoritma Artificial Neural Network dan untuk apa fungsinya?

**Jawaban**

**Python OOP**

1. OOP adalah merupakan suatu metode pemrograman yang berorientasi kepada objek, bertujuan untuk mempermudah pengembangan sebuah program, bisa disebut sebagai blueprint, dengan prinsip-prinsip OOP seperti Encapsulation, Abstraction, Inheritance, Polymorphism.  
2 implementasinya :

```

#membuat suatu class OOP
class Orang:
    #membuat atribut dengan nilai default
    def __init__(self, nama='satria', umur=20, hobi='sepak bola', negara='Indonesia', kota='pekalongan'):
        self.nama = nama
        self.umur = umur
        self.hobi = hobi
        self.negara = negara
        self.kota= kota
    #sebuah method OOP
    def info_data(self):
        return f'{self.nama} umur {self.umur} tahun. hobi saya {self.hobi} dan tinggal di {self.kota}, {self.negara}.'

Orang1 = Orang() #memanggil object OOP
print(Orang1.info_data())#menampilkan method OOP

#membuat Child Class
class Menyapa(Orang):
    def __init__(self):
        super().__init__()#menerapkan semua atribut dari parents
    #method
    def halo(self):
        print("Halo guys")

Orang2 = Menyapa()#memanggil object class menyapa
print(Orang2.halo())# menampilkan metode chill class


```

satya umur 20 tahun. hobi saya sepak bola dan tinggal di pekalongan, Indonesia.  
Halo guys  
None

2. -Modules built-in functions python merupakan sebuah module yang telah dibuat oleh pengembang python sehingga kita bisa langsung pakai modul tersebut, seperti math,print,sys,dan fibo.  
-modules user-defined functions python merupakan sebuah module yang dibuat oleh pengembangan sendiri diluar dari built-in python dengan tujuan tertentu dan lebih spesifik sesuai dengan kebutuhan program

Dari modules tersebut dapat digunakan diluar dari program utama, sehingga dapat digabungkan per scripts py sehingga dapat lebih simple, efisien, dan mudah dikembangkan programnya,

3. Hasil codingan:

```

import math # import math

print(math.sqrt(175)+ (4*math.sqrt(7)) + (1/3) + math.log(81,3)) #menampilkan hasil dari perhitungan|
```

28.145095132914648

4. Custom exception handling merupakan adalah event yang terjadi ketika program menemui suatu error sehingga akan melemparkan exception ke buatan handling sendiri.  
Contoh:

```
# membuat custom error
class Kesalahan(Exception):
    #melempar nilai ke kelas Nilai kosong
    pass #skip
class NilaiKosong(Error):
    #jika nilai 0
    pass #skip
try: #lakukan terlebih dahulu
    nilai = int(input("masukan nilai: ")) #input nilai
    if nilai ==0: # penyataan jika nilai 0
        raise NilaiKosong #lempar error
except NilaiKosong: #saat error maka dilakukan expect
    print("inputan tidak boleh 0, coba lagi")
    print()

→ masukan nilai: 0
inputan tidak boleh 0, coba lagi
```

## Artificial Intelligence

1. -Data structured adalah data dapat diproses, disimpan, dan diambil dalam format tetap karena memiliki aturan skema yang tetap sebelum dibuat, biasa disimpan dalam bentuk tabel, baris dan kolom, dan biasanya digunakan oleh RDMS.  
Contoh : data tabular , gaji pegawai , nama karyawan ,excel,spreadsheet.  
  
-Data unstructured merupakan data yang tidak memiliki bentuk atau struktur khusus.  
Contohnya : data yang berformat foto/gambar, video, dan audio.
2. -AI adalah program komputer yang memiliki kemampuan untuk memperoleh dan menerapkan suatu pengetahuan (knowledge) dan skill layaknya meniru cara berpikir manusia.  
-Machine learning adalah bagian dari AI yang pengembangan algoritma pemrograman dengan kemampuan belajar tanpa melakukan eksplisit terlebih dahulu,tetapi perlu adanya campur tangan manusia untuk melakukan fitur ekstrasi yang diharapkan mendapatkan insight sehingga dapat membuat keputusan berdasarkan data.  
-Deep learning adalah bagian dari machine learning yang memiliki kemampuan belajar mengadaptasi cara kerja otak manusia melalui jaringan neuron untuk memproses data yang begitu kompleks tanpa melakukan fitur ekstrasi.
3. Machine learning adalah bagian dari AI yang pengembangan algoritma pemrograman dengan kemampuan belajar tanpa melakukan eksplisit terlebih dahulu, dengan inputan Data, dan jawaban kemudian outputnya akan berupa aturan(model). Sedangkan

tradisional software adalah sebuah program yang memiliki inputan Data dan aturan dengan outputnya adalah jawaban, sehingga harus melakukan aturan pengkodean secara manual.

4. Pengaplikasian smart city:
  - a. Unlock kunci rumah memakai computer vision
  - b. Deteksi pelanggar lalu lintas menggunakan computer vision
  - c. Google nest menggunakan natural language processing

Pengaplikasian bidang agriculture:

- a. Prediksi cuaca menggunakan machine learning
- b. Pendekripsi penyakit tanaman menggunakan computer vision
- c. Klasifikasi kualitas tanaman menggunakan computer vision

## **Maths for AI**

1. AI pada dasarnya dibuat oleh matematika dibagian belakangnya dengan ilmu yang menerapkan empat bidang matematika yang berperan dalam AI, yakni statistika, kalkulus dan optimisasi, aljabar linier, dan probabilitas, sehingga ilmu matematika merupakan hal penting yang wajib diketahui untuk membuat suatu AI yang dapat menyelesaikan pemecahan masalah.
2. Vektor adalah besaran yang merepresentasikan dengan nilai dan arah. Sedangkan skalar adalah besaran yang merepresentasikan nilai saja.
3. Jarak Euclidean merupakan metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 (dua) buah titik dalam Euclidean space (meliputi bidang euclidean dua dimensi, tiga dimensi, atau bahkan lebih) biasanya digunakan untuk masalah yang memiliki data yang tidak besar. Sedangkan Jarak Manhattan digunakan untuk menghitung perbedaan absolut (mutlak) antara koordinat sepasang objek, biasanya digunakan untuk data yang besar
4. Kegunaan matriks dalam hal AI bisa dijumpai pada penerapan computer vision, dengan matriks seluruh nilai pixel yang ada pada gambar akan dijadikan matriks yang kompleks kemudian akan bisa dilakukan pembelajaran melalui deep learning dengan arsitektur yang begitu kompleks sehingga AI dapat mengenali suatu gambar melalui model yang dibuat.

## **Machine Learning**

1. -Supervised Learning adalah Machine Learning model yang mempelajari data dengan label atau target dimana evaluasi model tersebut akan berdasarkan target.  
Contohnya : klasifikasi diabetes, klasifikasi bunga ,klasifikasi kelulusan mahasiswa, regresi prediksi saham dan score.
- Unsupervised Learning adalah Machine Learning Model yang mempelajari pola data tanpa adanya target data.  
Contohnya : clustering suatu kelompok , dimensionality reduce terhadap text mining, visualisasi big data.

2. a. Gathering data

Merupakan sebuah untuk mengidentifikasi semua data yang diperlukan untuk menyelesaikan masalah hal ini juga bisa disebut dengan Scope dengan melakukan pernyataan apa , dimana , siapa, mengapa masalah harus dipecahkan.

b. Data preparation

mengelakukn penyiapan suatu data yang dibutuhkan serta melakukan data exploration seperti kualitas data, format, korelasi,dan outlier.

d. Data wrangling

Yaitu melakukan preprocessing data untuk menghilangkan data data yang kosong duplikat , tidak valid, dan berisik sehingga saat model belajar akan memiliki hasil model yang efesien dan akurat.

e. Data analysis

Merupakan tahapan untuk menyiapkan dan memilih serta menganalisis suatu model terhadap permasalahan yang akan dipecahkan, seperti memilih model arsitektur yang cocok untuk klasifikasi , regresi atau clustering. Kemudian nantinya model yang dibuat juga harus dilakukan tahap evaluasi, dengan harapan terhindar dari underfitting dan overfitting.

f. Train model

Tahapan untuk melatih algoritma yang telah ditentukan dengan data latih yang telah disiapkan sehingga diharapkan mendapatkan model yang akurat.

g. Test model

Tahapan untuk mencoba model terhadap data-data baru yang belum pernah diuji apakah model dapat membuat prediksi dengan tepat atau tidak.

h. Deployment

Tahap terakhir dari life cycle AI setelah semua proses telah dilalui sehingga model dapat diterapkan untuk real-world system untuk pemecahan masalah.

3. Regression termasuk supervised learning adalah metode mencari pola dan memberikan nilai numerik padanya, dengan melihat hubungan antara satu atau lebih variabel bebas dan variabel terikat dapat dimodelkan dengan menggunakan analisis regresi, dengan bertujuan untuk menemukan suatu fungsi yang memodelkan data dengan meminimalkan error atau selisih antara nilai prediksi dengan nilai sebenarnya.

Contohnya : Untuk memprediksi nilai masa depan, volume penjualan , harga saham.

Sedangkan Clustering termasuk bagian unsupervised learning umumnya banyak digunakan untuk segmentasi, proses segmentasi biasanya tidak dapat dikerjakan dengan mudah untuk data dengan jumlah yang sangat besar, oleh karena itu dibutuhkan pendekatan analitis untuk membagi data tersebut menjadi bagian-bagian yang lebih kecil berdasarkan kemiripan karakteristiknya

4. Artificial Neural Network (ANN) atau jaringan syaraf tiruan adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan perilaku jaringan syaraf manusia. Fungsi dari algoritma ini yaitu menyelesaikan data yang begitu kompleks dengan cara meniru cara kerja otak manusia dalam melakukan pelatihan. Termasuk juga dalam bagian deep learning.

**Nama : Satria Nur Saputro**

**Kelas: Kangae**

### Basic Statistics

1. Apa beda statistik dan statistika?
2. Apa yang dimaksud dengan data semi terstruktur dan sebutkan 3 contohnya!
3. Jelaskan yang dimaksud populasi, sensus, sampel dan survey!
4. Sebutkan dan jelaskan pembagian dari statistik

### Deep Learning

1. Mengapa Deep Learning menjadi sangat populer hari ini?
2. Apakah harus selalu menggunakan Deep Learning? Jelaskan!
3. Apa yang dimaksud perceptron?
4. Jelaskan apa itu Activation Function dan 3 contohnya!

### AI Project Cycle

1. Sebutkan semua tahapan yang ada di project cycle dari AI!
2. Apa yang dimaksud dari problem scoping?
3. Bagaimana kita tahu data apa yang dibutuhkan?
4. Apa yang dimaksud dengan model dan bagaimana mengevaluasinya?

### Jawaban

#### Basic statistics

1. statistik adalah sebuah kumpulan data, angka atau informasi. Sedangkan statistika merupakan ilmu yang mempelajari bagaimana data, angka tersebut dikumpulkan diolah, dan dianalisis untuk menghasilkan sebuah informasi yang bisa digunakan dalam pengeambilan keputusan.
2. Data semi-terstruktur merupakan format data terstruktur dan tidak terstruktur. Walaupun belum diklasifikasi oleh repository tertentu (database), namun mengandung informasi yang penting.  
Contohnya adalah data dalam bentuk file csv, file xml, dan file json.

3. -Populasi adalah jumlah keseluruhan dari satuan-satuan atau individu-individu yang karakteristiknya dari data yang akan diteliti.  
-Sensus adalah cara terstruktur untuk mendapatkan informasi deskriptif tentang jumlah dari sebuah populasi.  
-Sampel adalah sebagian dari populasi tersebut yang haruslah dapat mewakili karakteristik dari keseluruhan populasi.  
-survei adalah metode pengumpulan data primer dengan memberikan pertanyaan-pertanyaan kepada responden individu atau kelompok untuk mengumpulkan informasi.
4. Berdasarkan pembagiannya:
  - A. Pembagian Statistik berdasarkan bentuk parameternya
    1. Statistik parametrik adalah parameter dari populasinya itu mengikuti suatu distribusi tertentu, seperti halnya distribusi normal serta mempunyai varians yang homogen
    2. Statistik nonparametrik merupakan parameter dari populasinya itu tidak mengikuti suatu distribusi tertentu. mempunyai distribusi yang bebas dari adanya persyaratan serta variansnya tidak perlu homogen.
  - B. Pembagian Statistik berdasarkan ruang lingkup penggunanya
    1. Statistik Sosial merupakan statistik yang diterapkan didalam ilmu sosial
    2. Statistik ekonomi merupakan statistik yang diterapkan dalam ilmu ekonomi.
    3. Statistik Pendidikan merupakan statistik yang diterapkan didalam ilmu dalam bidang pendidikan.
    4. Statistik kesehatan merupakan statistik yang diterapkan didalam bidang kesehatan.
    5. Statistik perusahaan merupakan statistic yang diterapkan didalam bidang perusahaan.
    6. Statistik pertanian merupakan statistik yang diterapkan didalam bidang pertanian.
  - C. Pembagian statistik berdasarkan cara pengolahan datanya.
    1. Statistika Deskriptif adalah statistika yang berkenaan dengan metode atau cara mendeskripsikan, menggambarkan, menjabarkan, atau menguraikan data sehingga mudah dipahami mengacu pada bagaimana menata atau mengorganisasi data, menyajikan, dan menganalisis data. Menata, menyajikan, dan menganalisis data dapat dilakukan dengan menentukan nilai rata-rata hitung, median, modus, standar deviasi, dan persen/proporsi.
    2. Statistika Inferensial adalah statistika yang berkenaan dengan cara penarikan kesimpulan berdasarkan data yang diperoleh dari sampel untuk menggambarkan karakteristik atau ciri dari suatu populasi biasanya dilakukan pengujian hipotesis dan pendugaan mengenai karakteristik atau ciri dari suatu populasi, seperti mean dan standar deviasi

## Deep Learning

1. Pertumbuhan era data dan AI telah maju sangat pesat hal tersebut mendorong untuk banyak pengembang membuat algoritma untuk mendapatkan insight dari sebuah data yang

kemudian digunakan untuk melakukan problem solving baik untuk komersial maupun non-komersial, pada dasar diatas mengapa deep learning sangat popular karena deep learning merupakan algoritma yang sangat efektif, efisien, dan powerful untuk melatih data yang besar dan kompleks serta memiliki tingkat akurasi yang jauh lebih baik dibanding algoritma machine learning, hal tersebut juga dapat digunakan dalam banyak ragam permasalahan di perusahaan hal ini lah mengapa deep learning menjadi popular pada era saat ini.

2. Tidak, kebutuhan setiap permasalahan, jumlah data, bentuk data, dan juga sumber daya resource merupakan faktor yang mempengaruhi apakah pemasalahan harus memakai deep learning atau tidak, hal tersebut menghindari dari sebuah nilai cost, dan ketidak mampuan pridiksi jika data terlalu sedikit, karena deep learning merupakan jaringan yang kompleks dan membutuhkan banyak inputan latihan.
3. Perceptron adalah Jaringan Syaraf Tiruan sebuah metode yang mampu melakukan proses perhitungan dengan mengenali variabel-variabel dalam pencocokan pola dan pada akhirnya hasil keluaran dari Jaringan dapat digunakan sebagai bahan pertimbangan pengambilan keputusan.
4. activation function adalah sebuah fungsi di neural network yang befungsi untuk menentukan apakah neuron tersebut harus “aktif” atau tidak berdasarkan dari weighted sum dari input dengan kata lain membantu jaringan menggunakan informasi penting dan menekan titik data yang tidak relevan.

Contoh : Relu , sigmoid, LeakyRelu,dan Tan

## AI Project Cycle

1. Life cycle AI:
  - a. Gathering data

Merupakan sebuah untuk mengidentifikasi semua data yang diperlukan untuk menyelesaikan masalah hal ini juga bisa disebut dengan Scope dengan melakukan pernyataan apa , dimana , siapa, mengapa masalah harus dipecahkan.
  - b. Data preparation

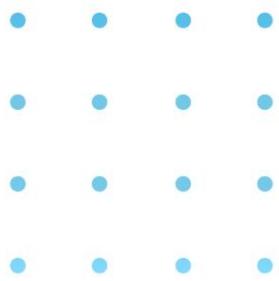
melakukan penyiapan suatu data yang dibutuhkan serta melakukan data exploration seperti kualitas data, format, korelasi,dan outlier.
  - a. Data wrangling

Yaitu melakukan preprocessing data untuk menghilangkan data data yang kosong duplikat , tidak valid, dan berisik sehingga saat model belajar akan memiliki hasil model yang efisien dan akurat.
  - b. Data analysis

Merupakan tahapan untuk menyiapkan dan memilih serta menganalisis suatu model terhadap permasalahan yang akan dipecahkan, seperti memilih model arsitektur yang cocok untuk klasifikasi , regresi atau clustering. Kemudian nantinya model yang dibuat juga harus dilakukan tahap evaluasi, dengan harapan terhindar dari underfitting dan overfitting.
  - c. Train model

- Tahapan untuk melatih algoritma yang telah ditentukan dengan data latih yang telah disiapkan sehingga diharapkan mendapatkan model yang akurat.
- d. Test model  
Tahapan untuk mencoba model terhadap data-data baru yang belum pernah diuji apakah model dapat membuat prediksi dengan tepat atau tidak.
  - e. Deployment  
Tahap terakhir dari life cycle AI setelah semua proses telah dilalui sehingga model dapat diterapkan untuk real-world system untuk pemecahan masalah.
2. Problem Scoping adalah sebuah lingkup masalah yang mengacu pada pemahaman masalah menemukan berbagai faktor yang mempengaruhi masalah, menentukan tujuan atau sasaran proyek terhadap masalah tersebut dalam pemecahannya. Dengan pembahasan seperti apa, siapa , dimana dan mengapa permasalahan harus dipecahkan.
  3. Hal ini tidak lepas dalam peran life cycle AI dan problem scoping yaitu data yang dibutuhkan dalam permasalahan masalah, data yang ditetapkan harus mewakili dari permasalahan apa yang akan dipecahkan, hal ini pada terdapat pada gathering data, data preparation, dan jawaban dari 4W problem scoping
  4. Model adalah sebuah hasil (aturan) dari machine learning atau deep learning yang telah diajarkan serta memiliki arsitektur yang telah ditentukan. Cara untuk mengevaluasi model dengan cara melakukan testing data yang belum diajarkan dan menggunakan confusion metrics untuk evaluasi model untuk setiap permasalahan seperti supervised learning akurasi presisi, dan recall untuk klasifikasi, dan MSE atau RMSE untuk regresi.

# AI Mastery Course

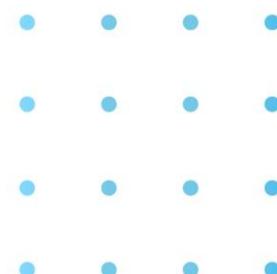


## Module 1

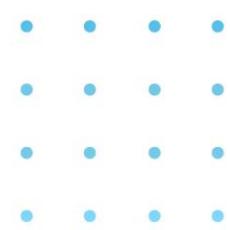
### Introduction to everything

#### Section I

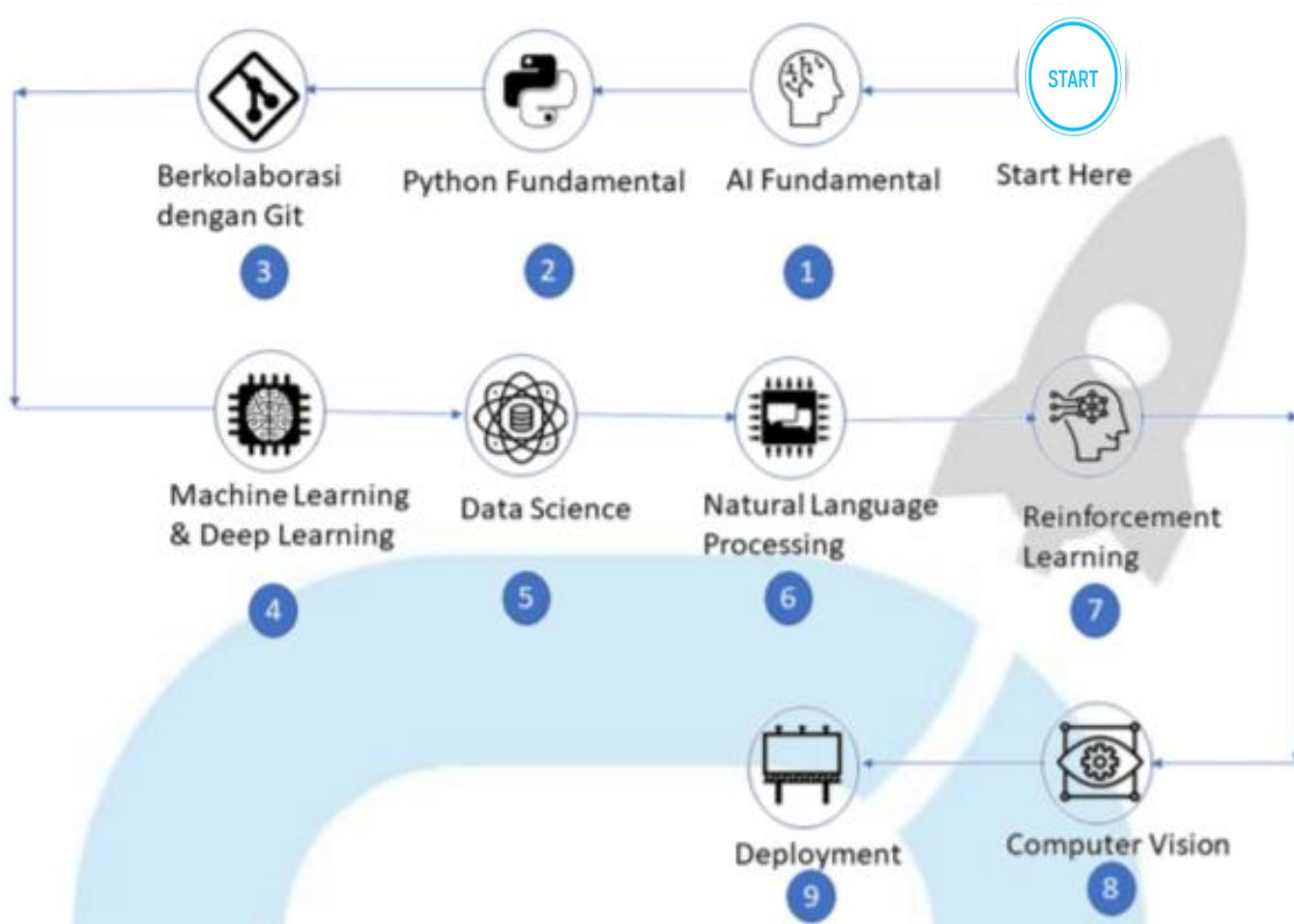
A Brief Introduction about Artificial Intelligence



# Welcome Abroad



# Learning Objectives



## About the Program



- 1000 mahasiswa/i dari seluruh Indonesia
- Idealnya, memiliki pemahaman terkait Bahasa pemrograman seperti C/C++ ataupun Python
- Tidak takut dengan matematika, kita akan banyak membutuhkan matematika untuk AI
- Jika Anda bukan berasal dari computer science, math, natural science or engineering atau tidak familiar dengan AI dan Python mohon untuk memberitahukan beritahukan ke coach kelas Anda

## About the Program



- Jam aktif (dalam WIB)
- Pagi: 08.00 – 11.30 (istirahat 30 menit)
- Siang: 13.00 – 16.30 (istirahat 30 menit)
- Toleransi keterlambatan 30 menit
- Jika, ada suatu keperluan yang membuat keterlambatan lebih dari waktu yang telah ditentukan mohon beritahukan coach kelas Anda
- Harus aktif dan semangat dalam belajar! . . .

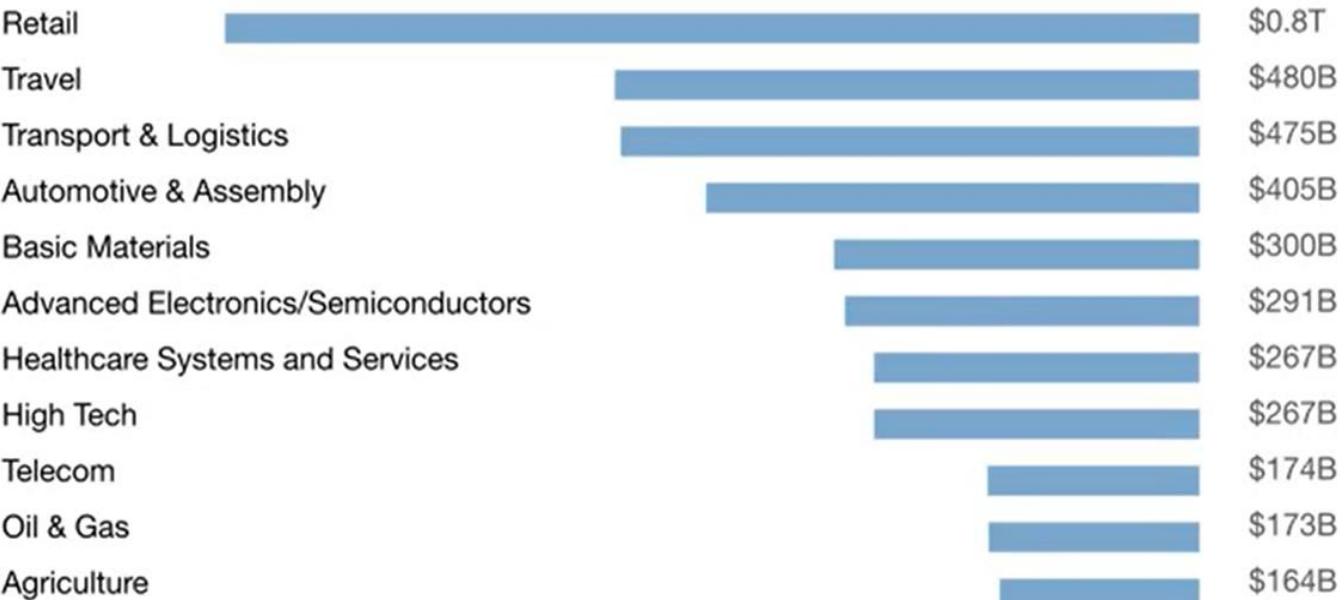


**Artificial Intelligence  
is Our Future!**

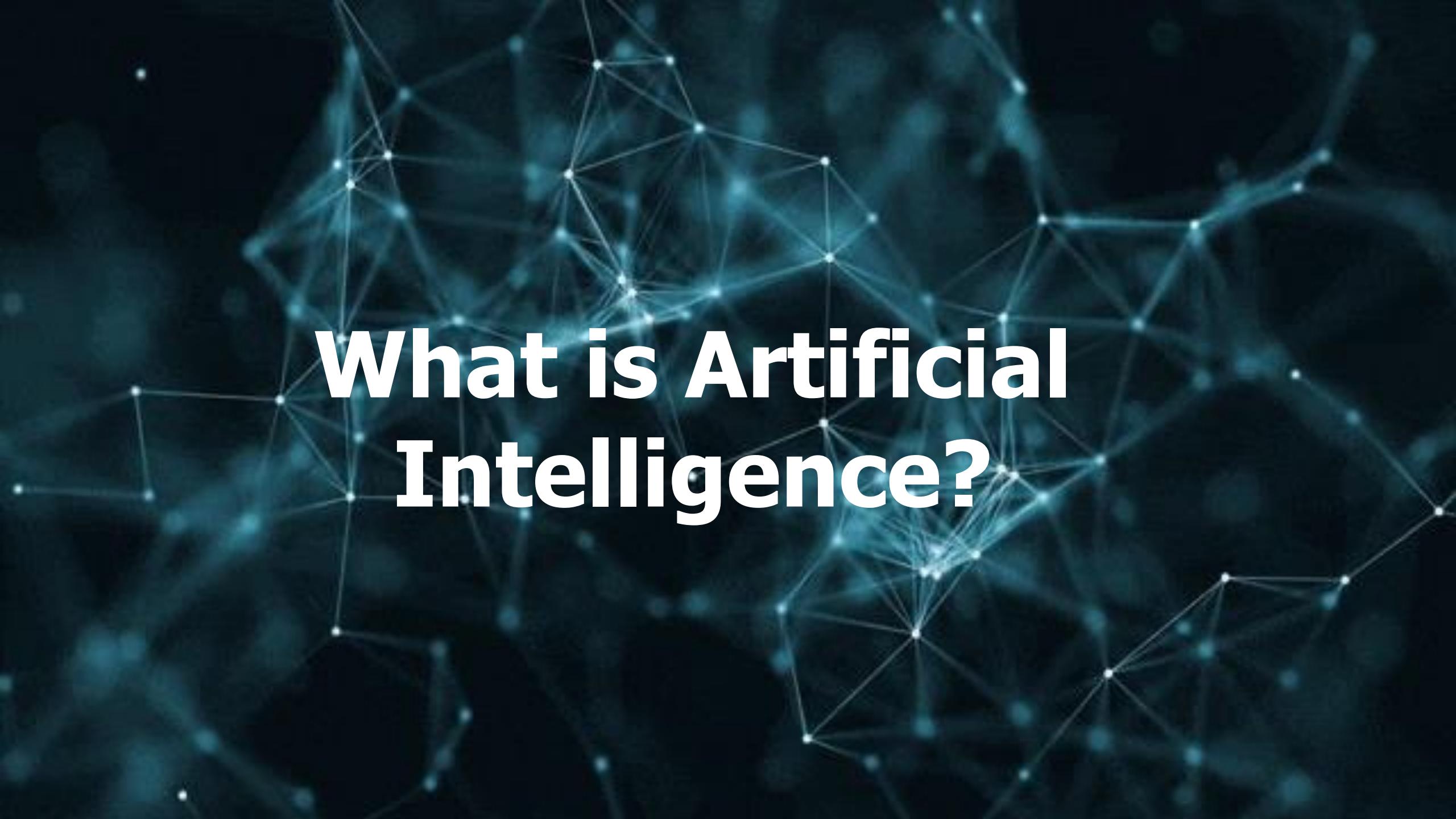
# AI Impact

AI value creation  
by 2030

\$13  
trillion



[Source: McKinsey Global Institute.]



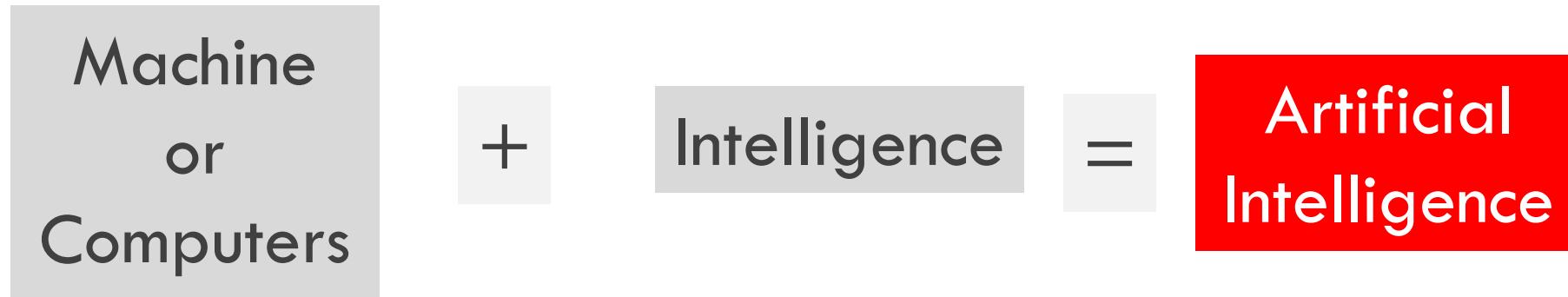
# What is Artificial Intelligence?

# What is AI?

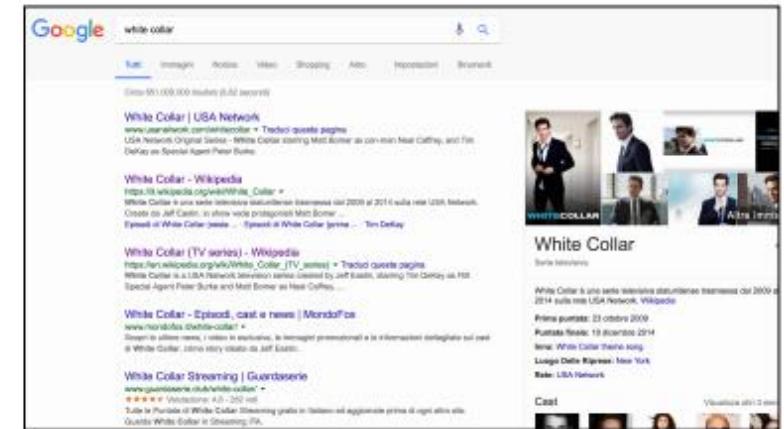
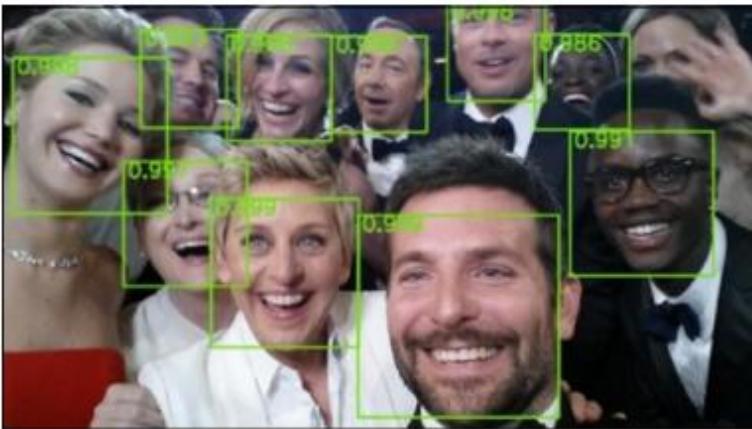
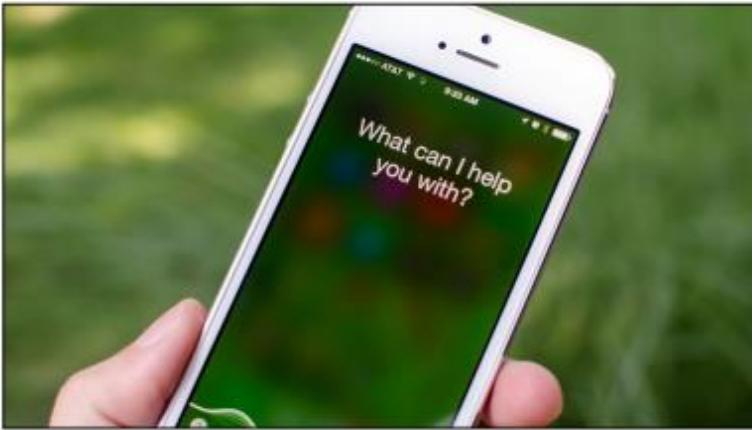
Kemampuan mesin untuk menirukan perilaku manusia yang cerdas



# AI System structure

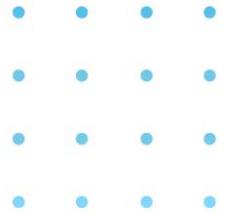


# AI in everyday products





# Kenapa Kita Harus Belajar Artificial Intelligence?



"We are entering a new world. The technologies of machine learning, speech recognition, and natural language understanding are reaching a nexus of capability. The end result is that we'll soon have artificially intelligent assistants to help us in every aspect of our lives." - **Amy Stapleton**

"I am telling you, the world's first trillionaires are going to come from somebody who masters AI and all its derivatives and applies it in ways we never thought of." - **Mark Cuban**

"Predicting the future isn't magic, it's artificial intelligence." - **Dave Waters**





Gary Kasparov



Lee Sedol

Manusia membuat mesin cerdas yang pada giliran nya, mengalahkan manusia dalam permainan nya sendiri

- Deep Blue vs Gary Kasparov (Feb 10, 1996 – May 11, 1997)
- Fan Hui vs AlphaGo (2015)
- Lee Sedol vs. AlphaGo (2016)
- Ke Jie vs AlphaGo (2017)





**Bagaimana mesin bisa  
menjadi cerdas dan  
mengalahkan manusia?**

<https://afiniti.com/corporate/rock-paper-scissors>



Sophia robot AI pertama yang memiliki kewarganegaraan



Can machines think? – The Imitation Game (Turing Test, 1950)

## Tujuan Khusus:

Dapatkah itu (AI) mencapai serangkaian tujuan terbatas yang terdefinisi dengan baik?

## Tujuan Umum:

Dapatkah ia (AI) mencapai serangkaian tujuan yang tidak dibatasi dengan jelas?

# AI Today is Amazing



# Bagaimana ini bermula?

- AI has a long history
  - Ancient Greece
    - Aristotle
  - Historical Figures Contributed
    - Ramon Lull
    - Al Khwarizmi
    - Leonardo da Vinci
    - David Hume
    - George Boole
    - Charles Babbage
    - John von Neuman
  - As old as electronic computers themselves (c1940)



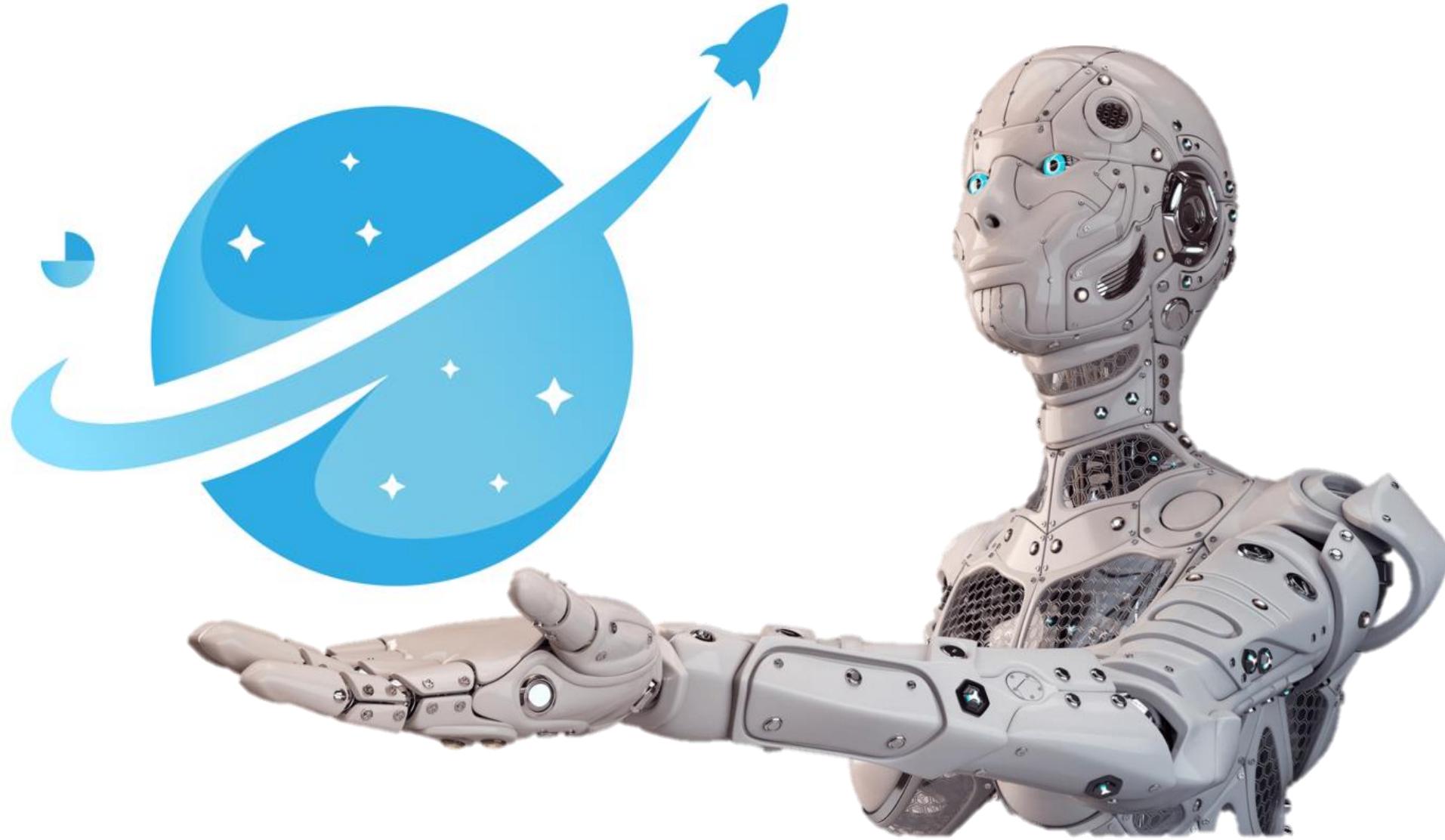
Al Khwarizmi



Charles Babbage



Ada Lovelace

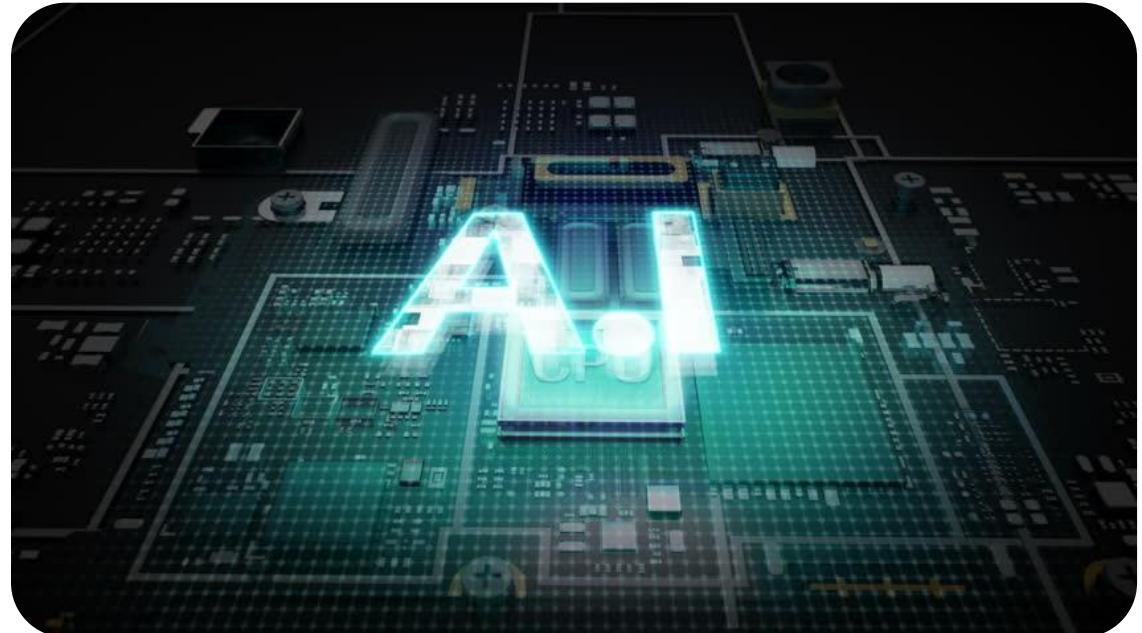


# Apa itu AI?

membuat komputer yang berpikir?

otomatisasi aktivitas yang kita kaitkan dengan pemikiran manusia, seperti pengambilan keputusan, pembelajaran ... ?

seni menciptakan mesin yang melakukan fungsi yang membutuhkan kecerdasan ketika dilakukan oleh orang ?



# Apa itu AI?



- studi tentang komputasi yang memungkinkan untuk memahami, bernalar, dan bertindak?
- bidang studi yang berusaha menjelaskan dan meniru perilaku cerdas dalam hal proses komputasi?
- cabang ilmu komputer yang mempelajari otomatisasi perilaku cerdas ?
- apa saja dalam Ilmu Komputer yang kita belum tahu bagaimana melakukannya dengan benar? (!)

# Definisi AI

- The study of how to make programs/computers do things that people do better
- The study of how to make computers solve problems which require knowledge and intelligence
- The exciting new effort to make computers think ... machines with minds
- The automation of activities that we associate with human thinking (e.g., decision-making, learning...)
- The art of creating machines that perform functions that require intelligence when performed by people
- The study of mental faculties through the use of computational models
- A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes
- The branch of computer science that is concerned with the automation of intelligent behavior

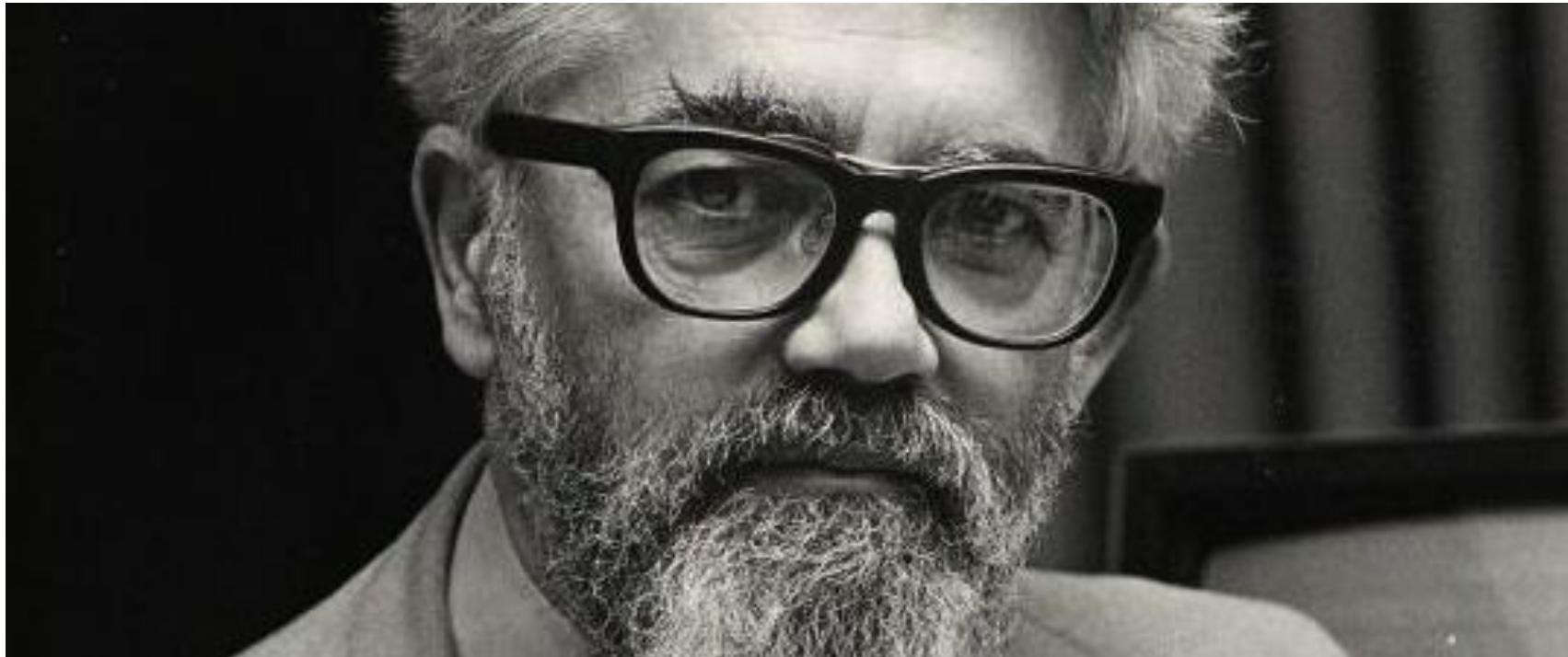
Thinking  
machines or  
machine  
intelligence



Studying  
cognitive  
faculties

Problem  
Solving and  
CS

# Father of AI



## **What is artificial intelligence?**

It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable.

[WHAT IS AI? \(12-Jul-1998\) \(stanford.edu\)](http://WHAT IS AI? (12-Jul-1998) (stanford.edu))

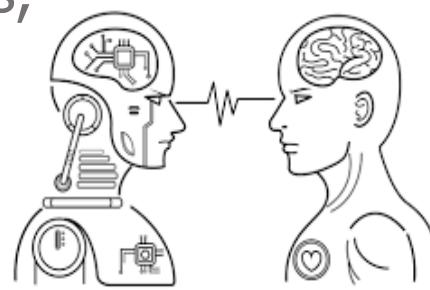
# Artificial Intelligence vs Natural Intelligence

## Artificial Intelligence

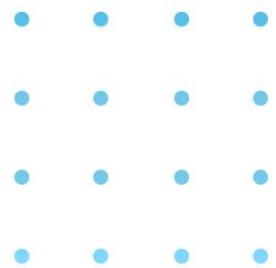
- Dapat definisikan. Ilmu yang mengembangkan metode untuk memecahkan masalah biasanya dikaitkan dengan kecerdasan manusia.
- Definisi alternatif:
  - membangun entitas atau agen cerdas;
  - membuat komputer berpikir atau berperilaku seperti manusia
  - mempelajari pemikiran manusia melalui model komputasi;
  - menghasilkan perilaku cerdas, penalaran, belajar.

## Natural Intelligence

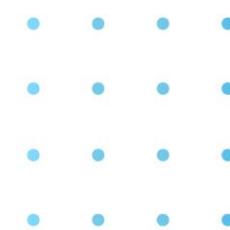
- Dapat didefinisikan. Kecerdasan – kapasitas untuk menciptakan hubungan antar gagasan. Atau kemampuan memecahkan masalah.
- Psikologi - istilah umum yang mencakup berbagai kemampuan mental, termasuk kemampuan untuk mengingat dan menggunakan apa yang telah dipelajari, untuk memecahkan masalah, beradaptasi dengan situasi baru, dan memahami dan memanipulasi realitas seseorang.



# Definisi AI

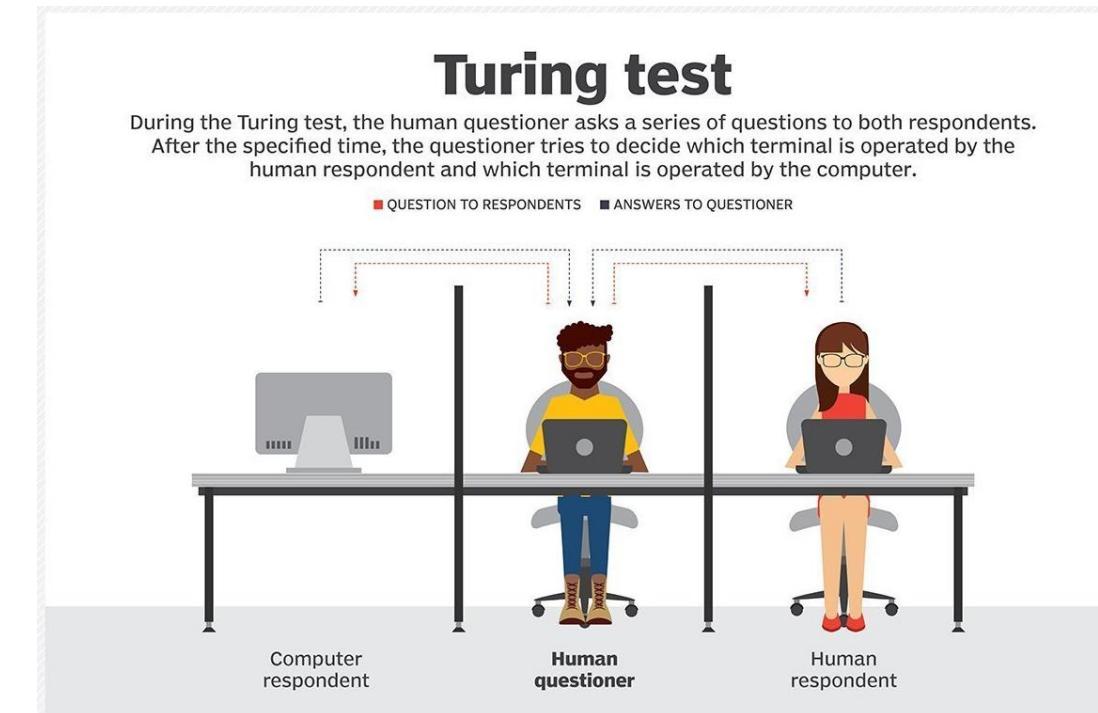


	HUMAN	RATIONAL
THOUGHT	Systems that think like humans	Systems that think rationally
BEHAVIOUR	Systems that act like humans	Systems that act rationally



# Bertindak Secara Manusiawi

- Bagaimana kita membedakan perilaku cerdas dari kecerdasan?
- Tes Turing: Menentukan apakah suatu program memenuhi syarat sebagai kecerdasan buatan dengan menginterrogasinya bersama dengan rekan manusianya.
- Program lulus jika manusia tidak dapat membedakan antara jawaban dari program komputer dan jawaban subjek manusia.
- Test ini belum lulus.

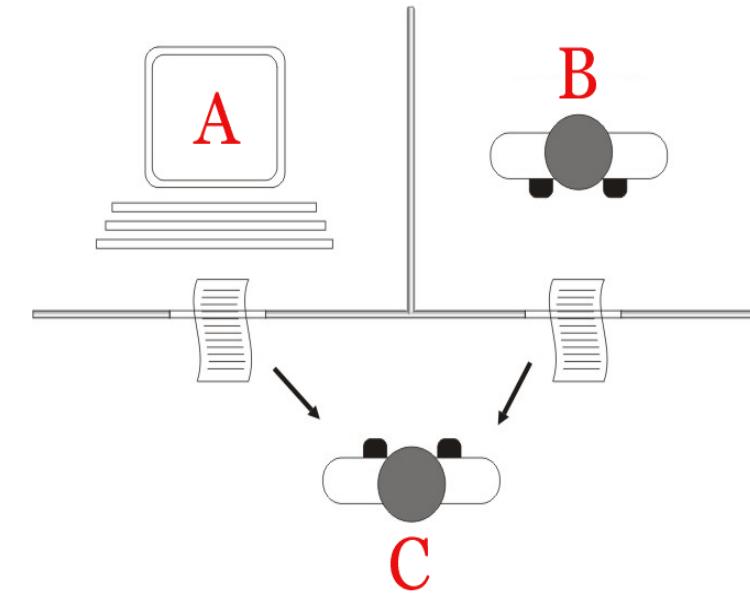


# Systems that act like humans: Turing Test



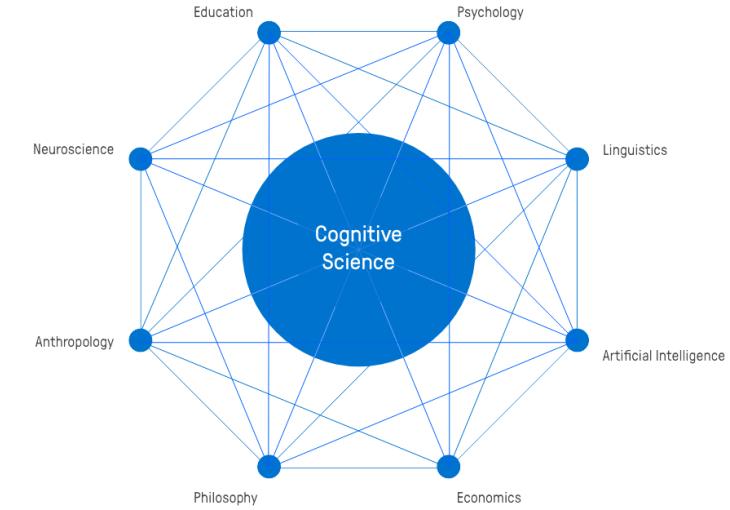
Alan Turing

Skenario dari test ini adalah bayangkan Anda memasuki sebuah ruangan yang terisolasi dari luar. Anda sebagai hakim manusia merupakan pemain C, bertindak sebagai interrogator. Di ruang terpisah ada pemain A dan B. Anda diminta untuk mengidentifikasi mana yang merupakan program komputer antara pemain A dan B dalam periode waktu tertentu. Anda dibatasi hanya untuk mengetik (menulis) apa yang apa yang anda ingin ke terminal yang terdapat di kedua sisi Anda. Anda diminta untuk mempelajari dan menganalisa hanya dari respon yang diberikan oleh pemain A dan pemain B.



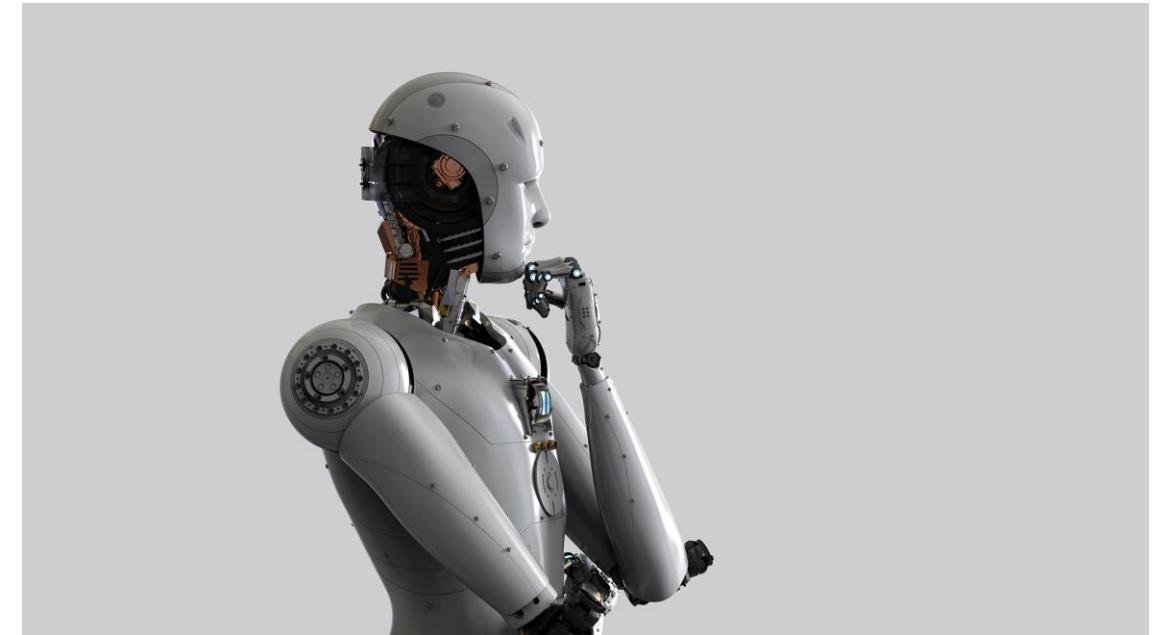
# Berpikir secara Manusiawi

- Cognitive Science: memodelkan proses pemikiran manusia.
- Melalui serangkaian eksperimen dan model komputasi, mencoba membangun penjelasan yang baik tentang apa yang kita lakukan ketika kita menyelesaikan tugas tertentu.
- Relevansi dengan AI: untuk memecahkan masalah yang dapat dilakukan oleh manusia (atau makhluk hidup lainnya)
- Pendekatan awal mencoba memecahkan masalah apa pun persis seperti yang akan dilakukan manusia. Sekarang kita tahu bahwa itu bukan pendekatan terbaik.



# Thinking Rationally

- Sistem yang mampu menalar, mampu membuat deduksi logis dari basis pengetahuan.
- Membutuhkan beberapa kapasitas untuk membuat kesimpulan logis, seperti "Semua manusia fana; Socrates adalah manusia; jadi Socrates fana". (Terdengar benar sebagai sebuah kalimat. Tapi, tidak rasional)

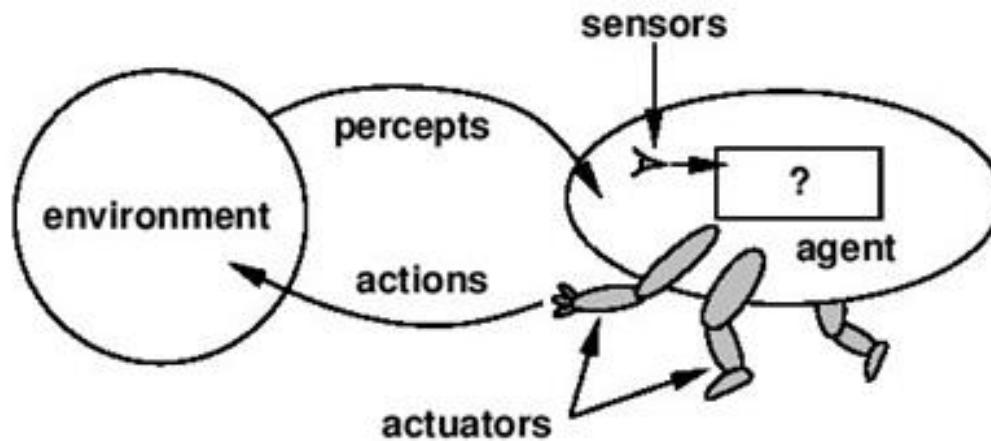


# Acting Rationally

- Banyak aplikasi AI mengadopsi pendekatan agen cerdas.
- Agen adalah entitas yang mampu menghasilkan tindakan.
- Dalam AI, agen rasional harus otonom, mampu memahami lingkungannya, beradaptasi, dengan tujuan tertentu.
- Paling sering agen adalah potongan kecil kode dengan kemahiran tertentu. Masalah diselesaikan dengan menggabungkan keterampilan beberapa agen.



# Konsep Agen



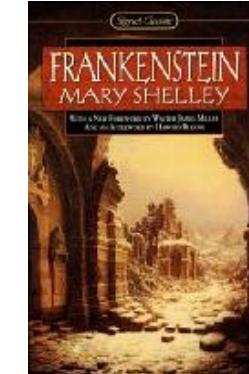
- $[f: P^* \rightarrow \mathcal{A}]$
- Program agen berjalan pada arsitektur fisik untuk menghasilkan f
- agen = arsitektur + program

- Agen adalah segala sesuatu yang dapat dilihat sebagai memahami lingkungannya melalui sensor dan bertindak atas lingkungan itu melalui aktuator
- Agen manusia: mata, telinga, dan organ lain untuk sensor; tangan, kaki, mulut, dan bagian tubuh lainnya untuk aktuator
- Agen robot: kamera dan pencari jangkauan inframerah untuk sensor;
- berbagai motor untuk aktuator

# State of the Art



1818



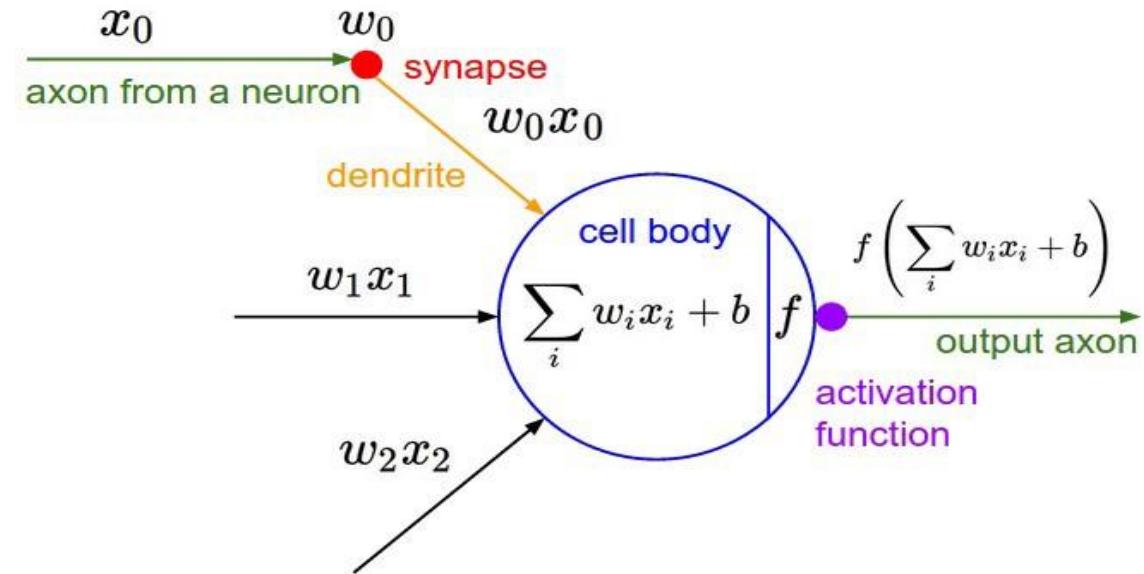
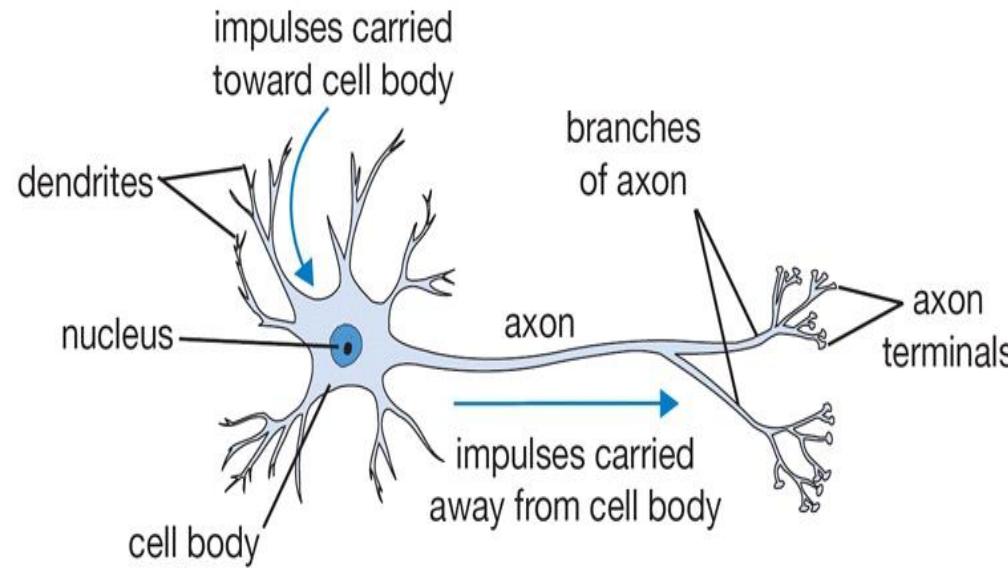
1931

# Revoluti

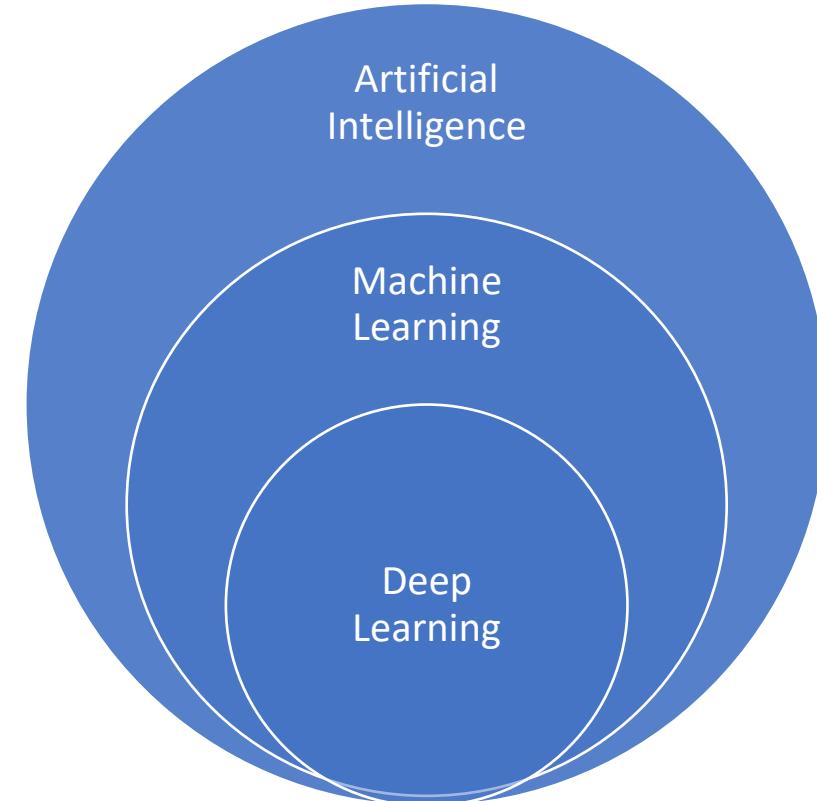


Sains dan teknologi menghasilkan revolusi industri yang pada gilirannya berhasil mengubah pola dan perilaku hidup kita sebagai manusia

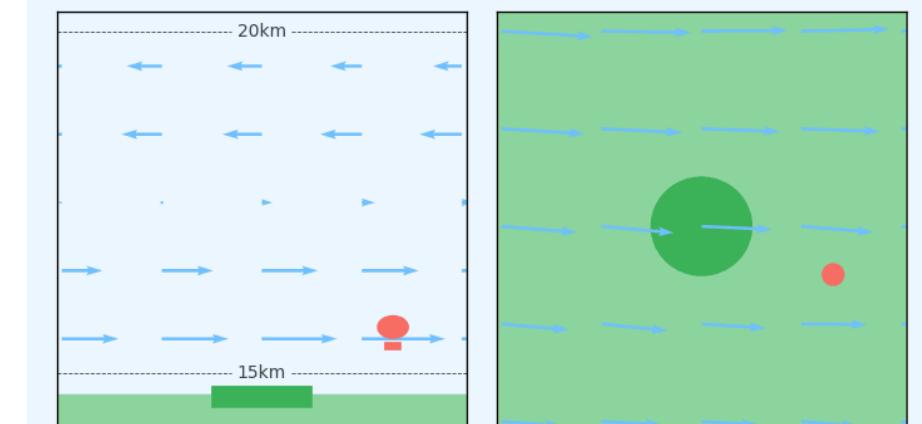
# The Main Idea



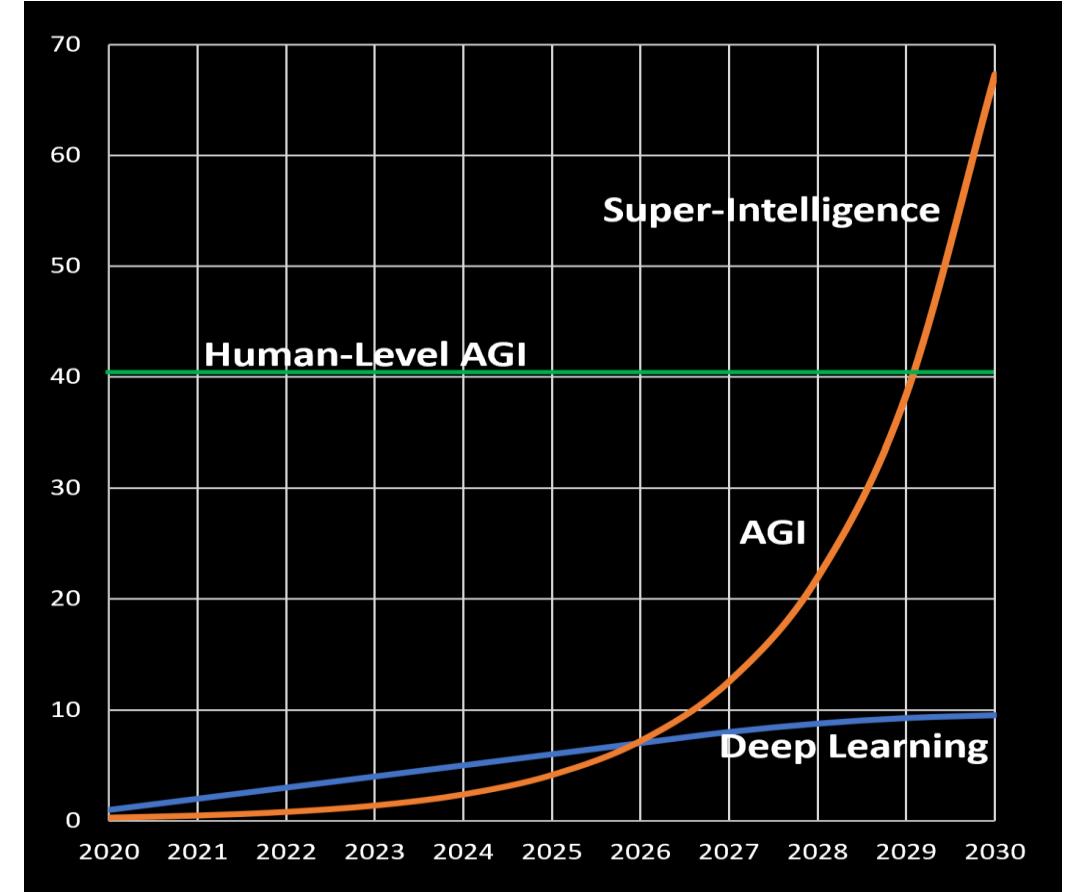
# AI, ML, DL



# ANI VS AGI



Hi, AI 🤖

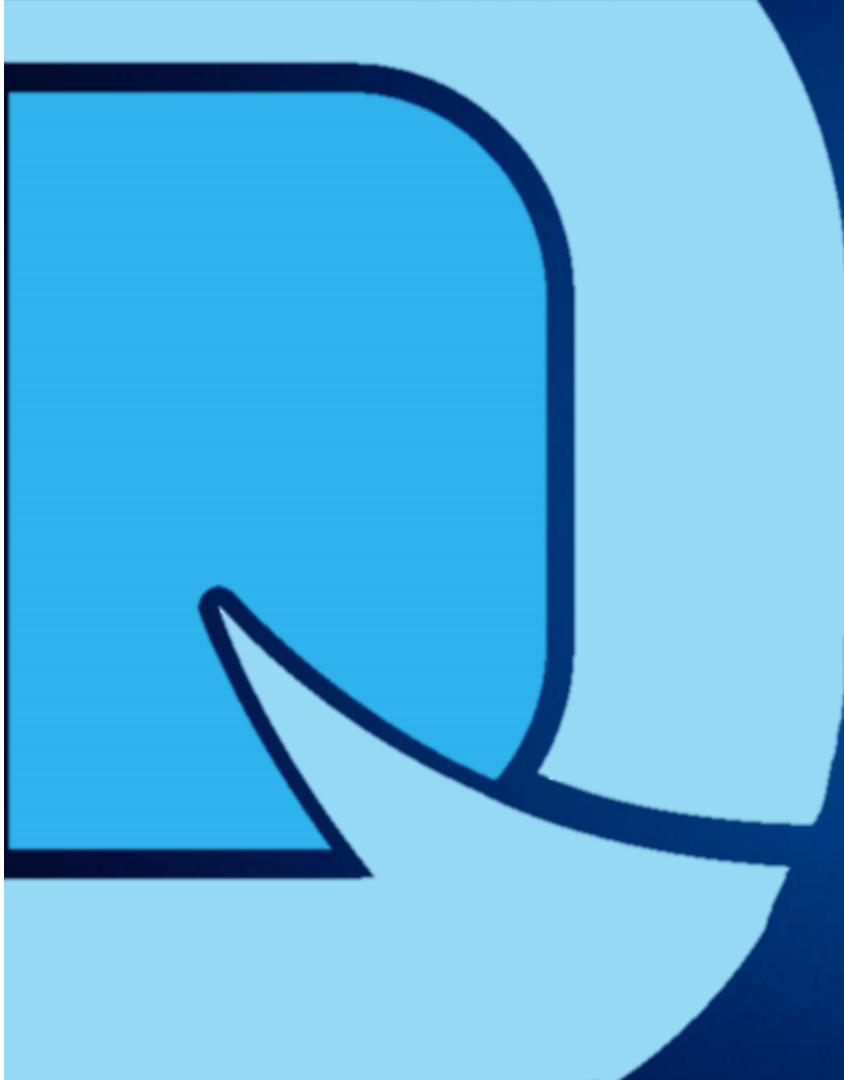




**Thanks!**

# Referensi

- [Artificial Intelligence Resources \(berkeley.edu\)](#)
- [WHAT IS AI? \(12-Jul-1998\) \(stanford.edu\)](#)
- [A Brief History of AI \(aitopics.org\)](#)
- [Lessons from a Restricted Turing Test \(harvard.edu\)](#)
- [Rock, Paper, Scissors | AfinitiDeep Blue vs Kasparov: How a computer beat best chess player in the world - BBC News – YouTube](#)
- [Match 1 - Google DeepMind Challenge Match: Lee Sedol vs AlphaGo – YouTube](#)
- [AlphaGo - The Movie | Full award-winning documentary – YouTube](#)
- [Can robots bond with people? A new study says yes | Woebot Health - YouTube](#)
- [AI Mastery 4 Individual \(orbitfutureacademy.id\)](#)
- [Artificial Intelligence Mastery Program | Merdeka Belajar - Kam... \(kemdikbud.go.id\)](#)
- [AI & Machine Learning \(intel.com\)](#)
- [Intel® Artificial Intelligence \(AI\) and Deep Learning Solutions](#)



# Module 1

## Introduction to everything

### Section

#### AI in Industry



## Sesi I

AI in Industry

# Mari Renungkan

Perusahaan yang paling populer saat ini?

**ALODOKTER**    **RedDoorz**

**bukalapak**

**gojek**

**airbnb**

**amazon**

**traveloka**

**tiket.com**

**Shopee**

**Grab**

**tokopedia**

**ALODOKTER**

bukalapak

**amazon**

**tiket.com**

**Shopee**

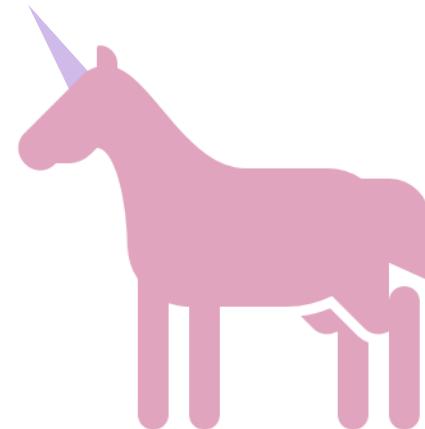
**tokopedia**

**RedDoorz**

airbnb

**traveloka**

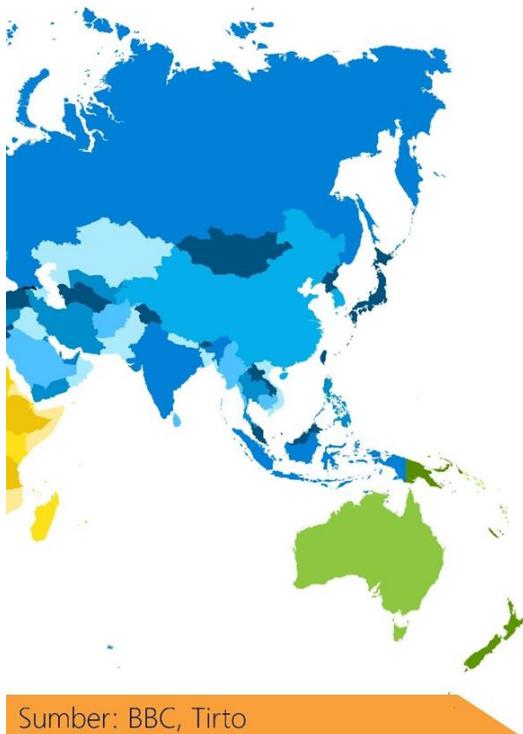
**Grab**



---

**Unicorn:** perusahaan rintisan yang nilai ekonominya/asset hingga US\$1 miliar atau Rp14.2 Triliun

**Startup:** perusahaan yang baru beroperasi atau masih meneliti untuk menemukan pasar maupun mengembangkan produknya

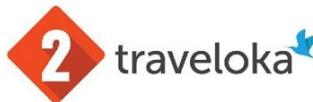


## Startup Unicorn Indonesia

Perusahaan dengan nilai valuasi di atas US\$1 miliar



1 GO-JEK  
Rp 53 triliun  
Nadiem Makarim  
2010



2 traveloka  
Rp 26.6 triliun  
Ferry Unardi  
2012



3 tokopedia  
Rp 15 triliun  
William Tanuwijaya  
2009



4 Bukalapak  
Rp 13.7 triliun  
Achmad Zaky  
2010

Sumber: BBC, Tirto

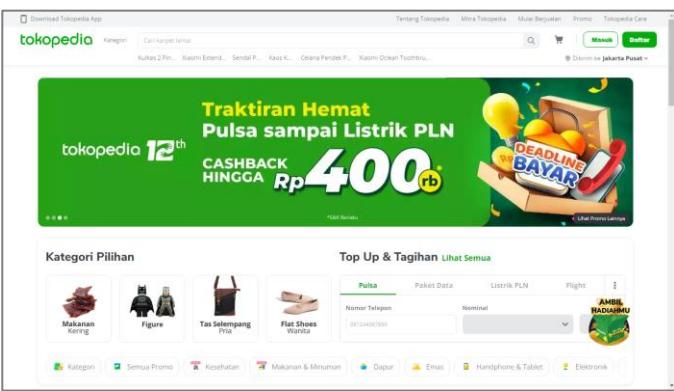
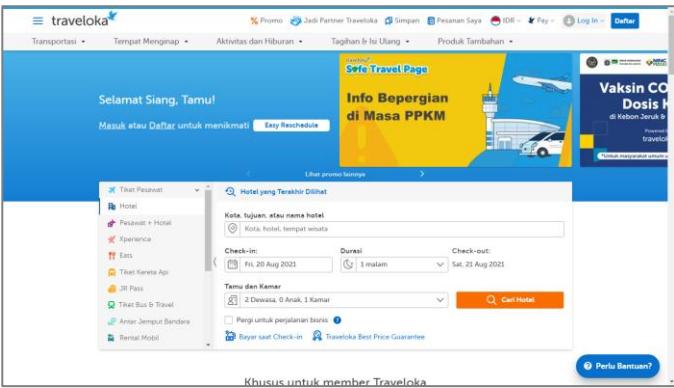
Infografees.com

# Mari Renungkan

## Persamaan dari perusahaan tersebut?

---

Setidaknya terdapat tiga persamaan mendasar



**Pertama**, bisa kita sebut, bahwa perusahaan-perusahaan tadi, sama-sama memiliki aplikasinya masing-masing. Entah itu aplikasi web atau aplikasi smartphone. Aplikasi tersebut juga bisa kita download dan pakai secara gratis.



**Kedua**, perusahaan tersebut beroperasi dengan mengandalkan internet. Kita hanya bisa menggunakan aplikasi mereka melalui internet



Image source: Tatanan Kehidupan Baru, Industri Manufaktur Dipantau Ketat - Ekonomi | (rri.co.id)

**Ketiga**, adalah yang paling penting. Mereka bukan perusahaan atau industri manufaktur (tidak memproduksi sesuatu). Contoh: Tokopedia tidak membuat produknya sendiri untuk dijual. Gojek tidak memiliki armada ojek (motor)-nya sendiri. Dst

# Mari Renungkan

## Apa yang mereka lakukan & tawarkan?

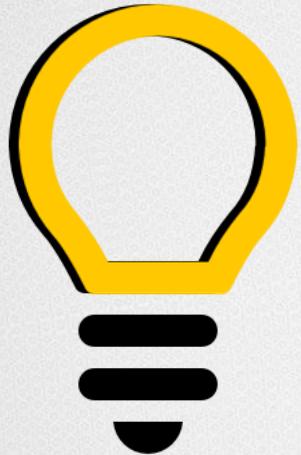
---

Lalu apa yang mereka **lakukan dan tawarkan** ke customer untuk mendapat profit?  
Mari kita pikirkan Kembali teman-teman! Menurut teman-teman, apa yang mereka lakukan dan tawarkan?



## Data Processing

Sebenarnya yang mereka lakukan dalam menjalankan usaha cukup sederhana, yakni hanya **memproses data** (data processing)



# Solution

Yang mereka tawarkan juga cukup sederhana. Mereka hanya menawarkan **insight/solusi** dari permasalahan customer.

# Contoh



## Ojek Online

---

Kita ambil contoh perusahaan ojek online **Gojek**. Kita analisis apa yang dilakukan dan ditawarkan oleh **Gojek** ke para customernya.

We have problem...



**Penumpang**

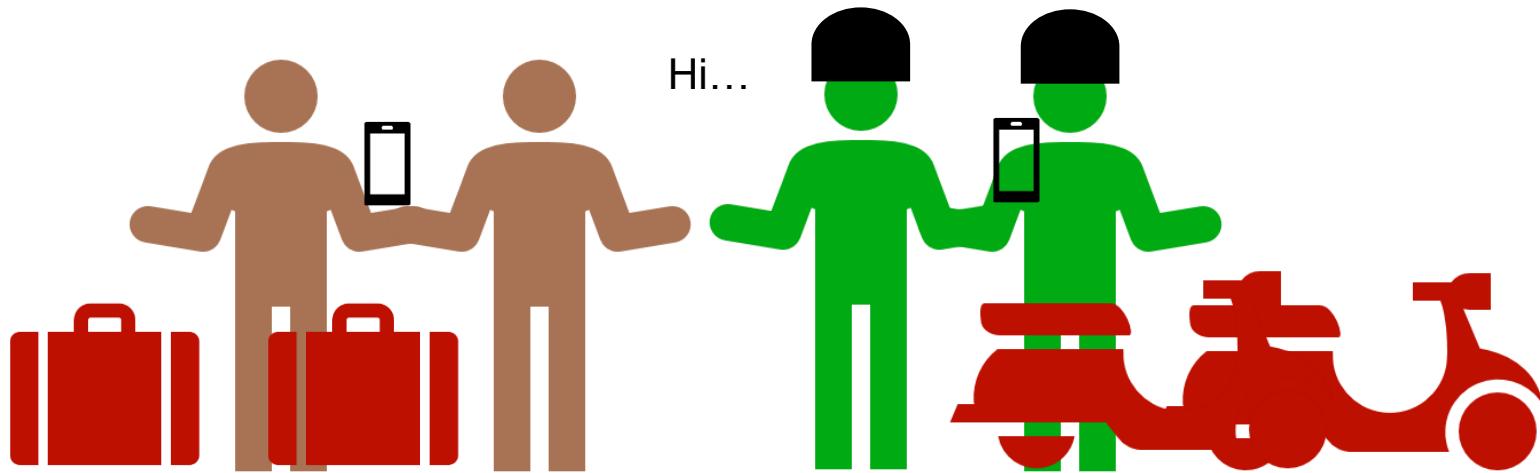


**Pengemudi**

Gojek, pada dasarnya, **menawarkan insight/solusi** permasalahan kepada para customernya (baik dari sisi penumpang atau dari sisi pengemudi.)

Jika kita analisis:

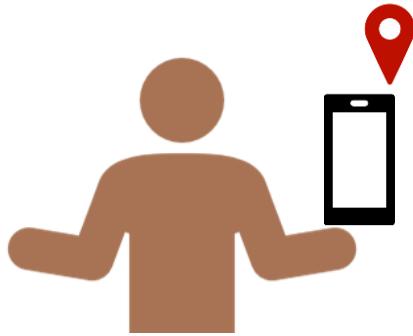
- Penumpang butuh kendaraan yang bisa mengantarkan mereka ke tujuan (butuh tumpangan); sedangkan
- Pengemudi memiliki kendaraan tetapi tidak memiliki penumpang (butuh uang)



Penumpang

Pengemudi

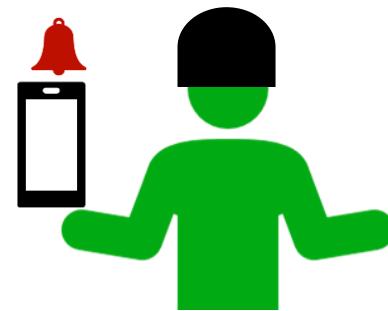
Insight/solusi yang ditawarkan Gojek adalah mempertemukan penumpang dengan pengemudi sehingga **problem** keduanya bisa teratasi



- Lokasi
- Tujuan



- Tarif
- Rute



- Lokasi
- Tujuan
- Tarif
- Rute

Lalu, apa **yang Gojek lakukan** untuk mempertemukan keduanya?

Gojek memproses data penumpang dan pengemudi.

- Pertama, Gojek akan mengambil data penumpang yang memesan tumpangan. Data berupa lokasi dan tujuan penumpang
- Kemudian, data-data tersebut akan diolah oleh Gojek sehingga menghasilkan tarif angkutan dan rute perjalanan terbaik
- Jika penumpang setuju dengan tarif dan rute yang ditawarkan, selanjutnya, Gojek akan mem-broadcast pesanan penumpang ke pengemudi terdekat
- Terakhir, pengemudi yang tertarik akan menerima pesanan penumpang



Pada akhirnya, penumpang dan pengemudi bisa dipertemukan dan problem keduanya terselesaikan.

- Penumpang bisa sampai ke tujuan berkat pengemudi
- Pengemudi bisa mendapat uang berkat penumpang

# Ayo Analisis!

# Kesempatan Besar



Dari hasil analisis Gojek dan contoh-contoh lain yang teman-teman paparkan, bisa kita lihat bahwa ada **kesempatan besar**.

Yang, memungkinkan kita mendapat profit dari hasil pemrosesan/pengolahan data. Seperti yang dilakukan oleh: Gojek, Tokopedia, Grab, Bukalapak, Traveloka, Amazon, Airbnb, dst



## Sesi II

Why is Data called The New Oil?

<https://www.youtube.com/watch?v=kG-Naum0Dvk>  
[https://www.youtube.com/watch?v=Hc83GcMKW\\_8](https://www.youtube.com/watch?v=Hc83GcMKW_8)

# “DATA IS THE NEW OIL”

-Clive Humby, 2006

Image Source: <https://www.linkedin.com/pulse/new-oil-satish-singh/>



*“Data is the new oil. It’s valuable, but if unrefined it cannot really be used. It has to be changed into gas, plastic, chemicals, etc to create a valuable entity that drives profitable activity; so must data be broken down, analyzed for it to have value.”*

**Clive Humby, 2006**

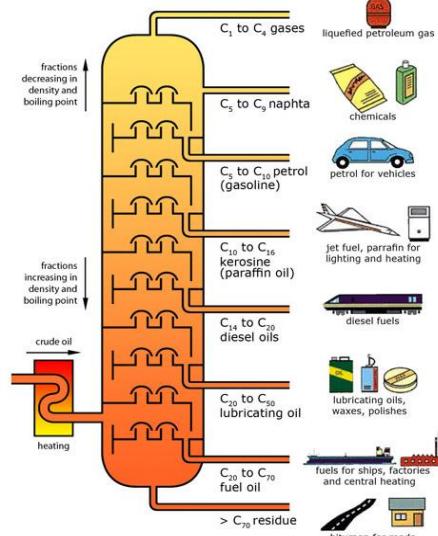
Image Source:

<https://www.oyd.sqv.mybluehost.me/interview-clive-humby/>



Image Source: the episode Search Committee from The Office

Jadi, jika suatu hari teman-teman disuruh mencari perbedaan diantara gambar data dan minyak mentah (oil). Teman-teman bisa dengan percaya diri menjawab, bahwa keduanya sama, sama-sama **berharga jika diproses**.



**AutomationForum.in**

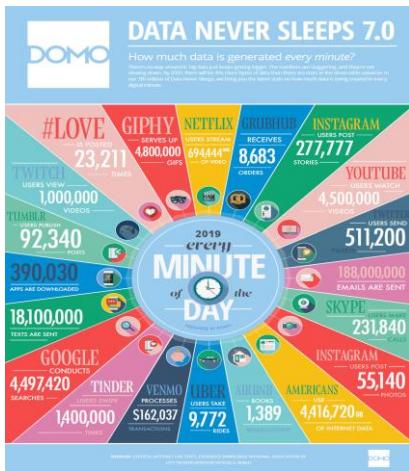
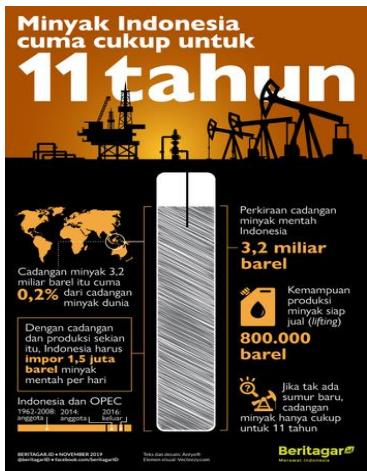
Minyak mentah berharga jika diproses menjadi: aspal, BBM, lilin, aftur, gas, dll.  
Contohnya seperti gambar di sebelah kiri.

Data juga berharga jika diproses menjadi insight/solusi.  
Contohnya seperti Gojek yang mampu mendapat profit dari hasil pemrosesan data penumpang dan pengemudi.

Lalu, apa perbedaan antara minyak mentah (oil) dan data? Coba teman-teman share pendapat kalian!



Data juga harus  
diolah agar bisa  
bernilai



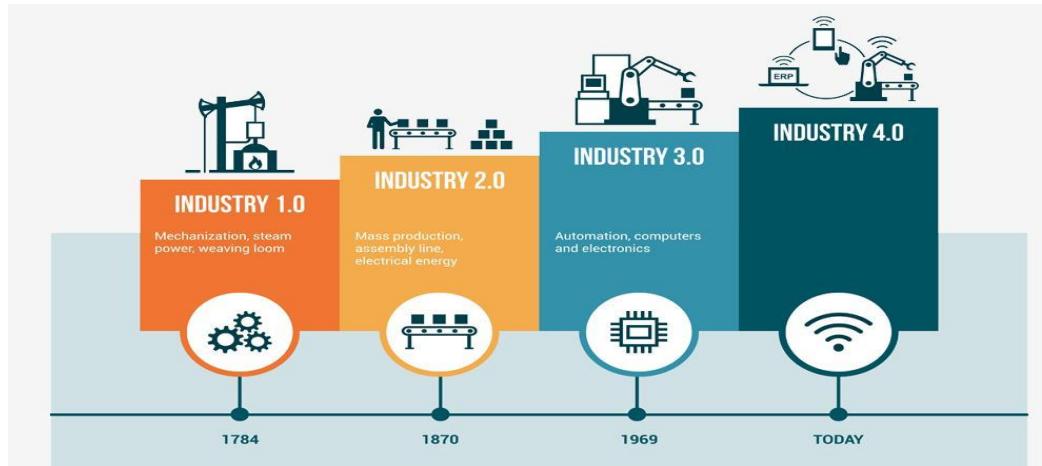
Perbedaan yang paling mencolok antara minyak mentah (oil) dan data bisa kita lihat di kedua infografis ini.

- Minyak mentah adalah sumber daya yang tidak dapat diperbaharui dan terus menipis
  - Data adalah sumber daya yang terus tumbuh dan bertambah sepanjang waktu ( $\infty$ ). Intinya, kesempatan mendapat profit dari hasil pemrosesan data juga semakin tinggi karena sumber dayanya tidak akan habis
  - Intinya, kesempatan mendapatkan profit dari hasil pemrosesan data juga semakin tinggi karena sumber dayanya tidak akan habis

# Data dan Revolusi Industri

<https://www.youtube.com/watch?v=wfAPL5Ixuk>

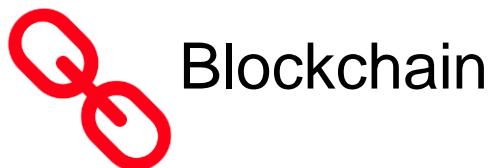
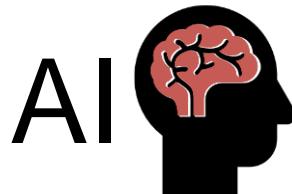
Saat ini. Pembahasan tentang Data tidak bisa dipisahkan dari Revolusi Industri.  
Menurut teman-teman apa itu Revolusi Industri?  
Mengapa Revolusi Industri berkaitan erat dengan data?



[https://www.youtube.com/watch?v=r\\_7Qos3KjIE](https://www.youtube.com/watch?v=r_7Qos3KjIE)

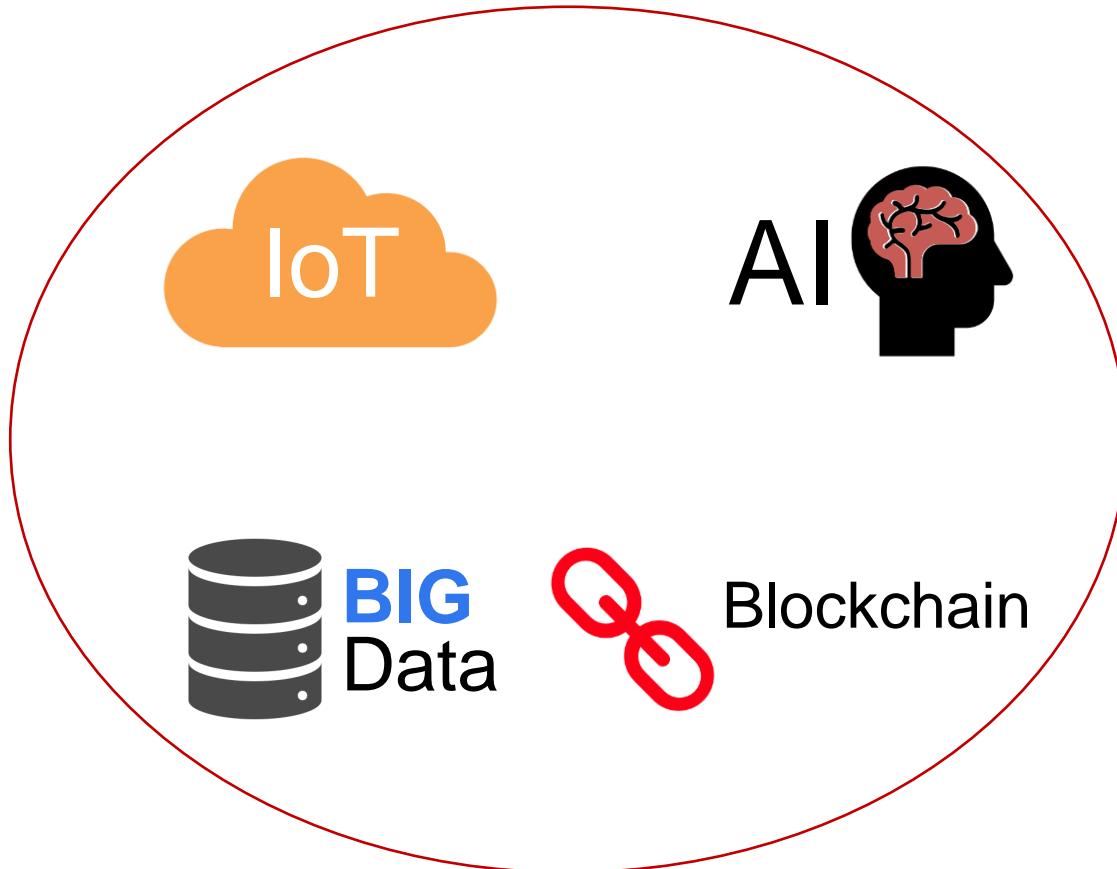
Revolusi industri sudah dimulai cukup lama, yakni dari abad ke-17, berawal dari

- Revolusi Industri 1.0 di mana mulai digunakannya teknologi **mesin uap** dan **mesin tenun** di industry (mechanical production -> steam and water)
- Revolusi Industri 2.0 di mana mulai digunakannya teknologi **listrik** dan **perakitan** di industry (mass production -> electricity)
- Revolusi Industri 3.0 di mana mulai digunakannya teknologi **komputer** dan **robot** di industry (automated production)
- Hingga saat ini yakni Revolusi Industri 4.0. Menurut teman-teman, teknologi apa yang digunakan pada Revolusi Industri 4.0 ini?  
(intelligent production -> IoT, Cloud, BigData )



Contoh teknologi yang booming di Revolusi Industri 4.0, yakni:

- **Internet of Things (IoT):** teknologi yang memungkinkan sebuah objek memiliki kemampuan untuk mentransmisikan atau mengirimkan data melalui jaringan (internet) tanpa bantuan manusia
- **Artificial Intelligence (AI):** teknologi yang membuat mesin/komputer cerdas melalui data, sehingga mampu mengerjakan tugas manusia.
- **Blockchain:** teknologi penyimpanan sekaligus pengamanan data digital.
- **Big Data;** kita akan membahas teknologi ini di sesi berikutnya.



# Data

Jika kita tarik benang merah, maka teknologi-teknologi ini sangat erat kaitannya dengan **Data**

# “Era of Data”

Karena teknologi-teknologi yang muncul pada Revolusi Industri 4.0 sangat erat kaitannya dengan data, maka Revolusi 4.0 sering disebut juga sebagai “**Era of Data**” (Era Data)



## Sesi III

Big Data

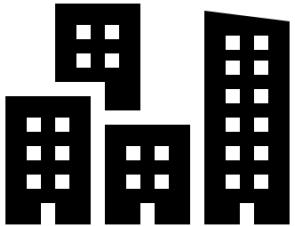


### Setiap industri pasti memerlukan bahan baku. Contohnya:

- Industri makanan memerlukan daging, bumbu, dan tepung
- Industri garmen memerlukan kapas dan pewarna
- Industri mebel memerlukan kayu, lem, dan paku
- Industri otomotif memerlukan baja dan cat
- Dsb

# Bahan Baku & Sumber Industri 4.0?

# Sumber Data Sebelum Era Data



NAMA MATA KULIAH	PROGRAM STUDI	KELAS
KOMPUTER & GRAFS	TEKNIK SIREN	A1
PROBABILITIK DAN STOKASTIK	TEKNIK ELEKTRO	A1
KALKULUS II	TEKNIK ELEKTRO	A1
KESELAMATAN DAN KESEHATAN KERJA	TEKNIK KIMIA	A1
DISKRIMINAN DAN KONSEP	TEKNIK MESIN	A1
MATEMATIKA TEKNIK II	TEKNIK MESIN	A1
PERANCANGAN TATA LETAK FASILITAS	TEKNIK INDUSTRI	A1
PERENCANAAN DAN PEMERINTAHAN	ARISTEKSIUR	A1
PENGANTAR PERENCANAAN KAWASAN	ARISTEKSIUR	A1
TEKNOLOGI BANGUNAN I	ARISTEKSIUR	A1
TEKNOLOGI BANGUNAN IV	ARISTEKSIUR	A1
TEKNOLOGI BANGUNAN V	ARISTEKSIUR	A1
ALJABAR LINER & MATRICK	TEKNIK INFORMATIKA	A1
ALJABAR LINER & MATRICK	TEKNIK INFORMATIKA	A2
ARISTETUR & ORGANISASI KOMPUTER	TEKNIK INFORMATIKA	A1
ARISTETUR & ORGANISASI KOMPUTER	TEKNIK INFORMATIKA	A2
MATEMATIKA TERAPAN II	TEKNIK OTOMOTIF ALAT BERAT (03)	A1
INTERAKSI MANAJAH II	TEKNIK OTOMOTIF	A1
AL ISLAM IV	TEKNIK ELEKTRO	A1
PENDIDIKAN PANCASILA	TEKNIK ELEKTRO	A1
TERMODINAMIKSA TEKNIK KIMIA	TEKNIK KIMIA	A1
TERMODINAMIKSA TEKNIK KIMIA	TEKNIK KIMIA	A1
ERGONOMI & PERANCANGAN SISTEM KERJA	TEKNIK INDUSTRI	A1
PENGANTAR ARISTETUR BERKELANJUTAN	ARISTEKSIUR	A1
ARISTETUR PERADABAN ISLAM	ARISTEKSIUR	A1
ARISTETUR PERADABAN ISLAM	ARISTEKSIUR	A1
BAHASA INGRIS II	TEKNIK INFORMATIKA	A1
BUDAYA ISLAM	TEKNIK INFORMATIKA	A1
ETIKA PROFESI	TEKNIK INFORMATIKA	A1
THERMODYNAMIKSA	TEKNIK OTOMOTIF ALAT BERAT (03)	A1



Kantor/Institusi

Kegiatan Bisnis

Administrator

## Sumber Data Setelah Era Data



Image Source: Arduino

Sensors

Sensor



Internet



Media Sosial

## Data Sebelum Era Data

- Data diinput secara manual
- Berbentuk tabular
- Data lebih terstruktur
- Ukuran data kecil
- Pertumbuhan data lambat

## Data Sesudah Era Data

- Data dibuat oleh user
- Bentuknya beragam
- Data tidak terstruktur
- Ukuran data besar
- Pertumbuhan data cepat

# Big Data

Lahir dari data tidak terstruktur yang tumbuh dengan cepat

# Big Data?

<https://www.youtube.com/watch?v=aC2CmTTZTVU>



**Tautan:**

<https://www.youtube.com/watch?v=aC2CmTTZTVU>

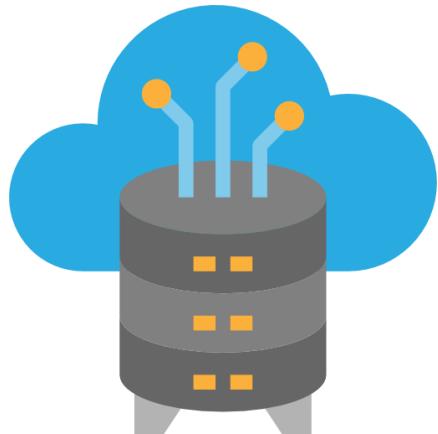
Bisa kita tarik kesimpulan, Big Data adalah sebuah istilah yang berkaitan dengan 3 sifat (3V), yakni:

Volume  
Ukuran

Velocity  
Kecepatan

Variety  
Keragaman

<https://www.youtube.com/watch?v=XQUyJfUscpk>



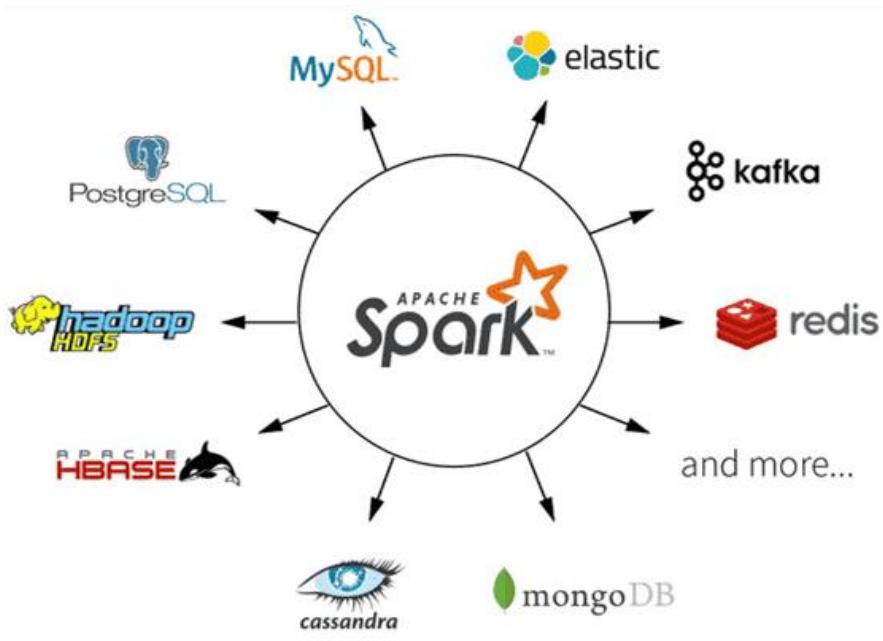
## Karakteristik Big Data sama dengan data sesudah era data, yaitu:

- Dibuat oleh user (User Generated Content) media sosial user.  
= Contohnya postingan
- User terlibat aktif dalam pertumbuhan data mengedit, mengomentari, menghapus posingan.  
= User dapat memuat,
- Datanya noisy (cenderung kotor atau tidak terstruktur) = User bebas membuat postingan dengan bahasa apapun.
- Mencakup wilayah yang luas  
postingan di manapun dan kapanpun ia suka.  
= User dapat membuat
- Informasi berharga hanya bersifat sementara  
Youtube  
Rekomendasi video hari ini akan berbeda dengan rekomendasi video esok hari (berubah tergantung selera user)  
= Contohnya: rekomendasi

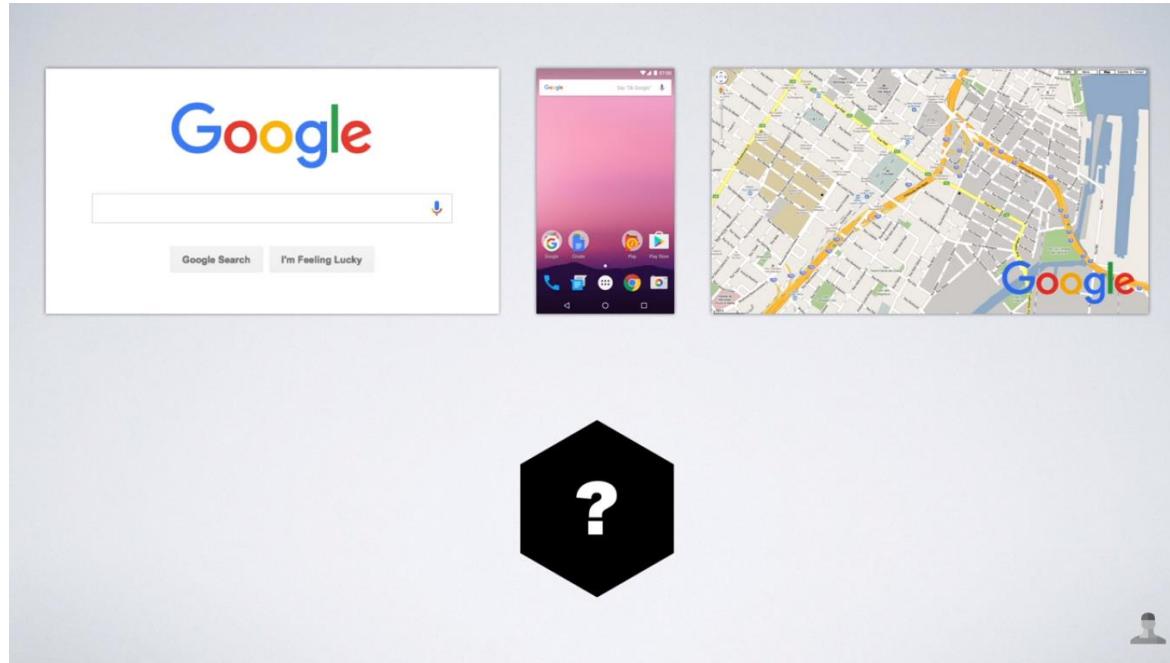


Tautan: <https://www.youtube.com/watch?v=XZmGGAbHqa0>

# Kesempatan Besar



<https://www.youtube.com/watch?v=fM39s4EN6pU>



**Tautan:** <https://www.youtube.com/watch?v=W8kA-EZReA0>

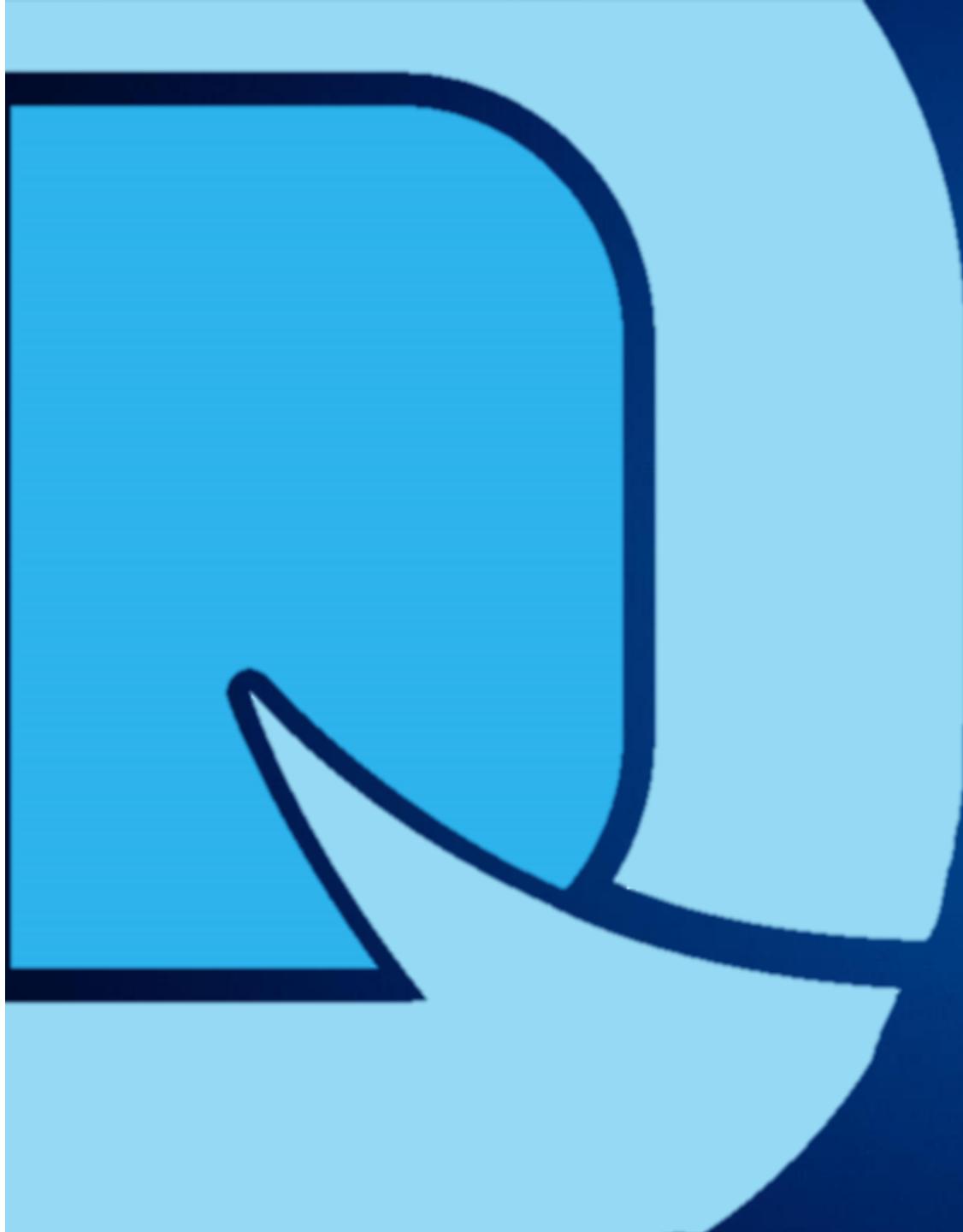
# Quiz

# Kenapa data disebut sebagai minyak mentah yang baru?

# Apa perbedaan Data dengan Big Data?

# Refleksi

# Terima Kasih



# Module 1

## Introduction to everything

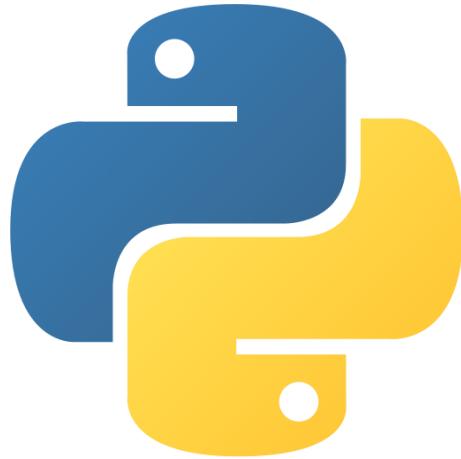
### Section

#### Python Introduction

# Let's Start Coding

Introduction to Programming

# What is Python?

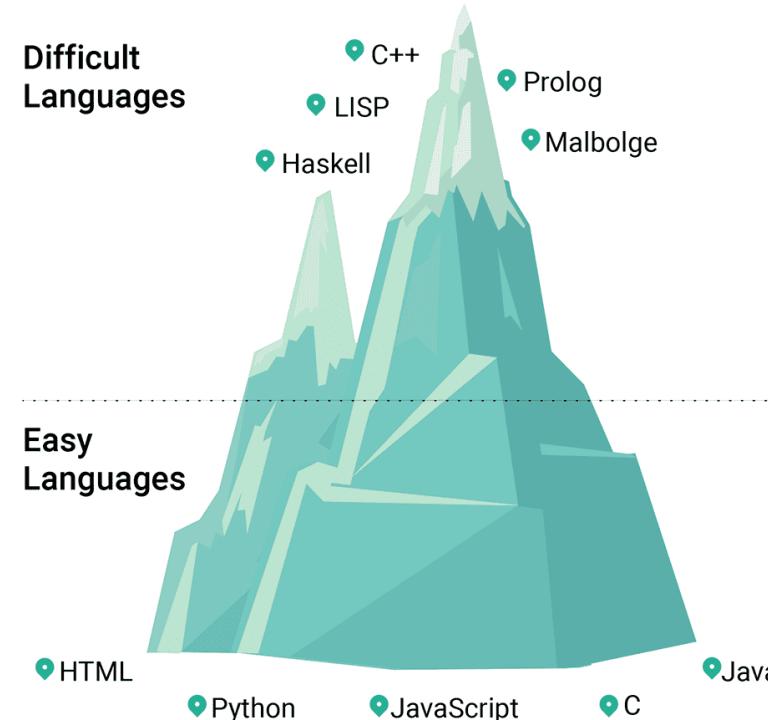


Python adalah bahasa pemrograman tingkat tinggi yang dinamis, serbaguna, dan mudah untuk dipelajari. Python juga dikenal sebagai “*glue language*”.

---

**Glue Language:** Bahasa pemrograman yang dapat dengan mudah berinteraksi dengan bahasa pemrograman lain beserta library-nya.

# How easy is Python?



Sumber Gambar:

<https://www.springboard.com/library/software-engineering/top-programming-languages/>

Berdasarkan survei yang dilakukan oleh wpengine, di USA, python menempati urutan ke-2 sebagai bahasa pemrograman paling mudah untuk dipelajari.

Python juga masuk ke dalam Top 5-bahasa pemrograman yang paling mudah dipelajari menurut Springboard.

# How easy is Python?

```
public class CobaCetak {  
    public static void main(String[] args) {  
  
        System.out.println("Hello Dunia");  
  
    }  
}
```

```
#include <iostream>  
using namespace std;  
main(){  
    cout<<"Hello Dunia";  
}
```

```
#include <stdio.h>  
main(){  
    printf("Hello Dunia");  
}
```

```
print("Hello Dunia")
```

# How easy is Python?

## Java

```
public class CobaCetak {  
    public static void main(String[] args) {  
  
        System.out.println("Hello Dunia");  
  
    }  
}
```

## C++

```
#include <iostream>  
using namespace std;  
main(){  
    cout<<"Hello Dunia";  
}
```

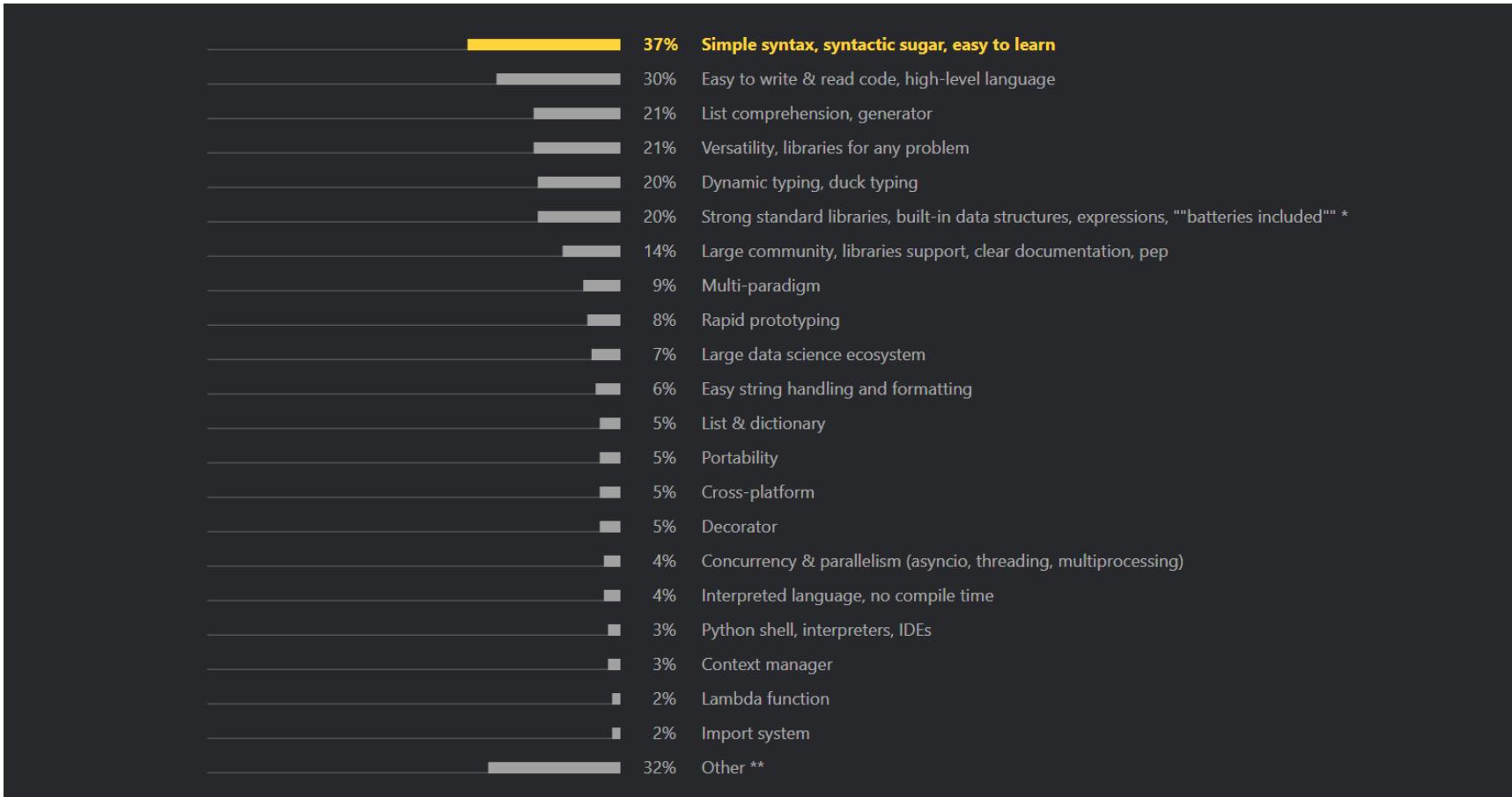
## C

```
#include <stdio.h>  
main(){  
    printf("Hello Dunia");  
}
```

## Python

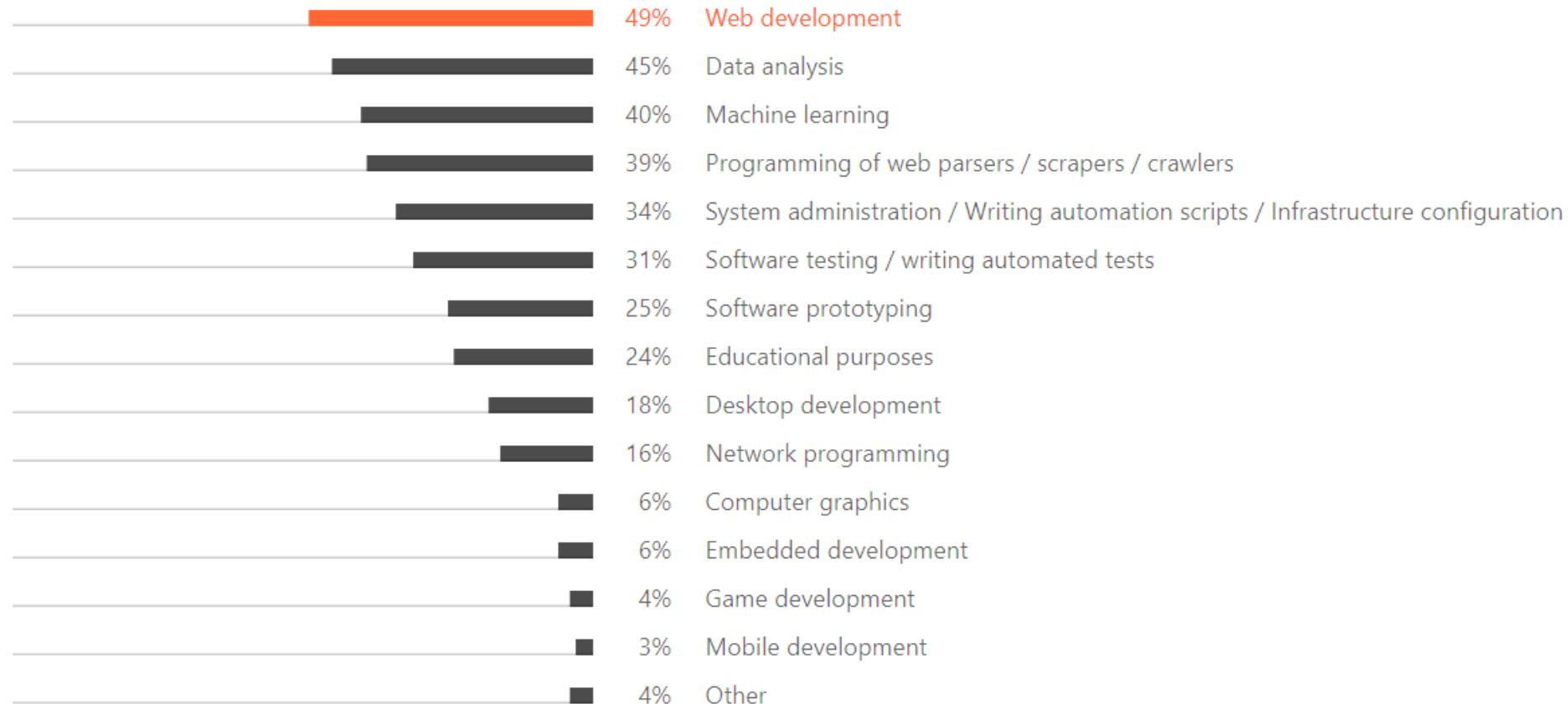
```
print("Hello Dunia")
```

# Most liked Python features



Sumber Gambar: <https://www.jetbrains.com/lp/python-developers-survey-2020/>

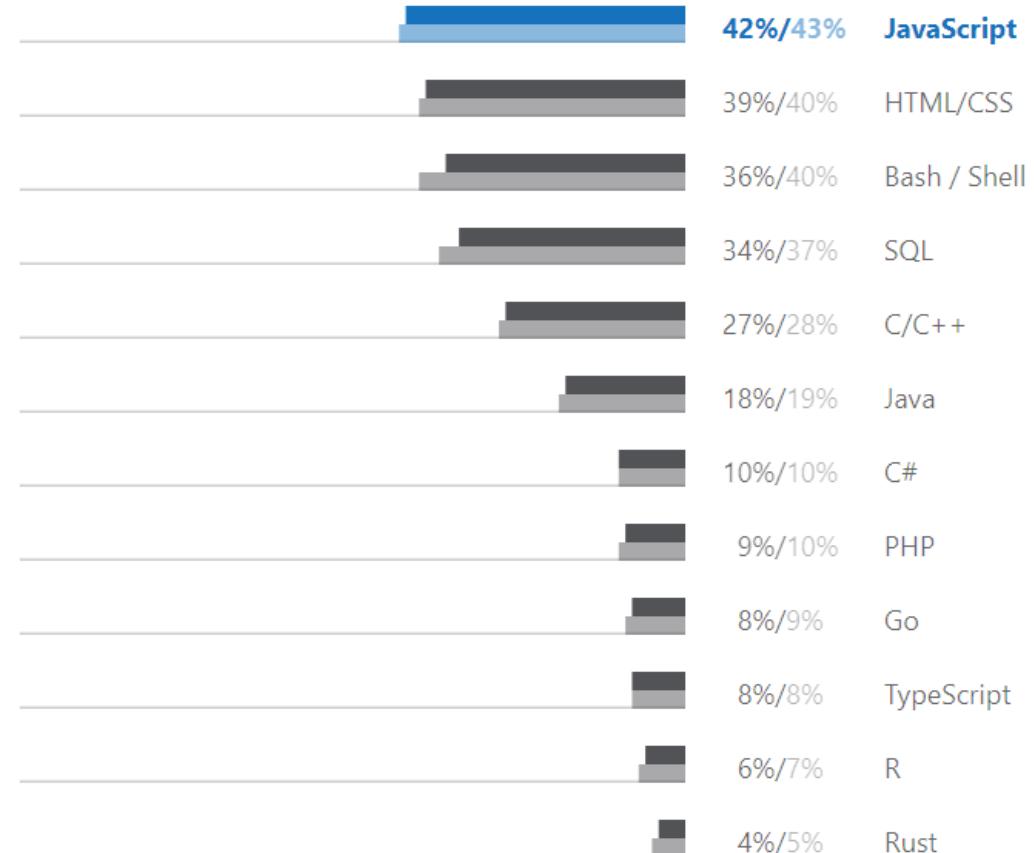
# Python Usage



Sumber Gambar: <https://www.jetbrains.com/lp/devecosystem-2021/python/>

python merupakan bahasa pemrograman paling populer untuk penelitian akademis.

# Python Usage



Sumber Gambar: <https://www.jetbrains.com/lp/python-developers-survey-2020/>



**“Python**  
*is an experiment in how much freedom  
programmers need. Too much freedom and  
nobody can read another’s code; too little and  
expressiveness is endangered”*

-Guido van Rossum -

Sumber Gambar: <https://gvanrossum.github.io/>

# “Python”?

# A Brief History of Python



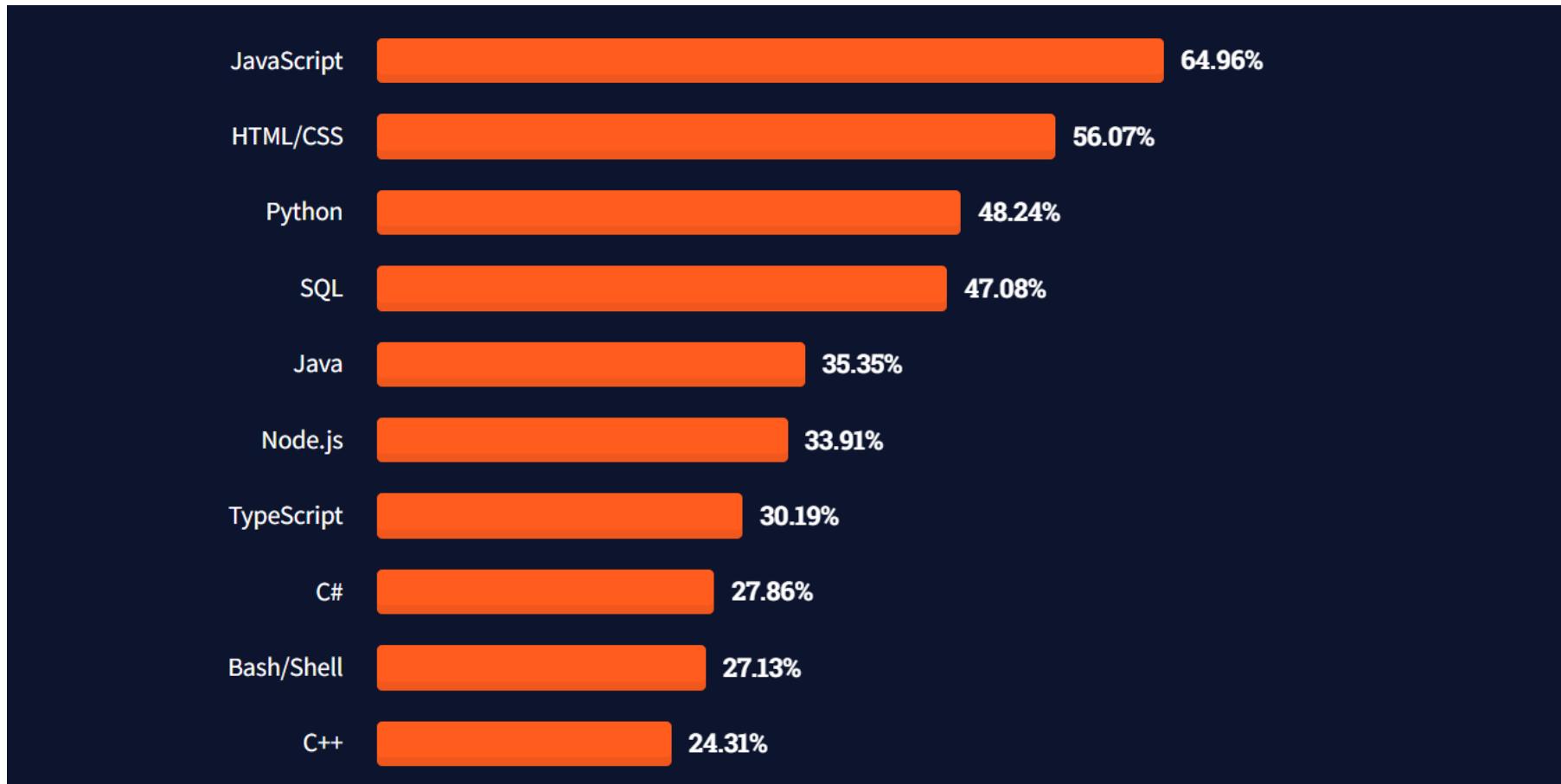
Nama 'python' berasal dari sebuah serial komedi BBC dari tahun 1970-an, yakni **Monty Python's Flying Circus**. Guido menganggap nama 'python' adalah nama yang singkat, unik, dan sedikit misterius.

Sumber Gambar: <https://www.imdb.com/title/tt0063929/>



Beberapa perusahaan besar yang menggunakan python

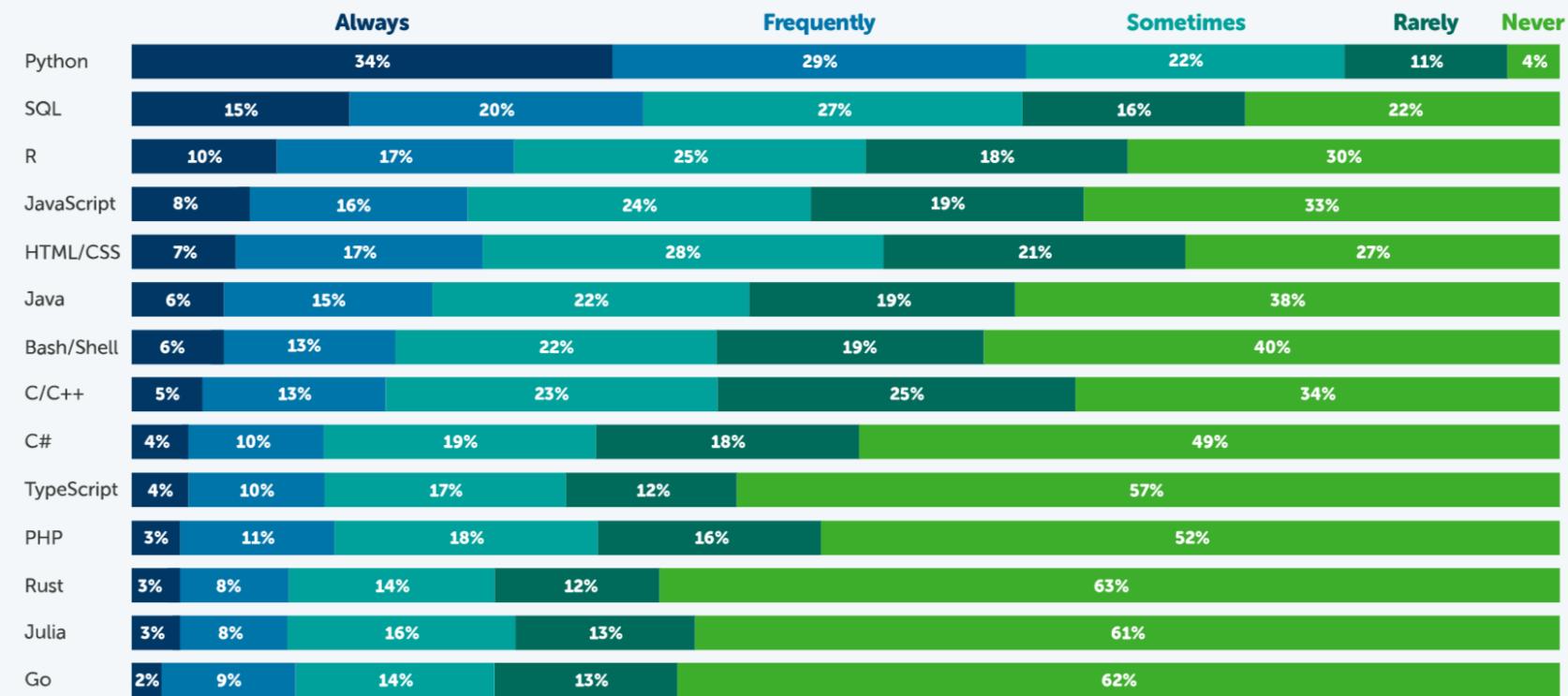
# How Popular is Python?



Sumber Gambar: <https://insights.stackoverflow.com/survey/2021>

# How Popular is Python?

63% of respondents said they always or frequently use Python, making it the most popular language included in this year's survey. In addition, 71% of educators are teaching Python, and 88% of students reported being taught Python in preparation to enter the data science/ML field.



Sumber Gambar: <https://www.anaconda.com/state-of-data-science-2021>

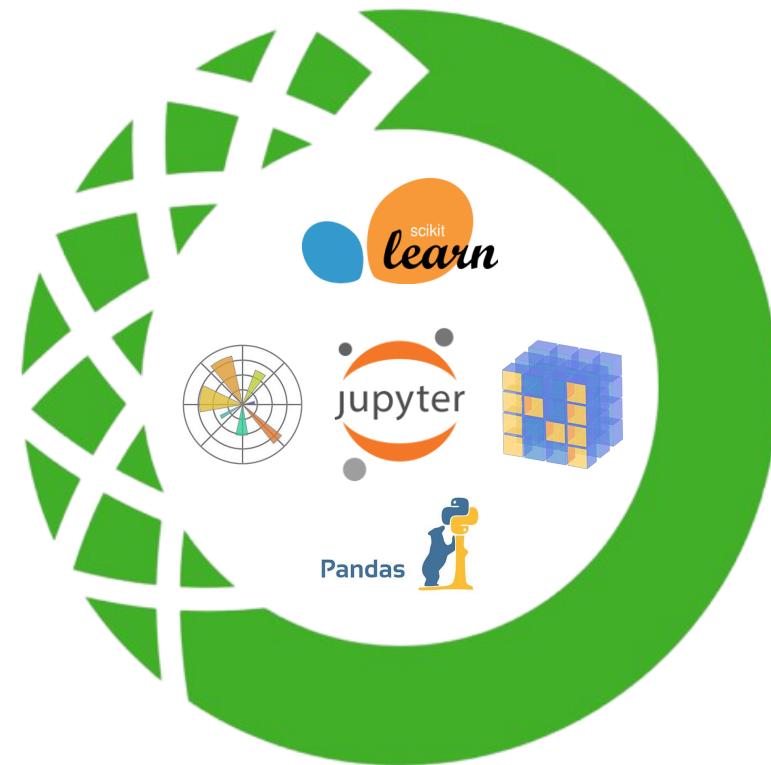
# What is Anaconda?



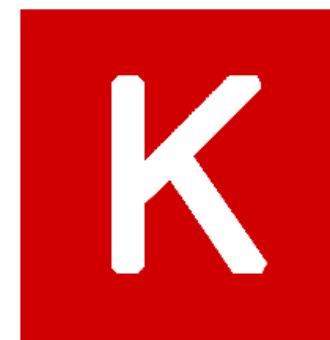
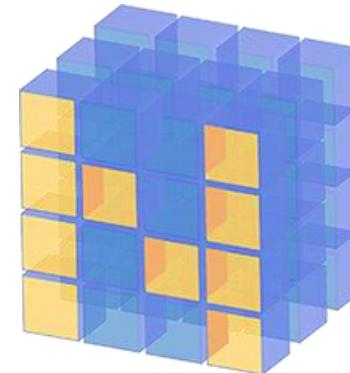
Anaconda adalah distribusi python/R, kumpulan package open-source, *package manager*, dan *environment manager* yang dibangun untuk data science.



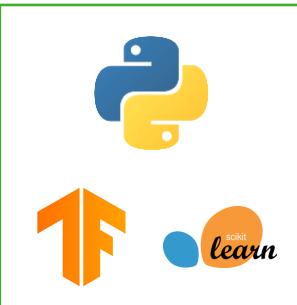
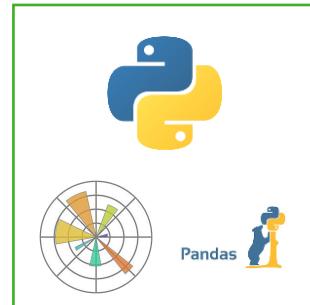
# Anaconda



# Popular Python Packages







Terdapat 4 metode yang dapat digunakan untuk menulis dan menjalankan program Python, antara lain:

1. Terminal
2. Terminal + Text Editor
3. IDE
4. IDE untuk penelitian dan pengembangan

# Terminal

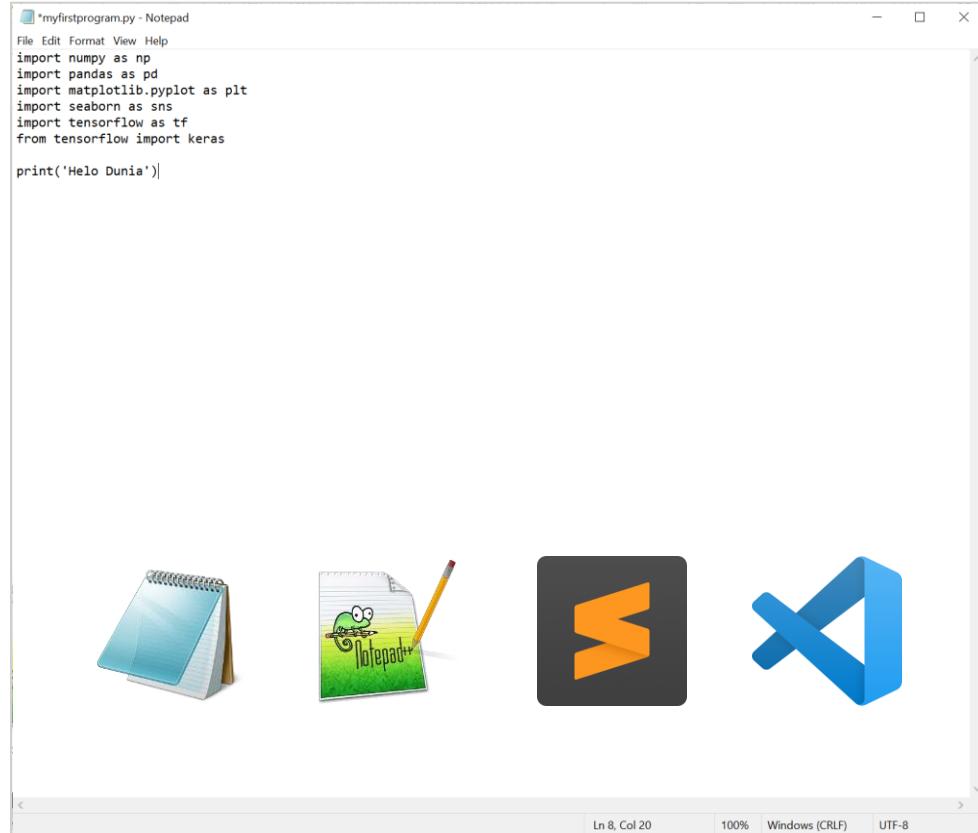
```
Command Prompt - python
Microsoft Windows [Version 10.0.19042.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\yusup>python
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello Dunia')
Hello Dunia
>>>
```

# Terminal + Text Editor

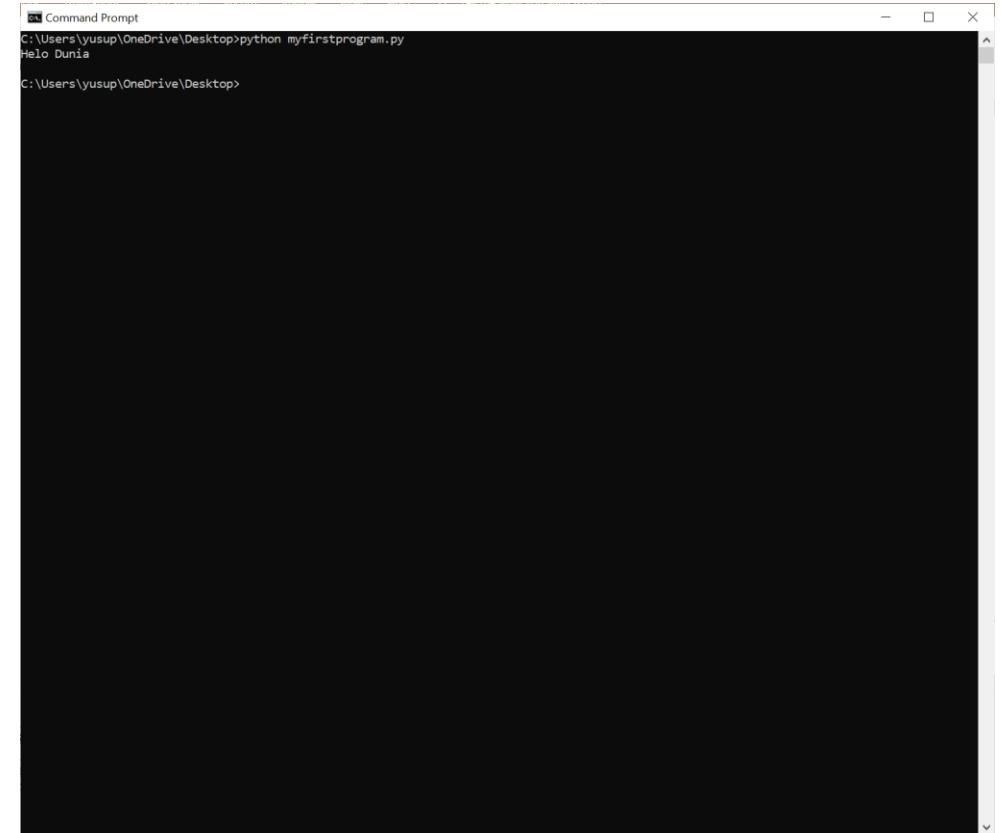


\*myfirstprogram.py - Notepad

```
File Edit Format View Help
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow import keras

print('Hello Dunia')
```

The screenshot shows a Windows Notepad window titled "myfirstprogram.py - Notepad". The code in the editor is a simple Python script that prints "Hello Dunia" to the console. Below the editor, there are four icons representing different text editors: a spiral notebook, a green document with a pencil, the Notepad+ logo, and the Visual Studio Code logo.



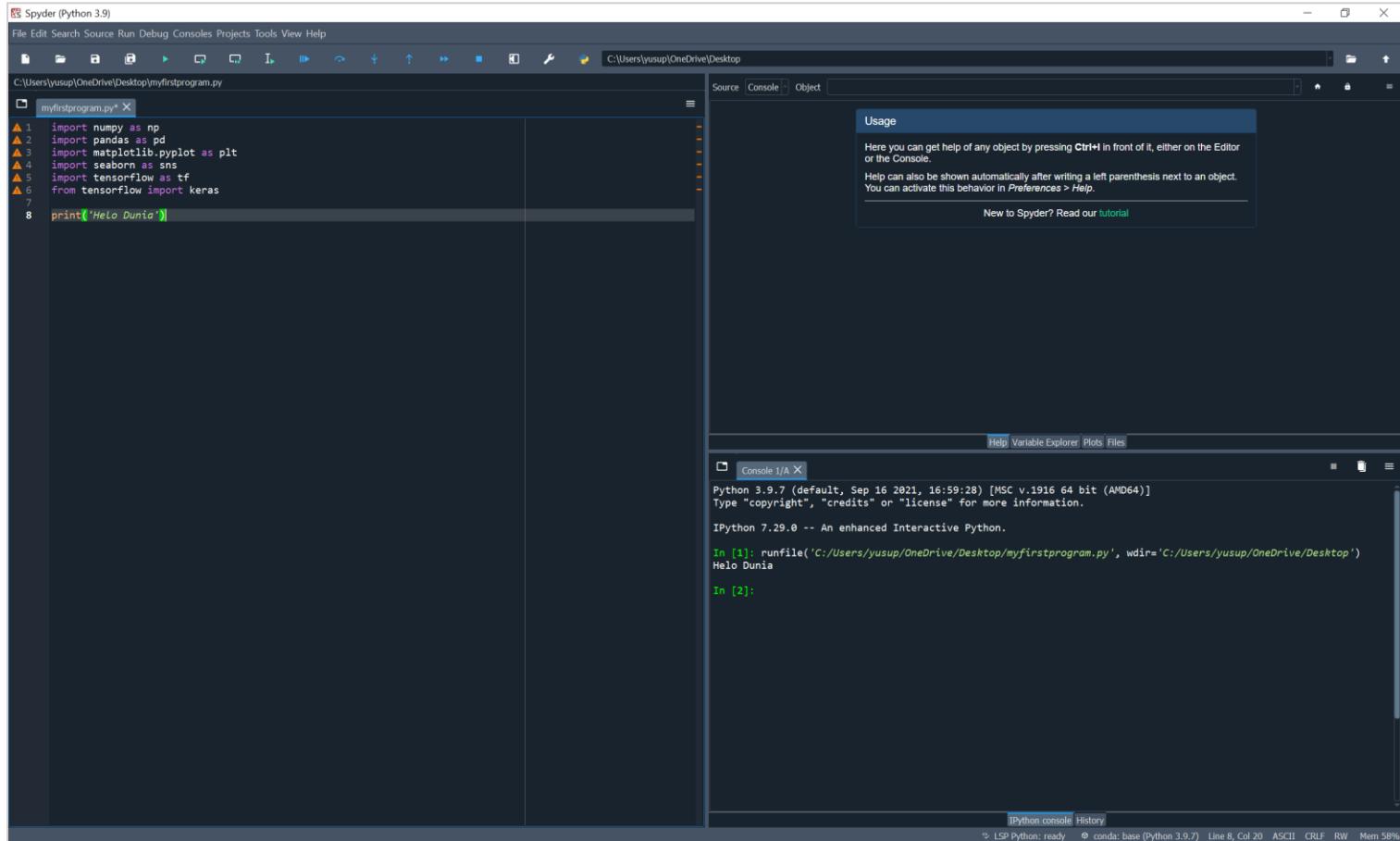
Command Prompt

```
C:\Users\yusup\OneDrive\Desktop>python myfirstprogram.py
Hello Dunia

C:\Users\yusup\OneDrive\Desktop>
```

The screenshot shows a Windows Command Prompt window titled "Command Prompt". The user has run the command "python myfirstprogram.py", and the output is "Hello Dunia", which was printed by the Python script in the Notepad window above.

# IDE (Integrated Development Environment)



# IDE for Research and Development

The screenshot shows a Jupyter Notebook interface with the title bar "myfirstprogram - Jupyter Noteb: x +". The URL "localhost:8888/notebooks/OneDrive/Desktop/myfirstprogram.ipynb" is visible. The notebook contains two code cells:

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow import keras
```

**Import Library**

```
In [2]: print('Hello Dunia')
```

**Cetak 'Hello Dunia'**

```
Hello Dunia
```

In [ ]: |

# What is Jupyter Notebook?



Jupyter Notebook adalah salah satu IDE untuk penelitian dan pengembangan, di bidang data science, yang berbasis web. IDE ini bersifat open sources.



Jupyter Notebook tidak hanya bekerja seperti IDE kabanyakan, namun juga mampu digunakan untuk presentasi dan edukasi.

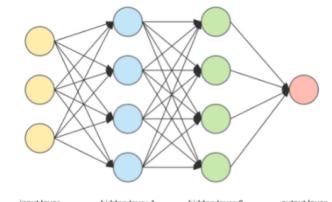
Jupyter Notebook mendukung **markdown**. Hal ini membuat kita dapat menambahkan heading, paragraph, table, dan image ke dalam program kita melalui komponen HTML.

jupyter myfirstprogram Last Checkpoint: 12 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Presentasi Tentang Deep Learning



Sumber Gambar: [https://miro.medium.com/proxy/1\\*Gh5PS4R\\_A5drI5ebd\\_gNrg@2x.png](https://miro.medium.com/proxy/1*Gh5PS4R_A5drI5ebd_gNrg@2x.png)

**Deep Learning**

Merupakan bagian dari machine learning yang menggunakan *Deep Neural Networks* untuk menyelesaikan permasalahan pada domain machine learning. Dasar dari deep learning adalah neural network, yang digabungkan untuk membentuk jaringan saraf yang dalam.

**Macam-Macam Arsitektur Deep Learning**

Arsitektur Deep Learning	Penjelasan Singkat
Convolutional Neural Network (CNN)	Sangat cocok digunakan pada data 2 dimensi
Recurrent Neural Network (RNN)	Memiliki kemampuan untuk mempelajari urutan

**Import Library yang Dibutuhkan**

```
In [1]:  
import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
import seaborn as sns  
import tensorflow as tf  
from tensorflow import keras
```

jupyter myfirstprogram Last Checkpoint: 12 minutes ago (unsaved changes) Not Trusted Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help

Presentasi Tentang Deep Learning

Sumber Gambar: [https://miro.medium.com/proxy/1\\*Gh5PS4R\\_A5drI5ebd\\_gNrg@2x.png](https://miro.medium.com/proxy/1*Gh5PS4R_A5drI5ebd_gNrg@2x.png)

Deep Learning

Merupakan bagian dari machine learning yang menggunakan *Deep Neural Networks* untuk menyelesaikan permasalahan pada domain machine learning. Dasar dari deep learning adalah neural network, yang digabungkan untuk membentuk jaringan saraf yang dalam.

Macam-Macam Arsitektur Deep Learning

Arsitektur Deep Learning	Penjelasan Singkat
Convolutional Neural Network (CNN)	Sangat cocok digunakan pada data 2 dimensi
Recurrent Neural Network (RNN)	Memiliki kemampuan untuk mempelajari urutan

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from tensorflow import keras
```

Umumnya, program yang ditulis menggunakan Jupyter Notebook akan disimpan dalam file dengan format .ipynb.

Tetapi, kita juga menyimpan program yang kita tulis dalam format .py apabila dibutuhkan.

Name	Date modified	Type	Size
myfirstprogram.ipynb	09/02/2022 18:21	Jupyter Source File	4 KB
myfirstprogram.py	06/02/2022 9:19	Python Source File	1 KB

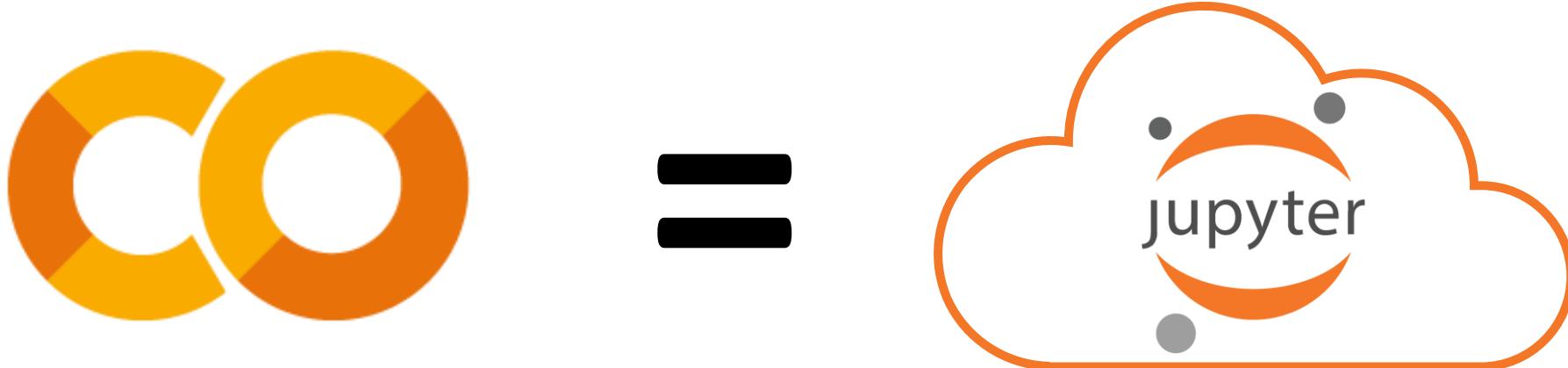
# Anaconda Installation

**Download Anaconda:**

<https://www.anaconda.com/products/individual>

# Jupyter Notebook User Interface

# What is Colab?



myfirstprogram.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings RAM Disk Editing

Presentasi Tentang Deep Learning

input layer      hidden layer 1      hidden layer 2      output layer

Sumber Gambar: [https://miro.medium.com/proxy/1\\*Gh5PS4R\\_A5drI5ebd\\_gNrg@2x.png](https://miro.medium.com/proxy/1*Gh5PS4R_A5drI5ebd_gNrg@2x.png)

Deep Learning

Merupakan bagian dari machine learning yang menggunakan *Deep Neural Networks* untuk menyelesaikan permasalahan pada domain machine learning. Dasar dari deep learning adalah neural network, yang digabungkan untuk membentuk jaringan saraf yang dalam.

Macam-Macam Arsitektur Deep Learning

Arsitektur Deep Learning	Penjelasan Singkat
Convolutional Neural Network (CNN)	Sangat cocok digunakan pada data 2 dimensi
Recurrent Neural Network (RNN)	Memiliki kemampuan untuk mempelajari urutan

Import Library yang Dibutuhkan

```
[1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
#from tensorflow import keras
```

0s completed at 11:15 AM

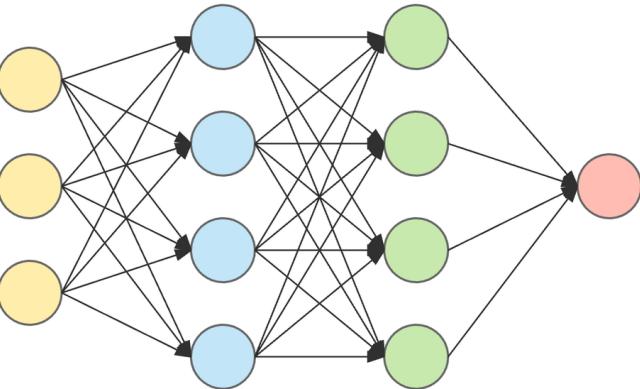
myfirstprogram.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Settings RAM Disk Editing

Presentasi Tentang Deep Learning

Heading



Image

input layer      hidden layer 1      hidden layer 2      output layer

Sumber Gambar: [https://miro.medium.com/proxy/1\\*Gh5PS4R\\_A5dr15ebd\\_gNrg@2x.png](https://miro.medium.com/proxy/1*Gh5PS4R_A5dr15ebd_gNrg@2x.png)

Deep Learning

Merupakan bagian dari machine learning yang menggunakan *Deep Neural Networks* untuk menyelesaikan permasalahan pada domain machine learning. Dasar dari deep learning adalah neural network, yang digabungkan untuk membentuk jaringan saraf yang dalam.

Paragraph

Macam-Macam Arsitektur Deep Learning

Arsitektur Deep Learning	Penjelasan Singkat
Convolutional Neural Network (CNN)	Sangat cocok digunakan pada data 2 dimensi
Recurrent Neural Network (RNN)	Memiliki kemampuan untuk mempelajari urutan

Table

Import Library yang Dibutuhkan

```
[1]: import numpy as np
       import pandas as pd
       import matplotlib.pyplot as plt
       import seaborn as sns
       #import tensorflow as tf
       #from tensorflow import keras
```

Code

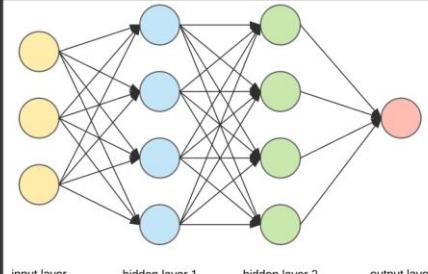
12.03

lab.research.google.com

myfirstprogram.ipynb

RAM Disk

## Presentasi Tentang Deep Learning



input layer      hidden layer 1      hidden layer 2      output layer

Sumber Gambar:  
[https://miro.medium.com/proxy/1\\*Gh5PS4RA5drI5ebd\\_gNrg@2x.png](https://miro.medium.com/proxy/1*Gh5PS4RA5drI5ebd_gNrg@2x.png)

## Deep Learning

Merupakan bagian dari machine learning yang menggunakan *Deep Neural Networks* untuk menyelesaikan permasalahan pada domain machine learning. Dasar dari deep learning adalah neural network, yang digabungkan untuk membentuk jaringan saraf yang dalam.

Macam-Macam Arsitektur Deep Learning

Arsitektur Deep Learning	Penjelasan
Convolutional Neural Network (CNN)	Sangat cocok digunakan untuk gambar
Recurrent Neural Network (RNN)	Memiliki kemampuan memori

Import Library yang Dibutuhkan

```
[1] import numpy as np
    import pandas as pd
    #import matplotlib.pyplot as plt
    #import seaborn as sns
    #import tensorflow as tf
    #from tensorflow import keras
```

12.03

lab.research.google.com

myfirstprogram.ipynb

RAM Disk

## Deep Learning

Merupakan bagian dari machine learning yang menggunakan *Deep Neural Networks* untuk menyelesaikan permasalahan pada domain machine learning. Dasar dari deep learning adalah neural network, yang digabungkan untuk membentuk jaringan saraf yang dalam.

Macam-Macam Arsitektur Deep Learning

Arsitektur Deep Learning	Penjelasan
Convolutional Neural Network (CNN)	Sangat cocok digunakan untuk gambar
Recurrent Neural Network (RNN)	Memiliki kemampuan memori

Import Library yang Dibutuhkan

12.04

lab.research.google.com

myfirstprogram.ipynb

RAM Disk

## Import Library yang Dibutuhkan

```
[1] import numpy as np
    import pandas as pd
    #import matplotlib.pyplot as plt
    #import seaborn as sns
    #import tensorflow as tf
    #from tensorflow import keras
```

## Cetak 'Hello Dunia'

```
[2] print('Hello Dunia')
```

Hello Dunia

# Jupyter Notebook vs Colab



- Tidak membutuhkan akses internet
- Hanya bisa diakses melalui computer
- Tidak bisa share code untuk kolaborasi
- Perlu melakukan konfigurasi
- Terhubung dengan local storage komputer
- Sangat bergantung pada hardware computer yang dimiliki

- Membutuhkan akses internet
- Bisa diakses melalui berbagai device
- Bisa share code untuk kolaborasi
- Tidak perlu melakukan konfigurasi
- Terhubung dengan penyimpanan google drive
- Menggunakan hardware yang telah disediakan google

# Colab User Interface

---

**Colab:** [https://colab.research.google.com/?utm\\_source=scs-index](https://colab.research.google.com/?utm_source=scs-index)

# My First Program//Activity

# What is Program?

Program adalah urutan instruksi (*code*) yang menentukan bagaimana cara komputer melakukan komputasi.

Komputasi di sini dapat berupa:

- Sesuatu yang berhubungan dengan matematika
- Komputasi simbolis

# What is Programming?

Programming adalah aktivitas menulis intruksi (*code*) untuk dijalankan oleh komputer. Contoh instruksi seperti:

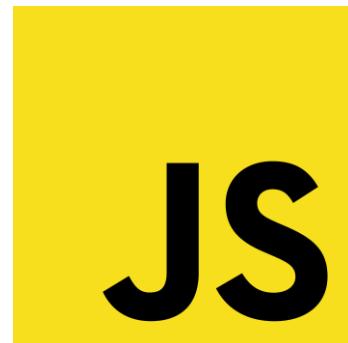
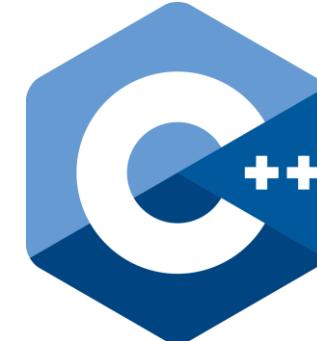
- Mencetak “Hello World”
- Mengambil data dari internet (*scraping*)
- Membaca isi file

Code dapat ditulis menggunakan bahasa pemrograman (*programming language*).

Programming language adalah alat/tools yang digunakan untuk menulis code.

Programming language tersusun atas serangkaian simbol yang berfungsi menerjemahkan pikiran kita menjadi instruksi yang dapat dipahami komputer.

# Programming Language



Dahulu, programming sulit dilakukan, karena menggunakan bahasa pemrograman tingkat rendah (*low-level programming language*), seperti bahasa assembly.

Code lebih dekat ke kode mesin (biner: 0 dan 1) ketika ditulis menggunakan low-level language.

# High-Level Programming Language

Pada saat ini, umumnya code ditulis menggunakan bahasa pemrograman tingkat tinggi (*high-level programming language*), seperti Python, Java, C, C++, dll.

Code lebih dekat ke bahasa manusia (English) ketika ditulis menggunakan *high-level language*.

## Low-Level Programming Language

```
global _main
extern _printf

section .text
_main:
    push message
    call _printf
    add esp, 4
    ret
message:
    db 'Hello, World', 10, 0
```

## High-Level Programming Language

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

Sumber: <https://qastack.id/programming/1023593/how-to-write-hello-world-in-assembler-under-windows>

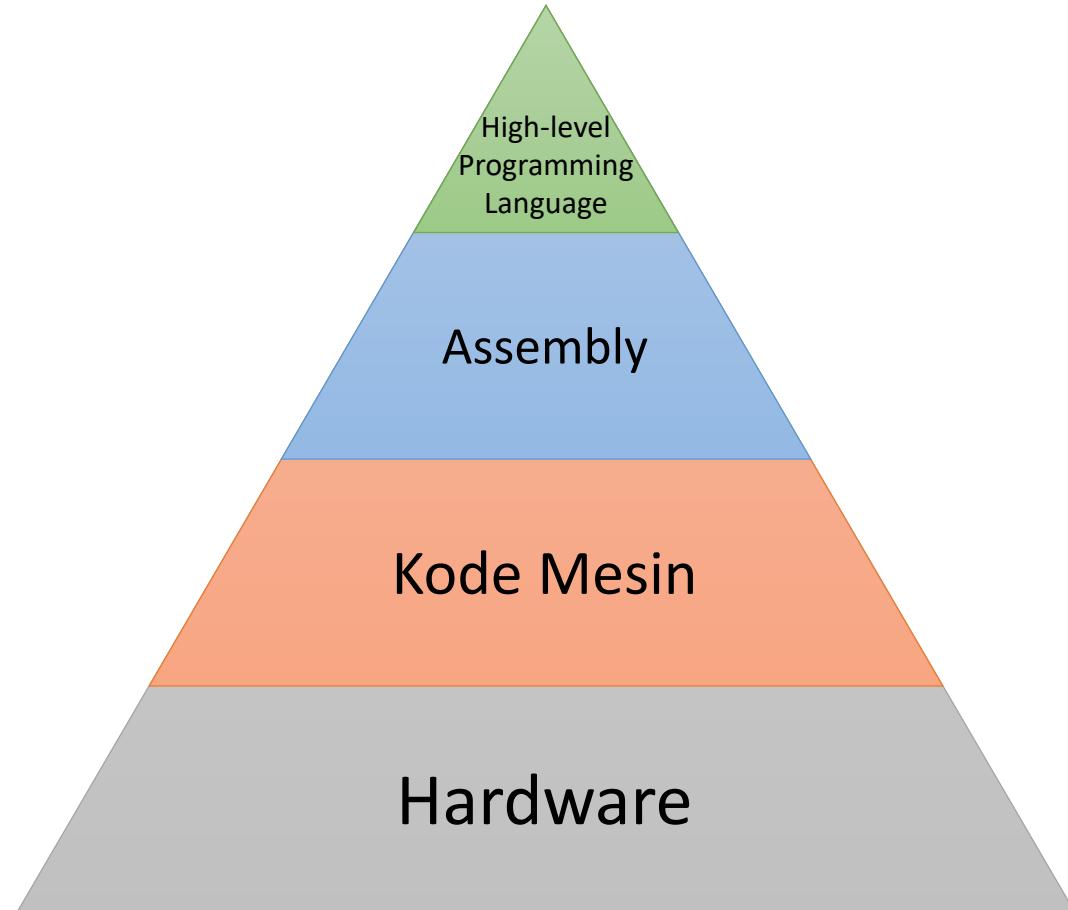
# How Programming Languages Work

Programming language dibangun di atas abstraksi. Abstraksi adalah konsep untuk mengelola kompleksitas. Contoh:

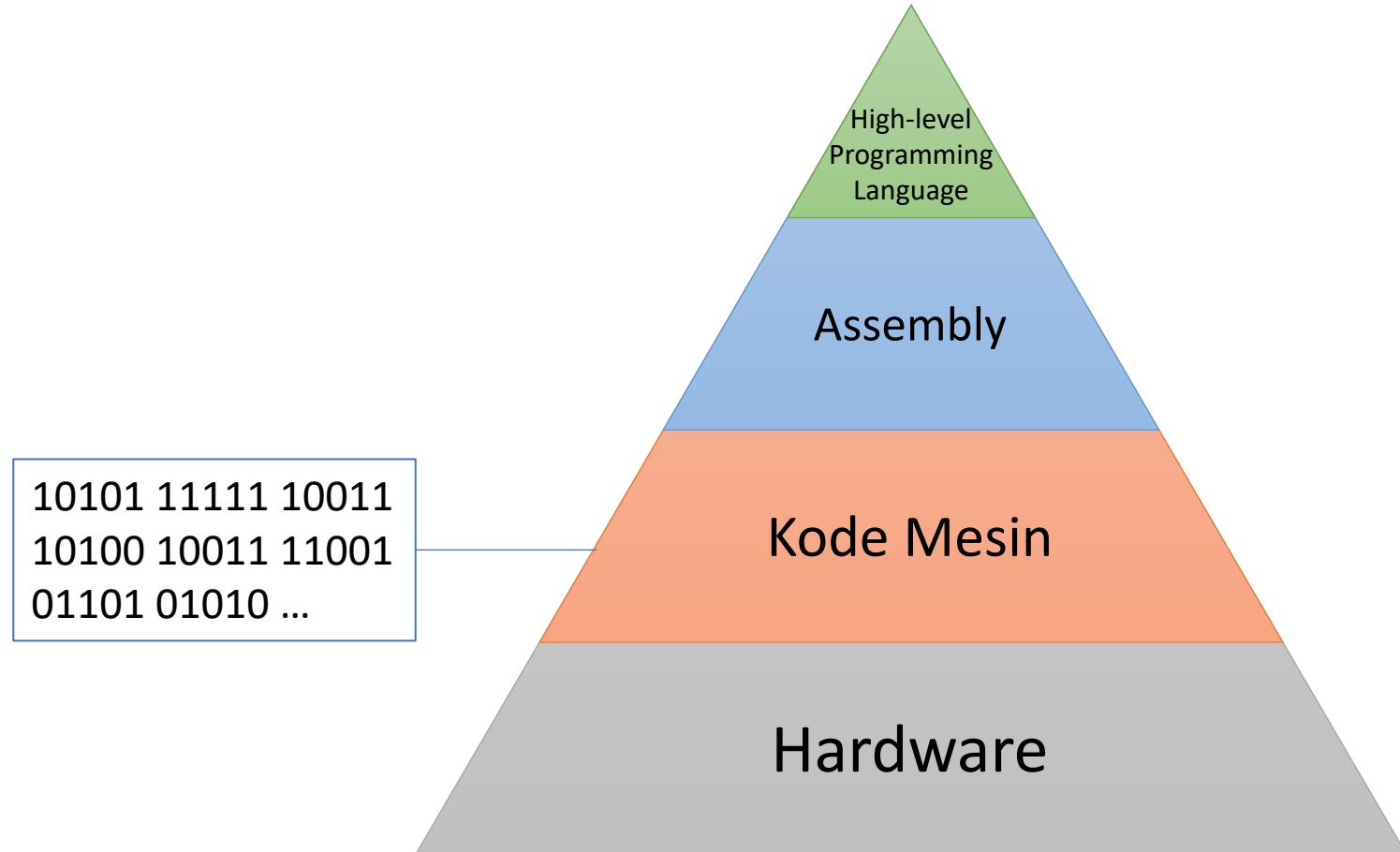
Ketika menulis program Python, kita tidak perlu tahu bagaimana cara Python berinteraksi dengan operating system (OS), bagaimana cara OS berinteraksi dengan hardware, dan bagaimana cara hardware mengeksekusi binary.

Pada dasarnya, kita melakukan programming di atas beberapa layer abstraksi.

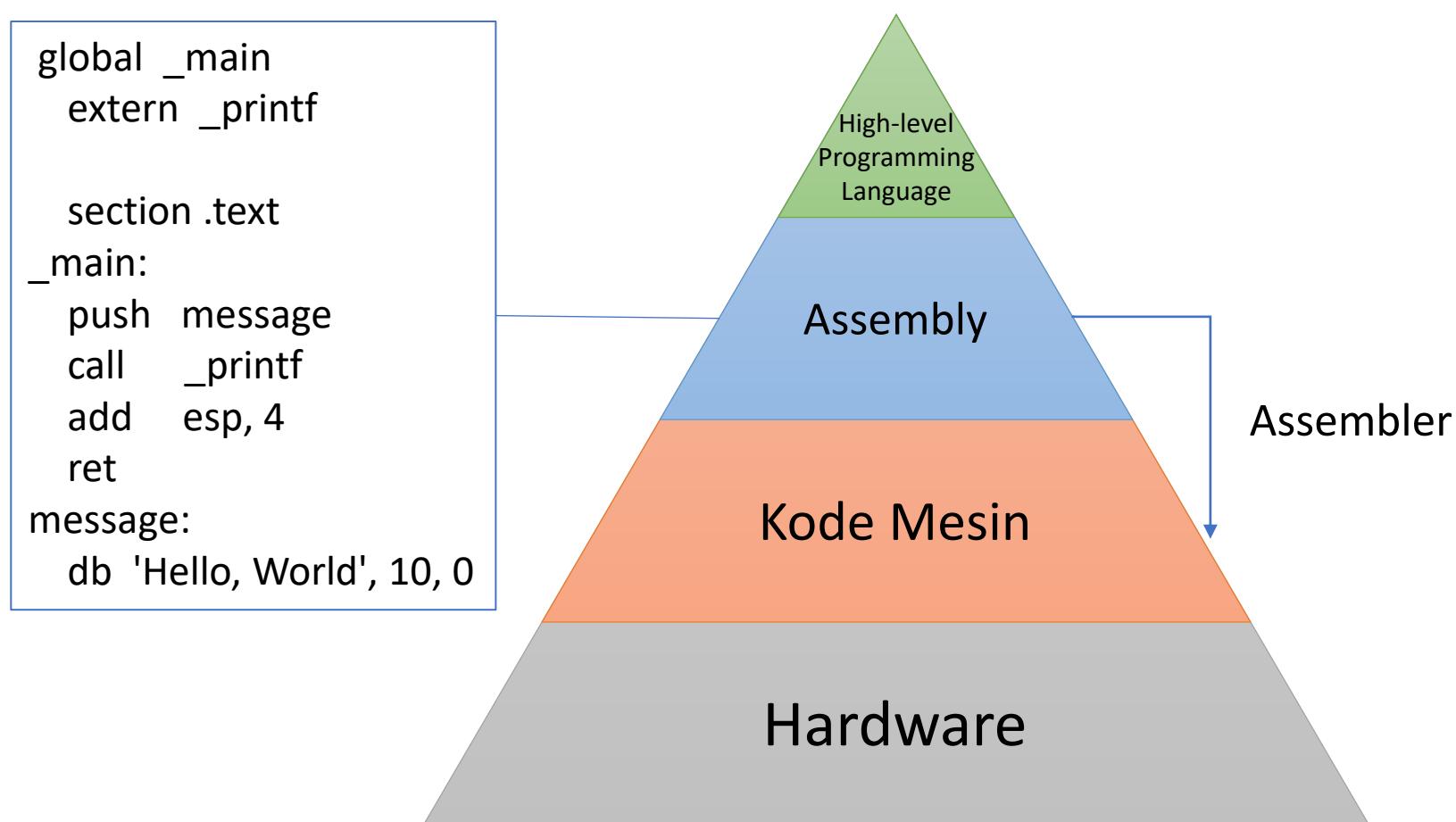
# How Programming Languages Work



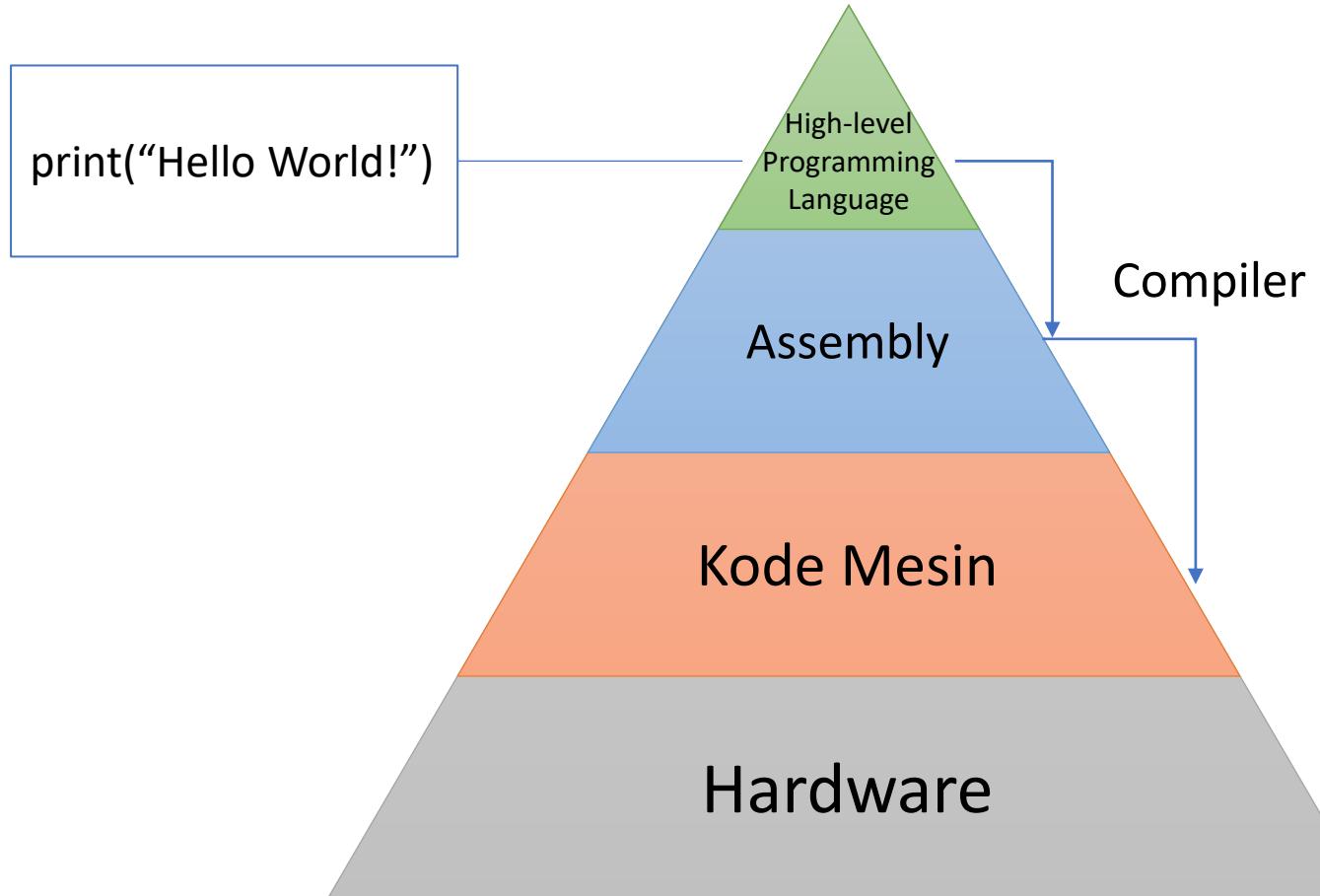
# How Programming Languages Work



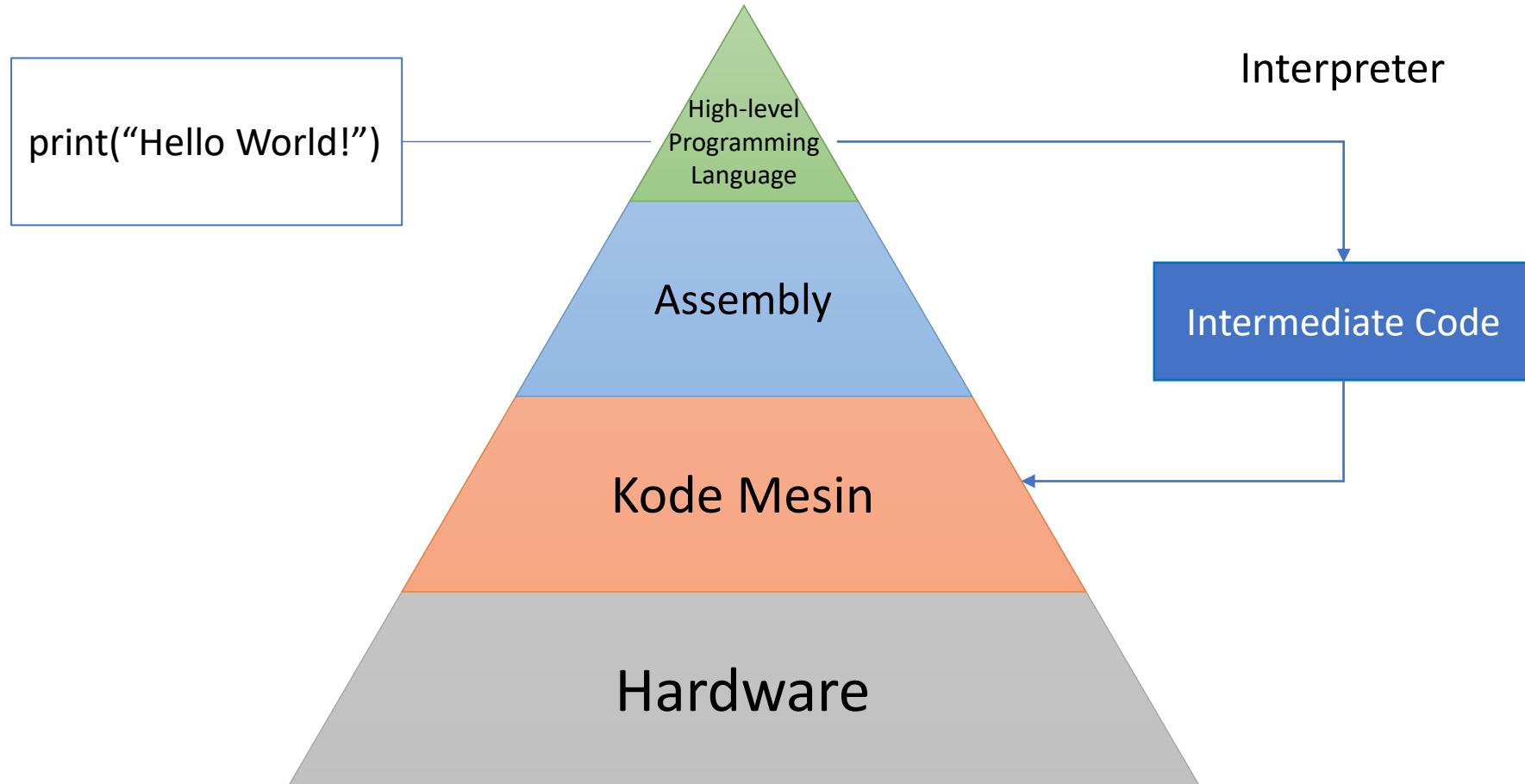
# How Programming Languages Work



# How Programming Languages Work



# How Programming Languages Work



Semakin tinggi level programming language, maka:

- semakin jauh dari kode mesin;
- semakin banyak abstraksi di bawahnya; dan
- semakin lambat berjalan.

- Effective Computation in Physics oleh Anthony Scopatz & Kathryn D. Huff
- Think Python - How to Think Like a Computer Scientist, 2<sup>nd</sup> Edition, Version 2.4.0, oleh Allen Downey
- The Self-taught Programmer oleh Cory Althoff



```
from heart import gratitude
```

```
x = "bye"
```

```
y = "until we meet again"
```

```
Thank_You = gratitude(feeling=[happy, super, hopeful]).fit(x, y)
```

```
print(Thank_You)
```

# Module 1

Introduction to everything

## Section

Data, Ethics, and Bias

# Apa Itu Data ?

Data bisa berupa banyak hal.

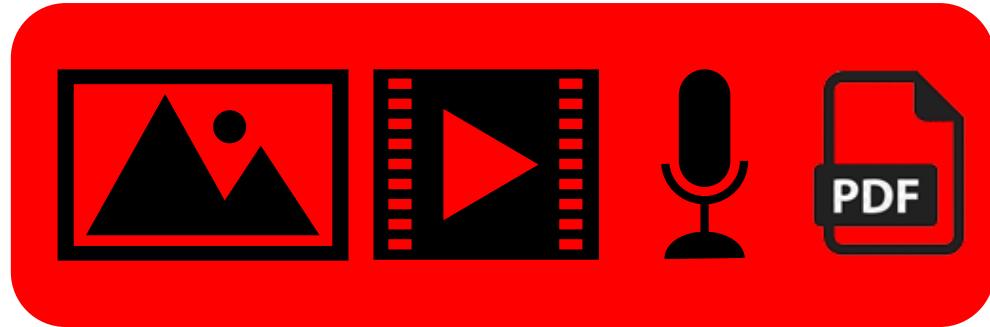
Dengan Kecerdasan Buatan itu harus menjadi kumpulan fakta:

Tipe	Contoh
Nomor	Harga Angka. Tanggal.
Ukuran	Pengukuran. Tinggi. Erat.
Perkataan	Nama dan Tempat.
Pengamatan	Menghitung Mobil.
Deskripsi	Ini Dingin, Ini Panas.

## STRUCTURED DATA



## UNSTRUCTURED DATA



**“Data terstruktur direpresentasikan dalam skema yang jelas sehingga mudah untuk dianalisa maupun diintegrasikan dengan data terstruktur lainnya. Sedangkan data tidak terstruktur direpresentasikan dalam berbagai bentuk sehingga sangat sulit untuk dianalisis maupun diintegrasikan dengan sumber data lain”**



C. Afifanto (2015). Integrasi Data Terstruktur dan Tidak Terstruktur dalam Sistem Inteligensi Bisnis Integrasi Data Terstruktur dan Tidak Terstruktur dalam Sistem.  
<https://doi.org/10.13140/RG.2.1.3251.8242>

## Structured Data vs Unstructured Data

Can be displayed  
in rows, columns and  
relational databases



Numbers, dates  
and strings



Estimated 20% of  
enterprise data (Gartner)



Requires less storage



Easier to manage  
and protect with  
legacy solutions



Cannot be displayed  
in rows, columns and  
relational databases



Images, audio, video,  
word processing files,  
e-mails, spreadsheets



Estimated 80% of  
enterprise data (Gartner)



Requires more storage



More difficult to  
manage and protect  
with legacy solutions



## STRUCTURED DATA

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEART T-LIGHT HO	6	01/12/2010 08:26	2,55 17850		United Kingdom
536365	71053	WHITE METAL LANTERN	6	01/12/2010 08:26	3,39 17850		United Kingdom
536365	844068	CREAM CUPID HEARTS COAT HANGE	8	01/12/2010 08:26	2,75 17850		United Kingdom
536365	84029G	KNITTED UNION FLAG HOT WATER B	6	01/12/2010 08:26	3,39 17850		United Kingdom
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	01/12/2010 08:26	3,39 17850		United Kingdom
536365	22752	SET 7 BABUSHKA NESTING BOXES	2	01/12/2010 08:26	7,65 17850		United Kingdom
536365	21790	GLASS STAR FROSTED T-LIGHT HOLDE	6	01/12/2010 08:26	4,25 17850		United Kingdom
536366	22633	HAND WARMER UNION JACK	6	01/12/2010 08:28	1,85 17850		United Kingdom
536366	22632	HAND WARMER RED POLKA DOT	6	01/12/2010 08:28	1,85 17850		United Kingdom
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	01/12/2010 08:34	1,69 13047		United Kingdom
536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	01/12/2010 08:34	2,1 13047		United Kingdom
536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	01/12/2010 08:34	2,1 13047		United Kingdom
536367	22749	FELTCRAFT PRINCESS CHARLOTTE D	8	01/12/2010 08:34	3,75 13047		United Kingdom
536367	22310	IVORY KNITTED MUG COSY	6	01/12/2010 08:34	1,65 13047		United Kingdom
536367	84969	BOX OF 6 ASSORTED COLOUR TEASP	6	01/12/2010 08:34	4,25 13047		United Kingdom
536367	22623	BOX OF VINTAGE JIGSAW BLOCKS	3	01/12/2010 08:34	4,95 13047		United Kingdom

## UNSTRUCTURED DATA



# Kecerdasan butuh Data

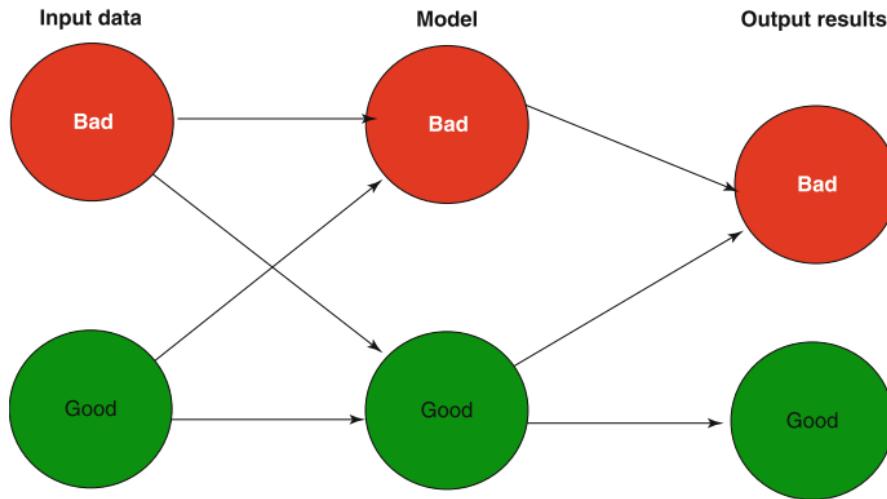


## Kecerdasan manusia membutuhkan data:

Broker real estate membutuhkan data tentang rumah yang dijual untuk memperkirakan harga.

## Kecerdasan buatan membutuhkan data:

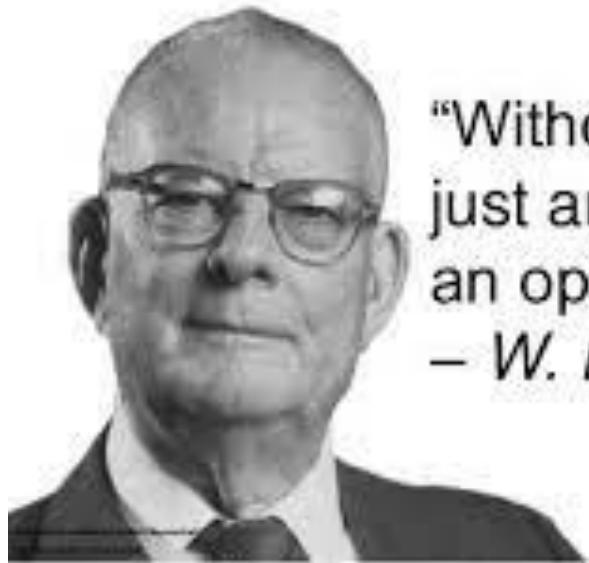
Program komputer juga membutuhkan data untuk memperkirakan harga.



### ***GIGO (Garbage In, Garbage Out) paradigm***



Naqa, Issam El, Ruijiang Li, & Murphy, M. J. (2015). Machine Learning in Radiation Oncology. (I. El Naqa, R. Li, & M. J. Murphy, Eds.), Machine Learning in Radiation Oncology. Cham: Springer International Publishing.  
<https://doi.org/10.1007/978-3-319-18305-3>



“Without data you’re  
just another person with  
an opinion.”  
– *W. Edwards Deming*

(Statistician)



## **Session II**

### **What is Data (2)**

## Kuantitatif vs Kualitatif

Data Kuantitatif adalah :

- 55 Mobil
- 15 Meter
- 35 Anak

Data Kualitatif adalah berdeskriptif :

- Hal ini dingin
- Hal ini panjang
- Itu menyenangkan



## Kualitas Data

Menurut Mark Mosley (2008), dalam bukunya “**Dictionary of Data Management**” kualitas data dilihat dari:

- **Accuracy** → Keakuratan data/ seberapa besar data bisa dipertanggungjawabkan
- **Completeness** → Kelengkapan data
- **Relevancy** → Memuat informasi yang dibutuhkan untuk analisa
- **Timelines** → Data mewakili informasi di waktu tersebut (jika membahas saat ini maka data harus yang terkini atau yang lama tapi masih relevan hingga saat ini)
- **Consistency** → Adanya kekonsistensian (data yang merepresentasikan informasi yang sama meski disimpan di tempat yang berbeda tetaplah sama)

# Cara Memperoleh Data (Populasi vs Sampel)



**Populasi** adalah sekelompok individu yang menjadi objek pengamatan/penelitian.

**Sensus** adalah aktivitas untuk mengumpulkan informasi tentang setiap individu dalam populasi.

Namun kenyataannya, sensus sulit dilakukan karena membutuhkan waktu dan biaya, sehingga dilakukan pengambilan informasi dari sebagian anggota populasi namun dapat merepresentasikan keseluruhannya

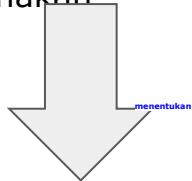
**Sampel** adalah himpunan bagian dari populasi yang cukup merepresentasikan keseluruhan populasi

# Cara Memperoleh Data (Sensus vs Survei)

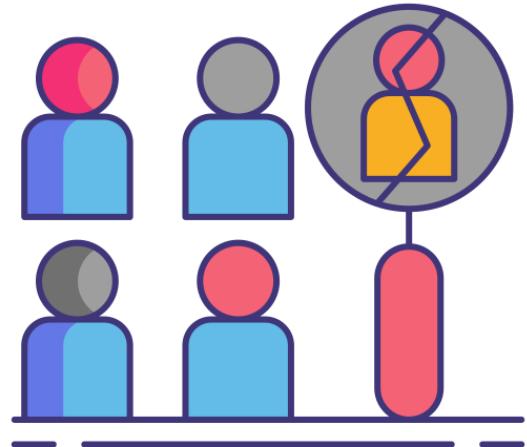
	<b>Sensus</b>	<b>Survei</b>
	Data yang dikumpulkan adalah data <b>populasi</b>	Data yang dikumpulkan adalah data <b>sampel</b>
Kelebihan	<ul style="list-style-type: none"><li>- Lebih akurat karena mencakup semua elemen</li><li>- Terhindar dari bias sampel</li></ul>	<ul style="list-style-type: none"><li>- Menghemat waktu dan biaya</li></ul>
kekurangan	<ul style="list-style-type: none"><li>- Membutuhkan biaya yang besar dan waktu yang lama</li><li>- Tidak bisa dilakukan untuk populasi yang mendekati “tak hingga”</li></ul>	<ul style="list-style-type: none"><li>- Hasil merupakan estimasi, bukan yang sebenarnya, sehingga lebih subjektif</li><li>- Bisa terjadi bias dalam pengumpulan sampel</li></ul>

# Apa saja yang perlu diperhatikan dalam pengambilan sampel?

1. Tujuan penelitian
2. Kondisi populasi
3. Biaya, waktu, dan tenaga yang tersedia
4. Metode analisa/pemodelan yang akan digunakan

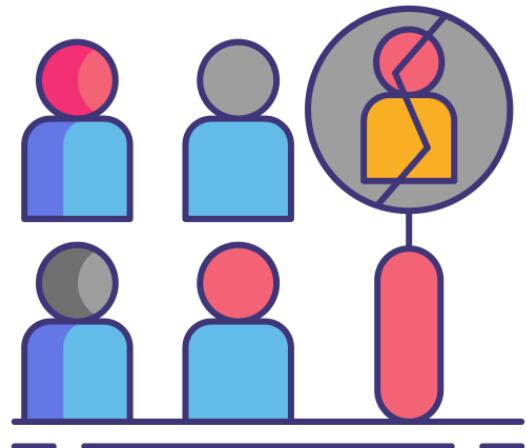


1. Ukuran Sampel
2. Kriteria Sampel
3. Teknik Sampling



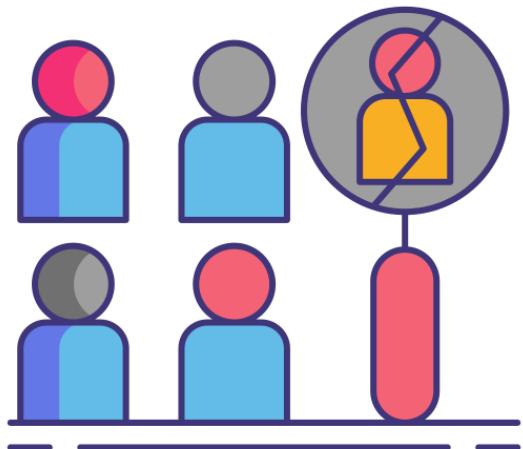
## Ukuran Sampel

- Semakin heterogen populasi, semakin besar sampel yang dibutuhkan
- Ada rumus tertentu yang bisa digunakan untuk menghitung minimal sampel (misalnya rumus slovin)
- Berdasarkan metode pemodelan yang akan digunakan: misalnya saat pake deep learning butuh data yang lebih banyak dari regresi linier



## Kriteria Sampel

- Disesuaikan dengan tujuan dan metode analisa/ pemodelan yang akan digunakan.
- Misalnya: ingin mengetahui pengaruh suatu perlakuan, maka perlu membandingkan sampel yang diberi perlakuan dan yang tidak.
- Berdasarkan metode analisa datanya, misalnya jika ingin memodelkan klasifikasi foto kucing atau anjing, maka diperlukan kriteria detail jenis kucing/anjing apa saja yang harus ada pada sampel (kalo hanya 1 jenis anjing saja tentu kurang baik dan menyebabkan overfitting). Tetapi jika hanya menguji signifikansi perbedaan keduanya tidak masalah jika 1 jenis saja

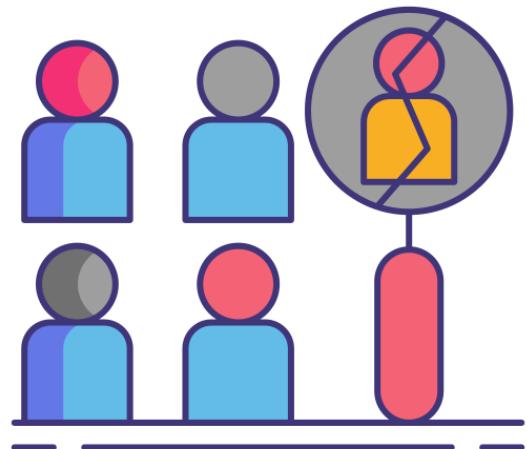


## Teknik Sampling

Ada beberapa jenis teknik sampling namun yang paling umum digunakan adalah:

### Sampel Acak (Random Sampling)

Sampel Acak, adalah sampel di mana setiap anggota populasi memiliki kesempatan yang sama untuk muncul dalam sampel.



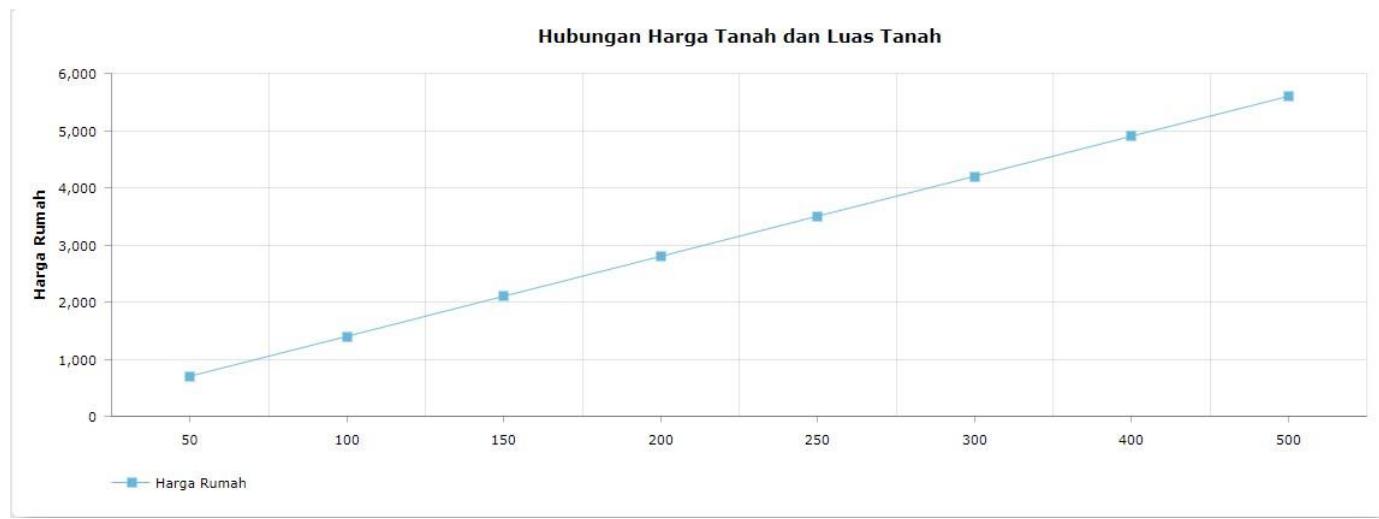
## Menyimpan Data

Data yang paling umum untuk dikumpulkan adalah Angka dan Pengukuran.

Seringkali data disimpan dalam array yang mewakili hubungan antara nilai.

Tabel ini berisi harga rumah versus ukuran

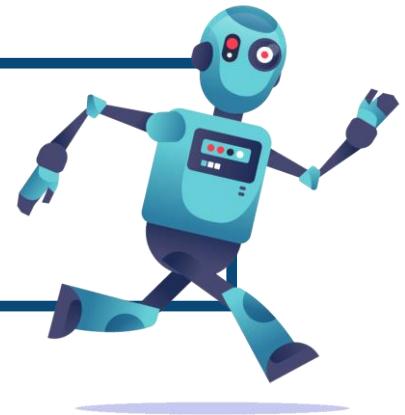
Harga (Juta)	700	1400	2100	2800	3500	4200	4900	5600
Ukuran (Meter)	50	100	150	200	250	300	400	500





- 1. Apa yang dimaksud dengan data yang terstruktur ?**
  
- 1. Apa yang dimaksud dengan data yang tidak terstruktur ?**

# WHAT IS AI ETHICS?



# What is AI Ethics?

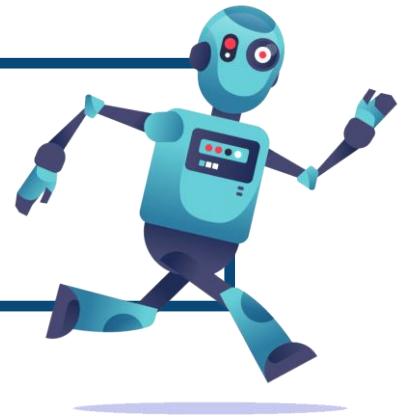


**Ethics / Etika** adalah seperangkat prinsip moral membantu kita membedakan antara benar dan salah



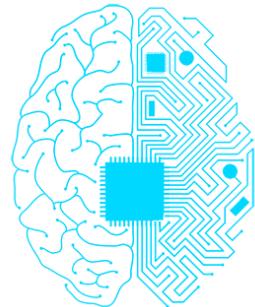
**AI Ethics** adalah **prinsip** dan **teknik** yang digunakan **sebagai pedoman** dalam **pengembangan** dan **penggunaan teknologi AI**.

# WHY IS AI ETHICS IMPORTANT?

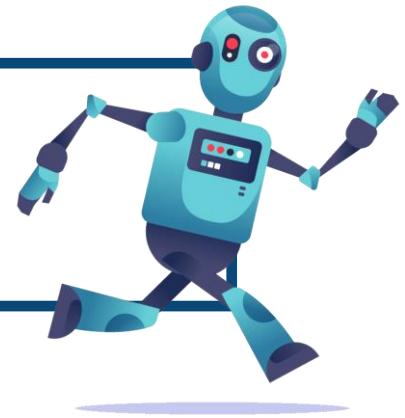


# Why is AI Ethics Important?

- Proyek AI yang tidak didesain dengan baik dapat mengakibatkan **bias** bahkan berpotensi membahayakan.
- Sebagai **pedoman** dalam mengembangkan dan menggunakan teknologi AI



# AI ETHICS PRINCIPLES



# AI Ethics Principles

## The Belmont Report

(National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research)



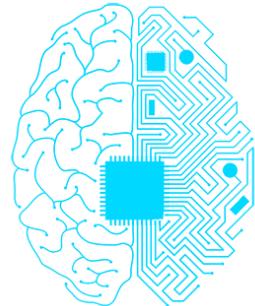
*Respect For Persons*



*Beneficence*



*Justice*



RESPECT



## *Respect For Persons*

- Menjunjung tinggi martabat manusia
- Atas dasar sukarela



## *Beneficence*

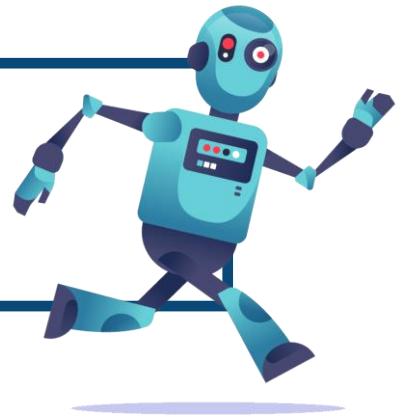
- Memaksimalkan potensi benefit
- Meminimalkan potensi bahaya



## *Justice*

- Meminimalkan kesenjangan potensi benefit dan potensi resiko terhadap populasi
- Meminimalkan bias individu atau sistem yang akan menggeser potensi risiko ke bagian tertentu dari populasi

# GAME TIME!

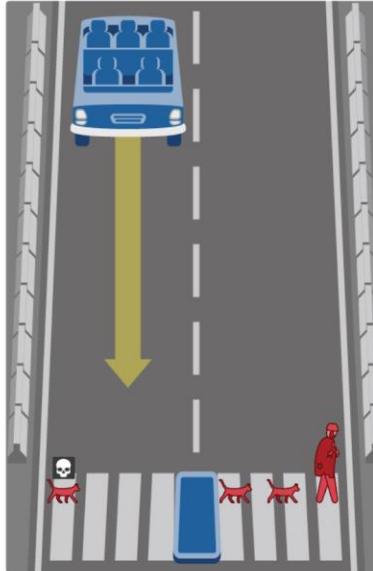


# Game: Trolley Problem

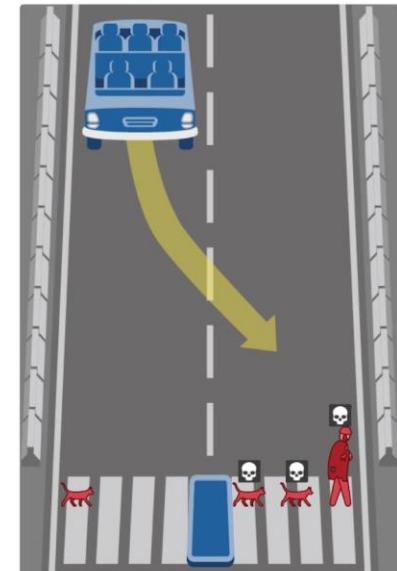


Bayangkan kita sedang mengembangkan Self-Driving Car.

What should the self-driving car do?



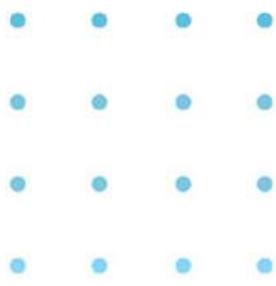
Show Description



Show Description

<http://moralmachine.mit.edu/>

# AI Mastery Course



## Module 1

Introduction to everything

### Section

Python Introduction





# Learning Objectives

Di akhir modul ini, kita akan dapat:

- Memahami kenapa bahasa pemrograman Python itu popular
- Memahami berbagai tools untuk bahasa pemrograman Python
- Mengenal Jupyter notebooks
- Mengenal sintaks umum di bahasa pemrograman Python



# Agenda





01

## INTRO I

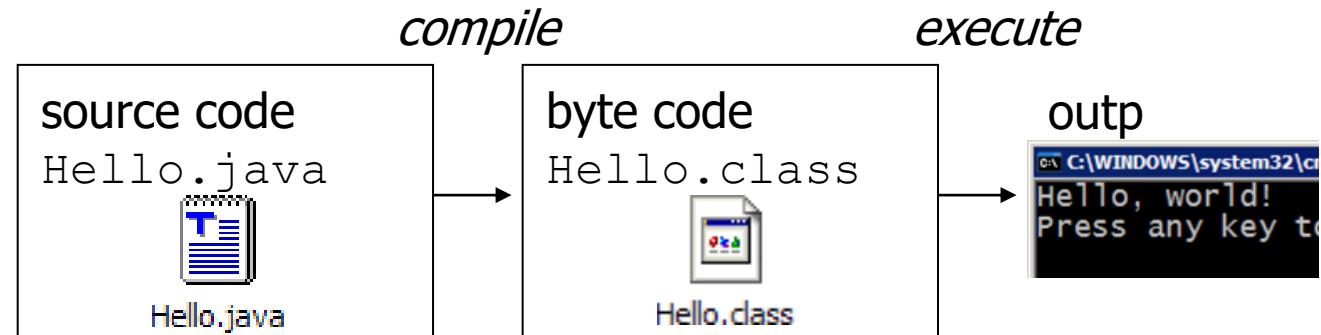
- Apa itu Python?
- Siapa yang menggunakan Python?
- Mengapa Python?

# Apa itu Python?

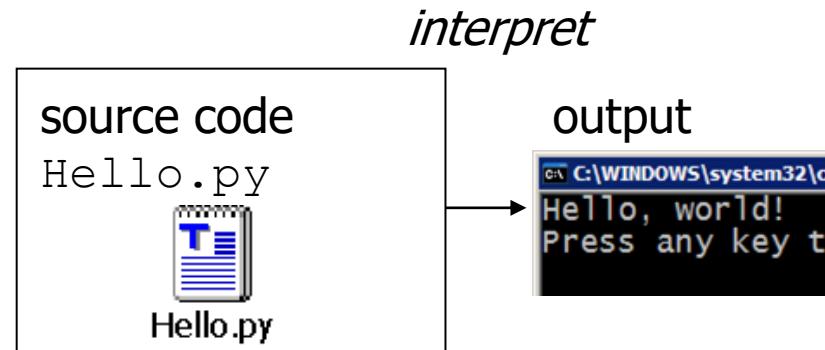
**Bahasa pemrograman open-source** yang bersifat dinamis dengan keutamaan pada **kesederhanaan** dan **produktivitas**. Python memiliki sintaks elegan yang alami untuk dibaca dan mudah untuk ditulis.

# Compiler v/s Interpreter

- Banyak bahasa pemrograman mengharuskan kita untuk mengkompilasi (menerjemahkan) program ke dalam bentuk yang dimengerti mesin.



- Python merupakan bahasa yang langsung diinterpretasi/dimengerti oleh mesin tanpa kompilasi.



# Siapa yang menggunakan Python?

The popular YouTube video sharing system is largely written in Python

Google makes extensive use of Python in its web search system

Dropbox storage service codes both its server and client software primarily in Python

The Raspberry Pi single-board computer promotes Python as its educational language



RaspberryPi

## COMPANIES USING PYTHON



BitTorrent™



NASA uses Python for specific Programming Task



The NSA uses Python for cryptography and intelligence analysis

**NETFLIX**

BitTorrent peer-to-peer file sharing system began its life as a Python Program

Netflix and Yelp have both documented the role of Python in their software infrastructures

# Mengapa Python?

Sederhana  
dan elegan

Powerful

Free

Dukungan  
Komunitas

# Python itu mudah

## Java

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

## Python

```
print("Hello World")
```



02

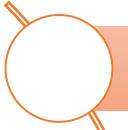
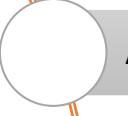
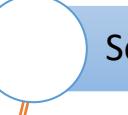
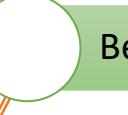
## INTRO II

- Sejarah Python
- Versi Python
- Aplikasi-aplikasi dari python

# Sejarah Python



**Guido van  
Rossum**

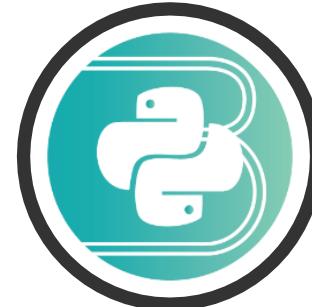
-  Diciptakan pada 1980-an, dan diterapkan oleh Guido van Rossum awal 90-an di Belanda
-  Awalnya diberi nama Monty Python
-  Berbentuk open sourced sejak awal hadir
-  Sebagai bentuk scripting language dengan fitur kaya
-  Bersifat scalable, object oriented dan functional sejak awal hadir
-  Digunakan oleh beberapa organisasi seperti Google & Facebook sejak masa awal berdiri

# Versi Python



Python 2.x  
legacy

VS



Python 3.x  
present and future of  
the language

# Aplikasi-aplikasi dari Python



Web Development



Software testing



Web Scraping



Data Analysis



## 03 SETUP

- Distribusi Python & IDEs
- Setup Python

# Distribusi Python

- Bundel perangkat lunak yang berisi penerjemah (interpreter) dan pustaka standar (basic library) yang biasa dikenal sebagai distribusi python.
- Mayoritas distribusi python populer memiliki manajer paket untuk menginstal atau memutakhirkan pustaka/paket python.
- Beberapa distribusi yang paling populer:

Anaconda	Enthought Canopy	ActivePython	WinPython
By Continuum Analytics 200+ packages PM – conda Use – ML, analytics, dev	By Enthought Inc 70+ packages CLI – canopy_cli Use – analytics, statistics	By ActiveState 50+ packages PM - PyPM Use – analytics, ML	Open-source PM – pip Only for windows OS Use – dev, analytics

# Python IDEs

- IDE (integrated development environment) adalah editor kode yang dilengkapi beberapa fitur seperti linting kode, debugging, dan pengujian.
- Ada beberapa editor kode dan IDE yang tersedia untuk python, beberapa di antaranya yang populer adalah –

## Jupyter Notebook

- Tool berbasis web untuk menulis kode python bersifat open-source dan analytics friendly

## Spyder

- Python IDE teroptimasi untuk data science workflows

## Pycharm

- Python IDEs, yang khusus digunakan untuk software development berbasis python

## Atom

- Python IDE untuk membuat desktop apps dan mendukung python

## Google Colab

- Code editor berbasis web dan dilengkapi dengan GPU power

# Setting up anaconda for python

1. Download Anaconda installer. - <https://www.anaconda.com/download/#windows>
2. Klik 2x installer untuk menjalankan.
3. Jika mengalami masalah selama penginstalan, nonaktifkan sementara perangkat lunak anti-virus selama penginstalan, lalu aktifkan kembali setelah penginstalan selesai. Jika Anda menginstal untuk all users, hapus instalasi Anaconda dan instal ulang hanya untuk pengguna Anda dan coba lagi.
4. Klik Berikutnya, Baca ketentuan lisensi dan klik "Saya Setuju".
5. Pilih instalasi untuk "Just Me" kecuali Anda menginstal untuk semua pengguna (yang memerlukan hak Administrator Windows) dan klik Berikutnya.



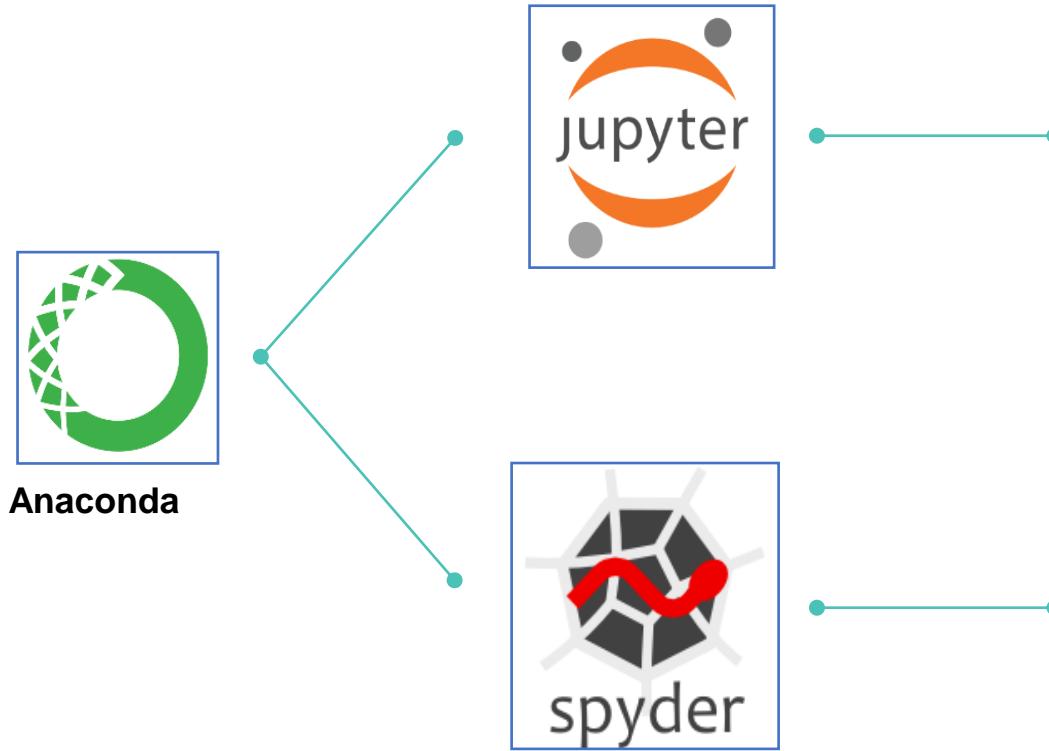
# Setting up anaconda for python

- 1.Pilih folder tujuan untuk menginstal Anaconda dan klik tombol Next.
- 2.Pilih apakah akan menambahkan Anaconda ke PATH environment variable Anda. Kami  
menyarankan untuk tidak menambahkan Anaconda ke PATH environment variable,  
karena ini dapat mengganggu perangkat lunak lain. Sebagai gantinya, gunakan  
perangkat lunak Anaconda dengan membuka Anaconda Navigator atau Anaconda  
Prompt dari Start Menu.

Detil installation guide - <https://docs.anaconda.com/anaconda/install/windows/>



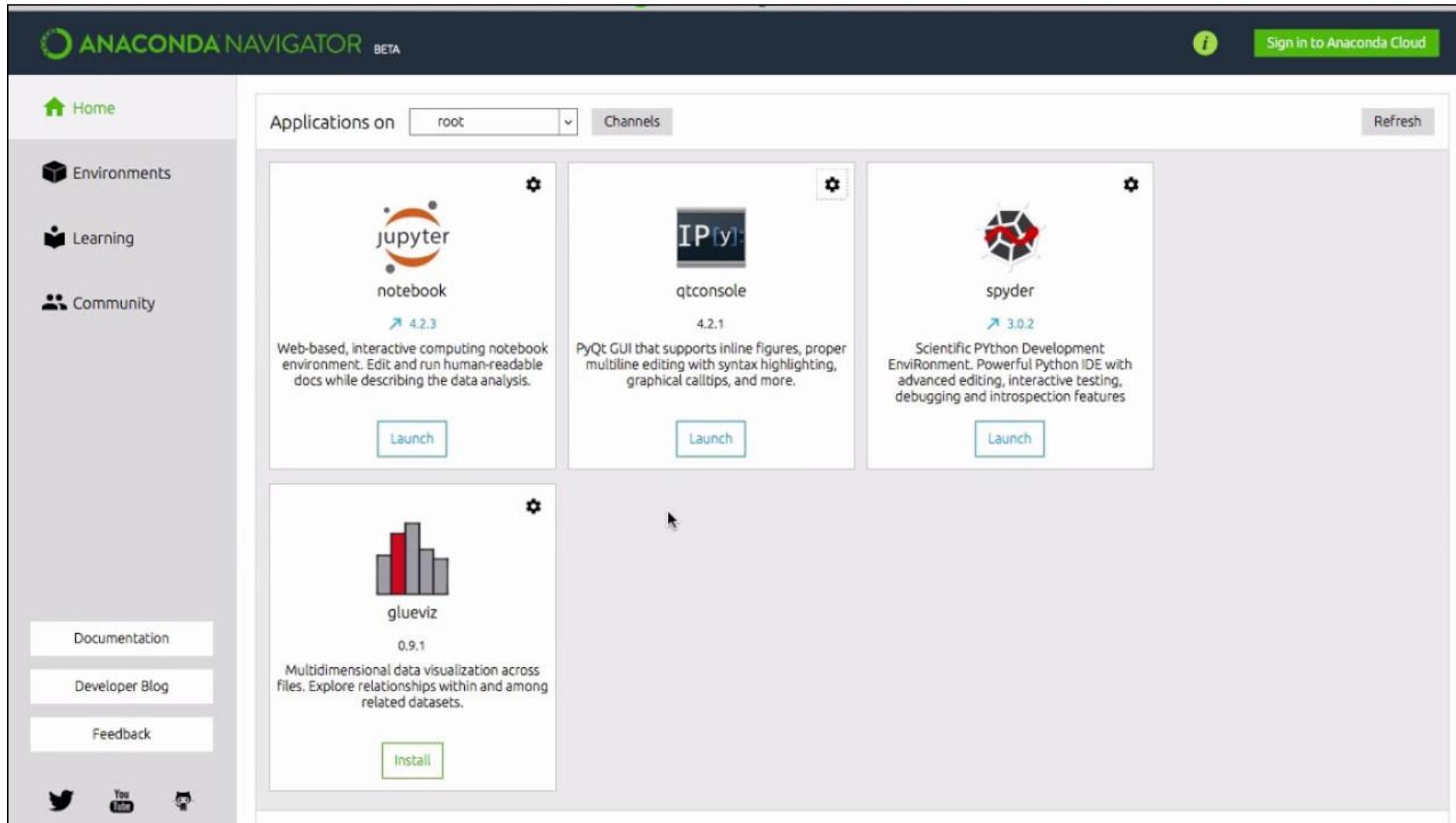
# Anaconda



Tool berbasis interface yang dapat digunakan untuk menulis catatan serta melakukan analisis data dalam file yang sama

IDE (Integrated Development Environment) untuk scientific programming di Python

# Anaconda Navigator





04

## TOOL

- Mengenal Jupyter notebook
- Sintaks dasar pada Python

# Mengenal Jupyter notebook

## Jupyter Notebook

- Notebook Jupyter adalah dokumen yang berisi kode python serta elemen teks yang lengkap seperti teks, judul, persamaan, gambar, dan tautan.

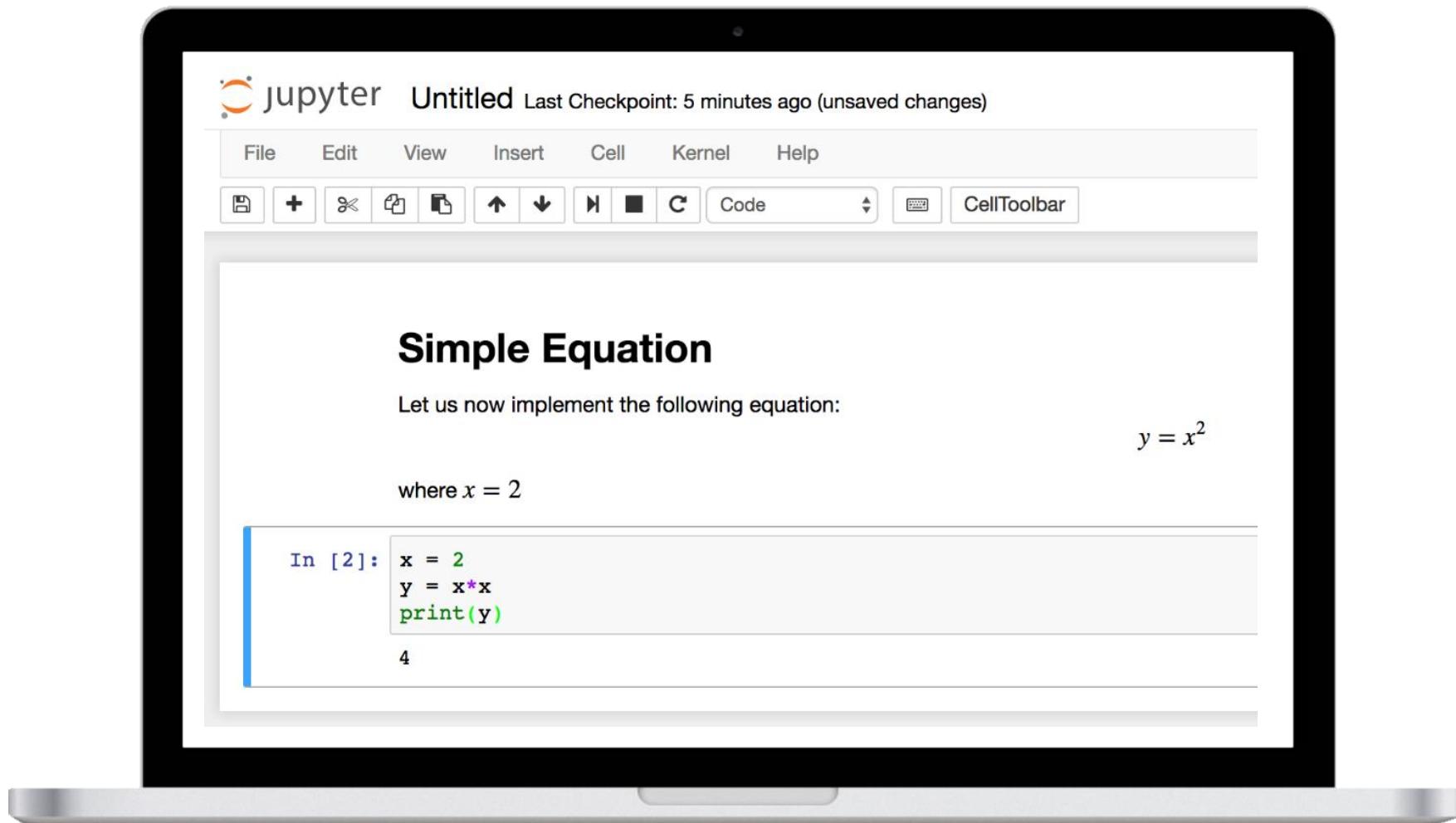
## Jupyter Notebook App

- aplikasi server-klien yang memungkinkan pengeditan dan menjalankan dokumen notebook melalui browser web
- dapat dijalankan di desktop lokal yang tidak memerlukan akses internet
- Bisa juga diinstall di remote server dan diakses melalui internet

## Kernel

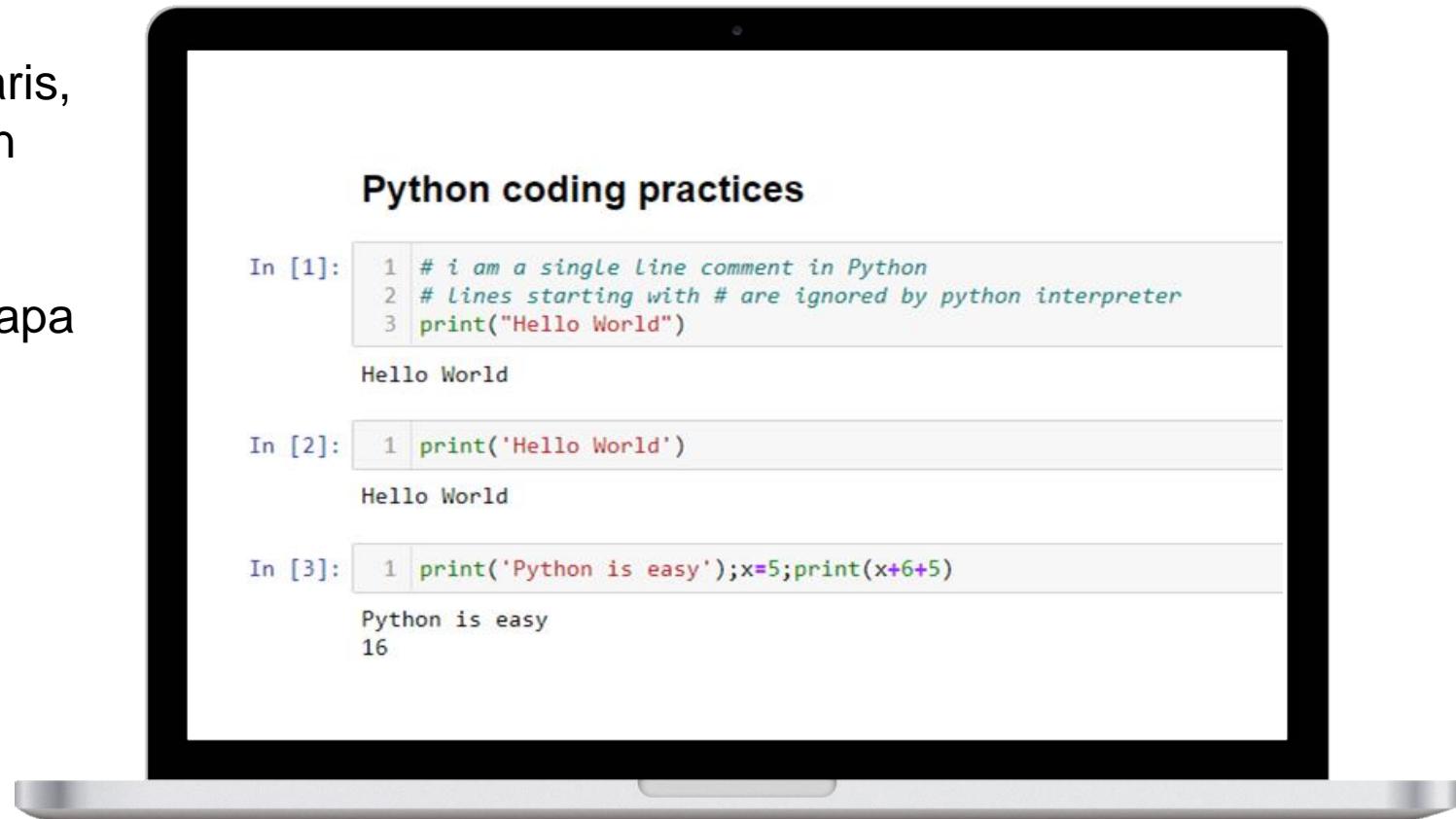
- Mesin komputasi yang mengeksekusi kode
- Kernel Ipython digunakan di anaconda untuk mengeksekusi kode

# Programming di Jupyter notebook



# Python – coding practices

- # digunakan untuk komentar satu baris, kode yang ditulis setelah # diabaikan oleh python Interpreter
- titik koma ( ; ) memungkinkan beberapa pernyataan pada satu baris
- tunggal ('), ganda ("") menunjukkan string



The image shows a laptop screen displaying a Jupyter Notebook titled "Python coding practices". The notebook contains three code cells:

- In [1]:**

```
1 # i am a single line comment in Python
2 # Lines starting with # are ignored by python interpreter
3 print("Hello World")
```

Hello World
- In [2]:**

```
1 print('Hello World')
```

Hello World
- In [3]:**

```
1 print('Python is easy');x=5;print(x+6+5)
```

Python is easy  
16

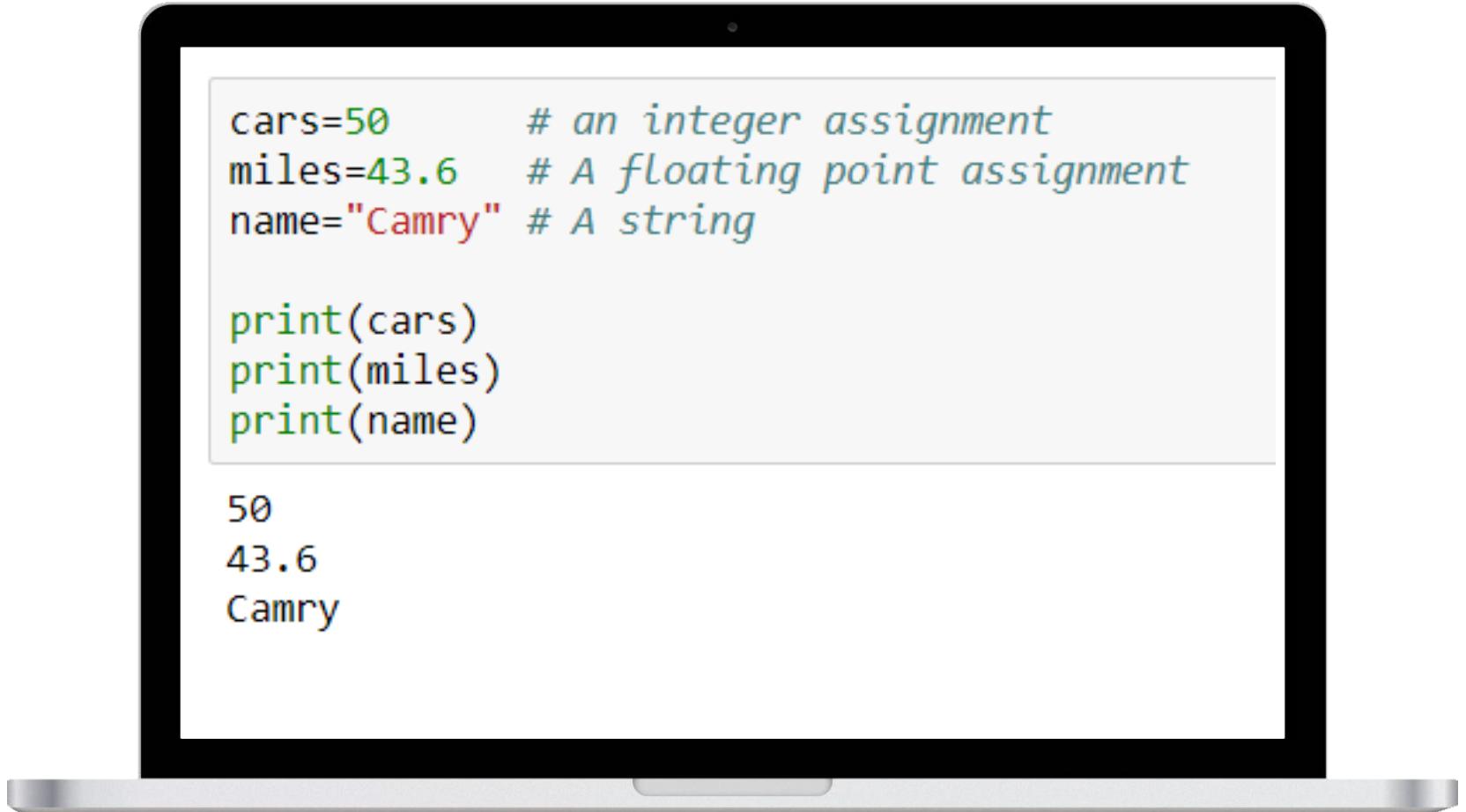
# Variables

Variabel adalah lokasi memori yang dicadangkan untuk menyimpan nilai

cars = 50

miles =  
43.60

name =  
"Camry"



The image shows a laptop screen with a white background. On the screen, there is a code editor window containing Python code. Below the code editor, there is a terminal window showing the output of the code. The code is as follows:

```
cars=50      # an integer assignment
miles=43.6   # A floating point assignment
name="Camry" # A string

print(cars)
print(miles)
print(name)
```

The terminal output below the code shows the results of the print statements:

```
50
43.6
Camry
```



05

## KESIMPULAN

- Kuis
- Ringkasan

## Pertanyaan

**Manakah dari pernyataan berikut yang TIDAK benar tentang python?**

- A. Python itu gratis
- B. Python tidak perlu lagi di-install di mesin karena sudah ter-install di mesin
- C. Python sangat powerful – memiliki banyak perangkat library
- D. Python sangat mudah digunakan dan dipahami

## Pertanyaan

**Manakah dari pernyataan berikut yang TIDAK benar tentang python?**

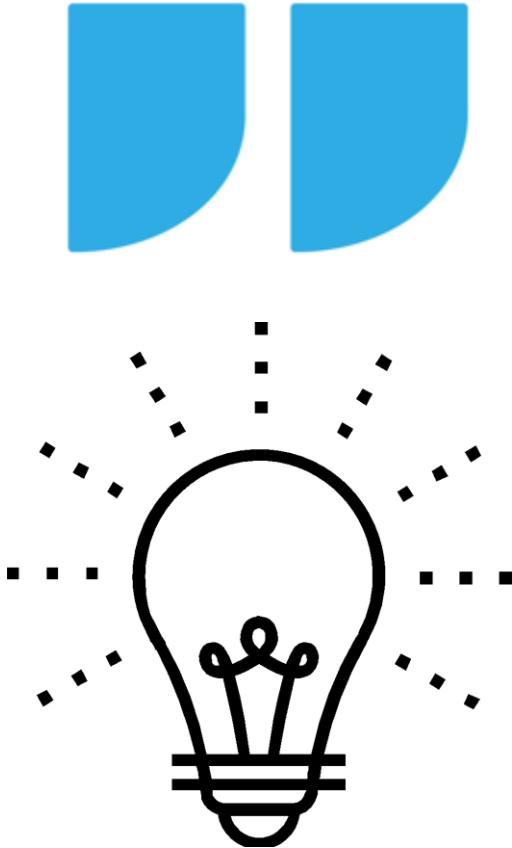
- A. Python itu gratis
- B. Python tidak perlu lagi di-install di mesin karena sudah ter-install di mesin
- C. Python sangat powerful – memiliki banyak perangkat library
- D. Python sangat mudah digunakan dan dipahami



Jawaban: B

# Ringkasan

- Python adalah bahasa pemrograman yang populer, mudah digunakan, mudah diperbarui dan dipelihara
- Merupakan open-source, kuat, portabel dan memiliki kumpulan perpustakaan yang lengkap
- Python saat ini digunakan untuk berbagai tugas, beberapa aplikasi populer termasuk analisis data, pembelajaran mesin, scrapping web, pengembangan aplikasi web, pengembangan layanan backend
- Python 2.x adalah legacy, python 3.x digunakan pada hari ini dan di masa depan
- Anaconda adalah distribusi python yang populer dengan editor kode seperti notebook Jupyter dan spyder IDE





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

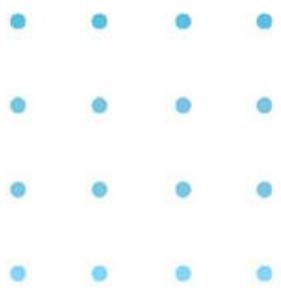
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  Orbit Future Academy
-  @OrbitFutureAcademyIn1
-  OrbitFutureAcademy
-  Orbit Future Academy

# AI Mastery Course



## Module 1

Introduction to everything

### Section

Python operators & control flow





# Learning Objectives

Di akhir modul ini, kita akan dapat:

- Memahami Python operators
- Mengeksplorasi input function di Python
- Menggunakan conditional programming dan loops
- Memahami bagaimana menggunakan list comprehension di Python



# Agenda

- 01 OPERATORS
  - Python operators
  - Input function
- 02 LOGIC
  - Control flow: if else
- 03 LOOP
  - Control flow: for loop
  - Control flow: while loop
- 04 COMPREHENSION
  - List comprehension
  - Dictionary comprehension
- 05 KESIMPULAN
  - Kuis
  - Ringkasan

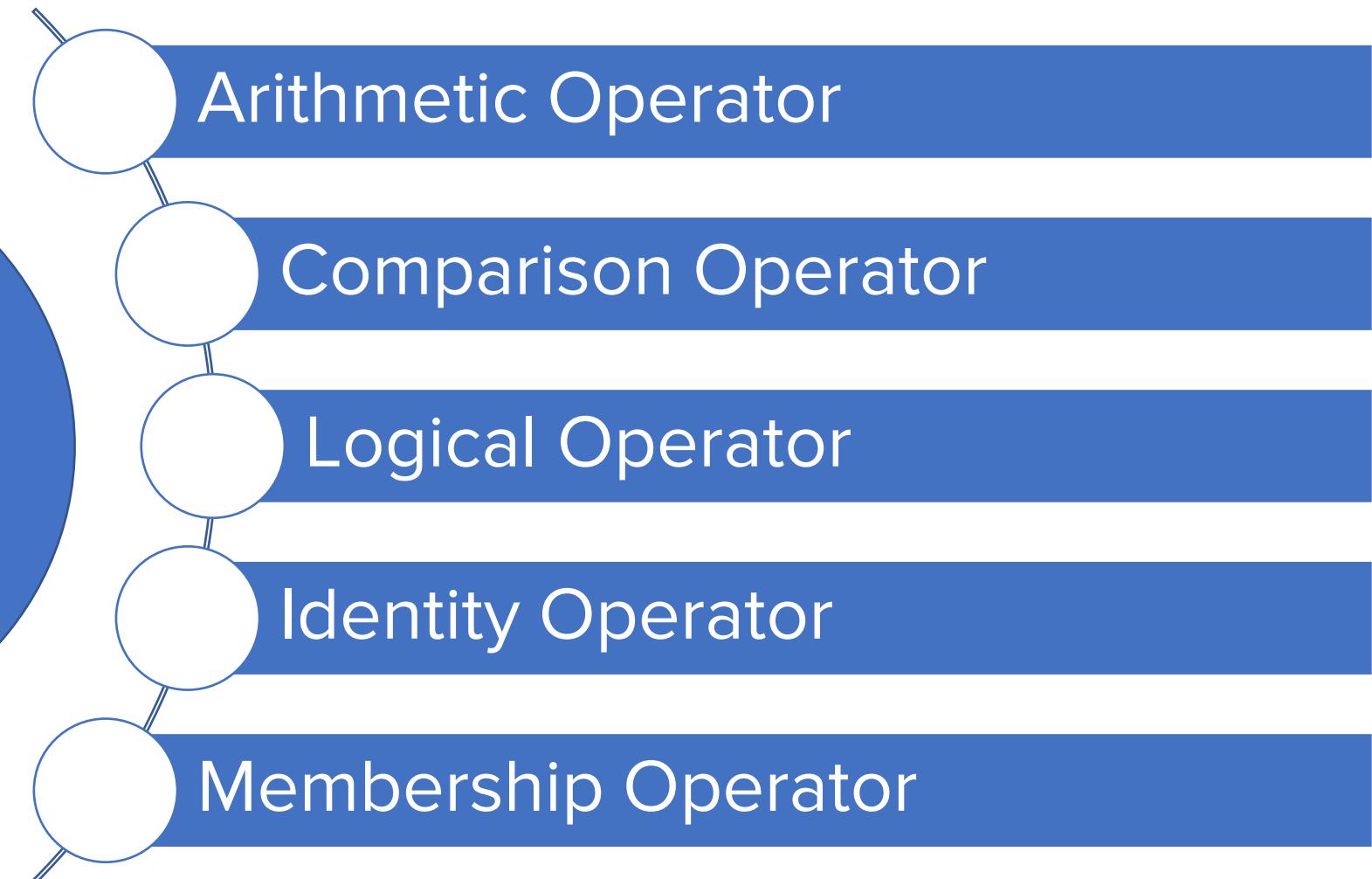


01 OPERATORS

- Python operators
- Input function



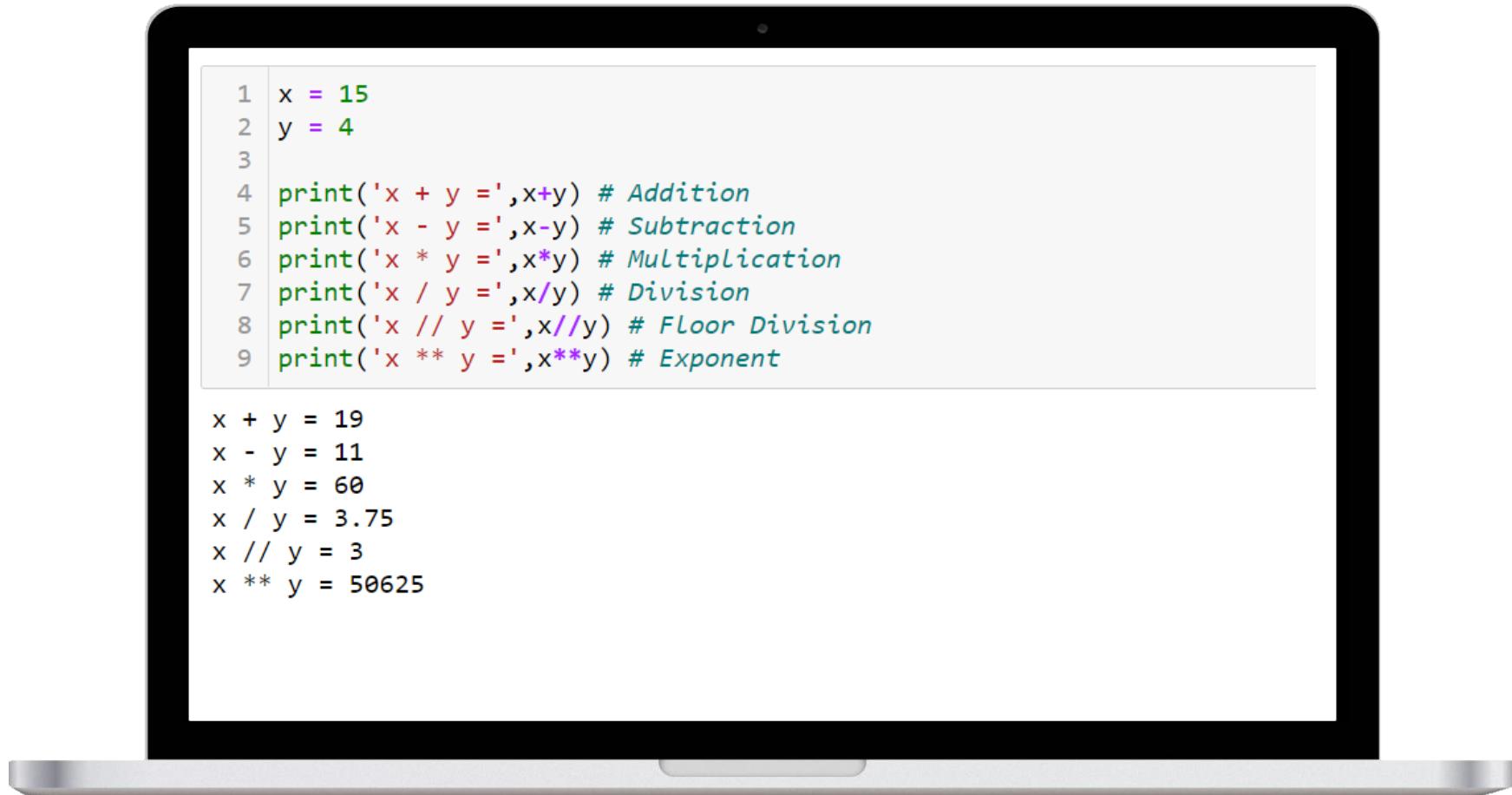
# Python operators



# Arithmetic operators

Operator	Definisi	Contoh
+	Menambahkan 2 operands atau unary plus	$x + y + 2$
-	Mengurangkan operand sisi kiri dengan operand sisi kanan atau unary minus	$x - y - 2$
*	Mengkalikan 2 operands	$x * y$
/	Membagikan operand sisi kiri dengan operand sisi kanan (selalu menghasilkan float)	$x / y$
%	Modulus - hasil sisa dari pembagian operand sisi kiri dengan operand sisi kanan	$x \% y$ (hasil sisa dari $x/y$ )
//	Floor division - Pembagian yang menghasilkan pembulatan angka	$x // y$
**	Exponent - operand sisi kiri dipangkatkan dengan operand sisi kanan	$x^{**}y$ ( $x$ dipangkatkan $y$ )

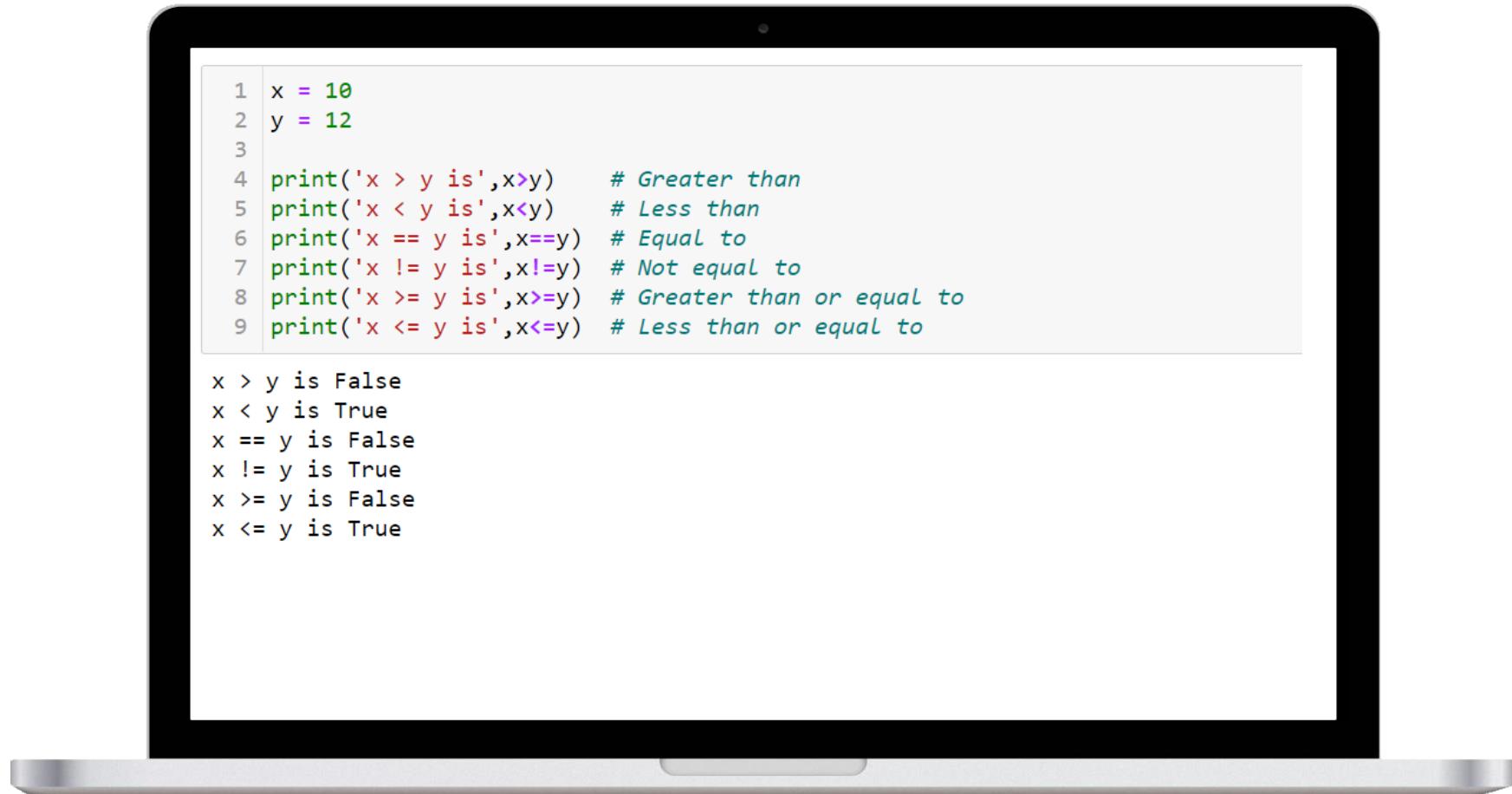
# Arithmetic operators



# Comparison operators

Operator	Definisi	Contoh
>	Lebih besar dari - True jika operand sisi kiri lebih besar dari sisi kanan	$x > y$
<	Kurang dari - True jika operand sisi kiri lebih kecil dari sis kanan	$x < y$
==	Sama dengan - True jika kedua operand sama besar	$x == y$
	Tidak sama dengan - True jika kedua operand tidak sama dengan	$x != y$
>=	Lebih besar dari atau sama dengan - True jika operand sisi kiri lebih besar dari atau sama dengan operand sisi kanan	$x >= y$
<=	Lebih kecil dari atau sama dengan - True jika operand sisi kiri lebih kecil dari atau sama dengan operand sisi kanan	$x <= y$

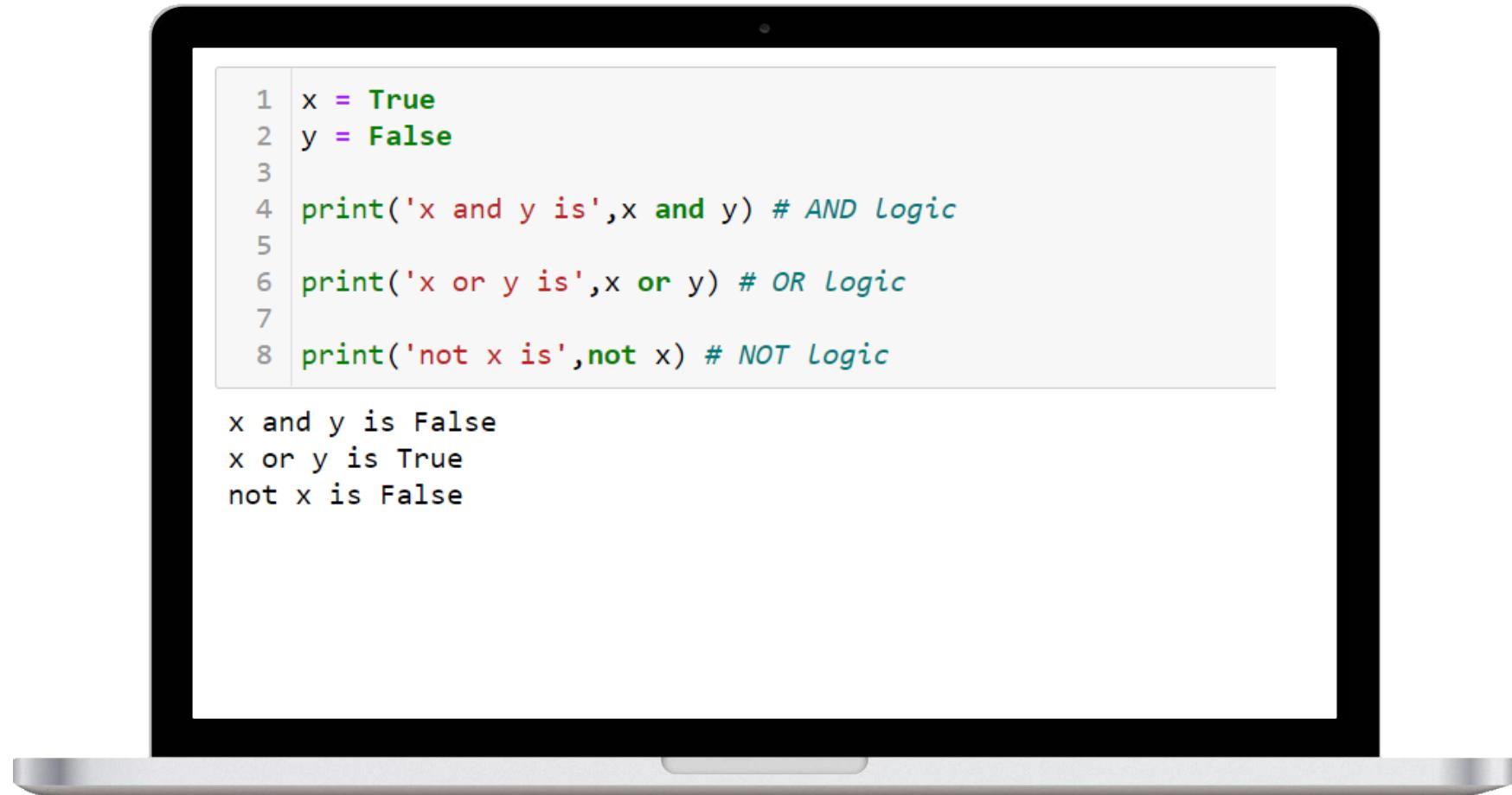
# Comparison operators



# Logical operators

Operator	Definisi	Contoh
and	True jika kedua operand true	x and y
or	True jika salah satu operand true	x or y
not	True jika operand false	not x

# Logical operators



# Identity operators

Operator	Definisi	Contoh
is	True jika operand bersifat identik	x is True
is not	True jika operand tidak bersifat identik	x is not True

# Identity operators

```
1 x1 = 5
2 y1 = 5
3 x2 = 'Hello'
4 y2 = 'Hello'
5 x3 = [1,2,3]
6 y3 = [1,2,3]
7
8 print(x1 is not y1) # Identity is not same?
9
10 print(x2 is y2) # Identity is same?
11
12 print(x3 is y3) # Identity is same?
```

```
False
True
False
```

# Membership operators

Operator	Definisi	Contoh
in	True jika value/variable ada di dalam sequence	5 in x
not in	True jika value/variable tidak ada di dalam sequence	5 not in x

# Membership operators

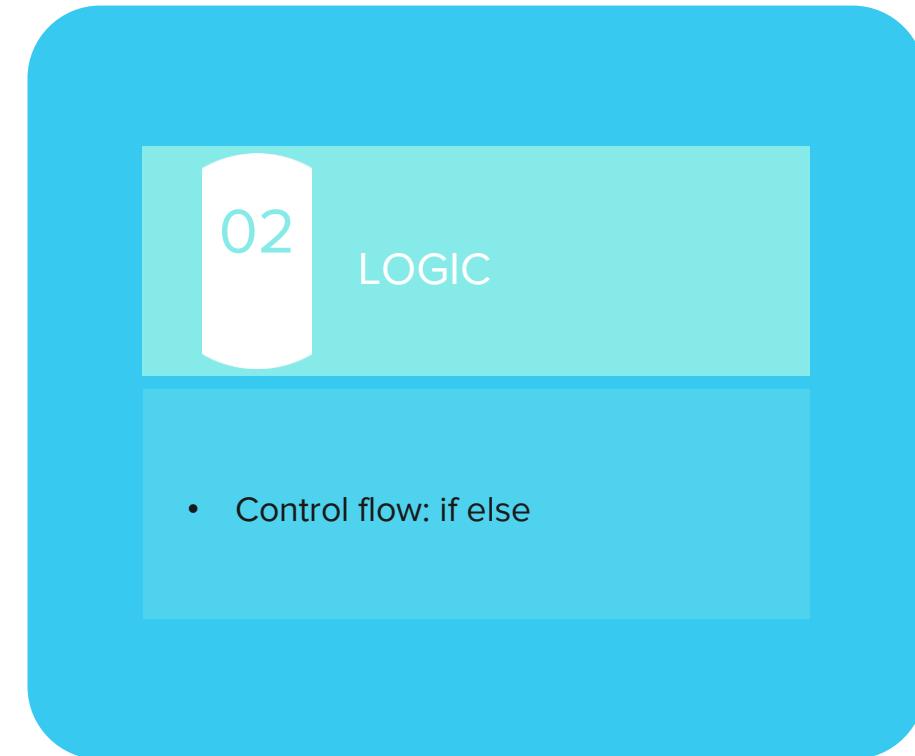
```
1 x = 'Hello world'  
2 y = {1:'a',2:'b'}  
3  
4 print('H' in x)          # Is member?  
5  
6 print('hello' not in x)  # Is not member?  
7  
8 print(1 in y)           # Is member?  
9  
10 print('a' in y)         # Is member?
```

```
True  
True  
True  
False
```

# Input function di python

- `input()` function digunakan untuk menerima input dari user
- Nilai yang diberikan sebagai input, maka fungsi input akan mengubahnya ke bentuk string
- Jika nilai integer diberikan, fungsi input tetap akan mengubahnya ke bentuk string



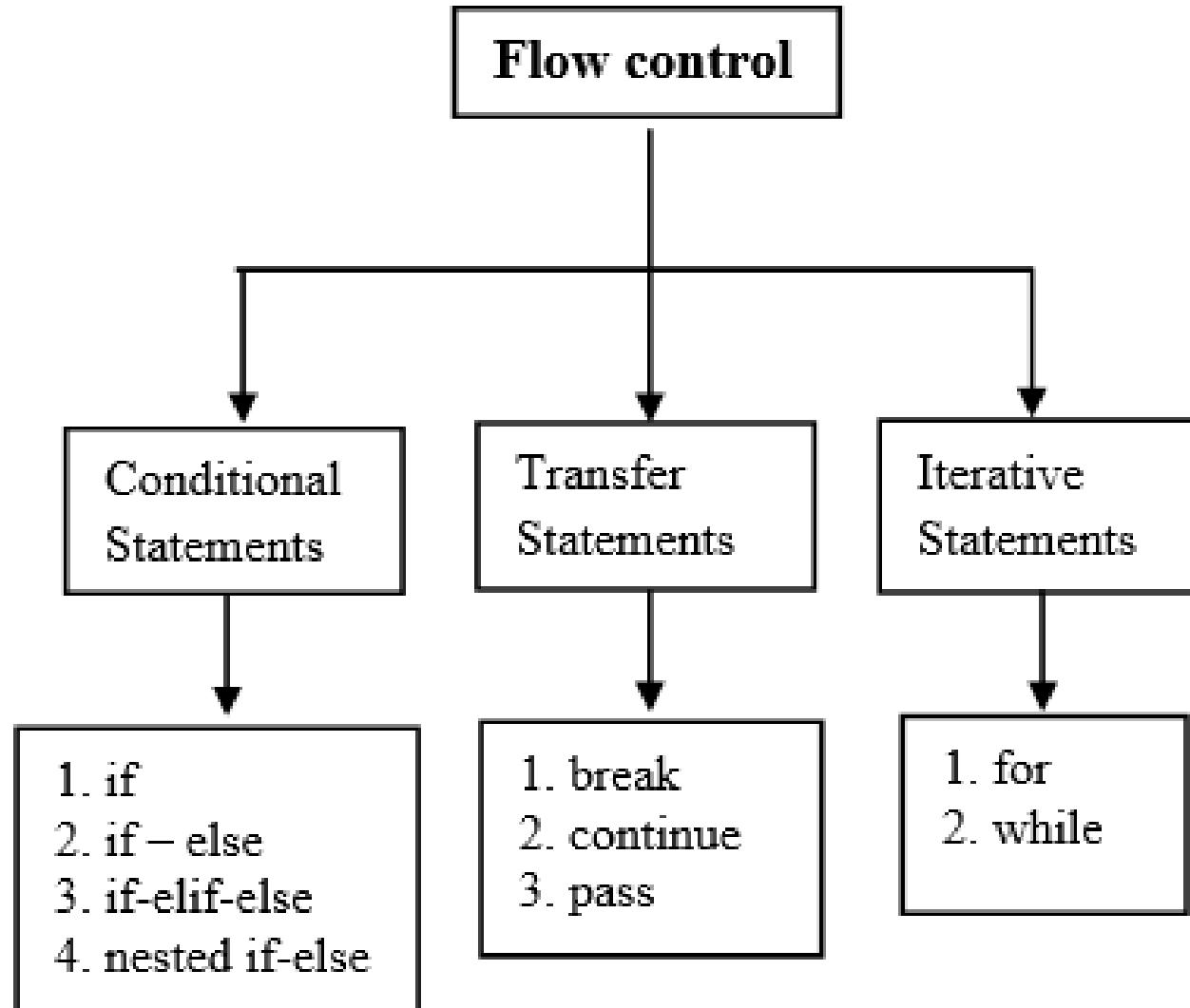


02 LOGIC

- Control flow: if else



# Python: Control Flow



# Control flow: if else

- Pernyataan bersyarat memutuskan aliran program berdasarkan beberapa aturan
- Ini dilakukan dengan bantuan pernyataan if-else

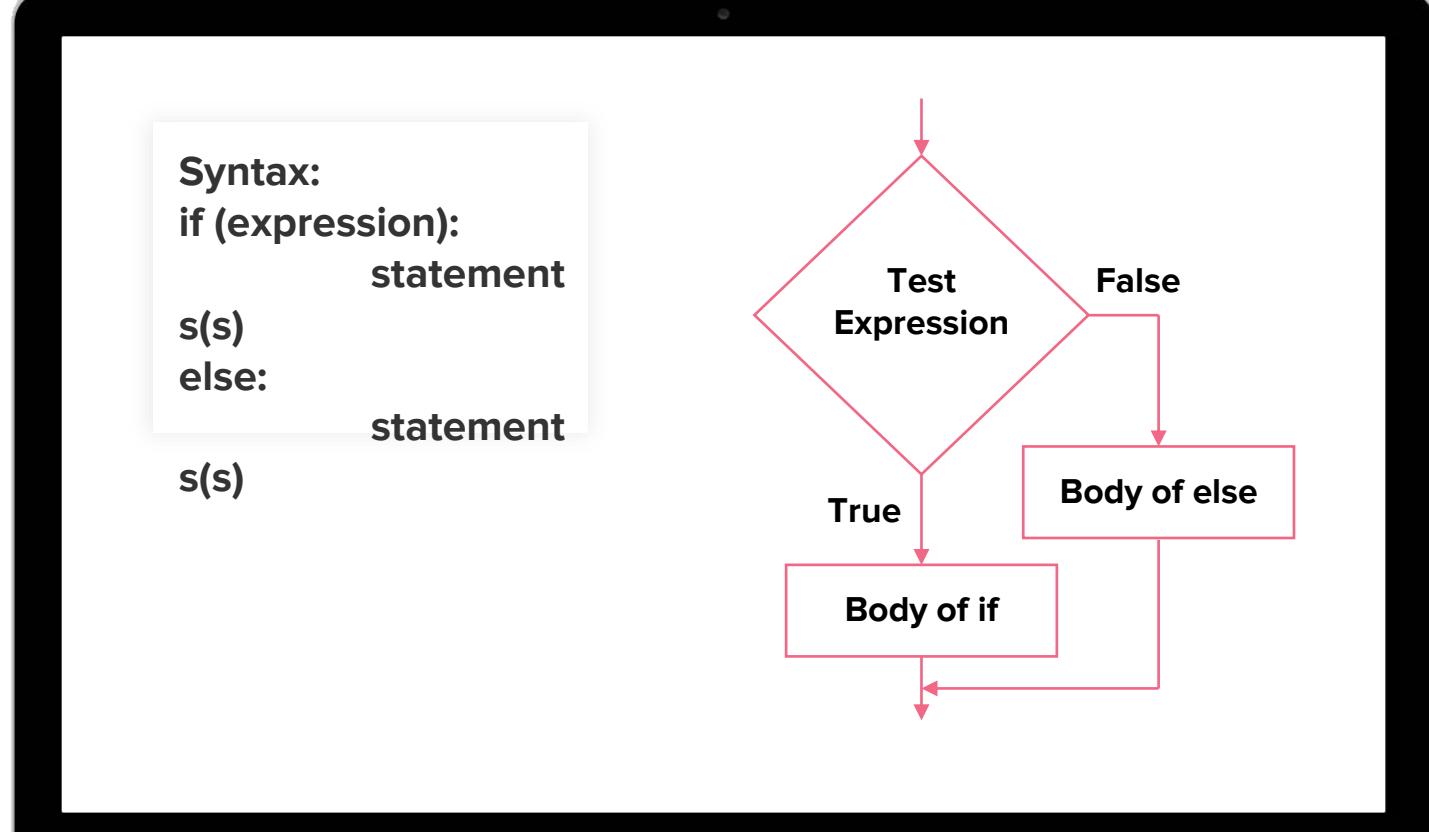


# Control flow: if else

- Pernyataan python if digunakan untuk mengeksekusi blok kode secara kondisional
- Jika pernyataan berisi ekspresi logis atau kondisi, tergantung pada keputusan yang dibuat



# Control flow: if else



# Control flow: if else



Syntax:

```
if (expression):
    statement
(s)
elif (expression):
    statement
(s)
elif (expression):
    statement
(s)
else:
    statement
(s)
```

```
graph TD
    Start(( )) --> If{if condition}
    If -- True --> BodyIf[Body of if]
    BodyIf --> End(( ))
    If -- False --> Elif{elif condition}
    Elif -- True --> BodyElif[Body of elif]
    BodyElif --> End
    Elif -- False --> Else[Body of else]
    Else --> End
```

The flowchart shows the execution path of an if-elif-else block. It begins with an 'if condition' diamond. If the condition is 'True', it leads to the 'Body of if' rectangle, which then leads to a final exit point. If the 'if' condition is 'False', it proceeds directly to the next diamond, 'elif condition'. If this 'elif' condition is 'True', it leads to the 'Body of elif' rectangle. If both 'if' and 'elif' conditions are 'False', it leads to the 'Body of else' rectangle. Finally, all paths converge at a common exit point.

# Control flow: if else

nested  
if else

```
var = 50
if var < 200:
    print ("Expression value is less than 200")
    if var == 150:
        print ("Which is 150")
    elif var == 100:
        print ("Which is 100")
    elif var == 50:
        print ("Which is 50")
    elif var < 50:
        print ("Expression value is less than 50")
else:
    print ("Could not find true expression")

print ("Good bye!")
```

```
Expression value is less than 200
Which is 50
Good bye!
```



03

## LOOP

- Control flow: for loop
- Control flow: while loop

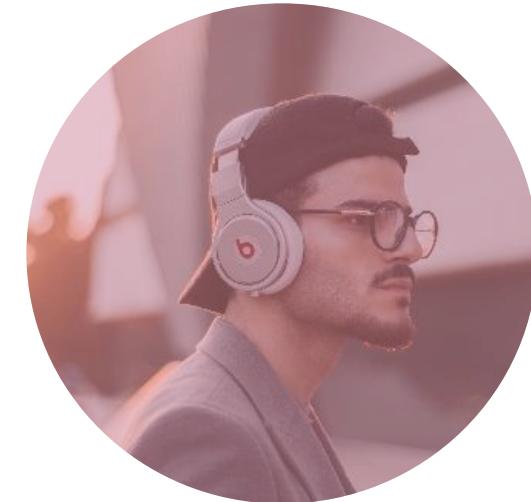


# Loop

- Loop adalah blok kode yang dilakukan berulang kali dalam jumlah tertentu
- Ada dua implementasi loop di Python
- For loop – digunakan ketika jumlah pengulangan tetap
- While loop – digunakan ketika pengulangan akan dilakukan sampai suatu kondisi menjadi benar

**Contoh penerapan konsep loops pada music player dan playlist**

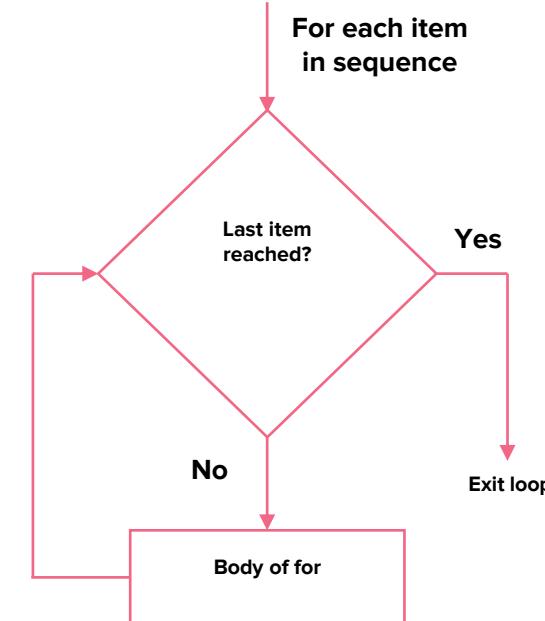
```
while (user does NOT press stop  
AND NOT end of playlist)  
{  
    get the next song  
    play the song  
}
```



# Loop: for loop

- for loop digunakan untuk mengulangi urutan yang berupa list, tuple, dictionary, atau set

```
Syntax:  
for (initialization;  
      condition;  
      propagation)  
{  
    statements(s)  
}
```



# Loop: for loop

Fungsi range() membuat urutan angka untuk diterasi

Ini dapat digunakan untuk membuat urutan penambahan/pengurangan



```
1 lst = ["wiley", 'python', "pyspark", "data", "Engineering"]
2 for i in range(len(lst)):
3     print(lst[i])
wiley
python
pyspark
data
Engineering

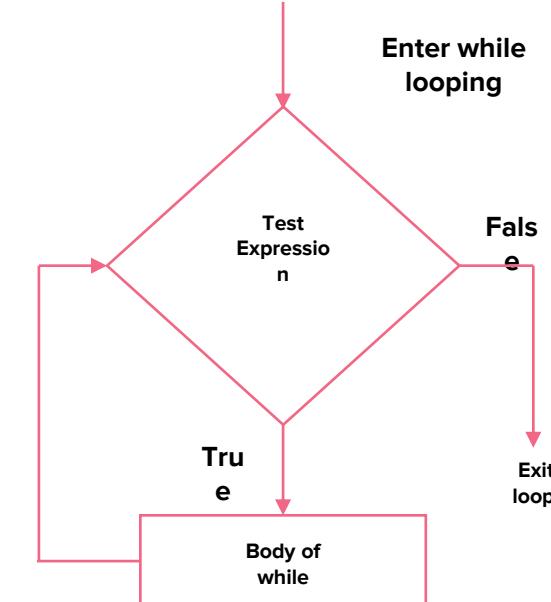
1 for j in range(0,10):
2     print(j)
0
1
2
3
4
5
6
7
8
9
```

# Loop: while loop

## while Loop

While loop mengeksekusi blok kode di dalamnya, hingga kondisi mengembalikan True dan keluar dari loop setelah kondisi mengembalikan False

```
Syntax:  
while  
(condition)  
{  
statements(s)  
}
```



# Loop: break

The break statement, seperti di bahasa C, akan menghentikan proses loop dimana break berada

```
1 for letter in 'for loop python':  
2  
3     # break the Loop as soon it sees 'p'  
4     if letter == 'p':  
5         break  
6  
7     print('Current Letter :', letter)
```

```
Current Letter : f  
Current Letter : o  
Current Letter : r  
Current Letter :  
Current Letter : l  
Current Letter : o  
Current Letter : o
```

# Loop: continue

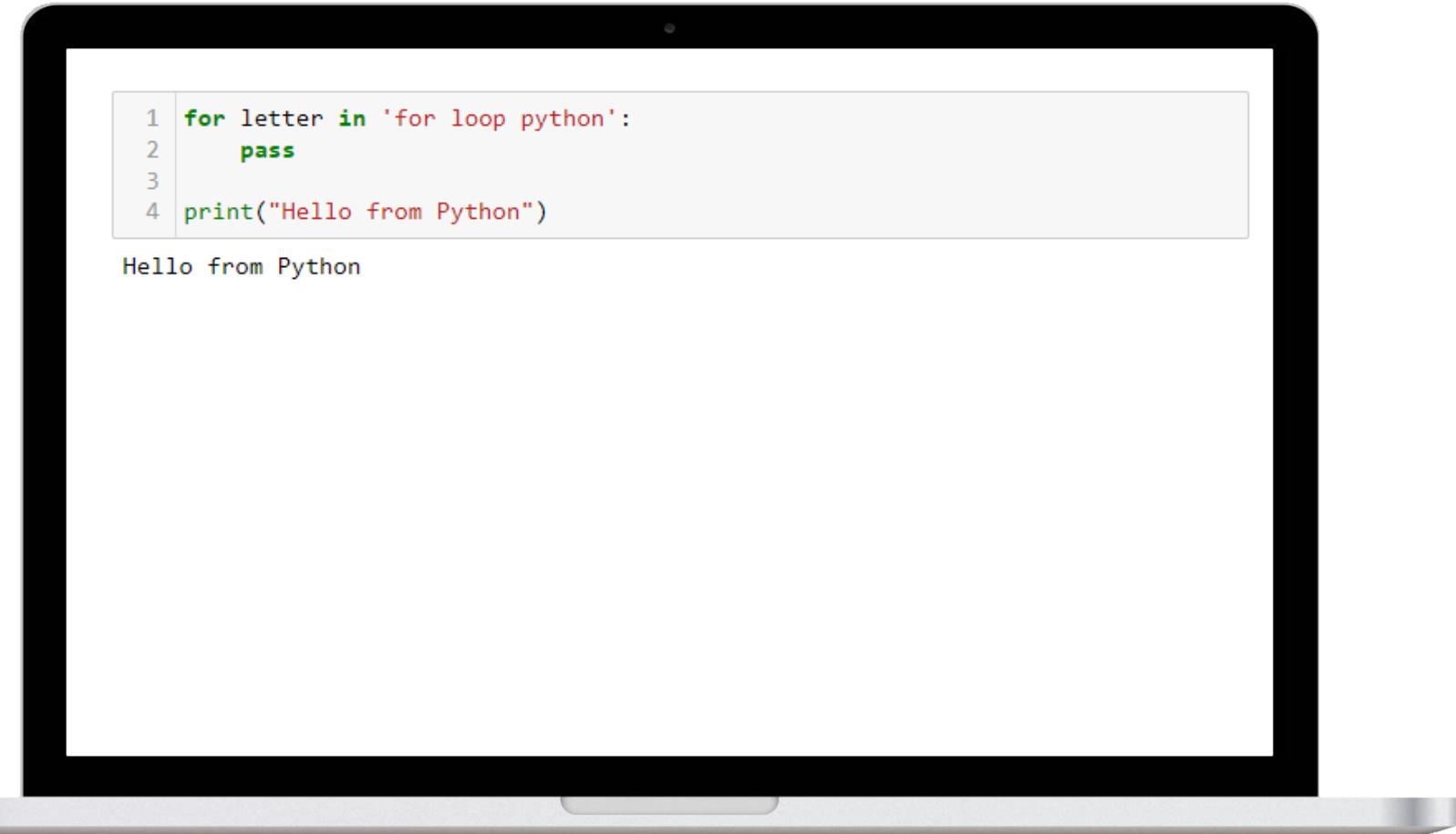
Continue mengembalikan kontrol ke awal loop, digunakan bersama loop for dan while.

```
1 for letter in 'for loop python':  
2  
3     # break the loop as soon it sees 'p'  
4     if letter == 'p':  
5         continue  
6     print('Current Letter :', letter)
```

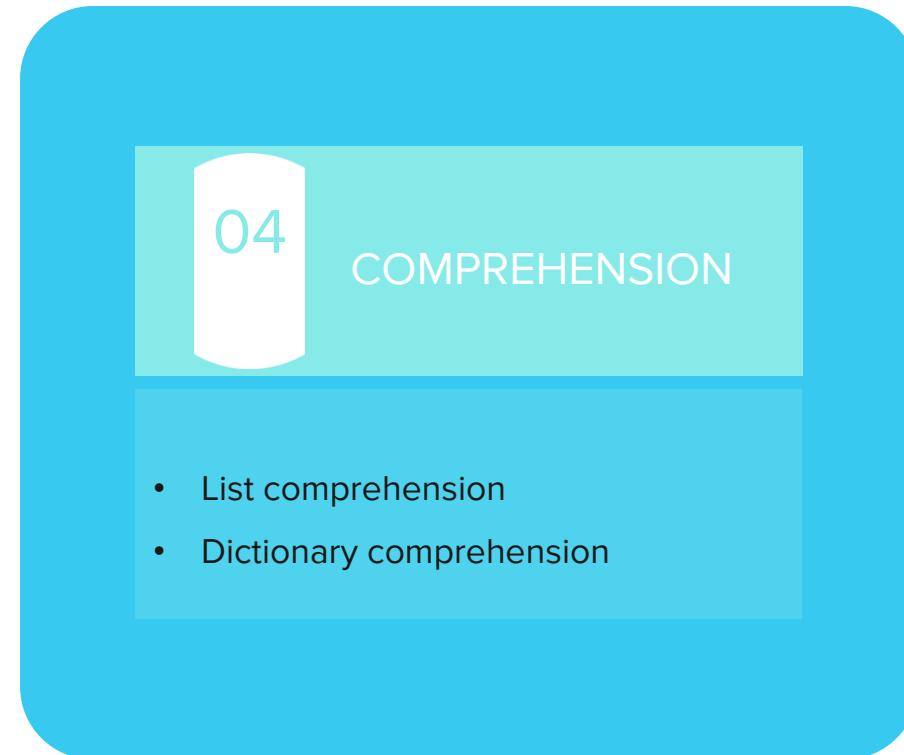
```
Current Letter : f  
Current Letter : o  
Current Letter : r  
Current Letter :  
Current Letter : l  
Current Letter : o  
Current Letter : o  
Current Letter :  
Current Letter : y  
Current Letter : t  
Current Letter : h  
Current Letter : o  
Current Letter : n
```

# Loop: pass

Pass digunakan untuk menulis loop kosong. Pass juga digunakan untuk pernyataan kontrol kosong, fungsi dan kelas.



```
1 for letter in 'for loop python':  
2     pass  
3  
4 print("Hello from Python")  
  
Hello from Python
```



04 COMPREHENSION

- List comprehension
- Dictionary comprehension



# List comprehension

List comprehension menawarkan sintaks yang lebih pendek ketika membuat list baru berdasarkan nilai dari list yang sudah ada

Kita juga dapat menambahkan conditional statements pada list comprehension

```
1 # list comprehension without condition
2 # [f(x) for x in sequence]
3 mylist= [2,3,6,5,9]
4 sqlist = [x**2 for x in mylist]
5 print(sqlist)
6
7
8 # list comprehension with if condition
9 # [f(x) for x in sequence if condition]
10 sqevenlist = [x**2 for x in mylist if x%2==0]
11 print(sqevenlist)
12
13 # list comprehension with if condition and else
14 # template 3: [f(x) if condition else g(x) for x in sequence]
15 # for even numbers - get square, odd number get cube
16 customlist = [x**2 if x%2==0 else x**3 for x in mylist]
17 print(customlist)
18
```

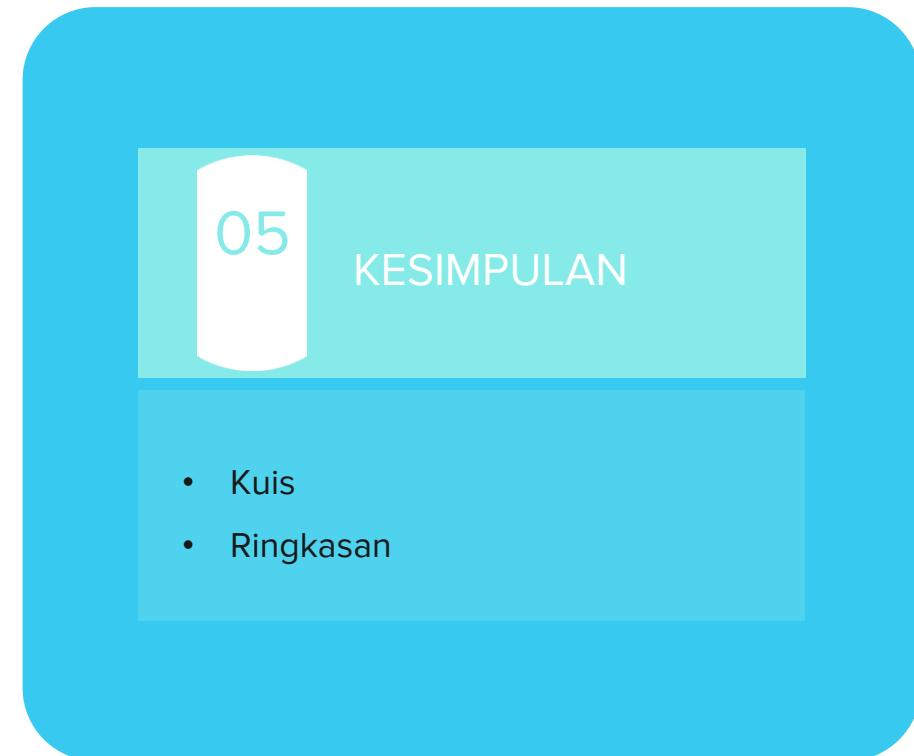
[4, 9, 36, 25, 81]  
[4, 36]  
[4, 27, 36, 125, 729]

# Dictionary comprehension

A dictionary comprehension berbentuk {key: value for (key, value) in iterable}

```
1 # Dictionary comprehension
2 # without condition: {key:value for vars in sequence}
3 mydic = {"apple":20,"mango":30,'banana':50,'grapes':40,'watermelon':80}
4 # add 10 with price of each fruit
5 newdic = {key:value+10 for (key,value) in mydic.items()}
6 print(newdic)
7
8 # adding conditions: {key:value for vars in sequence if condition}
9 # extracting dictionary of fruits having price more than 50
10 newdic = {key:value for (key,value) in mydic.items() if value>50}
11 print(newdic)
12
13 # adding conditions: {key:(value if condition else value2) for vars in sequence}
14 # increasing prices for fruits having price less than 50
15 newdic = {key:(value+10 if value<50 else value) for (key,value) in mydic.items()}
16 print(newdic)

{'apple': 30, 'mango': 40, 'banana': 60, 'grapes': 50, 'watermelon': 90}
{'watermelon': 80}
{'apple': 30, 'mango': 40, 'banana': 50, 'grapes': 50, 'watermelon': 80}
```



05 KESIMPULAN

- Kuis
- Ringkasan

## Pertanyaan

Manakah dari keyword berikut yang digunakan untuk menginterupsi eksekusi loop?

- A. Break
- B. Continue
- C. Pass
- D. Wait



## Pertanyaan

Manakah dari keyword berikut yang digunakan untuk menginterupsi eksekusi loop?

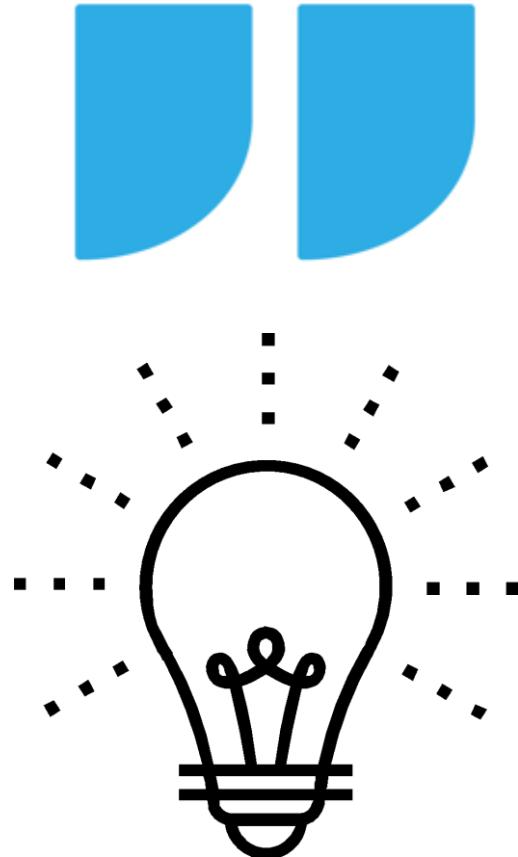
- A. Break
- B. Continue
- C. Pass
- D. Wait

Answer: A



# Kesimpulan

- Operator di Python digunakan untuk melakukan manipulasi data
- Fungsi Input() dapat digunakan untuk menerima input dari pengguna
- Python control flow structure berupa – if else, for loop dan while loop
- Loop digunakan untuk mengulang eksekusi kode yang ada
- List comprehension adalah cara yang efektif untuk menggambarkan dan membangun list berdasarkan list yang sudah ada
- Secara umum, list comprehension lebih ringan dan sederhana daripada fungsi dan list





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# Module 1

Introduction to everything

## Section

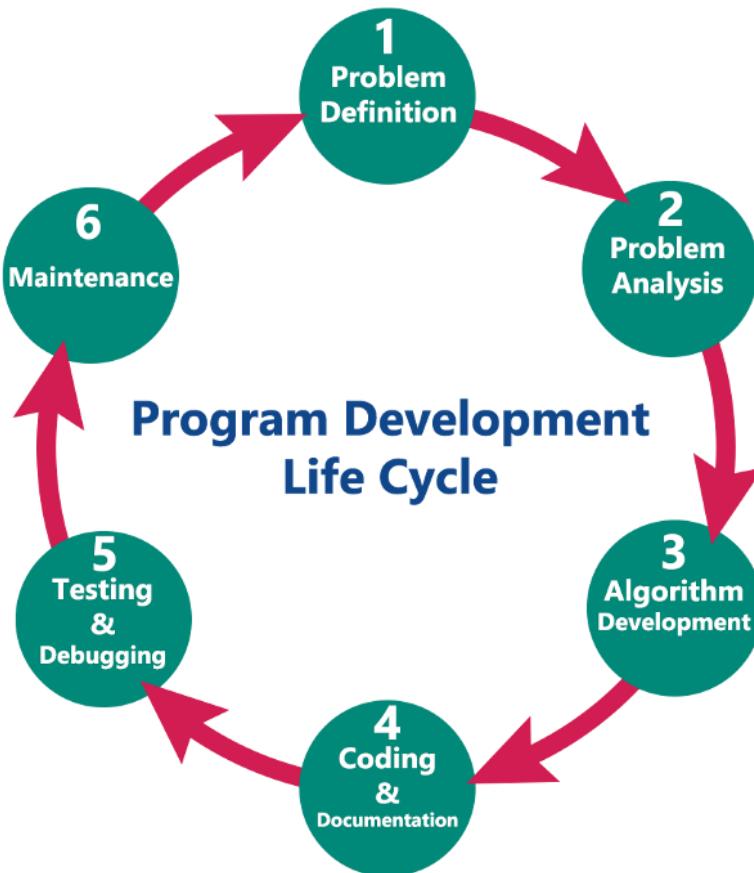
Algoritma, Flowchart, dan  
Pengkondisian Lanjutan

# Algoritma, Flowchart, dan Pengkondisian Lanjutan



# Session I

## Algoritma, Flowchart, dan Pemrograman (Python Lanjutan)



Sumber: <https://www.dictio.id/t/apa-saja-langkah-langkah-yang-dilakukan-dalam-pemrograman/12667/2>

# Dasar Penyusunan Algoritma

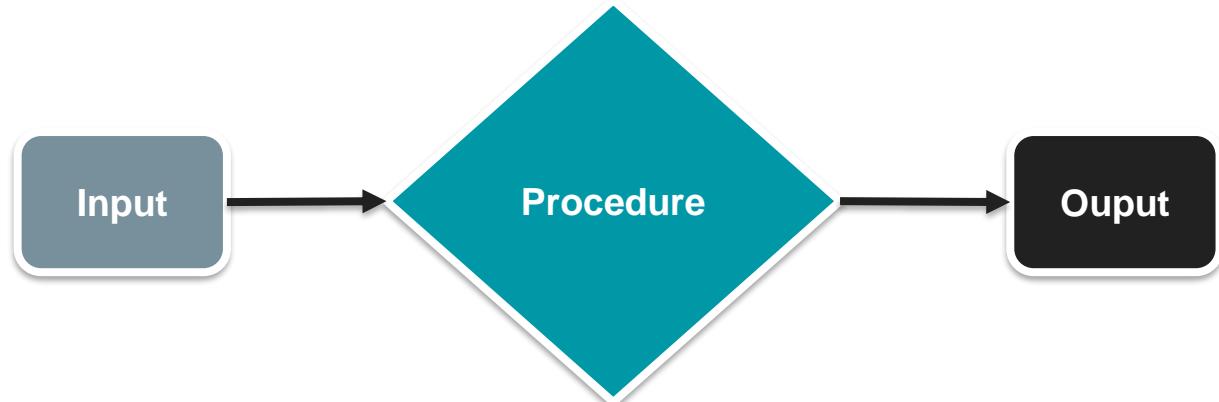
Finiteness

Definitiness

Masukan

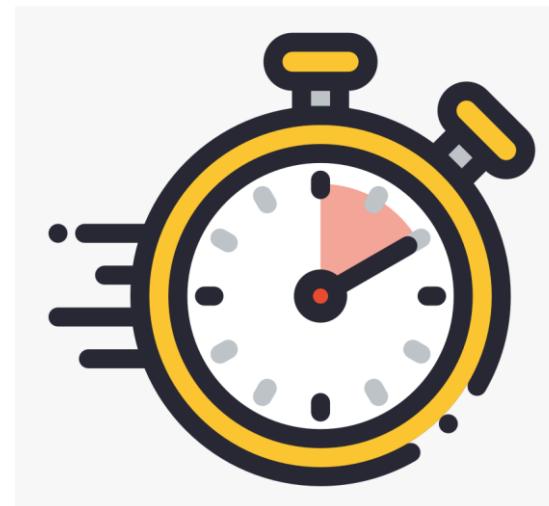
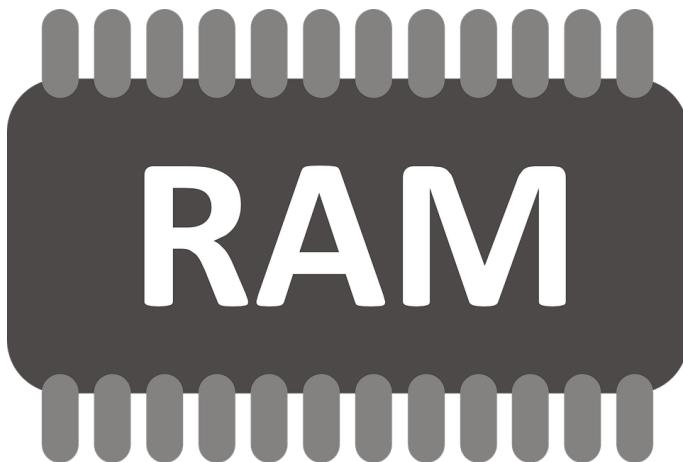
Keluaran

Efektivitas



Menurut Knuth (1973) dan juga Horowitz dkk. (1999)

# Bagaimana Algoritma yang efisien menurut kamu?

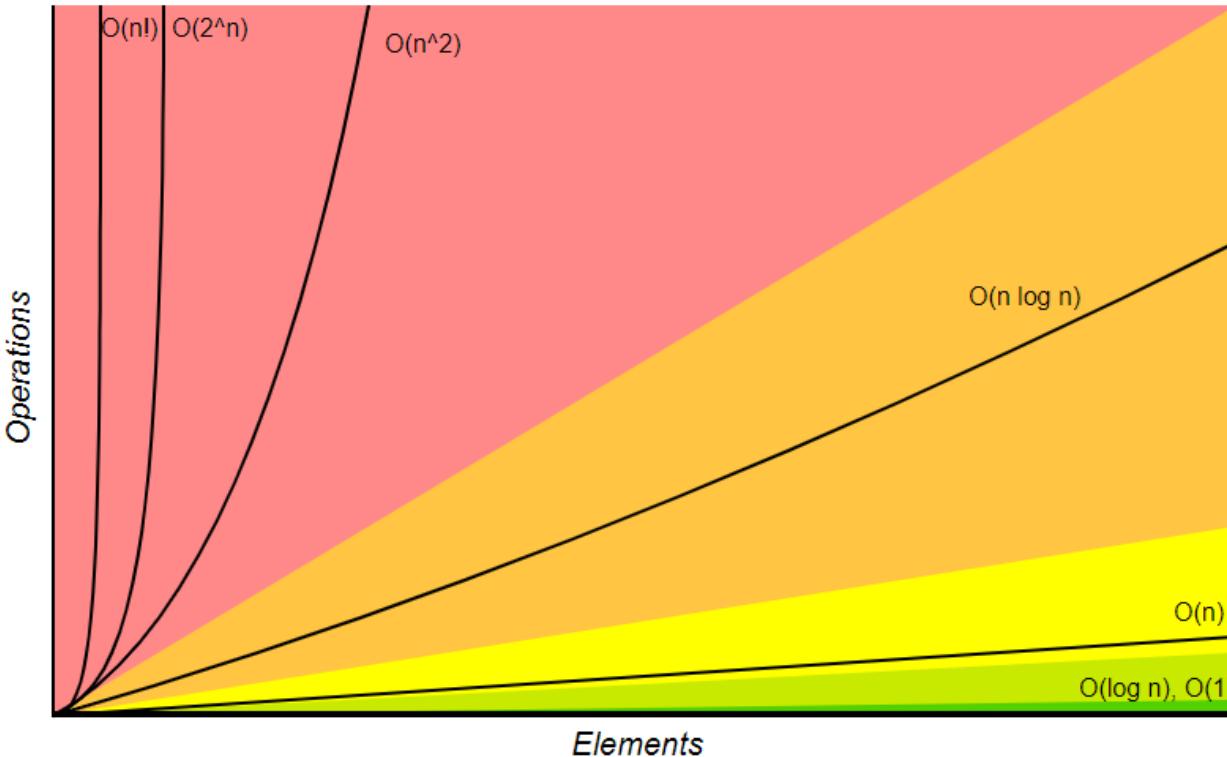


Data type	Description
bool_	Boolean (True or False) stored as a byte
int_	Default integer type (same as C long; normally either int64 or int32)
intc	Identical to C int (normally int32 or int64)
intp	Integer used for indexing (same as C ssize_t; normally either int32 or int64)
int8	Byte (-128 to 127)
int16	Integer (-32768 to 32767)
int32	Integer (-2147483648 to 2147483647)
int64	Integer (-9223372036854775808 to 9223372036854775807)
uint8	Unsigned integer (0 to 255)
uint16	Unsigned integer (0 to 65535)
uint32	Unsigned integer (0 to 4294967295)
uint64	Unsigned integer (0 to 18446744073709551615)
float_	Shorthand for float64
float16	Half-precision float: sign bit, 5 bits exponent, 10 bits mantissa
float32	Single-precision float: sign bit, 8 bits exponent, 23 bits mantissa
float64	Double-precision float: sign bit, 11 bits exponent, 52 bits mantissa
complex_	Shorthand for complex128
complex64	Complex number, represented by two 32-bit floats
complex128	Complex number, represented by two 64-bit floats

Contoh alokasi penggunaan  
memori untuk data type  
pada library Numpy

# Big-O Complexity Chart

Horrible Bad Fair Good Excellent



$O(N!)$	Factorial
$O(2^N)$	Exponential
$O(N^3)$	Cubic
$O(N^2)$	Quadratic
$O(N \log N)$	$N \times \log N$
$O(N)$	Linear
$O(\log N)$	Logarithmic
$O(1)$	Constant

Sumber: <https://www.bigocheatsheet.com/> dan <https://towardsdatascience.com>

## Common Data Structure Operations

Data Structure	Time Complexity								Space Complexity	
	Average				Worst					
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion		
Array	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	
Stack	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	
Queue	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	
Singly-Linked List	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	
Doubly-Linked List	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta(1)$	$\Theta(n)$	
Skip List	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n \log(n))$	
Hash Table	N/A	$\Theta(1)$	$\Theta(1)$	$\Theta(1)$	N/A	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	
Binary Search Tree	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	
Cartesian Tree	N/A	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	N/A	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	
B-Tree	$\Theta(\log(n))$	$\Theta(n)$								
Red-Black Tree	$\Theta(\log(n))$	$\Theta(n)$								
Splay Tree	N/A	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	N/A	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(n)$	
AVL Tree	$\Theta(\log(n))$	$\Theta(n)$								
KD Tree	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(\log(n))$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$	

# Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	
Quicksort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(\log(n))$
Mergesort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
Timsort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$
Heapsort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(1)$
Bubble Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Insertion Sort	$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Selection Sort	$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$	$O(1)$
Tree Sort	$\Omega(n \log(n))$	$\Theta(n \log(n))$	$O(n^2)$	$O(n)$
Shell Sort	$\Omega(n \log(n))$	$\Theta(n(\log(n))^2)$	$O(n(\log(n))^2)$	$O(1)$
Bucket Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n^2)$	$O(n)$
Radix Sort	$\Omega(nk)$	$\Theta(nk)$	$O(nk)$	$O(n+k)$
Counting Sort	$\Omega(n+k)$	$\Theta(n+k)$	$O(n+k)$	$O(k)$
Cubesort	$\Omega(n)$	$\Theta(n \log(n))$	$O(n \log(n))$	$O(n)$

Sumber: <https://www.bigocheatsheet.com/>



## Session II

**Menyusun Algoritma, Flowchart  
dan Pembuatan Program  
(Python Lanjutan)**

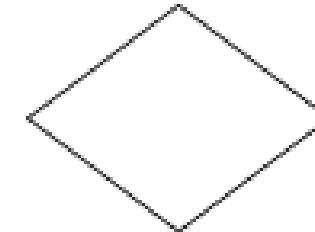
# Simbol standar Flowchart



Terminator



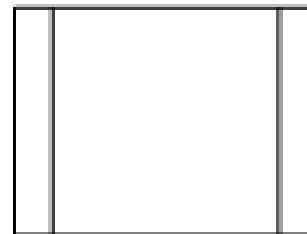
Proses



Keputusan



Input/Output



Proses terdefinisi



Konektor



Simbol Arus

SIMBOL TERMINAL		Menunjukkan permulaan (start) atau akhir (stop) dari suatu proses
SIMBOL INPUT/OUTPUT		Menunjukkan proses input/output yang terjadi
SIMBOL PROSES		Digunakan untuk menunjukkan kegiatan yang dilakukan oleh computer
SIMBOL KEPUTUSAN		Digunakan untuk memilih keputusan berdasarkan kondisi yang ada
SIMBOL ARUS		Digunakan untuk menghubungkan satu symbol dengan symbol lainnya. Simbol ini berfungsi untuk menunjukkan garis alir dari proses.

# Belajar menyusun algoritma dari beberapa kasus berikut

**Konversi bilangan  
Desimal (basis 10)  
ke Hexadecimal  
(basis 16), Octal  
(basis 8), dan  
Biner (basis 2)**

**Menghitung luas  
dan keliling  
bangun datar**

**Menghitung Volume dan  
Luas Permukaan Bangun  
Ruang**

# Algoritma Menghitung Luas & Keliling Persegi



Sisi



Apa yang kamu ketahui tentang persegi ini?

# Rumus Menghitung Luas & Keliling Persegi



**Rumus luas dan keliling Persegi adalah**

$$\text{Luas} = s^2$$

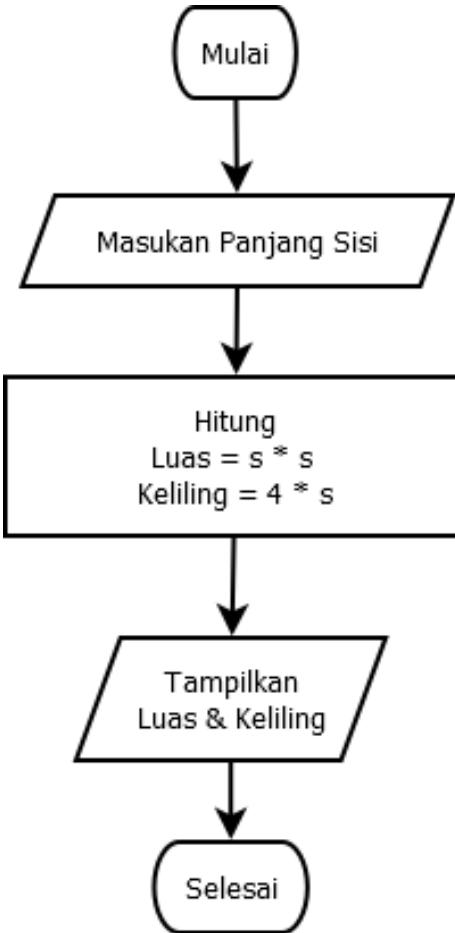
$$\text{Keliling} = 4 \times s$$

# Algoritma Menghitung Luas & Keliling Persegi



- Mulai
- Masukan sisi ( $s$ )
- Hitung luas =  $s^2$
- Hitung keliling =  $4 * s$
- Tampilkan Luas dan Keliling
- Selesai

# Flowchart Menghitung Luas & Keliling Persegi



# Code Menghitung Luas & Keliling Persegi



```
1 print("MENGHITUNG LUAS & KELILING PERSEGI")
2
3 s = float(input("\nMasukan Panjang Sisi dalam cm: "))
4
5 luas = s**2
6 keliling = 4 * s
7
8 print("\nLuas Persegi \t\t:",luas, "cm^2")
9 print("Keliling Persegi\t:",keliling, "cm")
```

# Rumus Menghitung Volume dan Luas Permukaan Bola

## Rumus Volume dan Luas Permukaan Bola

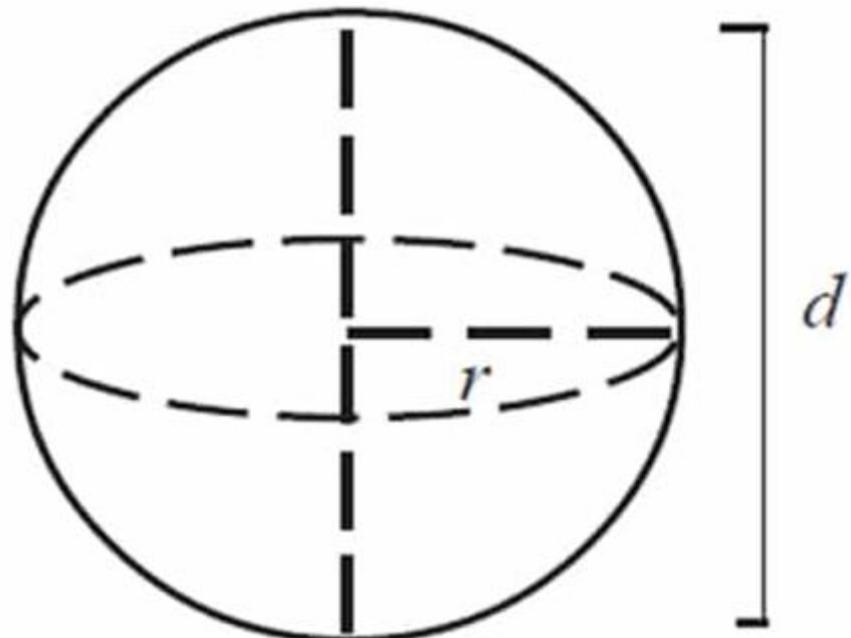
$$\text{Volume} = \frac{4}{3} \times \pi \times r^3$$

$$\text{Luas Permukaan} = 4 \times \pi \times r^2$$

# Algoritma Menghitung Volume & Luas Permukaan Bola

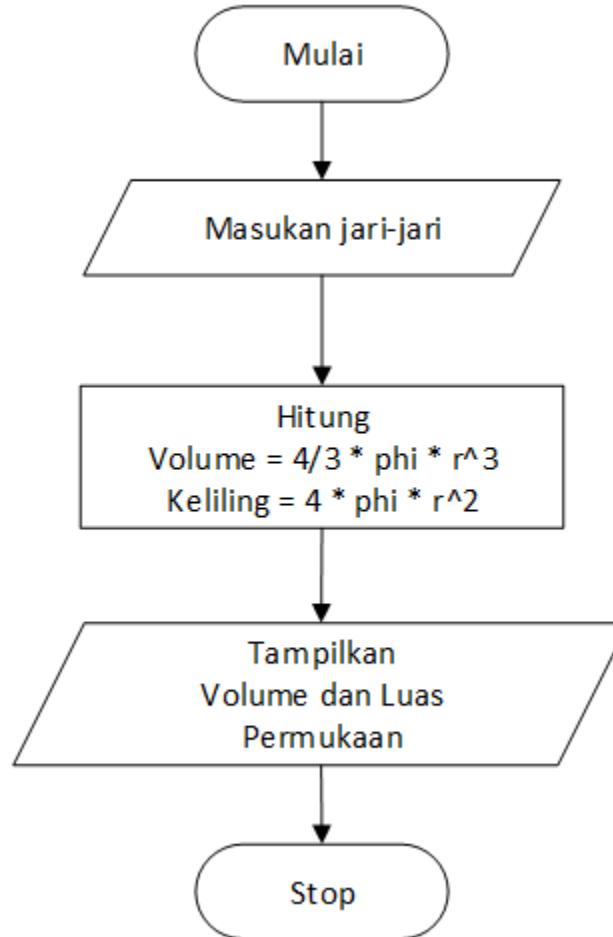
- Mulai
- Masukan jari-jari
- Hitung volume =  $4/3 * \pi * r^3$
- Hitung luas permukaan =  $4 * \pi * r^2$
- Tampilkan volume dan luas permukaan bola
- Selesai

# Algoritma Menghitung Volume & Luas Permukaan Bola



Apa yang kamu ketahui tentang bangun ruang ini?

# Flowchart Menghitung Volume & Luas Permukaan Bola



# Code Menghitung Volume & Luas Permukaan Bola



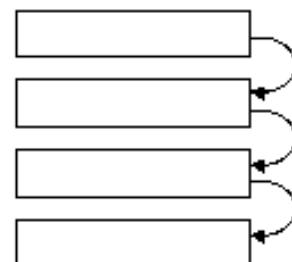
```
1 print("MENGHITUNG VOLUME & LUAS PERMUKAAN BOLA")
2
3 jari_jari = float(input("\nMasukan jari-jari dalam cm: "))
4
5 phi = 3.14
6 volume = 4/3 * phi * jari_jari**3
7 luas_permukaan = 4 * phi * jari_jari**2
8
9 print("\nVolume Bola adalah \t\t:",volume, "cm^3")
10 print("Luas Permukaan Bola adalah\t:",luas_permukaan, "cm^2")
```



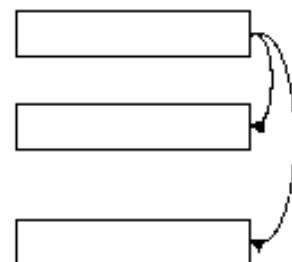
Sumber: Google

# Jenis Struktur Dasar Algoritma

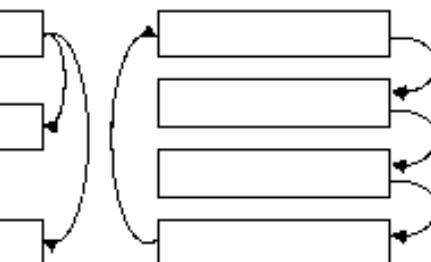
- Sekuensial
- Seleksi
- Pengulangan



runtunan  
(sequence)



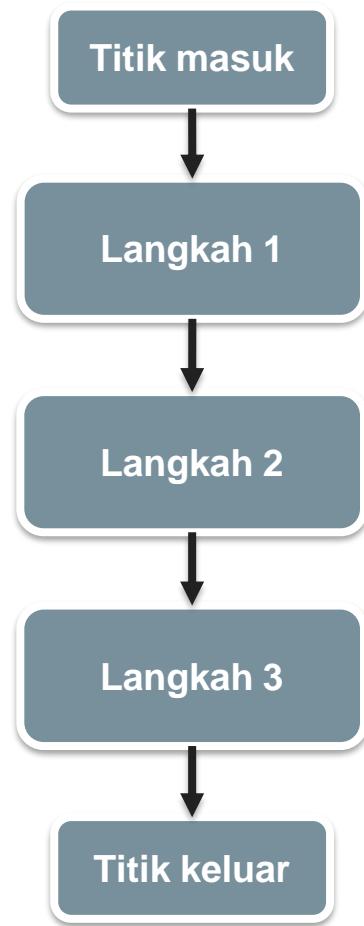
pemilihan  
(selection)



pengulangan  
(repetition)

Sumber Gambar: Google

## Struktur Sekuensial



## Struktur Seleksi

Titik masuk

Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills For Future Jobs

Ya

Kondisi  
benar

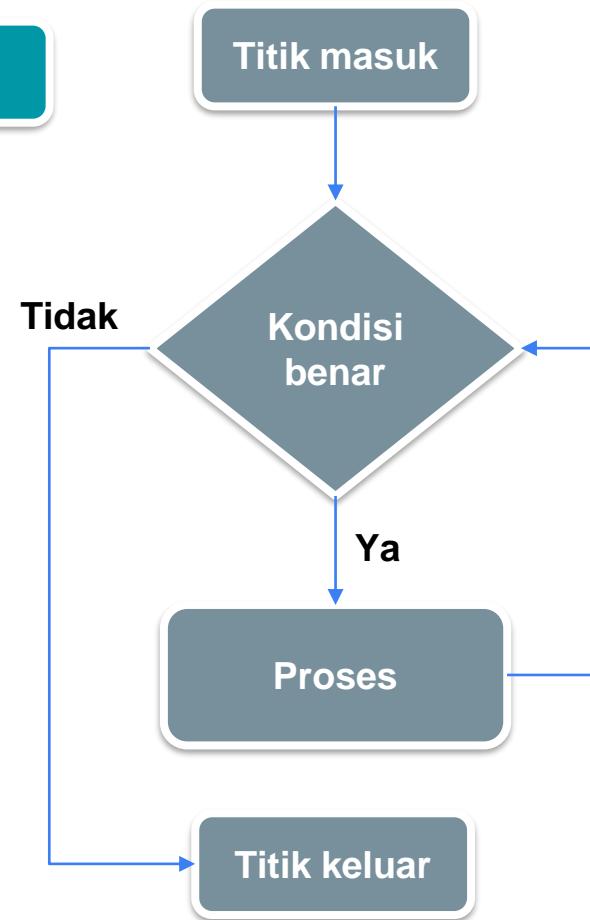
Tidak

Langkah 1

Langkah 2

Titik keluar

## Struktur Perulangan

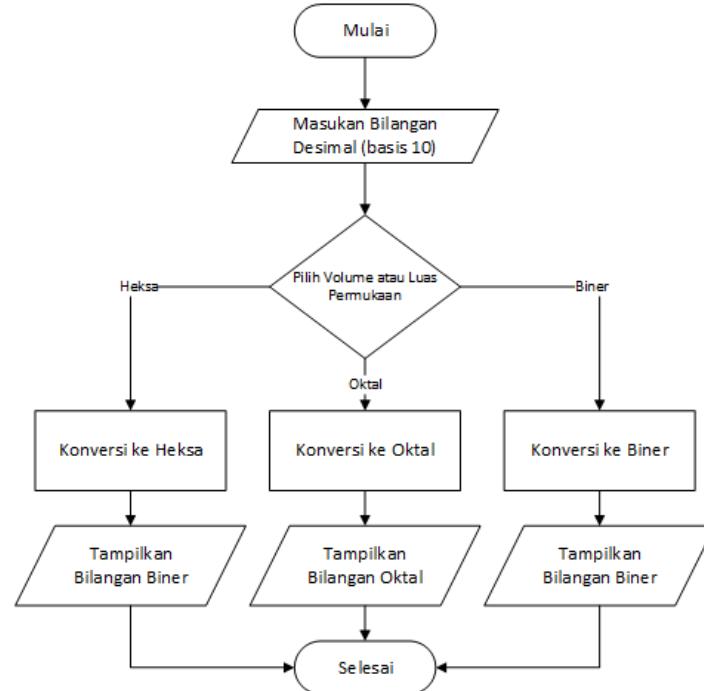


# Algoritma Percabangan Mengkonversi Bilangan



- Mulai
- Masukkan bilangan desimal yang ingin di konversi
- Pilih: a. Konversi ke Hexa, b. Konversi ke Oktal,  
dan c. konversi ke Biner
- Tampilkan hasil Konversi
- Selesai

# Algoritma Percabangan Mengkonversi Bilangan



# Algoritma Percabangan Menghitung Volume & Luas Permukaan Bola



- Mulai
- Masukan jari-jari
- Pilih: 1. Hitung Volume, 2. Hitung Luas Permukaan

Jika Volume

Hitung volume =  $4/3 * \pi * r^3$

Tampilkan Volume Bola

Selesai

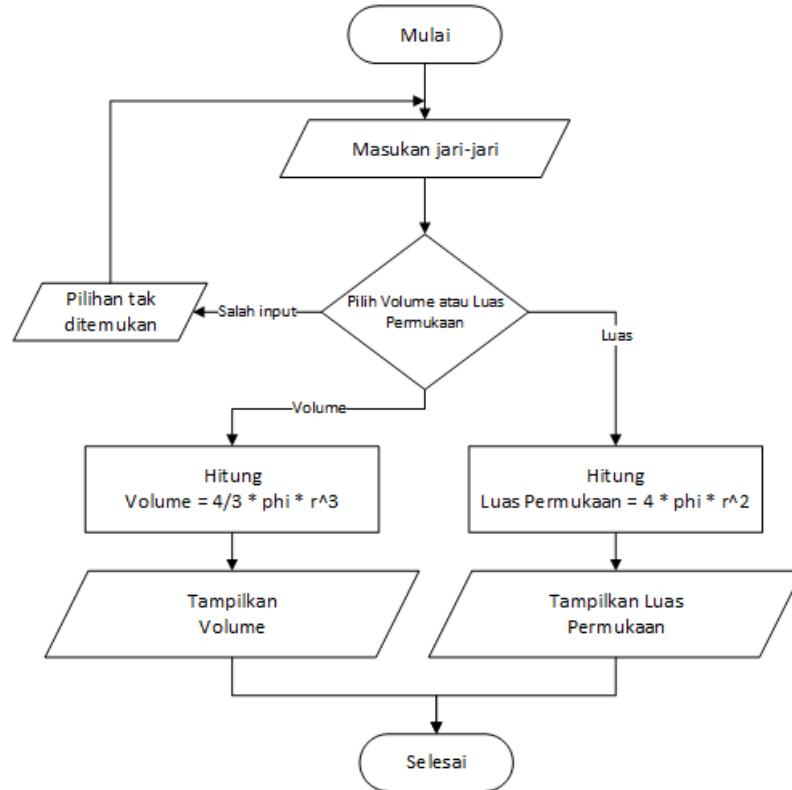
Jika Luas Permukaan

Hitung luas permukaan =  $4 * \pi * r^2$

Tampilkan luas permukaan bola

Selesai

# Flowchart Percabangan Menghitung Volume & Luas Permukaan Bola



# Algoritma Perulangan dan Percabangan Menghitung Volume & Luas Permukaan Bola



- Mulai
- Masukan pilihan untuk melanjutkan atau keluar dari looping
- Pilih: 1. Lanjut Menghitung, 2. Exit Looping

Jika Menghitung

Masukkan jari-jari

Pilih: 1. Hitung Volume, 2 Hitung Luas Permukaan

Jika Volume

Hitung volume =  $4/3 * \pi * r^3$

Tampilkan Volume Bola

Kembali ke Pilihan Melanjutkan Hitungan atau Exit

Jika Luas Permukaan

Hitung luas permukaan =  $4 * \pi * r^2$

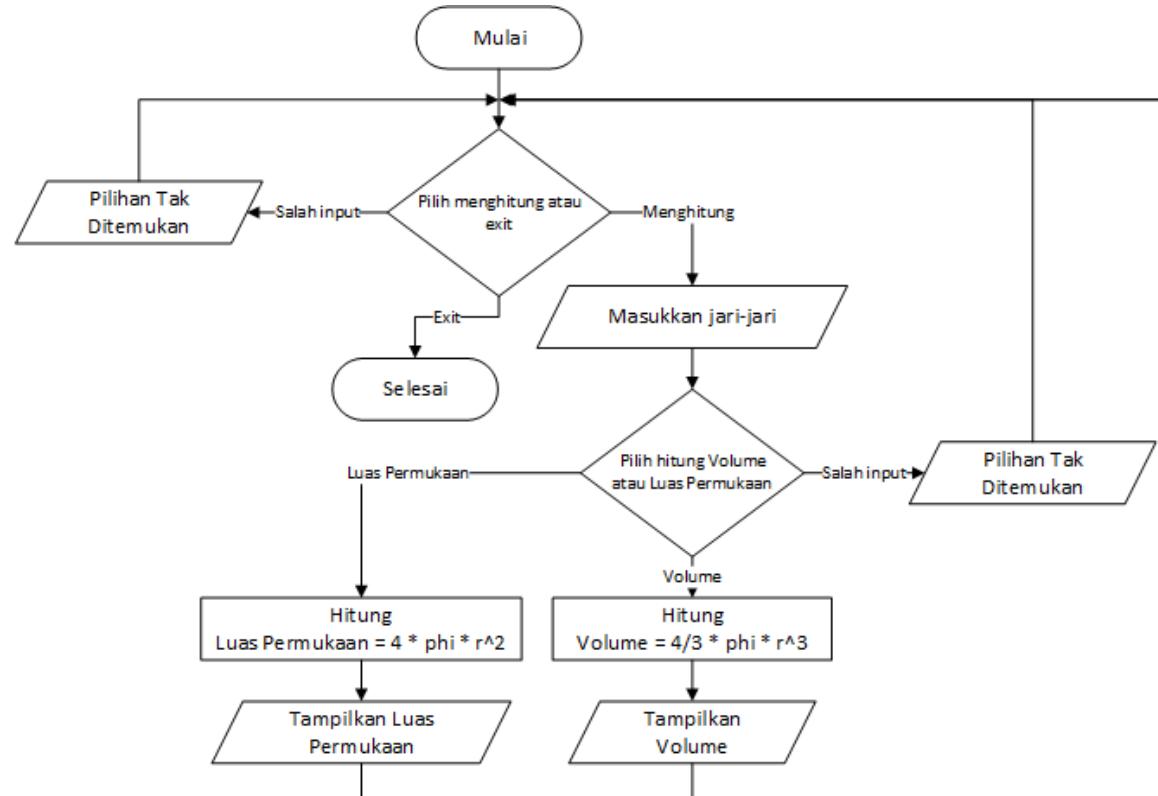
Tampilkan luas permukaan bola

Kembali ke Pilihan Melanjutkan Hitungan atau Exit

Jika pilih Exit

Selesai

# Flowchart Percabangan dan Perulangan Menghitung Volume & Luas Permukaan Bola



# Activity

## IF Bersarang (Nested IF)

Jika suatu kondisi if terpenuhi, maka kita bisa menambahkan syarat lagi di dalamnya untuk seleksi yang lebih spesifik/mendalam.

# Sintak dalam IF Bersarang Tanpa Percabangan



```
if (kondisi) :  
    if (kondisi):  
        -perintah yang akan dieksekusi jika kondisi terpenuhi-
```

# Sintak dalam IF Bersarang Dengan Percabangan

```
if (kondisi) :  
    if (kondisi) :  
        -perintah yang akan dieksekusi jika kondisi terpenuhi-  
    else :  
        -perintah yang akan dieksekusi jika kondisi terpenuhi-  
else :  
    -perintah yang akan dieksekusi jika kondisi terpenuhi-
```

# Merangkum Materi

# Quiz

1. Apa syarat algoritma yang efisien?
2. Apa itu Big O (n)?
3. Kesulitan apa yang kalian peroleh dari menyusun algoritma, membuat flowchart, hingga program?



THANK YOU

# AI Mastery Course



## Module 1

Introduction to everything

### Section

Python data types





# Learning Objectives

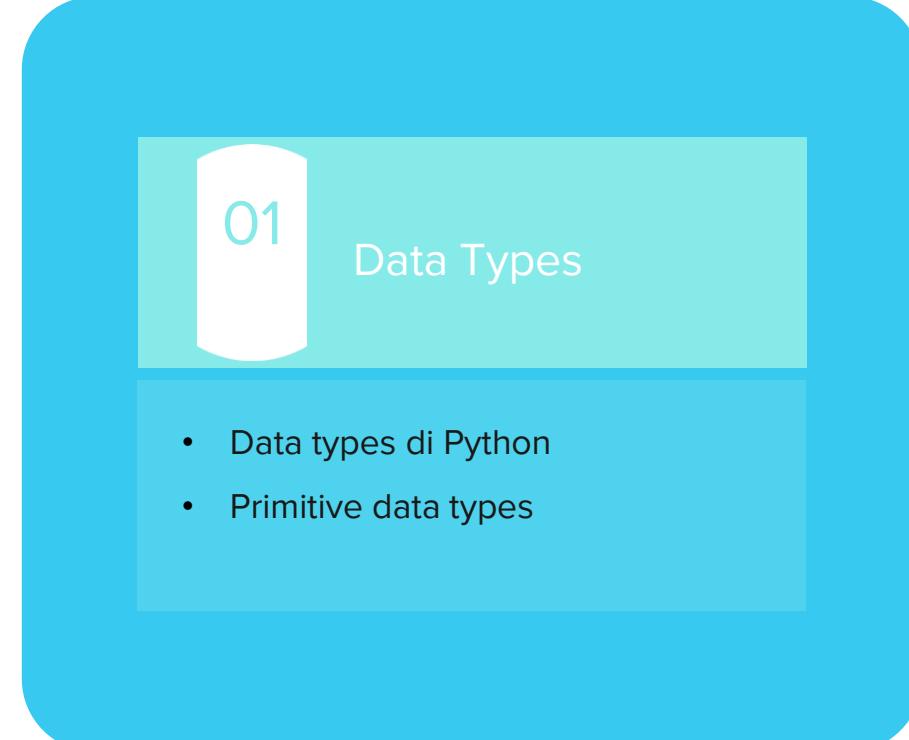
Di akhir modul ini, kita akan dapat:

- Memahami python data types
- Mengeksplorasi primitive data types di python
- Mempelajari bagaimana mendefinisikan, memanipulasi dan menggunakan non-primitive data types
- Memahami bagaimana menggunakan functions untuk type conversion



# Agenda

- 01 DATA TYPES
  - Data types di Python
  - Primitive data types
- 02 SEQUENCES
  - List
  - Tuple
  - String
- 03 MAPPING & SET
  - Dictionary
  - Set
- 04 CONVERSION
  - Type conversion – primitive
  - Type conversion – non-primitive
- 05 KESIMPULAN
  - Kuis
  - Ringkasan



01 Data Types

- Data types di Python
- Primitive data types

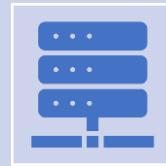
# Data structures in python

- Cara untuk mengatur dan menyimpan data
- Tipe data variabel menentukan operasi mana yang dapat diterapkan padanya



## Primitive data types

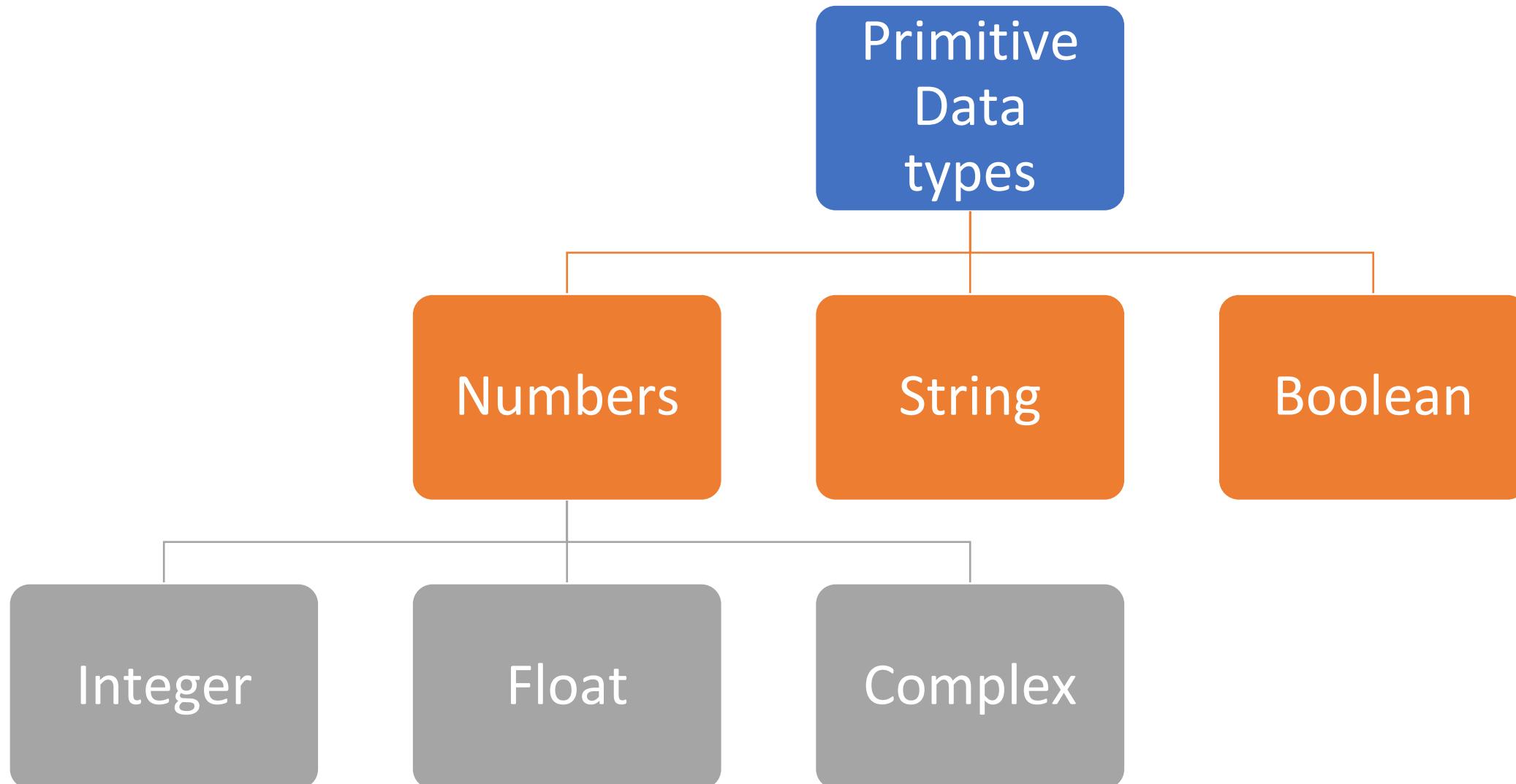
Struktur data dasar, blok bangunan untuk manipulasi data dan berisi nilai data yang murni dan sederhana.



## Non primitive data types

Berasal dari primitive data, digunakan untuk menyimpan kumpulan nilai.

# Primitive data types di Python



# Primitive data types

- `type()` fungsi dapat digunakan untuk mengetahui kelas mana yang dimiliki variabel atau nilai
- Bilangan bulat dapat memiliki panjang berapa pun, hanya dibatasi oleh memori yang tersedia
- Float in python akurat hingga 15 tempat desimal
- 5 adalah bilangan bulat, 5.0 adalah bilangan floating-point
- Bilangan kompleks ditulis dalam bentuk,  $a + bj$ , di mana  $a$  adalah bagian real dan  $b$  adalah bagian imajiner

The image shows a laptop screen displaying a Jupyter Notebook titled "Primitive data types in python". The notebook contains three code examples demonstrating different primitive data types in Python:

- Integer**:  
In [1]:  
1 x = 5  
2 print(x); type(x)  
5  
<class 'int'>
- Float**:  
In [2]:  
1 y = 2.5  
2 print(y); type(y)  
2.5  
Out[2]: float
- Complex**:  
In [3]:  
1 m = 5 + 8j  
2 print(m); type(m)  
(5+8j)  
Out[3]: complex

# Primitive data types

## String

- Teks diapit dengan tanda kutip tunggal atau ganda

## Boolean

- Berupa dua nilai Benar atau Salah

### String

```
In [4]: 1 k = "Hello from Python"  
2 print(k); type(k)
```

Hello from Python

```
Out[4]: str
```

### Boolean

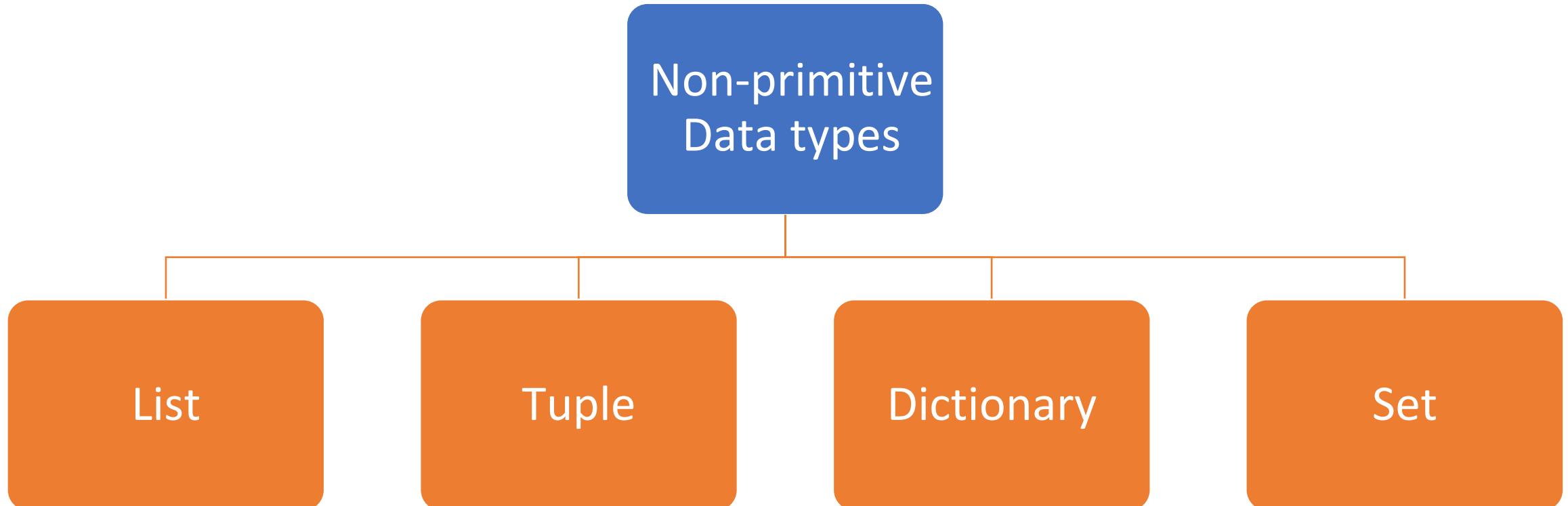
```
In [5]: 1 m = True  
2 n = False  
3 print(m); type(m)  
4 print(n); type(n)
```

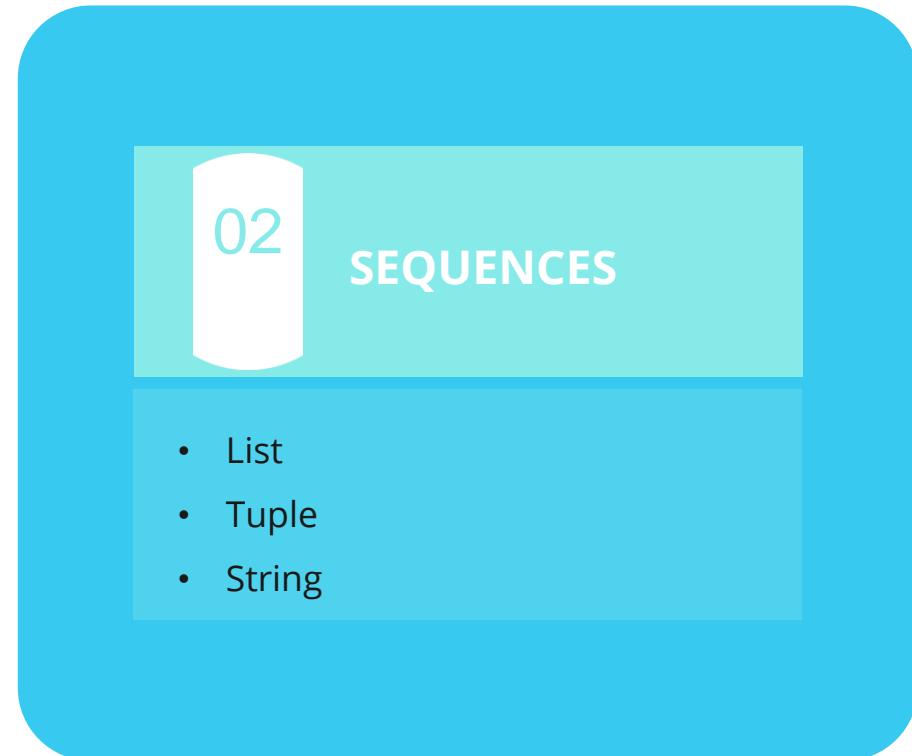
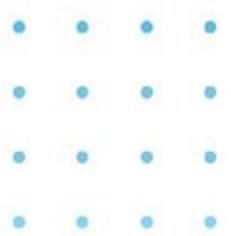
True

False

```
Out[5]: bool
```

# Non-primitive data types di Python





02 SEQUENCES

- List
- Tuple
- String



# List

”

berisi item yang dipisahkan dengan koma dan diapit dalam tanda kurung siku ([]).



♣

nilai yang disimpan dalam daftar dapat diakses menggunakan operator irisan ( [ ] dan [ : ] ) dengan indeks mulai dari 0 di awal daftar.



Mengakomodasi pengulangan elemen dan elemen dari tipe data yang berbeda



Tipe data yang bisa berubah – daftar dapat dimodifikasi

Mendukung pengindeksan negatif tanda plus ( + ) adalah operator rangkaian daftar, dan tanda asterisk ( \* ) adalah operator pengulangan.

# List

Pengindeksan dimulai dengan 0 (kiri ke kanan). Python juga mendukung pengindeksan negatif (kanan ke kiri).

```
1 mylist = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
2 tinylist = [123, 'john']
3
4 print(mylist)           # Prints complete list
5 print(type(mylist))     # Prints type of object as List
6 print(len(mylist))      # Prints length of the list
7 print(mylist[0])         # Prints first element of the list
8 print(mylist[1:3])       # Prints elements starting from 2nd till 3rd
9 print(mylist[2:])        # Prints elements starting from 3rd element
10 print(tinylist * 2)      # Prints list two times
11 print(mylist + tinylist) # Prints concatenated lists

['abcd', 786, 2.23, 'john', 70.2]
<class 'list'>
5
abcd
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
```

# List is mutable

Daftar memiliki berbagai metode yang tersedia yang dapat digunakan untuk memanipulasi daftar.

```
1 mylist = [ 'abcd', 786 , 2.23, 'john', 70.2 ]
2 tinylist = [123, 'john']
3
4 print(mylist)
5 mylist.append('wiley') # appending an element to a list
6 print(mylist)
7 mylist.insert(2,'python') # inserting an element at index 2
8 print(mylist)
9 mylist.remove(786) # removing an element from a list
10 print(mylist)
11 mylist.pop(3) # removing an element from a specific position
12 print(mylist)
13 mylist.extend(tinylist) # extending a list with elements from another list
14 print(mylist)

['abcd', 786, 2.23, 'john', 70.2]
['abcd', 786, 2.23, 'john', 70.2, 'wiley']
['abcd', 786, 'python', 2.23, 'john', 70.2, 'wiley']
['abcd', 'python', 2.23, 'john', 70.2, 'wiley']
['abcd', 'python', 2.23, 70.2, 'wiley']
['abcd', 'python', 2.23, 70.2, 'wiley', 123, 'john']
```

# Tuple

berisi item yang dipisahkan dengan koma dan diapit dalam tanda kurung () .



nilai yang disimpan dalam tuple dapat diakses menggunakan operator irisan ( [ ] ) dan ( [ : ] ) dengan indeks mulai dari 0 di awal tuple.



Mengakomodasi pengulangan elemen dan elemen dari tipe data yang berbeda



Mendukung pengindeksan negatif tanda plus ( + ) adalah operator rangkaian tuple, dan tanda asterisk ( \* ) adalah operator pengulangan.



Objek yang tidak dapat diubah – hanya baca

# Tuple

Tuple tidak mendukung modifikasi; itu adalah objek yang tidak dapat diubah.

```
tuple_items = ('Cars',786,2.23,'toyota',70.2)
tinytuple = (123,'camry')

print(tuple_items)                  #Prints complete tuple
print(tuple_items[0])              #Prints first element of complete tuple
print(tuple_items[1:3])            #prints elements starting from second till third
print(tuple_items[2:])             #Prints elemts starting from third till end
print(tinytuple*2)                 #Prints tuple two times
print(tuple_items+tinytuple)        #Prints concatenated tuples

('Cars', 786, 2.23, 'toyota', 70.2)
Cars
(786, 2.23)
(2.23, 'toyota', 70.2)
(123, 'camry', 123, 'camry')
('Cars', 786, 2.23, 'toyota', 70.2, 123, 'camry')
```

# List v/s Tuple

List	Tuple
Dapat didefinisikan menggunakan tanda kurung siku []	Dapat didefinisikan menggunakan tanda kurung ()
Bisa berubah	Tidak bisa berubah
Implikasi dari iterasi memakan waktu dalam list	Implikasi dari iterasi jauh lebih cepat dalam tuple
Kurang hemat memori	Lebih hemat memori
Kesalahan dan perubahan yang tidak terduga dapat dengan mudah terjadi dalam list	Kesalahan dan perubahan yang tidak terduga jarang terjadi pada tuple

# String



Merupakan primitive data types tapi mendukung banyak fungsi dari koleksi (list and tuple).



Dapat dibuat dengan melampirkan urutan karakter dalam sepasang tanda kutip tunggal atau ganda.



String adalah kumpulan abjad, kata, atau karakter lainnya.

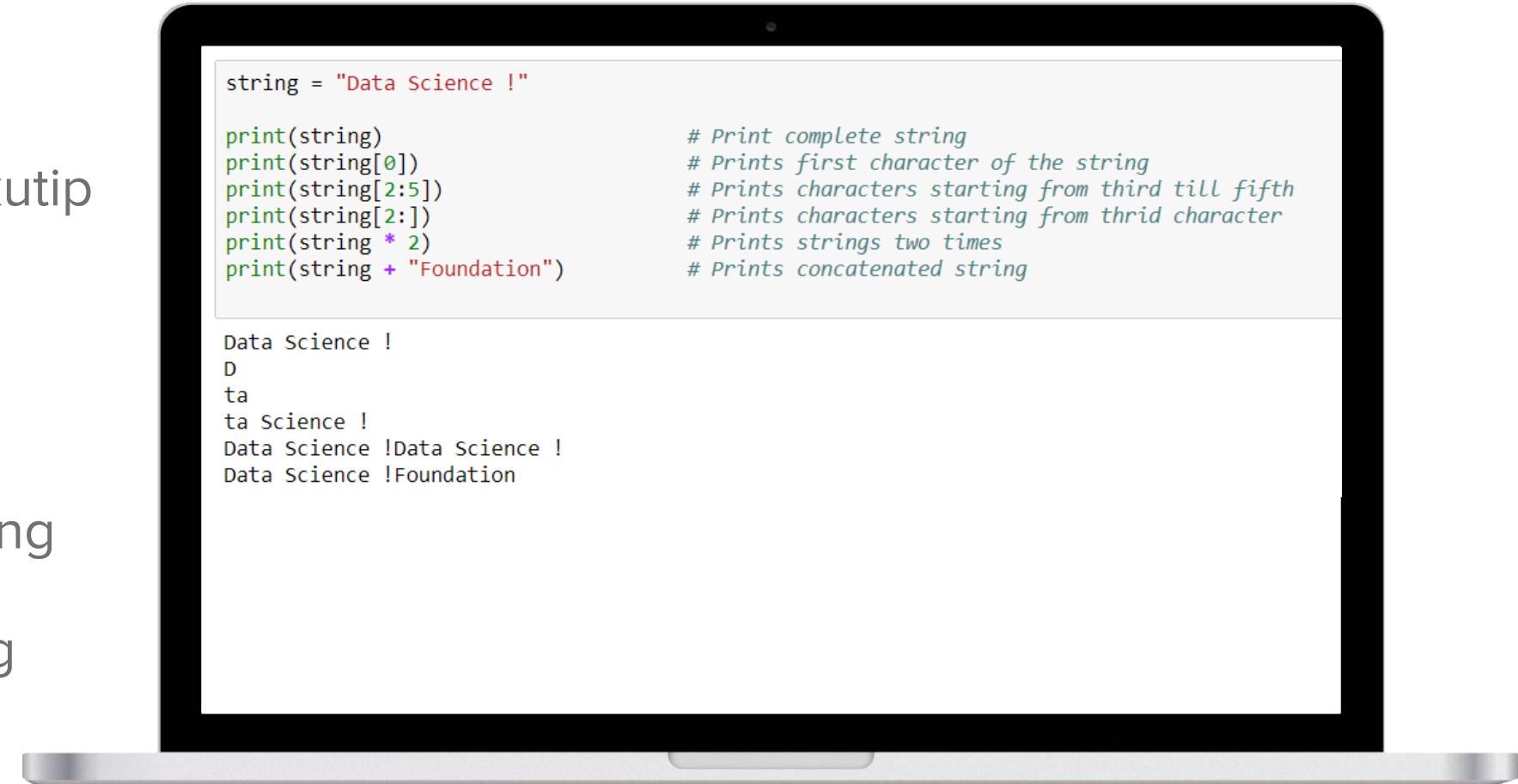


Tidak dapat diubah, kuat – memiliki banyak metode bawaan untuk eksplorasi dan manipulasi.

# String

- String multi-baris dilambangkan menggunakan tanda kutip tiga, "" atau """

Strings are immutable objects which means once we declare a string we can't update the already declared string



```
string = "Data Science !"

print(string)          # Print complete string
print(string[0])       # Prints first character of the string
print(string[2:5])     # Prints characters starting from third till fifth
print(string[2:])      # Prints characters starting from thrid character
print(string * 2)       # Prints strings two times
print(string + "Foundation") # Prints concatenated string

Data Science !
D
ta
ta Science !
Data Science !Data Science !
Data Science !Foundation
```



03

## MAPPING & SET

- Dictionary
- Set



# Dictionary



berisi item yang dipisahkan dengan koma dan diapit dalam tanda kurung kurawal {}.



Setiap item memiliki dua komponen – kunci dan nilai



'Kunci' mengidentifikasi item, dan 'nilai' menyimpan nilai item. Nilai dapat diakses dengan memasukkan kunci dalam tanda kurung siku ([]).



kunci tidak dapat diubah (angka, String, atau Tuples) dan unik, nilainya dapat berupa jenis apa pun.



Objek yang dapat berubah – dapat dimodifikasi

# Dictionary

- String multi-baris dilambangkan menggunakan tanda kutip tiga, "" atau """
- String adalah objek yang tidak dapat diubah yang berarti setelah kami mendeklarasikan string, kami tidak dapat memperbarui string yang sudah dideklarasikan.



```
dict={}
dict['one'] = "This is one"
dict[2] = "This is two"

tinydict = {'name':'john', 'code':6734 , 'dept':'sales'}

print(dict['one'])          # Prints value for 'one' key
print(dict[2])             # Prints value for 2 key
print(tinydict)            # Prints complete dictionary
print(tinydict.keys())     # Prints all keys of the dictionary
print(tinydict.values())    # Prints all the values of each key

This is one
This is two
{'name': 'john', 'code': 6734, 'dept': 'sales'}
dict_keys(['name', 'code', 'dept'])
dict_values(['john', 6734, 'sales'])
```

# Set



Kumpulan nilai yang dipisahkan dengan koma dan diapit dalam tanda kurung kurawal {}.



Tidak berurutan, bisa berubah dan tidak mendukung pengulangan elemen



Terutama digunakan untuk operasi teori himpunan seperti serikat pekerja, persimpangan dll.



Set adalah tipe data yang bisa berubah, sehingga dapat dimodifikasi. Frozenset tidak dapat diubah.

# Set

- Sets bersifat iretable
- Set juga bisa berubah, yaitu, kita dapat memodifikasi elemen set setelah pembuatan

```
1 setA = {2,4,5,6,5,6,5,4}           # Defining setA
2 setB = {1,2,4,8,8}                 # Defining setB
3
4 print(setA, type(setA))          # Prints the variable setA and its data type
5 print(len(setA))                 # Prints the length of the setA
6 setA.add(9) ; print(setA)         # Adding new value of the setA
7 setA.remove(6) ; print(setA)       # Removing 6 from setA
8 print(setA.union(setB))          # Union operation = setA U setB
9 print(setA.intersection(setB))    # Intersection operation = setA n setB

{2, 4, 5, 6} <class 'set'>
4
{2, 4, 5, 6, 9}
{2, 4, 5, 9}
{1, 2, 4, 5, 8, 9}
{2, 4}
```



04

## CONVERSION

- Type conversion – primitive
- Type conversion – non-primitive



# Type Conversion

Type Conversion

Implicit type conversion

Explicit type conversion

# Type Conversion

Function	Description
<b>int(x [,base])</b>	Mengubah x menjadi integer. base menspesifikasi the base jika x adalah string.
<b>long(x [,base] )</b>	Mengubah x menjadi long integer. base menspesifikasi the base jika x adalah string.
<b>float(x)</b>	Mengubah x menjadi floating-point number.
<b>complex(real [,imag])</b>	Membuat sebuah complex number.
<b>str(x)</b>	Mengubah object x menjadi sebuah representasi string.
<b>eval(str)</b>	Mengevaluasi sebuah string dan mengembalikan sebuah object.
<b>tuple(s)</b>	Mengubah s menjadi sebuah tuple.
<b>list(s)</b>	Mengubah s menjadi sebuah list.
<b>set(s)</b>	Mengubah s menjadi sebuah set.
<b>dict(d)</b>	Membuat sebuah dictionary. d merupakan sebuah sequence dari (key,value) tuples.
<b>frozenset(s)</b>	Mengubah s menjadi sebuah frozen set.
<b>chr(x)</b>	Mengubah sebuah integer menjadi sebuah character.
<b>ord(x)</b>	Mengubah sebuah single character menjadi integer value.
<b>hex(x)</b>	Mengubah sebuah integer menjadi sebuah hexadecimal string.
<b>oct(x)</b>	Mengubah sebuah integer menjadi sebuah octal string

# Type Conversion : primitive

```
1 x = 2.5; print(x,type(x),end="    >>>    ")
2 y = int(x) ; print(y, type(y)) # converting float to int
3 x = 1; print(x,type(x),end="    >>>    ")
4 y = bool(x) ; print(y, type(y)) # converting int to boolean
5 x = "124"; print(x,type(x),end="    >>>    ")
6 y = int(x) ; print(y, type(y)) # converting string to int
7 x = "2.5"; print(x,type(x),end="    >>>    ")
8 y = float(x) ; print(y, type(y)) # converting string to float
9 x = True; print(x,type(x),end="    >>>    ")
10 y = str(x) ; print(y, type(y)) # converting boolean to str
11
```

```
2.5 <class 'float'>    >>>    2 <class 'int'>
1 <class 'int'>    >>>    True <class 'bool'>
124 <class 'str'>    >>>    124 <class 'int'>
2.5 <class 'str'>    >>>    2.5 <class 'float'>
True <class 'bool'>    >>>    True <class 'str'>
```

# Type Conversion : primitive

```
1 x = [2, 5, 'Hi', 12, 2.5]; print(x, type(x), end = "\n    >>>    ")
2 y = tuple(x) ; print(y, type(y)) # converting List to tuple
3 x = [2, 5, 'Hi', 12, 2, 2.5, 5]; print(x, type(x), end = "\n    >>>    ")
4 y = set(x) ; print(y, type(y)) # converting List to set
5 x = (2, 5, 'Hi', 12, 2, 2.5, 5); print(x, type(x), end = "\n    >>>    ")
6 y = list(x) ; print(y, type(y)) # converting tuple to list
7 x = [['name', 'john'], ['age', 22], ['city', 'delhi']]; print(x, type(x), end = "\n    >>>    ")
8 y = dict(x) ; print(y, type(y)) # converting List to dict

[2, 5, 'Hi', 12, 2.5] <class 'list'>
    >>> (2, 5, 'Hi', 12, 2.5) <class 'tuple'>
[2, 5, 'Hi', 12, 2, 2.5, 5] <class 'list'>
    >>> {2.5, 2, 5, 12, 'Hi'} <class 'set'>
(2, 5, 'Hi', 12, 2, 2.5, 5) <class 'tuple'>
    >>> [2, 5, 'Hi', 12, 2, 2.5, 5] <class 'list'>
[['name', 'john'], ['age', 22], ['city', 'delhi']] <class 'list'>
    >>> {'name': 'john', 'age': 22, 'city': 'delhi'} <class 'dict'>
```



05

## KESIMPULAN

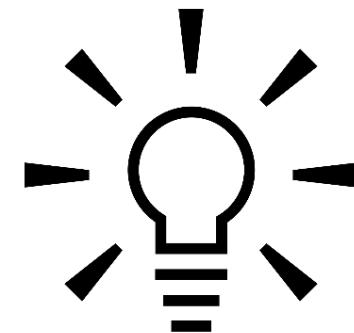
- Kuis
- Ringkasan

# Kuis

## Pertanyaan

Saat mengembangkan aplikasi kita ingin agar data diproses lebih cepat, dan membuat data hanya bisa dibaca saja. Tipe data yang dipilih untuk menyimpan data tersebut ?

- A. List
- B. Tuple
- C. Dictionary
- D. Set



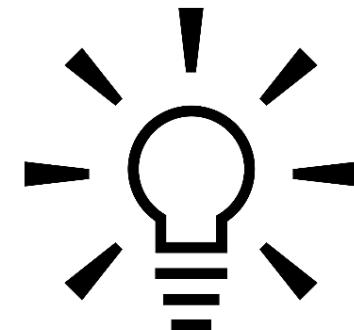
# Kuis

## Pertanyaan

Saat mengembangkan aplikasi kita ingin agar data diproses lebih cepat, dan membuat data hanya bisa dibaca saja. Tipe data yang dipilih untuk menyimpan data tersebut ?

- A. List
- B. Tuple
- C. Dictionary
- D. Set

Answer- B



# Ringkasan

- Struktur data adalah cara untuk menyimpan data dalam bahasa pemrograman
- Python memiliki lima tipe data primitif yang tidak dapat diubah – integer, float, string, Boolean, dan kompleks
- Python memiliki 4 tipe data non primitif – daftar, tupel, kamus, dan set
- Tuple tidak dapat diubah sedangkan daftar, kamus, dan set dapat diubah.
- Kita dapat mengetikkan tipe data primitif cor dari satu ke yang lain.





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

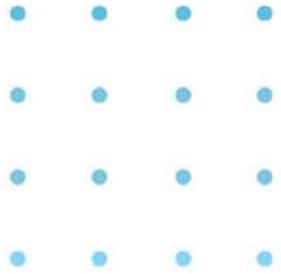
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  Orbit Future Academy
-  @OrbitFutureAcademyIn1
-  OrbitFutureAcademy
-  Orbit Future Academy

# AI Mastery Course



## Module 1

Introduction to everything

### Section

Python Functions, string manipulation & file handling





# Learning Objectives

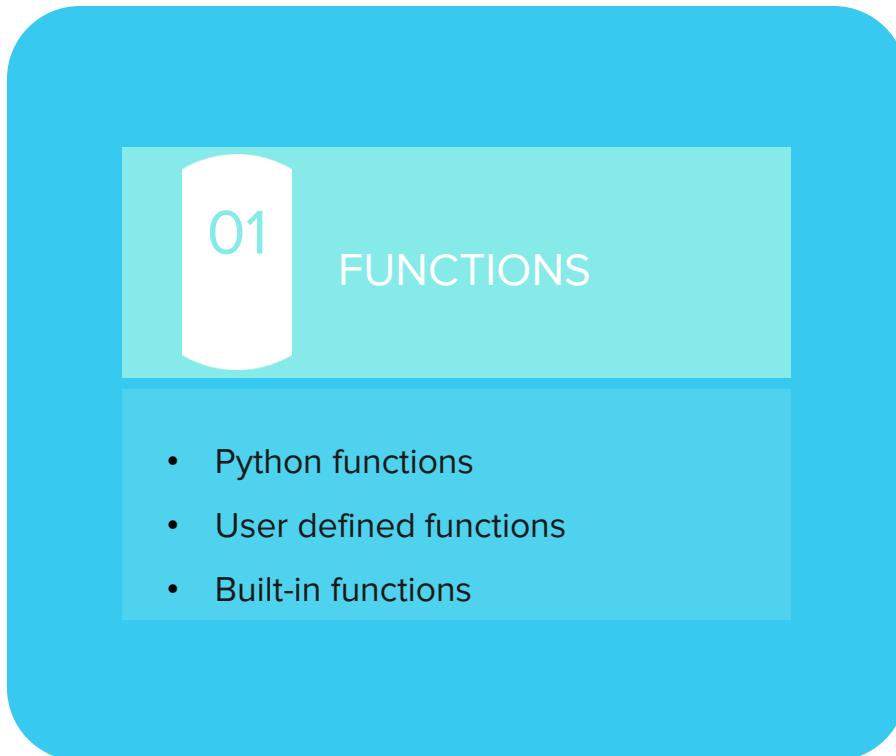
Di akhir modul ini, kita akan dapat:

- Menggunakan built-in functions yang populer di Python, juga membuat user-defined functions
- Memahami generators, menggunakan map dan filter functions
- Memahami string manipulation
- Memahami bagaimana menggunakan reading, writing dan appending data ke sebuah file di python



# Agenda

- 01 FUNCTIONS
  - Python functions
  - User defined functions
  - Built-in functions
- 02 ITERATORS
  - Generators
  - Map and filter
  - Lambda functions
- 03 STRINGS
  - String manipulation in python
- 04 FILE IO
  - File Handling with python
- 05 KESIMPULAN
  - Kuis
  - Ringkasan



01 FUNCTIONS

- Python functions
- User defined functions
- Built-in functions

# Functions di Python



## User defined functions

Python memungkinkan pengguna untuk membuat fungsi yang ditentukan pengguna



## Built-in functions

Fungsi Python yang sudah tersedia sejak melakukan instalasi Python tanpa harus membuat fungsi lagi



## Lambda functions

Fungsi Lambda, juga disebut fungsi anonim atau tanpa nama, biasanya digunakan untuk mendefinisikan logika sederhana

# User defined functions in python

Fungsi adalah metode utama untuk menata kode dan dapat digunakan kembali dalam Python.

Blok pernyataan yang menerima beberapa argumen, melakukan beberapa fungsi, dan menyediakan output

Keyword def digunakan untuk membuat fungsi yang ditentukan pengguna dengan python

Hasil dari fungsi dinyatakan dengan pernyataan return

## Syntax

```
def function_name(arg1, arg2, ..., argN):  
    <statements>  
    return
```

# User defined functions in python

Value yang dikembalikan oleh fungsi

The diagram shows a laptop screen displaying Python code. The code defines a function named `my_function` that adds two numbers. The function includes a docstring and a return statement. Below the function definition, an example call to the function is shown, resulting in the output value 7.

```
In [1]: 1 def my_function(num1, num2):
2     """ This function can be used to add two numbers
3     num1: int / float / complex
4     num2: int / float / complex
5     """
6     output = num1 + num2
7     return output

In [2]: 1 my_function(num1 = 2, num2 = 5)
Out[2]: 7
```

Annotations on the code:

- def keyword untuk fungsi**: Points to the `def` keyword in the first line of the code.
- Nama fungsi**: Points to the function name `my_function`.
- Input arguments**: Points to the parameters `num1` and `num2`. Subtext: "Fungsi bisa menerima satu atau lebih input argument".
- Function documentation string**: Points to the docstring starting with `"""`.
- Calling functions**: Points to the call to `my_function` in In [2].

# User defined functions in python

user bebas untuk mengubah argument ke 2 dari nilai defaultnya

Jika fungsi ini dipanggil dengan 1 argumen, nilai default 5 akan digunakan untuk argument ke 2

```
In [1]: 1 def my_function(num1, num2=5):
          2     output = num1 + num2
          3     return output

In [2]: 1 my_function(5,9)

Out[2]: 14

In [3]: 1 my_function(6)

Out[3]: 11
```

Argument tanpa default value

Argument dengan default value  
– menjadikannya optional input argument

# User defined functions in python

Global  
Variable  
Local  
Variable

Local variable dari  
function tidak bisa  
diakses dari luar  
fungsi

The image shows a laptop screen with a Jupyter Notebook interface. The code in cell In [1] defines a function that uses global variables:

```
In [1]: 1 x = 5
         2 def my_function(num1,num2):
         3     y = 2
         4     print(f"value of x is {x}")
         5     print(f"value of y is {y}")
         6     num3 = num1 + num2 + x
         7     return num3
```

Cell In [2] calls the function and prints its output:

```
In [2]: 1 my_function(4,3)
          value of x is 5
          value of y is 2
out[2]: 12
```

Cell In [3] attempts to print the local variable 'y' from within the function, which results in a NameError:

```
In [3]: 1 print(y)
          -----
NameError: name 'y' is not defined
Traceback (most recent call last)
<ipython-input-3-d9183e048de3> in <module>
      1 print(y)
```

A red bracket on the left side of the screen encloses the 'Local variable' text and the 'In [3]' cell area. A green arrow points from the text 'Global variable can be accessed inside a function' to the line 'print(f"value of y is {y}")' in the In [1] code.

# Built-in functions in python

## Standard

- `print()` - Mencetak objek ke text stream file, dipisahkan oleh sep dan diikuti oleh akhir
- `len()` - Mengembalikan panjang (jumlah item) dari suatu objek.
- `type()` - Dengan satu argumen, kembalikan tipe objek.
- `Id()` - Mengembalikan “identitas” suatu objek.
- `Input()` - Jika argumen prompt hadir, itu ditulis ke output standar tanpa tambahan baris baru.
- `help()` - Panggil sistem bantuan bawaan.

## Operasi Matematika

- `min()`- Mengembalikan item terkecil dalam iterable atau terkecil dari dua atau lebih argumen
- `max()` - Mengembalikan item terbesar dalam iterable atau terbesar dari dua atau lebih argumen.
- `sum()` - Menjumlahkan awal dan item dari iterable dari kiri ke kanan dan mengembalikan total.
- `pow()` - Kembalikan *base* ke *power exp*
- `round()` - Mengembalikan angka yang dibulatkan ke presisi ndigit setelah titik desimal
- `abs()` - Mengembalikan nilai absolut dari suatu angka.

# Built-in functions in python

## Type Conversion

- Primitive type conversion: int(), float(), bool(), str(), complex()
- Non-primitive type conversion: list(), tuple(), dict(), set()
- Metode conversion lainnya: ascii(), bin(), bytes(), hex(), chr(), bytearray(), ord()

## Utility based functions

- Sequence based functions: zip(), range(), map(), filter(), reversed(), enumerate(), iter(), sorted(), next()
- OOPs related functions: hasattr(), getattr(), setattr(), issubset(), isinstance(), object(), classmethod(), staticmethod(), super(), dir(),

# Built-in functions in python

```
1 mylist = [8,7,6,2,4,9,3]
2
3 print("mylist is ",mylist)
4
5 print("type of mylist is ",type(mylist))
6
7 print("length of mylist is ",len(mylist))
```

```
mylist is [8, 7, 6, 2, 4, 9, 3]
type of mylist is <class 'list'>
length of mylist is 7
```

```
1 print("sum of all values ",sum(mylist))
2
3 print("minimum value ",min(mylist))
4
5 print("Maximum value ",max(mylist))
6
7 print("Sorted list ",sorted(mylist))
```

```
sum of all values 39
minimum value 2
Maximum value 9
Sorted list [2, 3, 4, 6, 7, 8, 9]
```



02

## ITERATORS

- Generators
- Map and filter
- Lambda functions

# Generators

Generator adalah jenis fungsi khusus yang tidak mengembalikan nilai tunggal, melainkan mengembalikan objek iterator dengan urutan nilai. Dalam fungsi generator, yield statement biasanya digunakan daripada return statement.

## Syntax

```
def generator_function(arg1, arg2, ...,
                      argN):
    <statements>
    yield value1
    yield value2
```



# Generators

```
1 # A Python program to demonstrate use of
2 # generator object with next()
3
4 # A generator function
5 def simpleGeneratorFun():
6     yield 1
7     yield 2
8     yield 3
9
10 # x is a generator object
11 x = simpleGeneratorFun()
12 print("Type of x - ",type(x))
13
14 # Iterating over the generator object using next
15 print(next(x)) # In other option is x__next__()
16 print(next(x))
17 print(next(x))
```

```
Type of x - <class 'generator'>
```

```
1
2
3
```

# Generators – communicating with generators



next()

Fungsi next dapat meminta pada generator sebuah nilai berikutnya.



stopIteration

Built-in exception yang digunakan untuk exit dari sebuah Generator.



Send

Mengizinkan caller untuk komunikasi dengan generator. (biasanya digunakan untuk coroutines)

# Map

fungsi map() adalah fungsi tingkat lanjut, ia menerima fungsi lain dan urutan 'iterables' sebagai parameter dan memberikan output setelah menerapkan fungsi ke setiap iterable dalam urutan.

## Syntax

```
map(function, iterables)
```

# Map

```
1 # a function for data transformation, to be applied on an iterable object
2 def my_function(a):
3     return a*a
4 # iterable object
5 mylist = [2,4,5,9,6,3]
6
7 # Applying map
8 x = map(my_function, mylist) #x is the map object
9
10 print(x)
11 print(type(x))
12 print(list(x))

<map object at 0x00000245444EB2B0>
<class 'map'>
[4, 16, 25, 81, 36, 9]
```

# Filter

fungsi filter() adalah fungsi tingkat lanjut, digunakan untuk menghasilkan daftar keluaran nilai yang mengembalikan nilai true saat fungsi dipanggil.

## Syntax

```
filter(function, iterables)
```

# Filter

```
1 # a function for data filtering, to be applied on an iterable object
2 def get_even(a):
3     if a%2==0:
4         return True
5     else:
6         return False
7
8 # iterable object
9 mylist = [2,4,5,9,6,3,8,7]
10
11 # Applying filter
12 x = filter(get_even, mylist) #x is the filter object
13
14 print(x)
15 print(type(x))
16 print(list(x))
```

```
<filter object at 0x0000024544504C10>
<class 'filter'>
[2, 4, 6, 8]
```

# Lambda function

Fungsi Lambda juga disebut sebagai fungsi anonim adalah fungsi yang tidak memiliki nama. Fungsi lambda digunakan untuk merepresentasikan ekspresi pendek untuk waktu yang singkat.



## Syntax

```
lambda arguments: expression
```

# Lambda function

```
1 # Use of Lambda functions
2
3 square = lambda x:x**2
4
5 square(5)
```

25

```
1 # Using Lambda functions with map
2
3 mylist = [4,5,3,2,5,8]
4
5 squarelist = map(lambda x:x**2, mylist)
6
7 print(list(squarelist))
```

[16, 25, 9, 4, 25, 64]



03 **STRING**

- String manipulation in python

# String Manipulation



Dalam Python, String adalah array byte yang mewakili karakter Unicode.



String baris tunggal dapat dibuat menggunakan tanda kutip tunggal atau ganda. String multiline dapat dibuat menggunakan tanda kutip tiga.



Python menyediakan banyak seperangkat operator, fungsi, dan metode untuk bekerja dengan string.

# String manipulation menggunakan operators

Operator + melakukan operasi matematika dengan Python. Cukup menulis dua literal string bersama-sama dan menggabungkannya.

Operator \* dapat digunakan untuk mengulang string beberapa kali.

Operator "in" dapat digunakan untuk memeriksa apakah substring ada di dalam string atau tidak

```
1 # Python String Operations
2 str1 = 'Hello'
3 str2 = 'World!'
4
5 # using +
6 print('str1 + str2 = ', str1 + str2)
7
8 # using *
9 print('str1 * 3 =', str1 * 3)
```

```
str1 + str2 = HelloWorld!
str1 * 3 = HelloHelloHello
```

```
1 str3 = "Hello world from python"
2
3 # using "in"
4 print("world" in str3)
5
6 # using "not in"
7 print("Hello" not in str3)
```

```
True
False
```

# String Manipulation menggunakan built-in methods

Method	Description
capitalize()	Mengubah karakter pertama menjadi huruf besar
count()	Mengembalikan berapa kali nilai tertentu muncul dalam string
encode()	Mengembalikan versi string yang disandikan
endswith()	Mengembalikan nilai true jika string diakhiri dengan nilai yang ditentukan
find()	Mencari string untuk nilai yang ditentukan dan mengembalikan posisi di mana ia ditemukan
format()	Memformat nilai yang ditentukan dalam string
index()	Mencari string untuk nilai yang ditentukan dan mengembalikan posisi di mana ia ditemukan
join()	Mengubah elemen dari iterable menjadi string
lower()	Mengubah string menjadi huruf kecil
maketrans()	Mengembalikan tabel terjemahan untuk digunakan dalam terjemahan
partition()	Mengembalikan tupel di mana string dipisah menjadi tiga bagian
replace()	Mengembalikan string di mana nilai tertentu diganti dengan nilai tertentu
split()	Membagi string pada pemisah yang ditentukan, dan mengembalikan daftar
splitlines()	Membagi string pada jeda baris dan mengembalikan daftar
startswith()	Mengembalikan nilai true jika string dimulai dengan nilai yang ditentukan
strip()	Mengembalikan versi string yang dipangkas
swapcase()	Tukar kasus, huruf kecil menjadi huruf besar dan sebaliknya
title()	Mengonversi karakter pertama setiap kata menjadi huruf besar
translate()	Mengembalikan string yang diterjemahkan
upper()	Mengubah string menjadi huruf besar
zfill()	Mengisi string dengan jumlah nilai 0 yang ditentukan di awal

# String manipulation menggunakan built-in methods

```
1 # Common string methods
2 x = "hello World from PYTHON"
3
4 print(x.upper()) # converting to upper case
5 print(x.lower()) # converting to lower case
6 print(x.split(" ")) # splitting string to a list of words
7 print("wold found in x ",x.find("World")) # finding a substring
8 print(x.replace("World","Earth")) # replacing a word with other word
```

```
HELLO WORLD FROM PYTHON
hello world from python
['hello', 'World', 'from', 'PYTHON']
wold found in x  6
hello Earth from PYTHON
```

# String Manipulation menggunakan built-in methods

```
1 # Python string format() method
2
3 # default(implicit) order
4 default_order = "{}, {} and {}".format('John','Bill','Sean')
5 print('\n--- Default Order ---\n',default_order)
6
7 # order using positional argument
8 positional_order = "{1}, {0} and {2}".format('John','Bill','Sean')
9 print('\n--- Positional Order ---\n',positional_order)
10
11 # order using keyword argument
12 keyword_order = "{s}, {b} and {j}".format(j='John',b='Bill',s='Sean')
13 print('\n--- Keyword Order ---\n',keyword_order)
```

--- Default Order ---  
John, Bill and Sean

--- Positional Order ---  
Bill, John and Sean

--- Keyword Order ---  
Sean, Bill and John



04 FILE IO

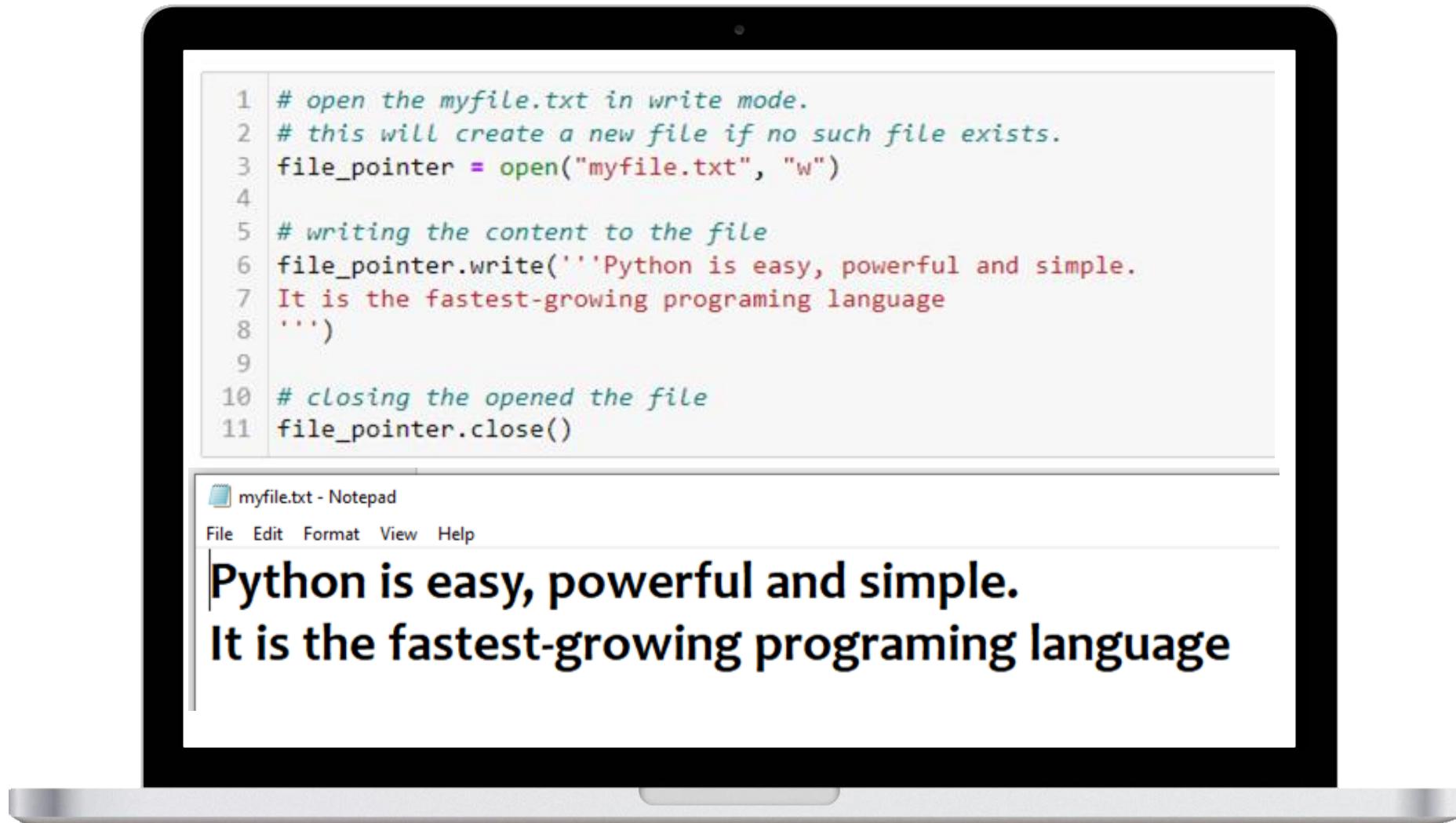
- File Handling with python

# File handling di Python

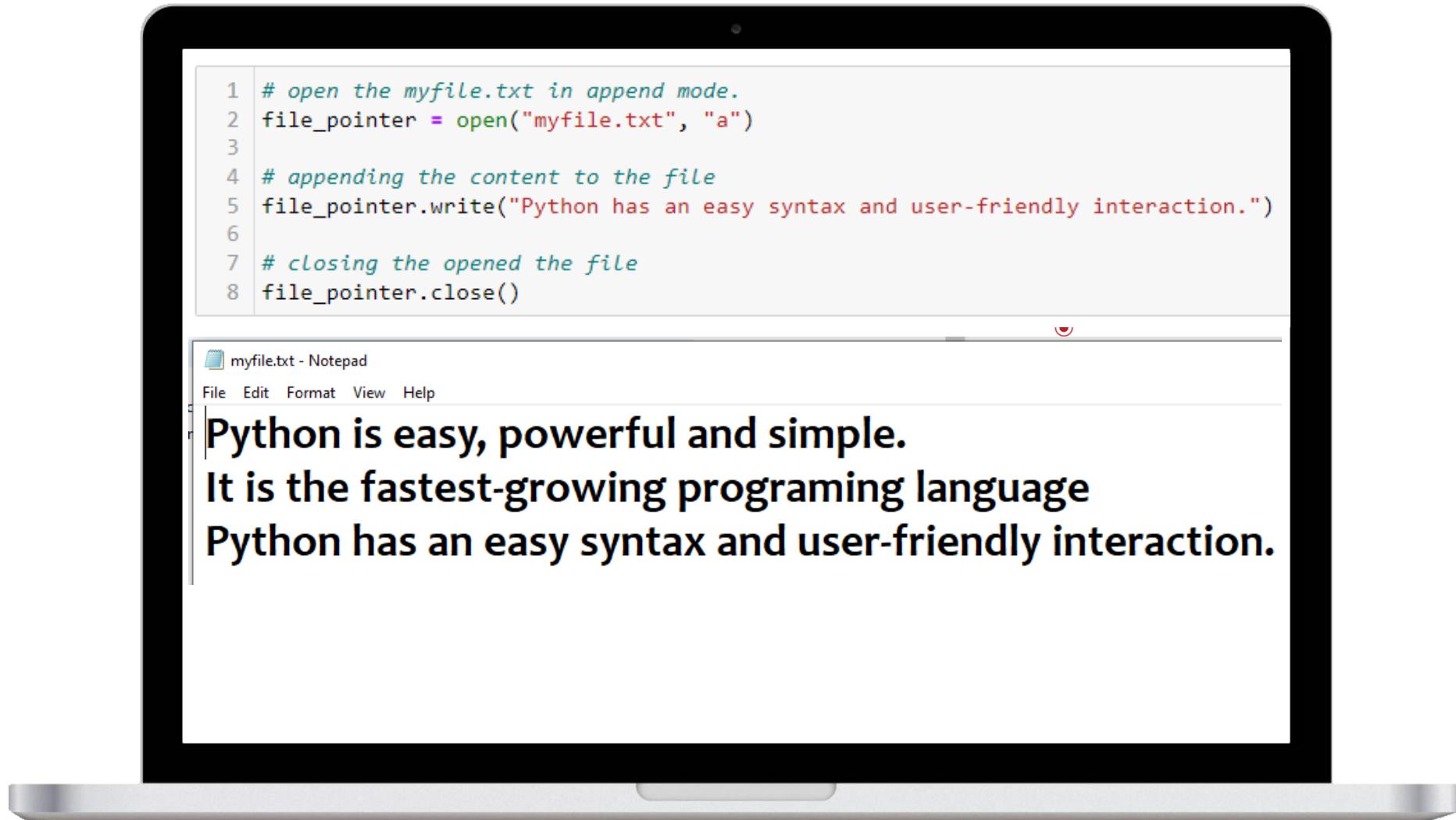
- fungsi open() menerima dua argumen, nama file dan mode akses di mana file diakses.
- Fungsi mengembalikan objek file yang dapat digunakan untuk melakukan berbagai operasi seperti membaca, menulis, dll.

Access mode	Operation	Position of file pointer in file
r	Read only	Beginning
rb	Read only in binary format	Beginning
r+	Read and write both	Beginning
rb+	Read and write both in binary format	Beginning
w	Write only, overwrite existing file, creates new if no file exist	Beginning
wb	Write only in binary format, overwrite existing file, creates new if no file exist	Beginning
w+	Write and read modes, overwrites existing file, creates new file if not no file exist	Beginning
wb+	Read and write in binary format	Beginning
a	Append mode, creates new file if no file exist	End
ab	Append mode in binary format, creates new file if no file exist	End
a+	Append and read, creates new file if no file exist	End
ab+	Append and read in binary format	End

# Menulis data di file baru



# Menambahkan data di file



# Membaca data dari sebuah file

```
1 # open the myfile.txt in read mode.  
2 # it will cause error, if no such file exist  
3 file_pointer = open("myfile.txt", "r")  
4  
5 # reading the content of the file  
6 data = file_pointer.read()  
7 print(type(data))  
8 print(data)  
9  
10 # closing the opened the file  
11 file_pointer.close()
```

```
<class 'str'>  
Python is easy, powerful and simple.  
It is the fastest-growing programming language  
Python has an easy syntax and user-friendly interaction.
```



05

## KESIMPULAN

- Kuis
- Ringkasan

# Kuis

## Pertanyaan

**Keyword apa yang digunakan untuk membuat fungsi generator?**

- A. Return
- B. Lambda
- C. Yield
- D. None of above



# Kuis

## Pertanyaan

**Keyword apa yang digunakan untuk membuat fungsi generator?**

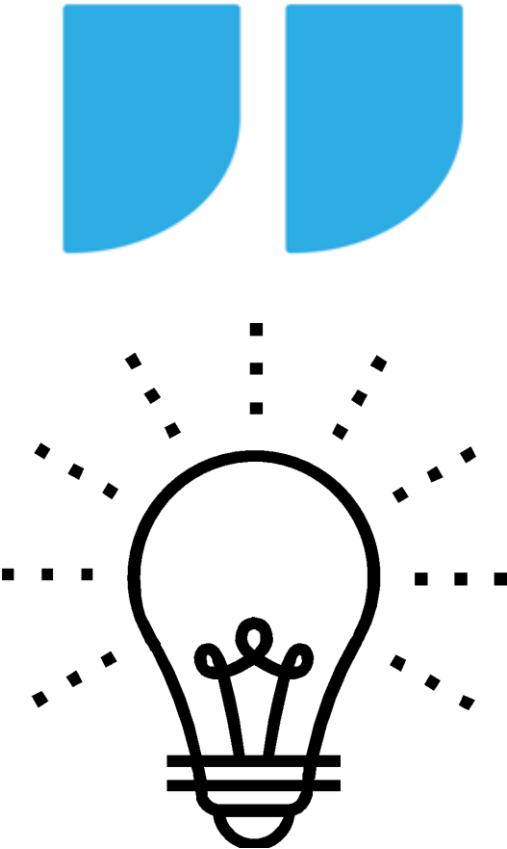
- A. Return
- B. Lambda
- C. Yield
- D. None of above



**Answer - C**

# Ringkasan

- Fungsi adalah metode utama dan paling penting untuk menata kode agar dapat digunakan kembali di Python
- Python Generator membantu dalam menghemat memori dengan melakukan pemrosesan data secara efisien termasuk dalam melakukan perhitungan.
- Fungsi map dan filter dapat digunakan untuk melakukan transformasi dan pemfilteran data pada objek yang dapat diubah
- Python mendukung penggunaan operator seperti +, \*, "in" dan "not in" untuk memanipulasi string, selain itu string python memiliki banyak metode bawaan yang dapat digunakan untuk memanipulasi string
- Fungsi Python open() dapat digunakan untuk menangani file, membantu dalam membaca, menulis, menambahkan data ke file





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  Orbit Future Academy
-  @OrbitFutureAcademyIn1
-  OrbitFutureAcademy
-  Orbit Future Academy



# Module 1

Introduction to everything

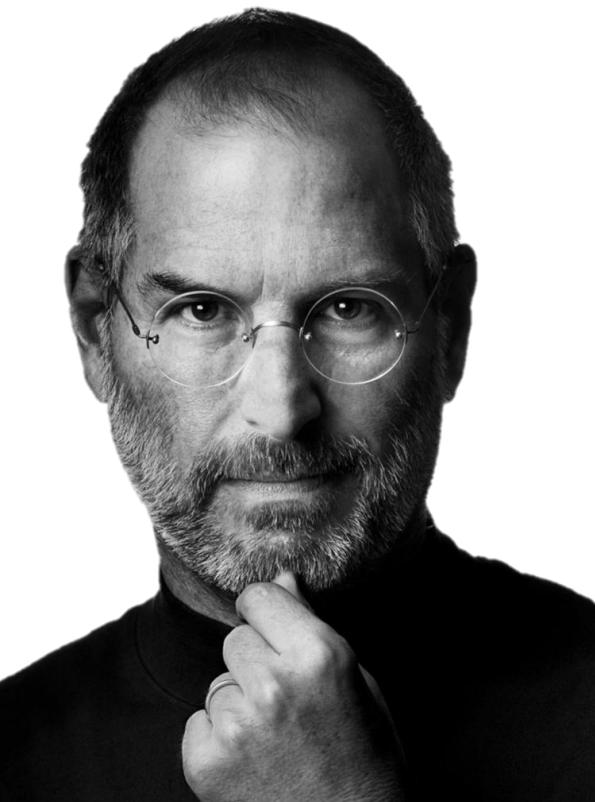
## Section

Basic Data Structure & Algorithm

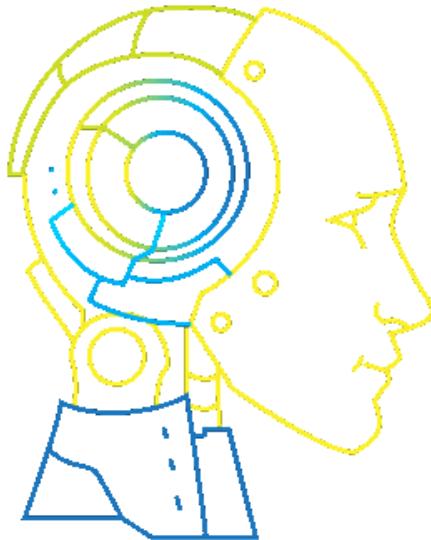
# Basic Data Structure & Algorithm

“To develop a complete mind: Study the science of art;  
Study the art of science. Learn how to see. Realize that  
everything connects to everything else.”

Leonardo da Vinci



**“Everybody should learn  
to program a computer,  
because it teaches you  
how to think”**  
**-Steve Jobs**



# Session I

## What is Data Structure?

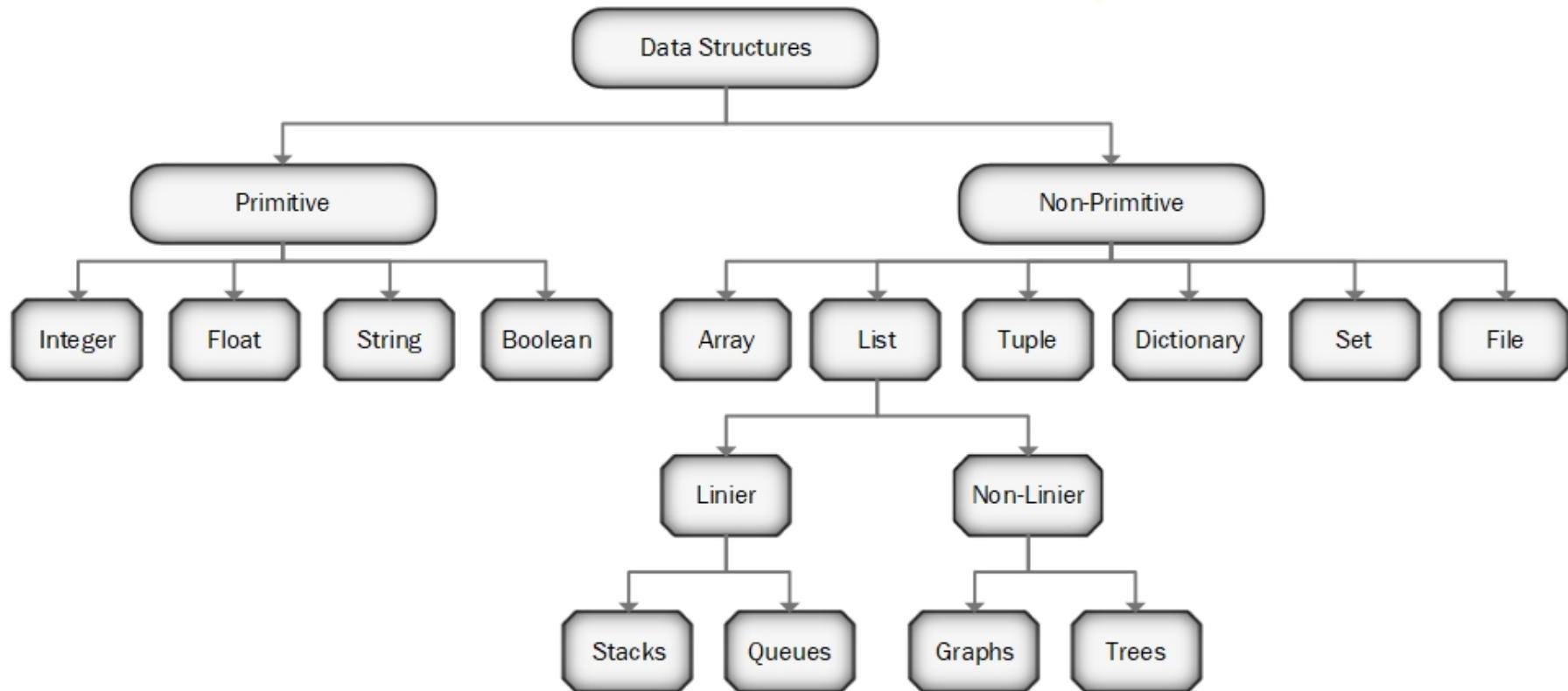
## What is Data Structure?

*Data Structure* adalah sebuah cara mengatur dan menyimpan data, sehingga dapat diakses dan dikerjakan dengan lebih efisien.

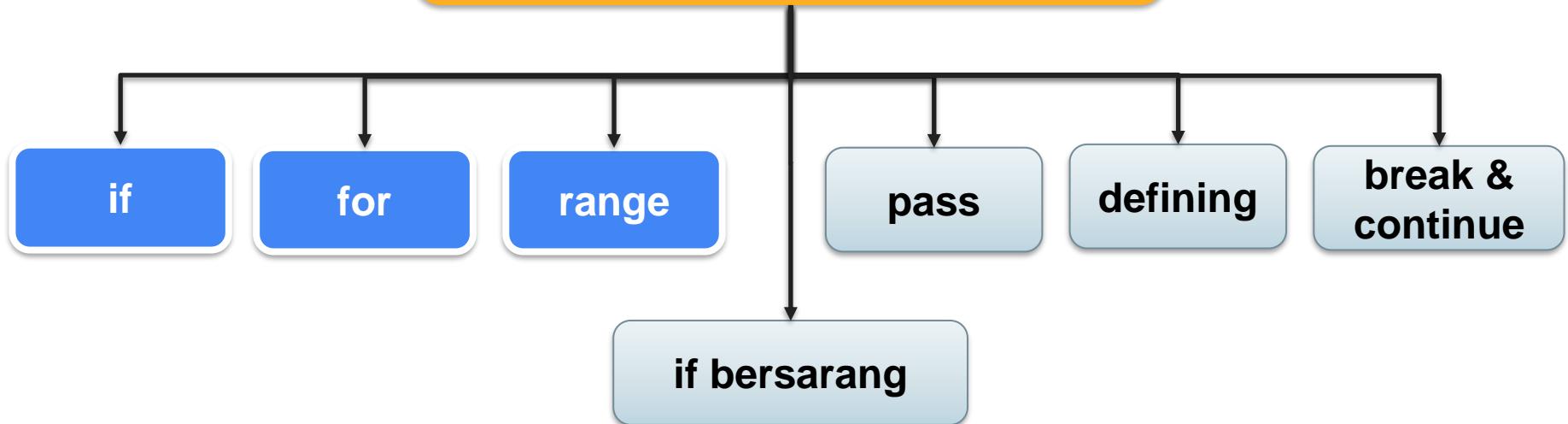
## Abstract Data Type and Data Structures

*Abstract Data Type* merupakan struktur data yang dapat membantu kita untuk fokus pada gambaran yang lebih umum, daripada fokus pada detailnya yang dapat membuat kita sulit untuk memahaminya.

**Data Structures** sebenarnya merupakan implementasi dari Tipe Data Abstrak atau ADT. Implementasi ini membutuhkan tampilan fisik data menggunakan beberapa kumpulan konstruksi pemrograman dan tipe data dasar.



# Kendali Program, Lanjutan



## Statement if bersarang

Statemen seleksi **if...else** bersarang untuk beberapa kasus dilakukan dengan menempatkan statemen **if...else** didalam statemen **if...else** lain.

## Contoh statement if bersarang

```
1  a = 4
2  b = 3
3  if(a == b):
4      if(a != 4):
5          print("Nilai A sama dengan B namun bukan angka 4")
6      else:
7          print("Nilai A sama dengan B")
8  elif(a > b):
9      print("Nilai A lebih besar dari B")
10 else:
11     print("Nilai A lebih kecil dari B")
```

Nilai A lebih besar dari B

## Statement pass

Statemen **pass** tidak melakukan proses apa-apa. Statement ini dapat digunakan ketika pernyataan diperlukan secara sintaksis tetapi program tidak memerlukan Tindakan apapun.

## Contoh pernyataan pass

```
1 x = 5
2 if (x<5):
3     x=3
4 else:
5     pass
6
```

## Defining Function

Pernyataan **def** merupakan sebuah definisi fungsi. Pernyataan **def** harus diikuti dengan nama fungsi dan daftar parameter formal dalam kurung. Pernyataan-pernyataan yang membentuk badan fungsi dimulai pada baris berikutnya, dan harus diindentasi (penulisan baris berikutnya harus menjorok kedalam satu tab).

## Penulisan Defining Function

Setelah tanda titik 2 dari **def** maka baris selanjutnya harus menjorok ke dalam 1 tab

```
def fib(n):      # write Fibonacci series up to n
    a, b = 0, 1
    while a < n:
        print(a, end=' ')
        a, b = b, a+b
```

## Contoh Defining Function

```
1 def fib(n):      # write Fibonacci series up to n
2     """Print a Fibonacci series up to n."""
3     a, b = 0, 1
4     while a < n:
5         print(a, end=' ')
6         a, b = b, a+b
7
8     # Now call the function we just defined:
9     fib(2000)
10
```

```
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

## Break & Continue



Pernyataan **break**, sama seperti dalam Bahasa C, yaitu digunakan untuk keluar dari looping for atau looping while. Sedangkan **continue** untuk melanjutkan ke iterasi berikutnya dari sebuah loop.

## Contoh Break

```
1 for n in range(2, 10):
2     for x in range(2, n):
3         if n % x == 0:
4             print(n, 'equals', x, '*', n//x)
5             break
6     else:
7         # loop fell through without finding a factor
8         print(n, 'is a prime number')
```

```
3 is a prime number
4 equals 2 * 2
5 is a prime number
5 is a prime number
5 is a prime number
6 equals 2 * 3
7 is a prime number
8 equals 2 * 4
9 is a prime number
9 equals 3 * 3
```

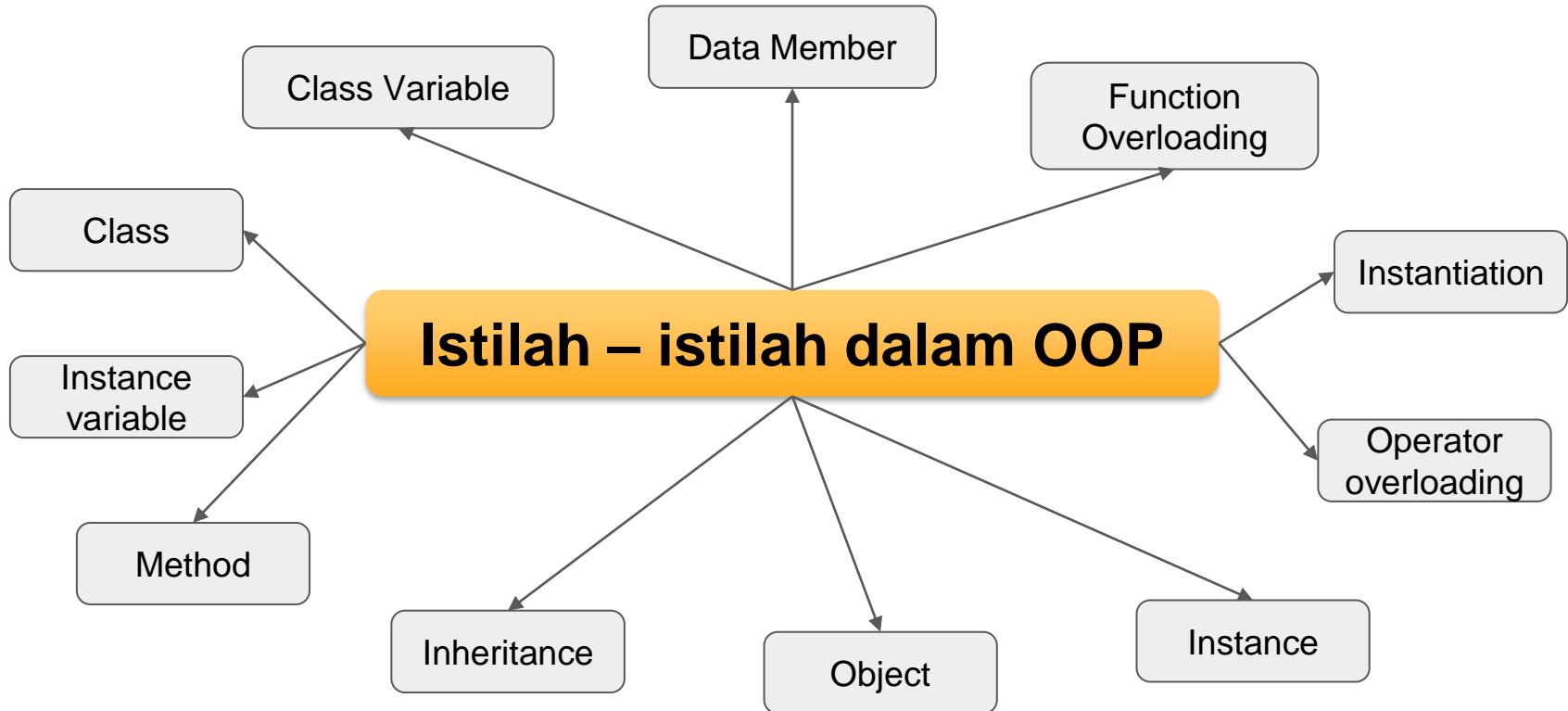
## Contoh Continue

```
1  for num in range(2, 10):
2      if num % 2 == 0:
3          print("Found an even number", num)
4          continue
5      print("Found an odd number", num)
```

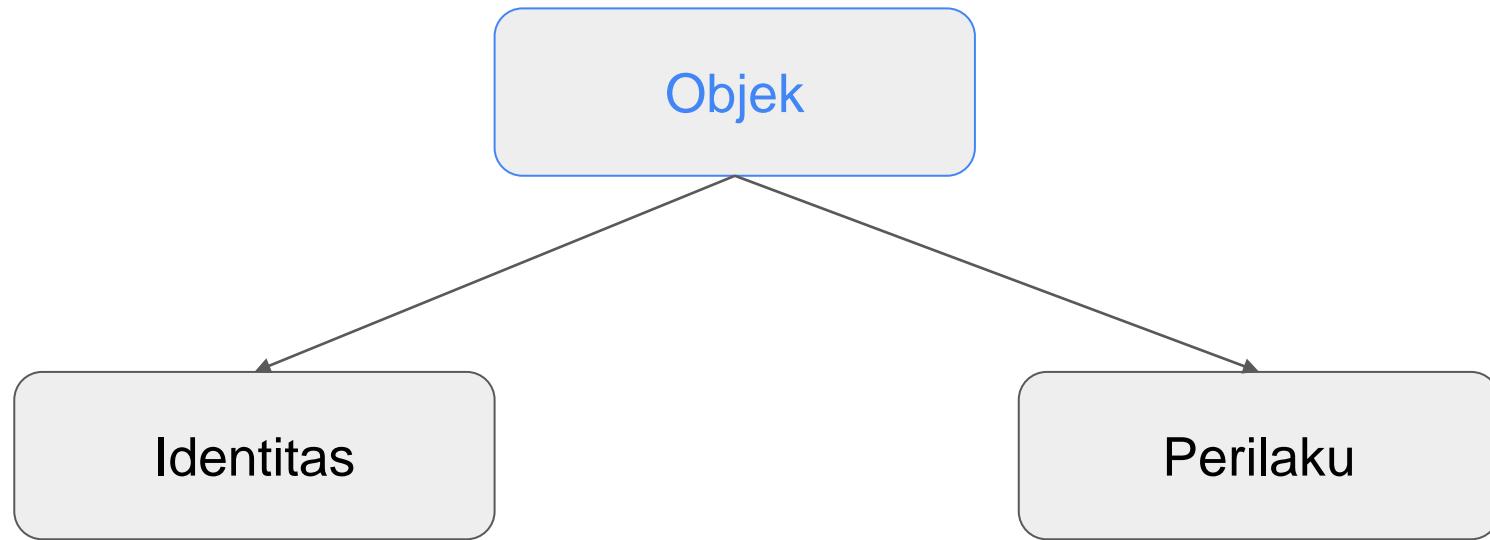
```
Found an even number 2
Found an odd number 3
Found an even number 4
Found an odd number 5
Found an even number 6
Found an odd number 7
Found an even number 8
Found an odd number 9
```

## Object-Oriented Programming (OOP) - Intro

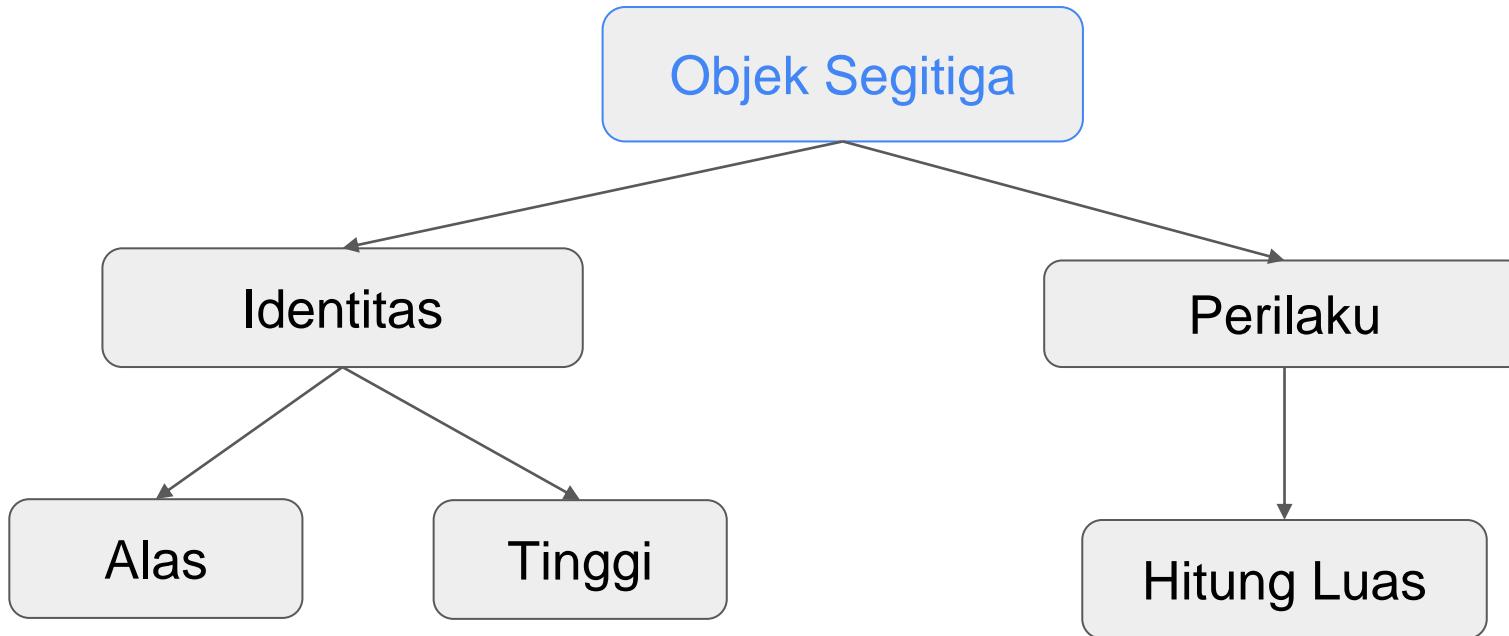
Pemrograman berorientasi objek atau dalam bahasa inggris disebut Object Oriented Programming (OOP) adalah paradigma atau teknik pemrograman di mana semua hal dalam program dimodelkan seperti objek dalam dunia nyata. Objek di dunia nyata memiliki ciri atau attribut dan juga aksi atau kelakuan (behaviour).



# Konsep Dasar OOP



# Konsep Dasar OOP



# Contoh program OOP

```
1  class Segitiga:  
2      def __init__(self, alas, tinggi):  
3          self.alas = alas  
4          self.tinggi = tinggi  
5  
6      def get_luas(self):  
7          return 0.5 * self.alas * self.tinggi  
8  
9      segitiga1 = Segitiga(5, 10)  
10     segitiga2 = Segitiga(10, 10)  
11  
12     print('luas segitiga1:', segitiga1.get_luas())  
13     print('luas segitiga2:', segitiga2.get_luas())  
  
luas segitiga1: 25.0  
luas segitiga2: 50.0
```



# Session II

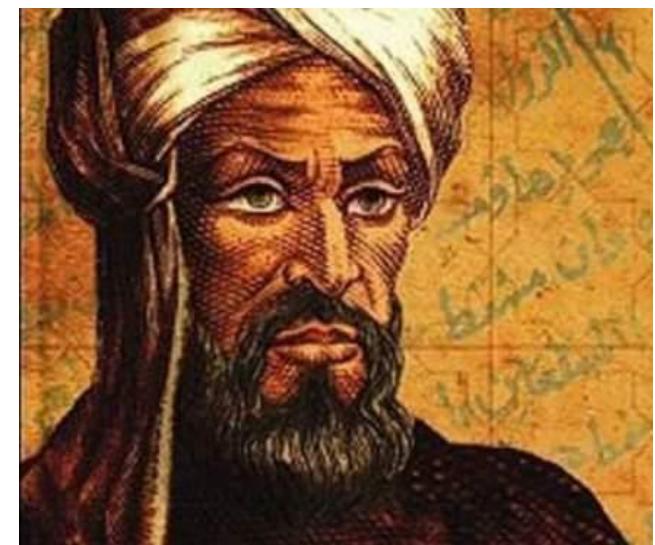
## What is Algorithm?

## What is Algorithm?

Algoritma adalah urutan langkah logis yang digunakan untuk menyelesaikan suatu masalah. Singkatnya, sebuah masalah harus diselesaikan dengan beberapa langkah yang logis.

## Sejarah Algoritma

Istilah algoritma berasal dari nama seorang pengarang berkebangsaan Arab bernama Abu Ja'far Mohammad ibn Musa al Khowarizmi (tahun 790 – 840), yang sangat terkenal sebagai 'Bapak Aljabar'. Beliau juga adalah seorang astronom, ahli geografi, dan sarjana di House of Wisdom di Baghdad.



## Ada 3 pendekatan dalam menyelesaikan algortima

Divide et Impera

Dyanamic programming

Greedy algorithms

# Dasar Penyusunan Algoritma

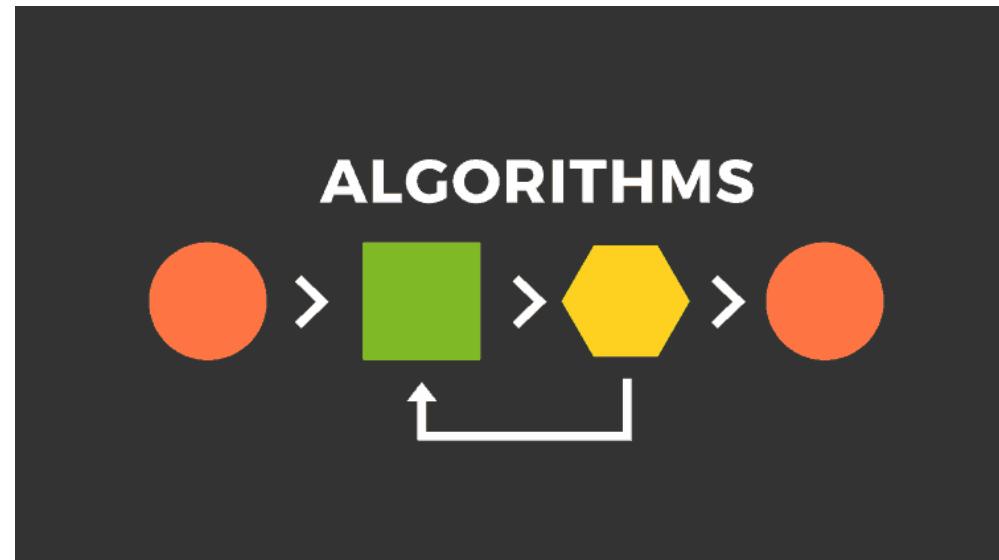
Finiteness

Definiteness

Masukan

Keluaran

Efektivitas

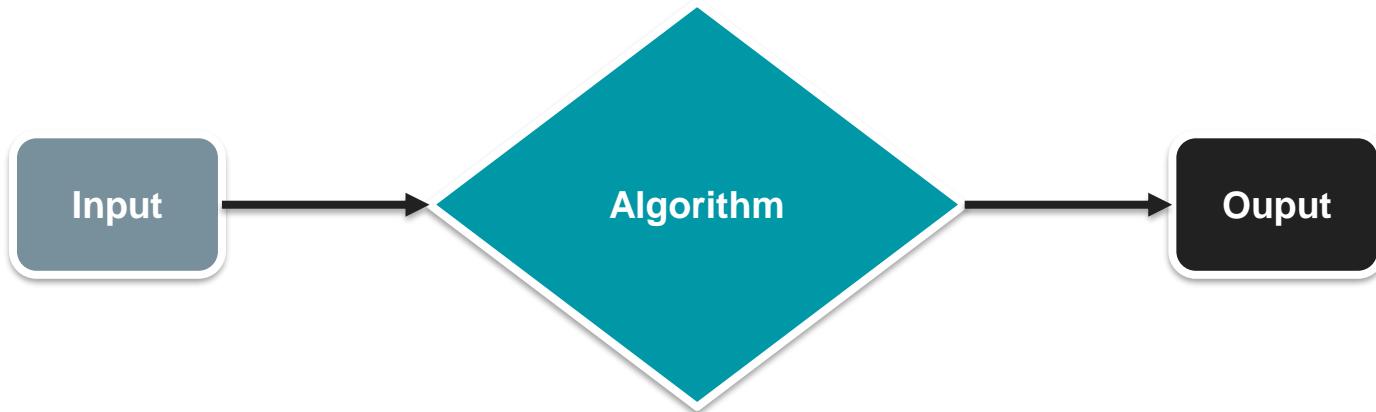


sumber gambar: google

Menurut Knuth (1973) dan juga Horowitz dkk. (1999)

Sebuah algoritma dikatakan benar, untuk berbagai ragam masukan, jika algoritma berakhir dengan keluaran yang benar. Pada keadaan seperti ini, algoritma menyelesaikan masalah komputasi yang diberikan.

Cormen, dkk. (1994)



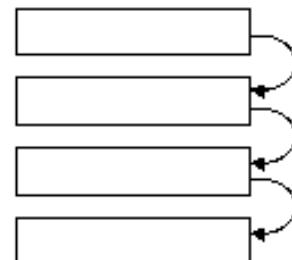
# Algoritma dalam kehidupan sehari-hari



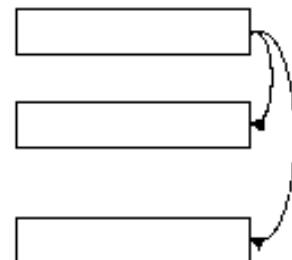
- Taruhlah makanan dalam wadah yang aman untuk *microwave*.
- Tutuplah pintu *microwave* dengan rapat.
- Tancapkan steker ke stop kontak.
- Putarlah knop ke posisi 5 menit.
- Tunggu sampai lampu mati dan ada bunyi 'ting'.
- Lepaskan steker dari stop kontak.
- Bukalah Pintu pemasak *microwave* dan keluarkan wadah yang berisi makanan tersebut.

# Jenis Struktur Dasar Algoritma

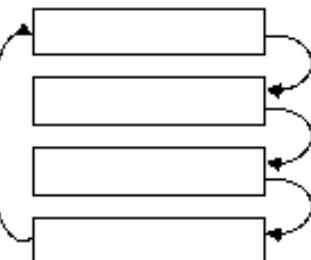
- **Sekuensial**
- **Seleksi**
- **Pengulangan**



runtunan  
(sequence)



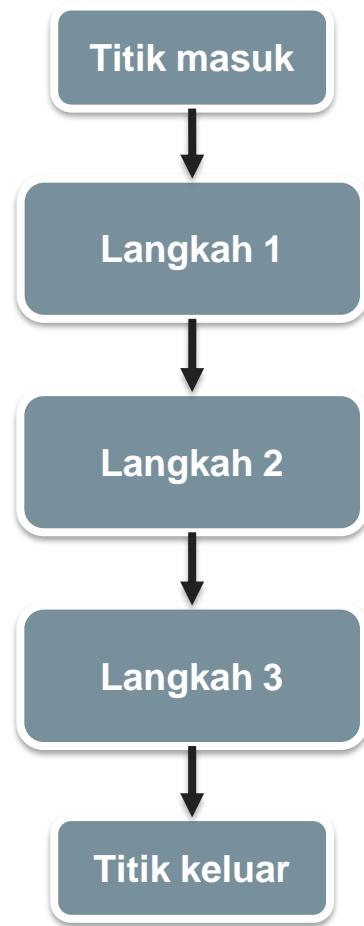
pemilihan  
(selection)



pengulangan  
(repetition)

Sumber Gambar: Google

## Struktur Sekuensial

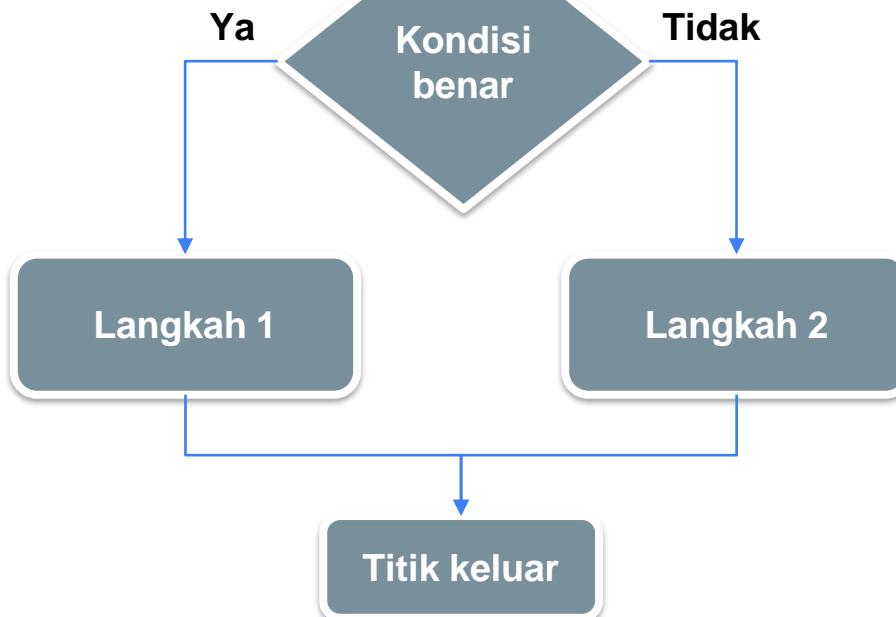


## Struktur Seleksi

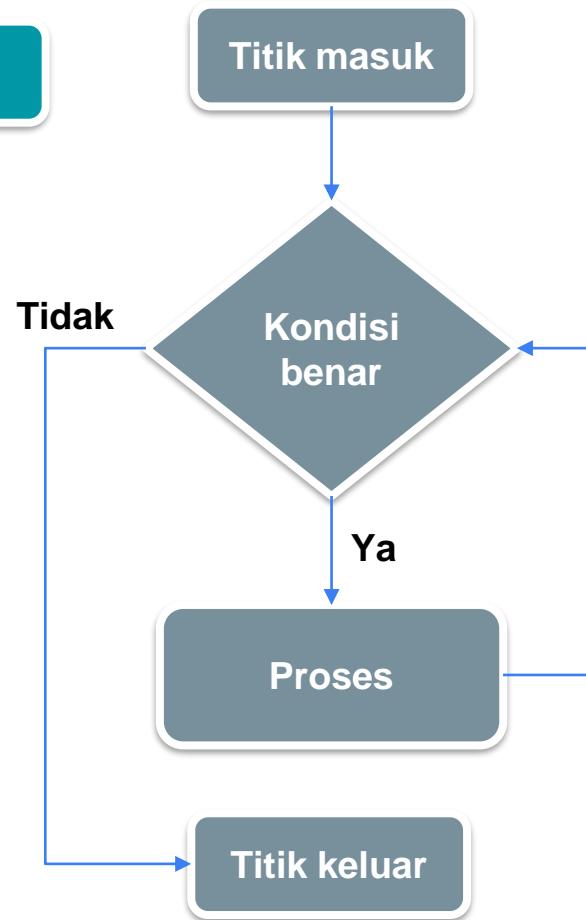
Titik masuk

Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For Future Jobs



## Struktur Perulangan



# Activity

# Algoritma Terapan



Tree Traversal Algorithm

Insertion Algorithm

Searching Algorithm

Deletion Algorithm

Sorting Algorithm

Merging Algorithm

# Tree Traversal Algorithm

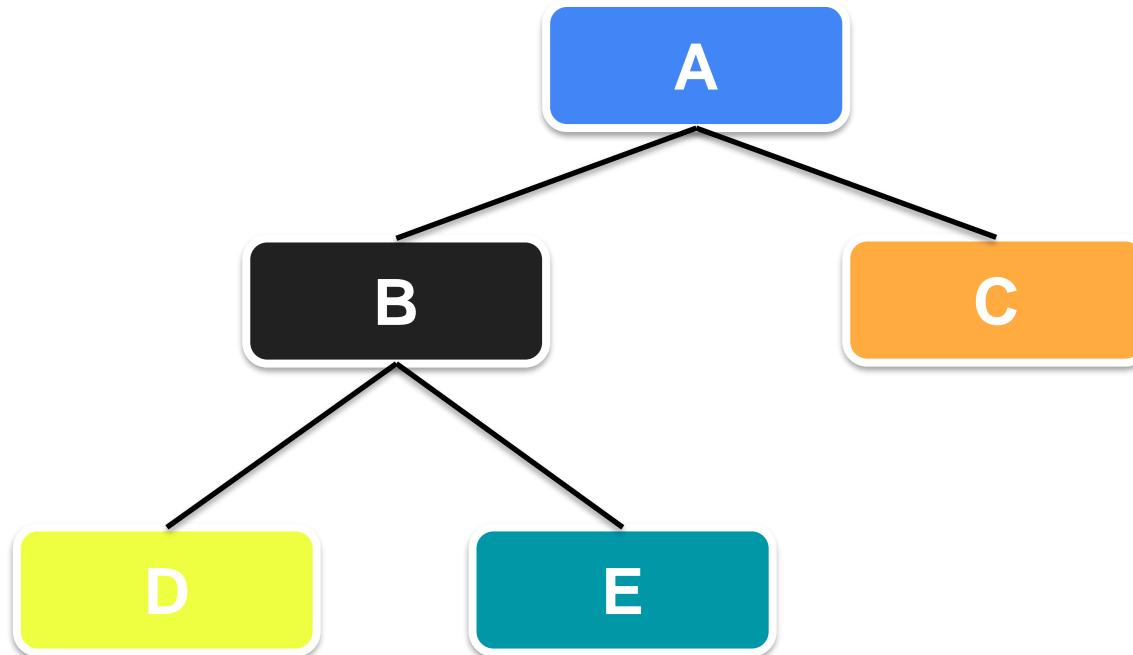
## Tree Traversal Algorithm



Trees di python adalah struktur data non-linear. Memiliki ciri-ciri akar dan simpul. Tree adalah kumpulan *element* yang saling terhubung secara hirarki (*one to many*). *Element* pada tree disebut *node*.

Contohnya adalah binary tree

# Tree Traversal Algorithm



# 3 Tipe Tree Traversal Algorithm

In-order traversal

D – B – E – A – C

Pre-order traversal

A – B – D – E – C

Post-order traversal

D – E – B – C – A

# Implementasi Algoritma Tree Traversal

Penyimpanan Data Statik

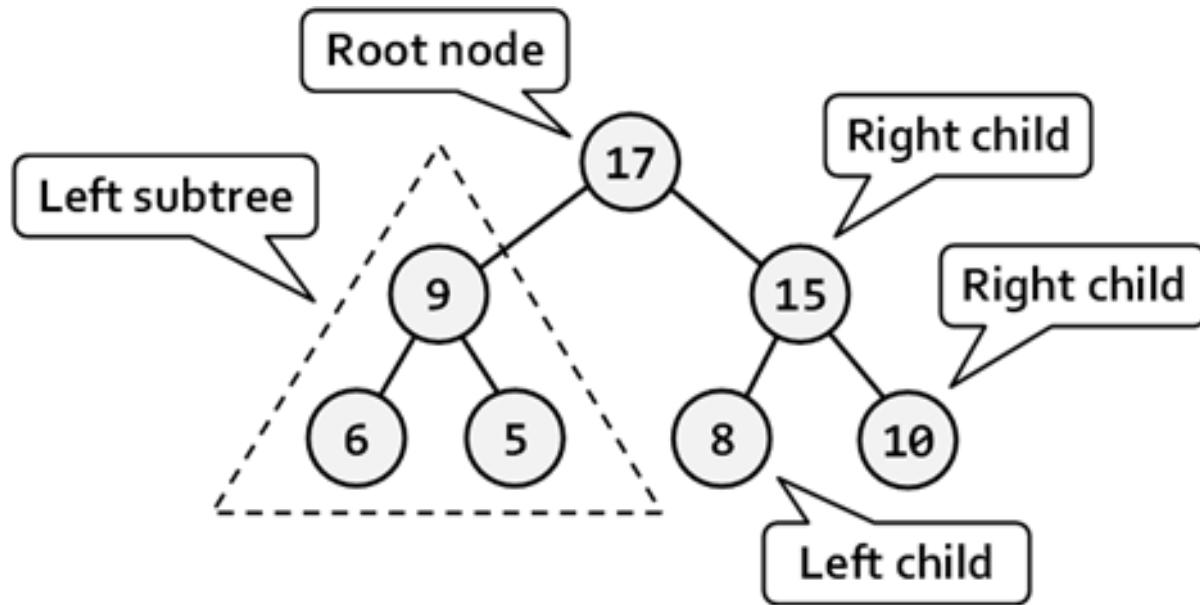


Penyimpanan Data Dinamik



Pencarian Data

## Binary Tree – Implementation



# Sorting Algorithm

## Sorting Algorithm



Algoritma pengurutan digunakan untuk mengurutkan data dengan urutan tertentu.

Contoh: Merge Sort dan Bubble Sort

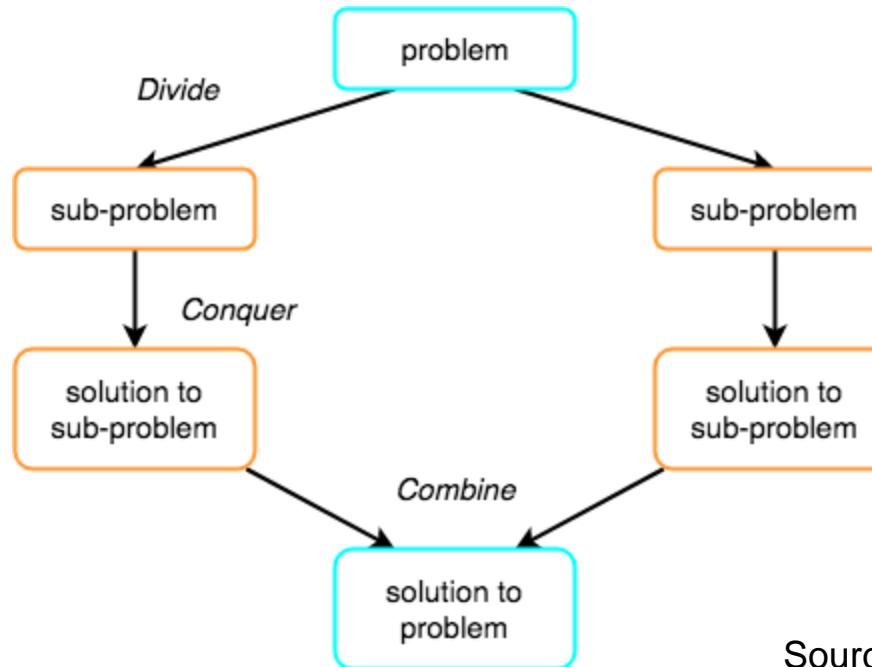
# Sorting Algorithm



Merge Sort

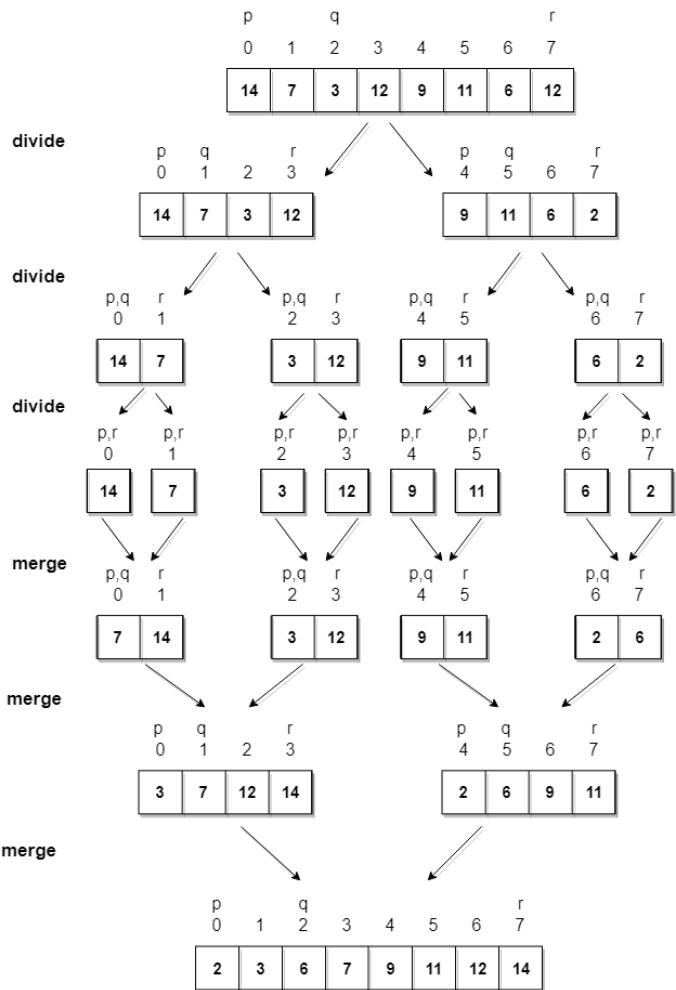
Bubble Sort

## Divide and Conquer in Merge Sort



Source: [studytonight.com](http://studytonight.com)

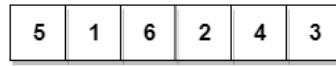
# How Merge Sort Works?



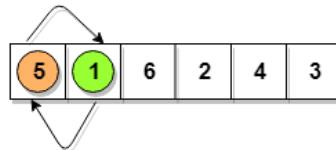
Source: [studytonight.com](http://studytonight.com)

# How Bubble Sort Works?

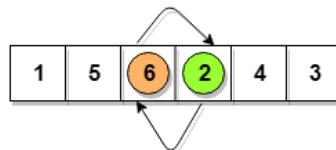
$5 > 1$   
so interchange



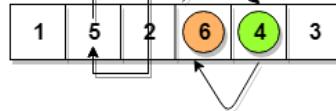
$5 < 6$   
No swapping



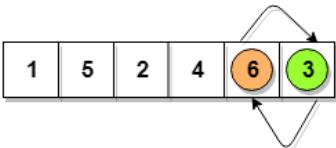
$6 > 2$   
so interchange



$6 > 4$   
so interchange



$6 > 3$   
so interchange



This is first insertion

similarly, after all the iterations, the array gets sorted

Source: [studytontonight.com](http://studytontonight.com)

# Searching Algorithm

## Searching Algorithm



Algoritma pencarian digunakan untuk mencari beberapa elemen yang ada dalam kumpulan data. Ada banyak jenis algoritma pencarian seperti “Linear Search”, “Binary Search”, “Exponential Search”, “Interpolation Search”, dan lain sebagainya. Tapi pada pertemuan ini kita hanya membahas “Linier Search” dan “Binary Search” saja.

# Searching Algorithm



Linear Search

Binary Search

# Contoh Linier Search

1

```
1 def lin_search(ourlist, key):  
2     for index in range(0, len(ourlist)):  
3         if (ourlist[index] == key):  
4             print("Nilai ditemukan")  
5             return index  
6         else:  
7             return "Nilai tidak ditemukan"  
8  
9 ourlist = [15, 1, 9, 3] ← List Nilai  
10  
11 lin_search(ourlist, 2) ← Nilai dicari  
12  
13  
14  
15
```

→ 'Nilai tidak ditemukan'

2

```
1 def lin_search(ourlist, key):  
2     for index in range(0, len(ourlist)):  
3         if (ourlist[index] == key):  
4             print("Nilai ditemukan")  
5             return index  
6         else:  
7             return "Nilai tidak ditemukan"  
8  
9 ourlist = [15, 1, 9, 3] ← List Nilai  
10  
11  
12 lin_search(ourlist, 9) ← Nilai dicari  
13  
14  
15
```

→ Nilai ditemukan  
2

# Contoh Binary Search

```
1 def bin_search(ourlist, key):
2     left = 0 # I assign left position to zero
3     right = len(ourlist)-1 # I assign right position by defining the length of ourlist minus one
4     matched = False
5     while(left<=right and not matched): # the loop will continue until the left element is less
6         # or equal to the right element and the matched is True
7         mid = (left+right)//2 # I find the position of the middle element
8         if ourlist[mid] == key: # if the middle element corresponds to the key element
9             matched = True
10        else: #otherwise
11            if key < ourlist[mid]: # if key element is less than the middle element
12                right = mid - 1 #I assign the position of the right element as mid - 1
13            else: #otherwise
14                left = mid + 1 #left position will become the middle position plus 1
15    return matched
16
17 print(bin_search([1, 3, 9, 15], 17))      [1, 3, 9, dan 15] adalah list data yang terdaftar
18 print(bin_search([1, 3, 9, 15], 3))
```

C) False  
True

[1, 3, 9, dan 15] adalah list data yang terdaftar

[17 dan 3] adalah nilai yang ingin kita cari

# Merangkum Materi

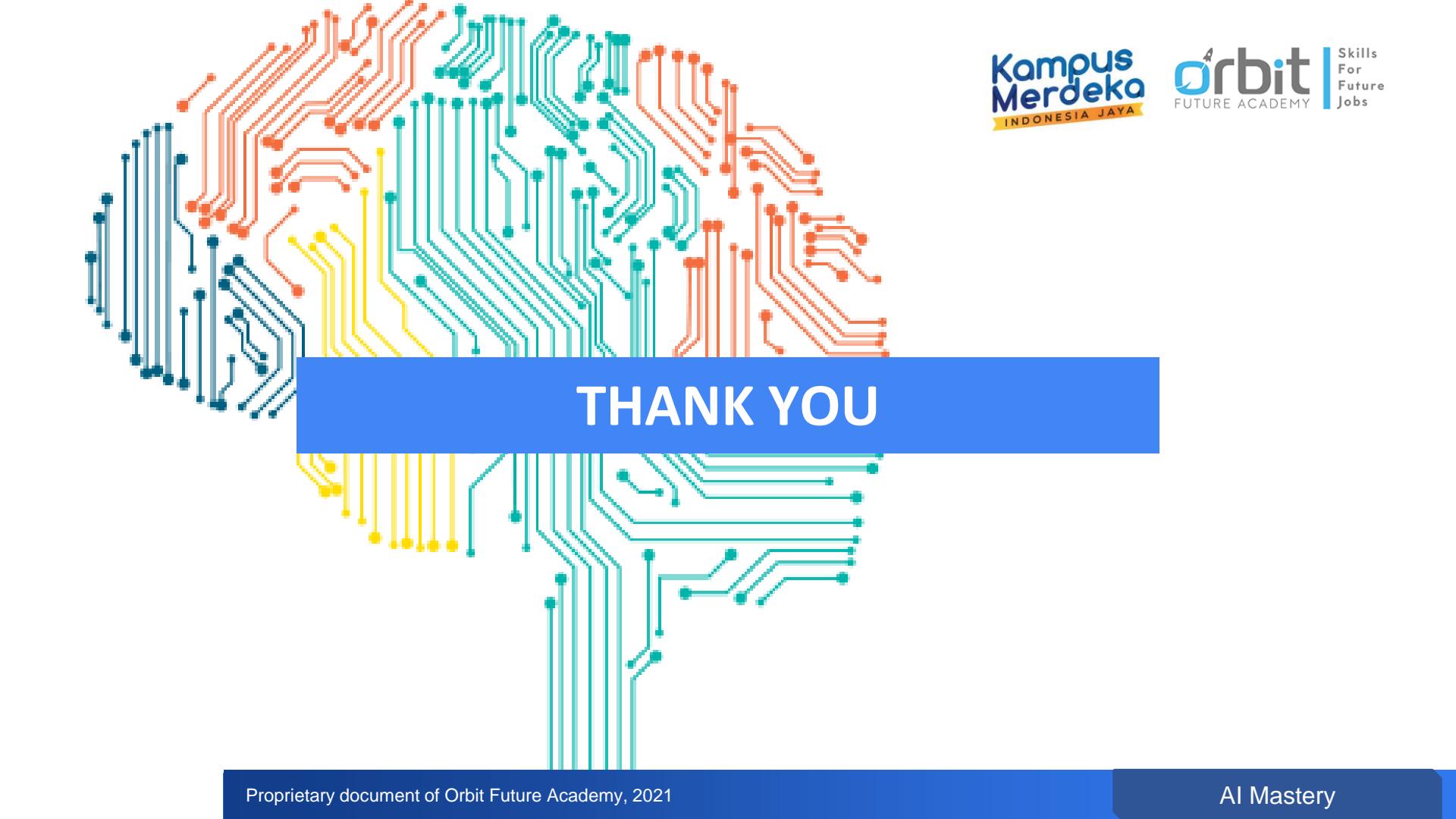
# Quiz

1. Apa itu data struktur?
2. Apa itu OOP?
3. Apa itu algoritma?
4. Apa perbedaan algoritma dan program?
5. Sebutkan 3 jenis struktur dasar algoritma!
6. Bagaimana cara kerja *Merge Sort* pada algoritma *Sorting*?

# Refleksi & Diskusi

Refrensi buku:

1. Algoritma & Pemrograman menggunakan C & C++ karangan Abdul Kadir



THANK YOU

# AI Mastery Course



## Module 1

Introduction to everything

### Section

OOPs and Common Packages





# Learning Objectives

- Di akhir modul ini, kita akan dapat:
  - Memahami object-oriented programming dan inheritance di Python
  - Membuat custom modules dan packages di Python
  - Menggunakan built-in modules seperti os, time dan math
  - Memahami exception handling di Python



# Agenda

- 01 OOPS**
  - Object oriented programming
  - OOPs terminology
  - Inheritance di Python
- 02 MODULES I**
  - Python custom modules
  - Python built in modules
- 03 MODULES II**
  - os
  - time
  - math
- 04 EXCEPTIONS**
  - Exception Handling
  - Custom Exception
- 05 KESIMPULAN**
  - Kuis
  - Ringkasan



01

## OOPS

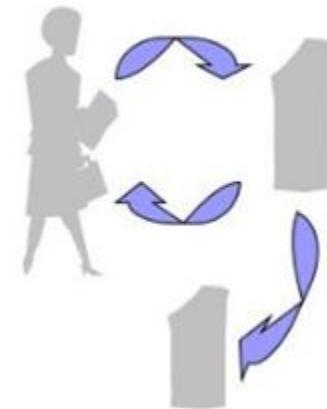
- Object oriented programming
- OOPs terminology
- Inheritance di Python

# Object oriented programming

Object-Oriented Programming (OOP) adalah suatu metode pemrograman yang berorientasi objek.

Sederhananya, OOP adalah paradigma pemrograman yang **merepresentasikan objek** seperti objek di sekitar kita dalam bentuk kode program.

- Procedural



Withdraw, deposit, transfer

- Object Oriented



Customer, money, account

# OOPs : class

Class merupakan blueprint atau prototipe yang digunakan untuk membuat objek. Biasanya, class terdiri dari beberapa atribut dan method.

Classes dibuat dengan keyword class.

Attributes adalah variables yang dimiliki sebuah class. Attributes selalu bersifat public dan bisa diakses menggunakan operator dot (.).

Method adalah fungsi yang melekat pada objek.

## Syntax

```
class ClassName:  
    # Statement-1 .  
    # Statement-N
```

# OOPs terminology

Object	Instance	Method	Constructor method	Class variable	Instance variable
<ul style="list-style-type: none"><li>• Instance unik dari struktur data yang ditentukan oleh kelasnya</li><li>• terdiri dari anggota data (variabel kelas &amp; variabel instan) &amp; metode</li></ul>	<ul style="list-style-type: none"><li>• Objek individu dari kelas tertentu</li></ul>	<ul style="list-style-type: none"><li>• Jenis fungsi khusus yang didefinisikan di dalam kelas</li></ul>	<ul style="list-style-type: none"><li>• Init() adalah keyword untuk membuat constructor</li><li>• Method yang selalu dijalankan ketika kelas dibuat</li></ul>	<ul style="list-style-type: none"><li>• Variabel yang dibagikan oleh semua instance kelas</li><li>• didefinisikan di dalam kelas tetapi di luar method kelas mana pun</li></ul>	<ul style="list-style-type: none"><li>• Variabel yang didefinisikan di dalam method dan hanya dimiliki oleh instance kelas saat ini. Biasanya diakses menggunakan keyword self</li></ul>

# Python class and objects

Membuat object dari class yang ada

keyword class yang digunakan untuk membuat sebuah class

```
1 class course:  
2     # defining the class attributes  
3     name = ""  
4     duration = 0
```

Nama Class

```
1 c1 = course() # creatig an object of type class course  
2 c1.name = "Python programming" # modifying the attribute  
3 c1.duration = 25  
4 print(c1.name)  
5 print(c1.duration)
```

Definisi Class

Python programming  
25

# Python class and objects

keyword class yang digunakan untuk membuat sebuah class

Membuat object dari  
class yang ada

```
1 class Mobil:  
2     # class attribute  
3     Roda = 4  
4     # initializer / instance attributes  
5     def __init__(self, warna, merek):  
6         self.warna = warna  
7         self.merek = merek  
8     # method 1  
9     def showDescription(self):  
10        print("Mobil ini warna ", self.warna, " merek ", self.merek)  
11    # method 2  
12    def changeColor(self, warna):  
13        self.warna = warna  
14 c = Mobil('Hitam', 'Tesla')  
15 # call method 1  
16 c.showDescription()  
17 # Cetak Mobil ini warna Hitam merek Tesla  
18 # call method 2 and set color  
19 c.changeColor('Putih')  
20 c.showDescription()  
21 # Cetak Mobil ini warna Putih merek Tesla
```

```
Mobil ini warna Hitam merek Tesla  
Mobil ini warna Putih merek Tesla
```

Definisi Class

# OOPs terminology : self keyword

01

"Self" mewakili instance dari kelas.

02

Method dari sebuah Class harus memiliki parameter tambahan pertama dalam definisinya. Kita tidak harus memberikan nilai untuk parameter ini.

03

Dengan menggunakan keyword "self" kita dapat mengakses atribut dan metode Class dengan python

04

"self" mirip dengan pointer di C++ dan reference di Java.

# Methods inside the class

Attribut Class  
Method yang didefinisikan di dalam Class dengan keyword self sebagai input

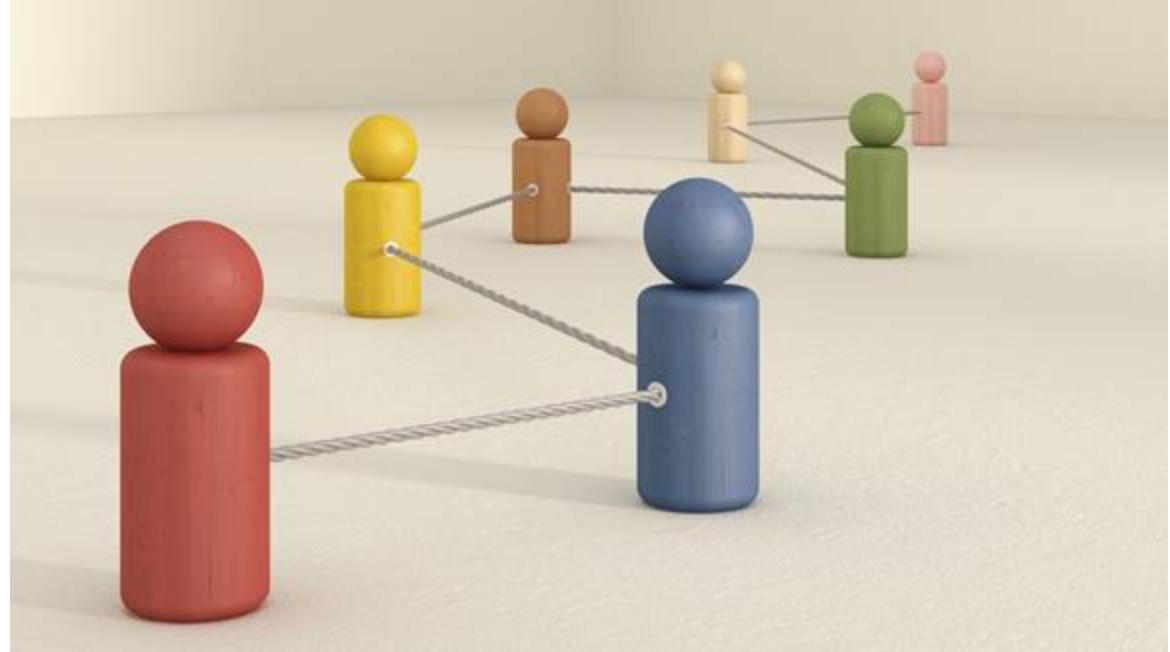
Mengakses attribute dari Class, Class attribute akan dibagikan ke berbagai instances dari Class yang sama

```
1 class Employee:  
2     "Common base class for all employees"  
3     empCount = 0  
4     def __init__(self, name, salary):  
5         self.name = name  
6         self.salary = salary  
7         Employee.empCount += 1  
8  
9     def display_count(self):  
10        print("Total Employee %d" % Employee.empCount)  
11  
12    def display_employee(self):  
13        print("Name : ", self.name, ", Salary: ", self.salary)  
14  
15 # This would create first object of Employee class  
16 emp1 = Employee("Johny", 9000)  
17  
18 #This would create second object of Employee class  
19 emp2 = Employee("Shafi", 4000)  
20  
21 emp1.display_employee()  
22 emp2.display_employee()  
23  
24 print("Total Employee %d" % Employee.empCount)  
25  
26 Name : Johny , Salary: 9000  
27 Name : Shafi , Salary: 4000  
28 Total Employee 2
```

Attribut Class (points to line 3: empCount = 0)  
Method yang didefinisikan di dalam Class dengan keyword self sebagai input (points to line 4: def \_\_init\_\_(self, name, salary):)  
Mengakses attribute dari Class, Class attribute akan dibagikan ke berbagai instances dari Class yang sama (points to line 15: emp1 = Employee("Johny", 9000))  
Instance attributes (points to line 6: self.name = name)

# Inheritance

- Inheritance adalah kemampuan satu kelas untuk memperoleh atau mewarisi properti dari kelas lain (yang sudah ada).
- Kelas yang baru terbentuk adalah kelas turunan (atau child class). Demikian pula kelas yang ada adalah kelas dasar (atau parent class).



# Inheritance

Menggunakan inheritance untuk kelas Employee di kelas Manager

Child class / Derived class



```
1 class Employee:
2     "Common base class for all employees"
3     empCount = 0
4     def __init__(self, name, salary):
5         self.name = name ; self.salary = salary
6         Employee.empCount += 1
7
8     def display_count(self):
9         print("Total Employee %d" % Employee.empCount)
10
11    def display_employee(self):
12        print("Name : ", self.name, ", Salary: ", self.salary)
13
14    class Manager(Employee):
15        manCount = 0
16        def __init__(self, name, salary, teamsize=0):
17            # invoking __init__() method of the parent class
18            super().__init__(name, salary)
19            self.teamsize = teamsize
20
21        def display_teamsize(self):
22            print("Team size %d" % self.teamsize)
23
24
25    emp1 = Employee("Johny", 9000) # creating object of Employee class
26    mgr1 = Manager("Shafi", 12000) # creating object of Manager class
27
28    emp1.display_employee()
29    mgr1.display_employee()
30    mgr1.display_teamsize()
31    print("Total Employee %d" % Employee.empCount)
```

Name : Johny , Salary: 9000  
Name : Shafi , Salary: 12000  
Team size 0  
Total Employee 2

Mengkases method dengan nama yang sama seperti class parent nya



02

## MODULES

- Python custom modules
- Python built in modules

# Python modules



Python modules adalah file dengan ekstensi .py yang berisi kode python yang dapat diimpor ke dalam program python lain.



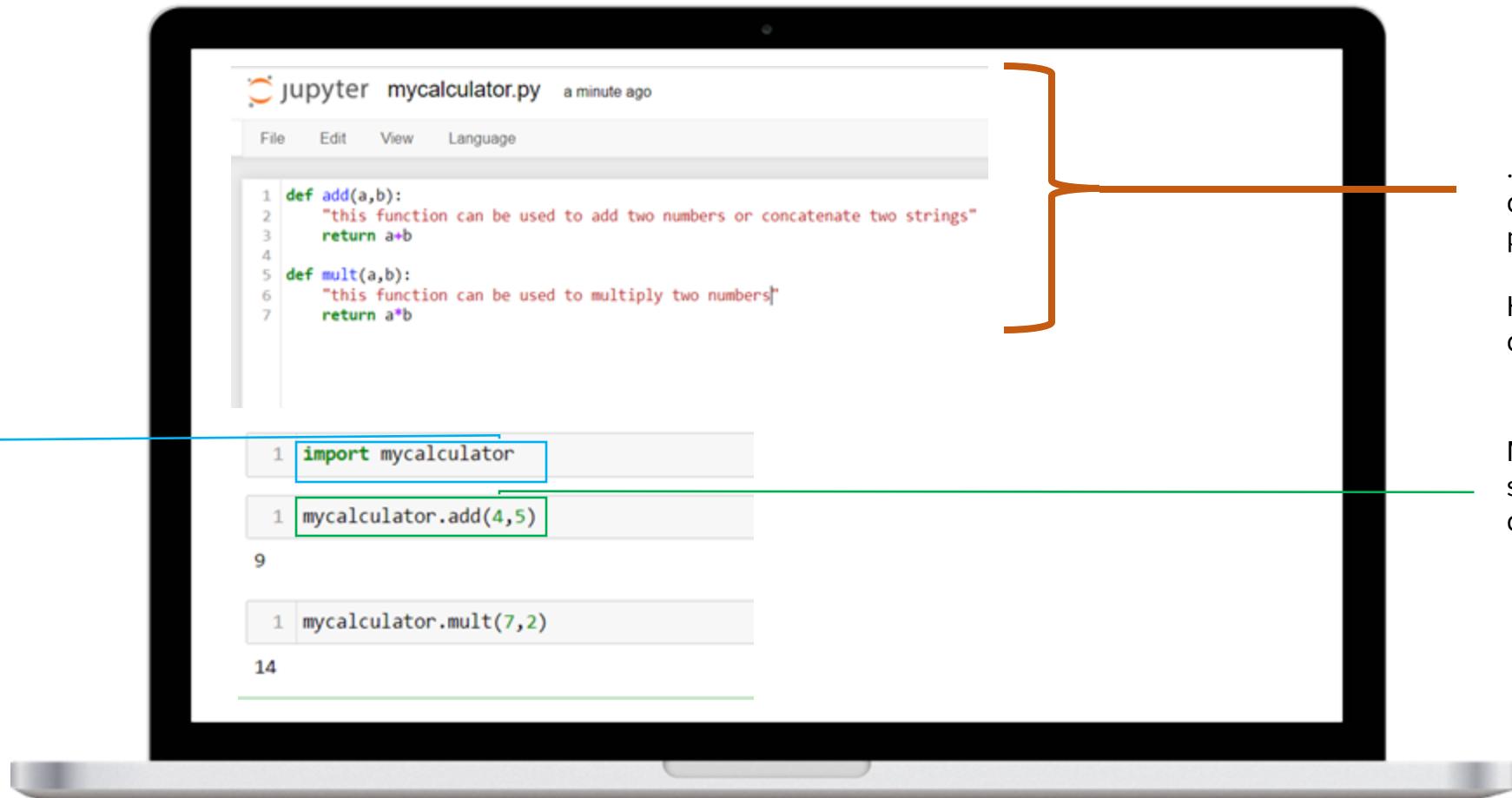
Modul Python berisi komponen berikut :

- Function definitions
- Class implementations
- Variables



Untuk menggunakan modul python, kita dapat menggunakan keyword **import**

# Python custom modules



Memuat  
modul python  
dalam kode  
menggunakan  
keyword  
import

.py file yang akan  
digunakan modul  
python

Harus ada di  
direktori yang tepat

Mengakses  
sebuah fungsi  
dari module

# Python built-in modules



Python memiliki banyak fungsi bawaan, selain itu banyak fungsi tersedia sebagai bagian dari perpustakaan yang dibundel dengan distribusi python, Dalam python ini dikenal sebagai built-in modules



Mayoritas modul python ditulis dalam bahasa C dan terintegrasi dengan python shell



Untuk melihat daftar semua modul yang tersedia, kita dapat menggunakan perintah `help('modules')`

# Python built-in modules

Memanggil  
built-in  
module  
dengan  
keyword  
import

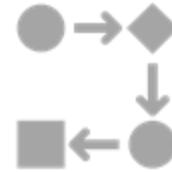


# Keuntungan dari modules



## Reusability

Menggunakan modul python, meningkatkan tingkat penggunaan ulang kode dalam python, fungsi/implementasi kelas yang sama dapat diimpor dalam banyak kode.



## Simplicity

Modul berfokus pada sebagian kecil dari masalah, daripada berfokus pada keseluruhan masalah.



## Scoping

Namespace terpisah ditentukan oleh modul yang membantu menghindari tabrakan antara pengidentifikasi.

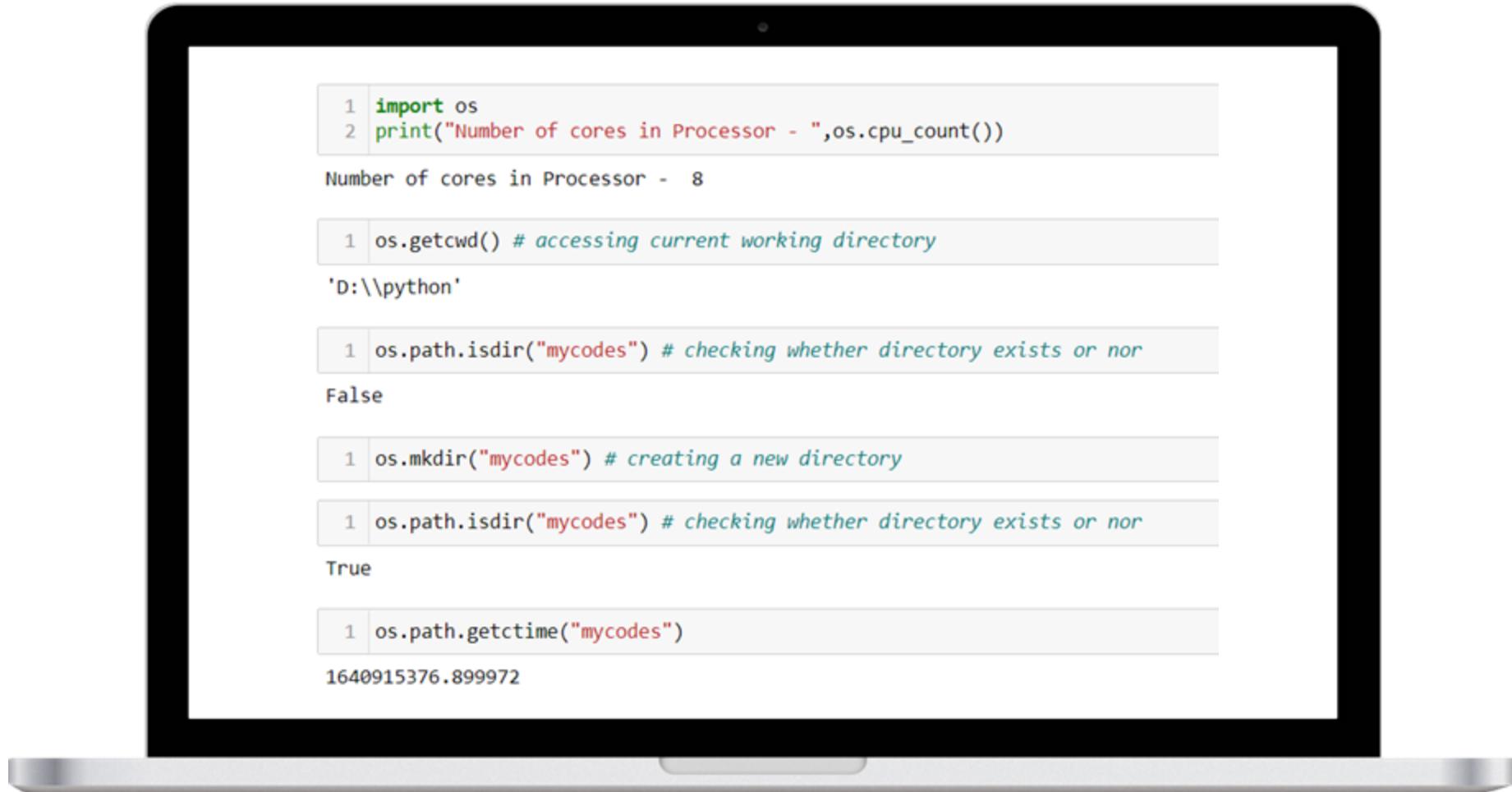


03

## MODULES

- os
- time
- math

# Python module: os



# Python module: time

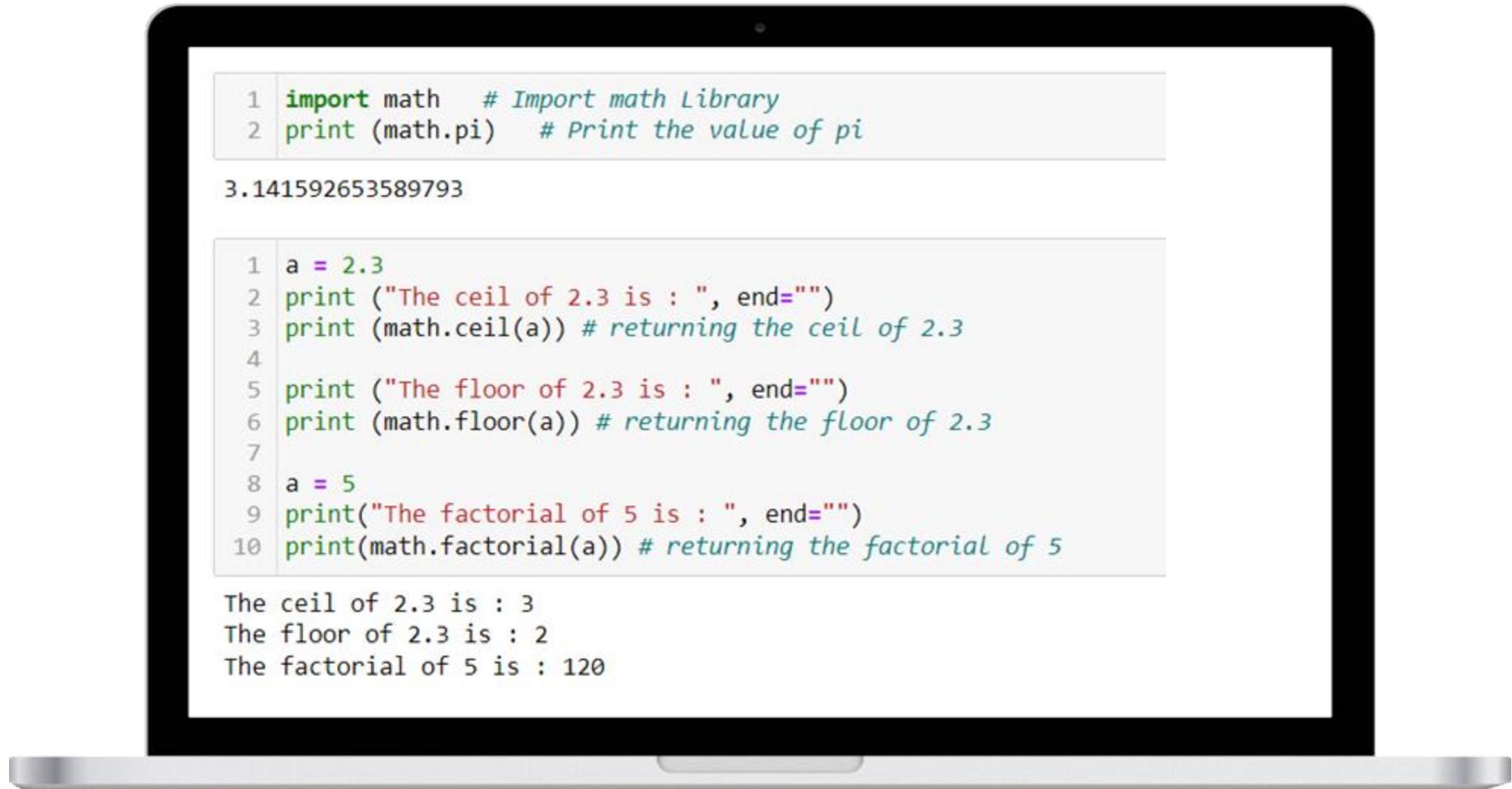
```
1 import time  
2  
3 # seconds passed since epoch  
4 local_time = time.ctime()  
5 print("Local time:", local_time)
```

Local time: Thu Dec 30 14:54:33 2021

```
1 print("This is printed immediately.")  
2 time.sleep(2.4)  
3 print("This is printed after 2.4 seconds.")
```

This is printed immediately.  
This is printed after 2.4 seconds.

# Python module: math





04

## EXCEPTIONS

- Exception Handling
- Custom Exception

# Python syntax errors v/s python exceptions

Syntax error

Tidak menggunakan titik dua (:) setelah kondisi tidak valid di Python

```
1 # initialize the age variable
2 age = 18
3
4 # checking if you are eligible to cast your vote or not
5 if (age < 18)
6 print("You are NOT eligible to cast your vote.")
```

```
File "<ipython-input-42-85eadb0eae2a>", line 5
    if (age < 18)
               ^
SyntaxError: invalid syntax
```

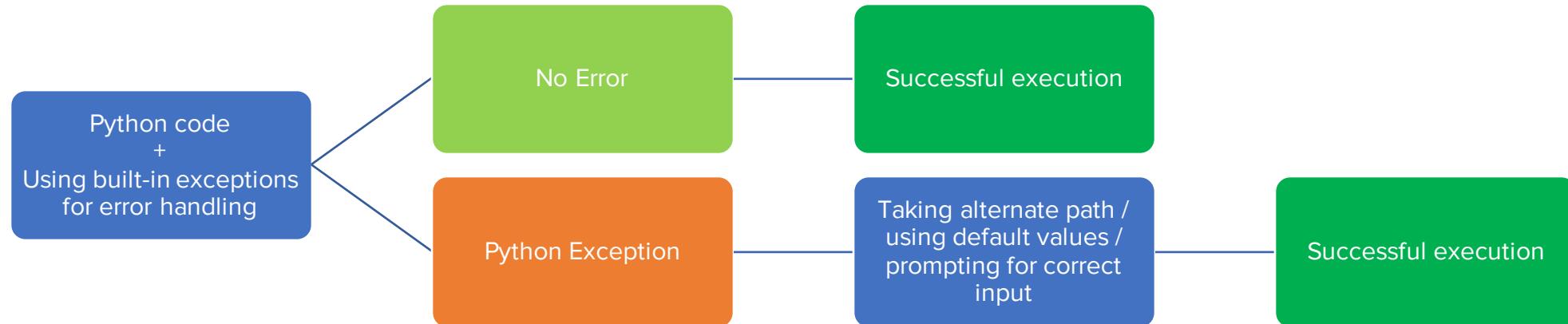
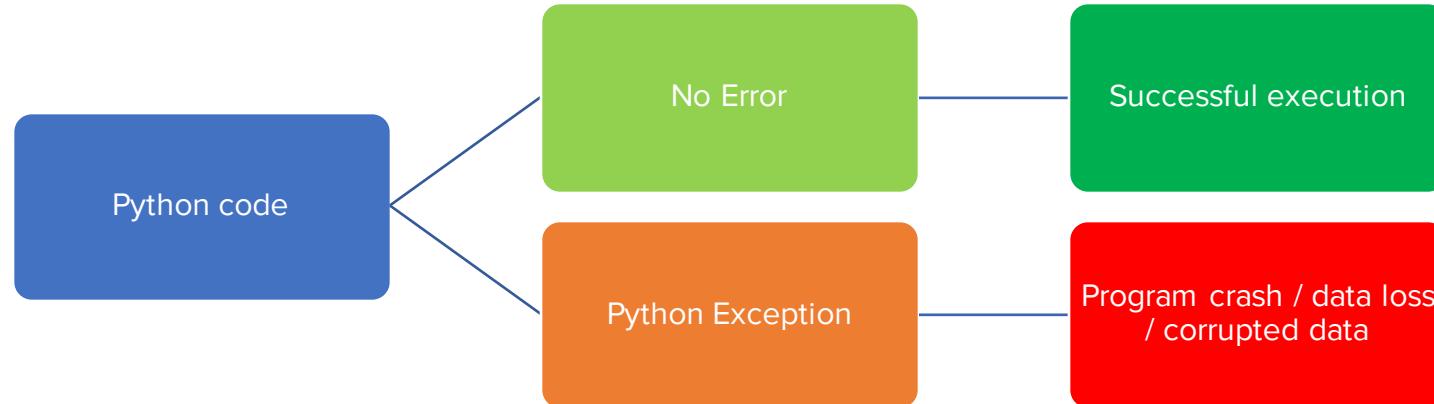
```
1 # initialize the age variable
2 age = 18
3
4 # dividng age by 0
5 out = age/0
```

```
-----
ZeroDivisionError                                Traceback (most recent call last)
<ipython-input-43-035f83291406> in <module>
      3
      4 # dividng age by 0
----> 5 out = age/0

ZeroDivisionError: division by zero
```

Membagi dengan nol akan memicu python exception

# Exception handling pada Python



# Custom exception in python

Raise dapat digunakan untuk memunculkan python exception

The image shows a laptop screen with a white background. On the screen, there is a code editor window and a terminal window. The code editor contains Python code that checks if a birth date is provided after today's date. If it is, it raises an Exception or a ValueError. The terminal window shows the execution of this code and the resulting error messages.

```
1 from datetime import datetime
2 # accessing the current date
3 current_date = datetime.now()
4 print ("Current date is: " + current_date.strftime('%Y-%m-%d'))
5
6 dateinput = input("Enter your birth date in yyyy-mm-dd format: ")
7 date_provided = datetime.strptime(dateinput, '%Y-%m-%d')
8 print ("Date provided is: " + date_provided.strftime('%Y-%m-%d'))
9
10 if (date_provided.date() > current_date.date()):
11     raise Exception("Birth date can't be higher than today")
```

Current date is: 2021-12-30  
Enter your birth date in yyyy-mm-dd format: 2022-01-05  
Date provided is: 2022-01-05

```
-----
Exception                                                 Traceback (most recent call last)
<ipython-input-44-52cf0bf6fef0> in <module>
      9
     10 if (date_provided.date() > current_date.date()):
--> 11     raise Exception("Birth date can't be higher than today")

Exception: Birth date can't be higher than today
```

```
1 if (date_provided.date() > current_date.date()):
2     raise ValueError("Birth date can't be higher than today")
```

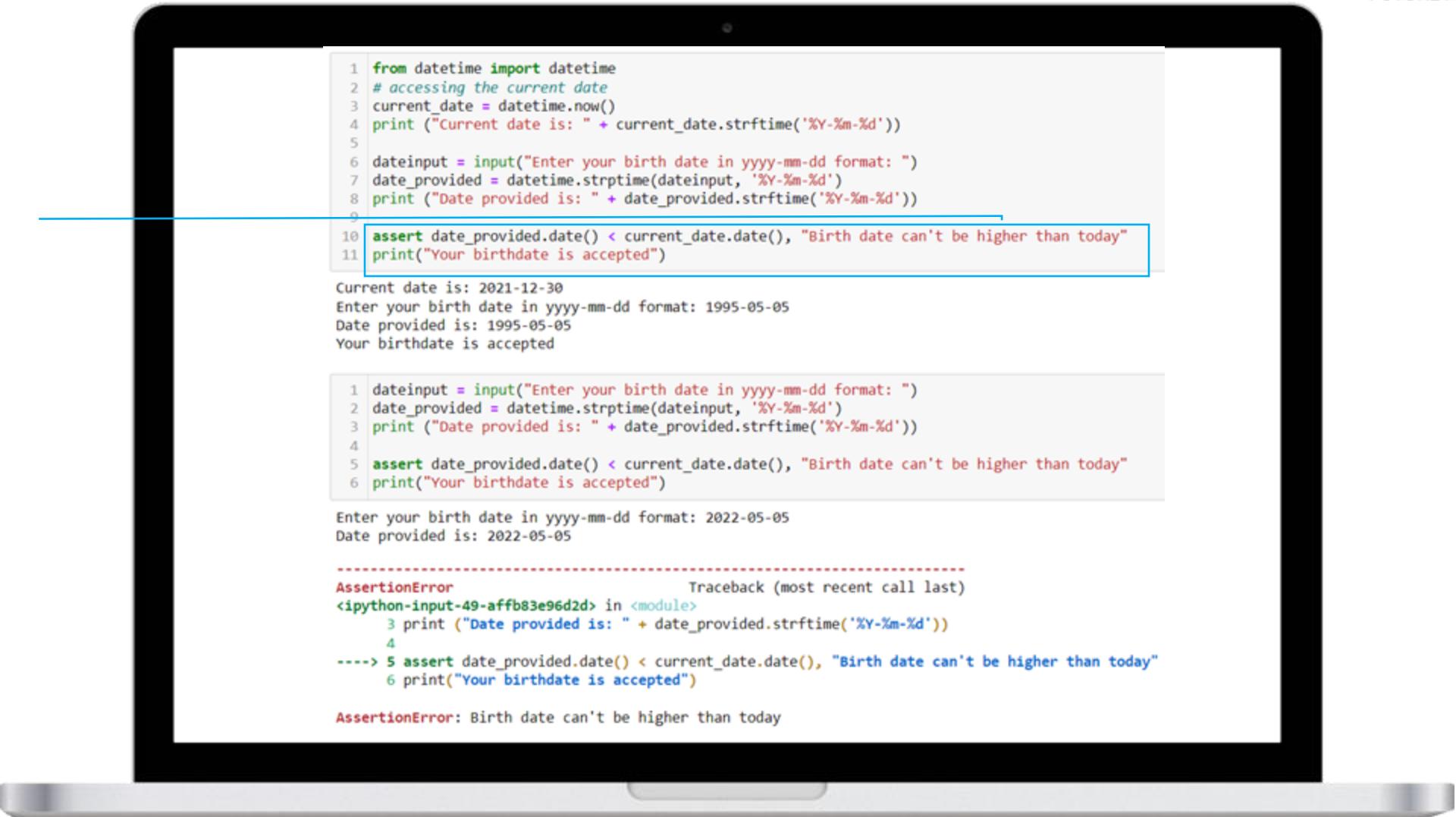
```
-----
ValueError                                              Traceback (most recent call last)
<ipython-input-45-746d45a5943c> in <module>
      1 if (date_provided.date() > current_date.date()):
--> 2     raise ValueError("Birth date can't be higher than today")

ValueError: Birth date can't be higher than today
```

Kita juga dapat menentukan jenis exception yang sesuai

# Assertion Error Exception

Keyword assert dapat digunakan untuk melempar kesalahan pernyataan



# Catching exceptions in python

## Syntax

```
try:  
    <--program code-->  
except <--Exception Type 1-->:  
    <--exception handling code-->  
except <--Exception Type 2-->:  
    <--exception handling code-->  
else:  
    <--program code to run if "try" block doesn't encounter any error-->  
finally:  
    <--program code that runs regardless of errors in the "try" or "else" block-->
```

# Catching exceptions in python

Eksekusi dimulai dengan kode di blok try

Blok else akan dieksekusi jika tidak ada kesalahan dalam blok try

```
1 try:  
2     f = open("testfile.txt", 'r')  
3 except FileNotFoundError as fne:  
4     print(fne)  
5     print ('Creating file...')  
6     f = open("testfile.txt", 'w')  
7     f.write('2')  
8 else:  
9     data=f.readline(1)  
10    print(data)  
11 finally:  
12     print ('Closing file')  
13     f.close()
```

```
[Errno 2] No such file or directory: 'testfile.txt'  
Creating file...  
Closing file
```

Blok except digunakan untuk menentukan jalur alternatif ketika python exception tertentu terjadi

Blok finally selalu dieksekusi apakah exception terjadi atau tidak

# Agenda

05

## KESIMPULAN

- Kuis
- Ringkasan

## Pertanyaan

metode mana yang digunakan untuk membuat konstruktor di kelas?

- A. \_\_enter\_\_()
- B. \_\_init\_\_()
- C. \_\_doc\_\_
- D. \_\_myconstr\_\_()



## Pertanyaan

metode mana yang digunakan untuk membuat konstruktor di kelas?

- A. \_\_enter\_\_()
- B. \_\_init\_\_()
- C. \_\_doc\_\_
- D. \_\_myconstr\_\_()

Answer - B



# Ringkasan

- Object oriented programming memungkinkan pendekatan pemrograman modular dan struktural yang pada gilirannya meningkatkan kesederhanaan kode, memudahkan pemeliharaan kode
- Python module adalah file .py yang berisi implementasi kelas, atribut, dan fungsi, dan dapat dimuat menggunakan keyword import
- Python memiliki banyak modul bawaan yang berguna seperti matematika, os, time, statistics, dll.
- Python memiliki banyak built-in code exceptions yang dapat berguna dalam memprogram tindakan alternatif ketika pengecualian tertentu terjadi.





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  Orbit Future Academy
-  @OrbitFutureAcademyIn1
-  OrbitFutureAcademy
-  Orbit Future Academy

# AI Mastery Course



*Artificial Intelligence  
Mastery Program*

## Module 1

Introduction to everything

### Section

Mathematics for AI



# Matematika Dasar untuk Kecerdasan Buatan

Pengenalan

## Peluang

Konsep dasar peluang, peluang bersyarat, teorema bayes

## Aljabar Linier

Matriks, Vektor, Nilai & Vektor Eigen

## Math for AI

## Kalkulus

Fungsi, Turunan, optimisasi

## Video Aljabar Linier di Artificial Intelligence

Image to Matrix :

<https://youtu.be/X0HXnHKPXSo>

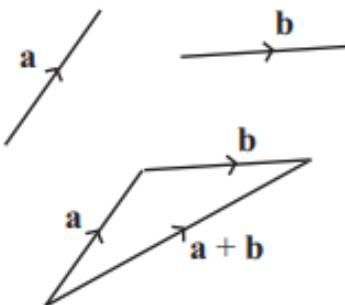
o

(menit ke 3:41 - 7:10)

# Aljabar Linier: Vektor

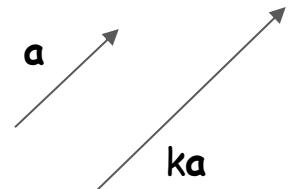
- **Vektor** → besaran yang memiliki nilai dan arah
- Dalam ilmu data, vektor adalah kumpulan nilai dari variabel tertentu
- **Skalar** → konstanta atau besaran yang memiliki nilai
- Vektor berdimensi sama dapat dioperasikan.

## Penjumlahan vektor



$$\mathbf{a} + \mathbf{b} = (a_1 + b_1, \dots, a_n + b_n)$$

## perkalian skalar



$$k\mathbf{a} = (ka_1, \dots, ka_n)$$

## Dot product

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$$

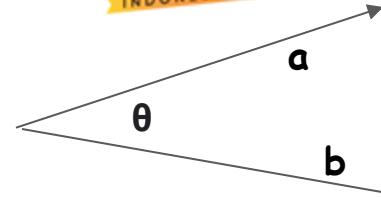
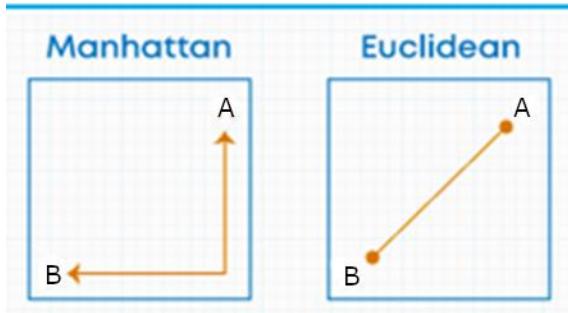
# Jarak & sudut antar vektor

- **Jarak Euclidean**

$$d(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}$$

- **Jarak Manhattan**

$$d(a, b) = \sum_{i=1}^n |a_i - b_i|$$



- **Sudut antar vektor (cosine similarity)**

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

# Aljabar Linier: Matriks

- **Matriks** → kumpulan nilai dari beberapa variabel yang disusun menjadi baris dan kolom. Matriks berukuran  $m \times n$  memiliki  $m$  baris dan  $n$  kolom:

Data pembelian motor

Nama	Kategori harga	Pekerjaan	Brand/Merk
A	1	2	1
B	2	1	1
C	3	1	2
D	2	2	2

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad a_{ij} \in \mathbb{R}$$



$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 3 & 1 & 2 \\ 2 & 2 & 2 \end{pmatrix}$$

# Operasi Matriks

- Penjumlahan matriks: A, B matriks berukuran mxn

$$A+B = (a_{ij}) + (b_{ij}) = (a_{ij}+b_{ij})$$

$$\begin{bmatrix} 1 & 3 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 1+5 \\ 1+7 & 0+5 & 0+0 \end{bmatrix} = \begin{bmatrix} 1 & 3 & 6 \\ 8 & 5 & 0 \end{bmatrix}$$

- Perkalian matriks dengan matriks:

A matriks berukuran mxn, B matriks berukuran nxp, maka C=AxB matriks berukuran mxp

$$2 \cdot \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 5 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 & 2 \cdot 8 & 2 \cdot -3 \\ 2 \cdot 4 & 2 \cdot -2 & 2 \cdot 5 \end{bmatrix} = \begin{bmatrix} 2 & 16 & -6 \\ 8 & -4 & 10 \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ cx + dy \end{bmatrix}$$
$$\begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix} \cdot \begin{pmatrix} 7 & 9 \\ 8 & 0 \end{pmatrix} = \begin{pmatrix} 1 \cdot 7 + 4 \cdot 8 & 1 \cdot 9 + 4 \cdot 0 \\ 2 \cdot 7 + 5 \cdot 8 & 2 \cdot 9 + 5 \cdot 0 \\ 3 \cdot 7 + 6 \cdot 8 & 3 \cdot 9 + 6 \cdot 0 \end{pmatrix} = \begin{pmatrix} 39 & 9 \\ 54 & 18 \\ 69 & 27 \end{pmatrix}$$

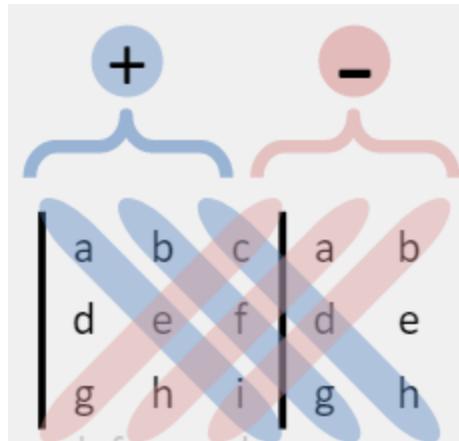
# Determinant & invers

- Misalkan  $A$  matriks persegi, determinant dari  $A$ :

$\det(A) =$

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

=



$$= aei + bfg + cdh  
- ceg - afh - bdi$$

- Misalkan  $A$  matriks persegi, invers dari  $A$  adalah matriks persegi  $A^{-1}$  dengan  $AA^{-1} = A^{-1}A = I$  (matriks identitas)

$$I_n = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

$$A^{-1} = \frac{1}{\det(A)} \text{adj}(A).$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

↑  
determinant

# Aljabar Linier: penerapan invers



Misalkan kita memiliki data berikut, lalu ingin mencari tahu seberapa besar pengaruh jam belajar (x), jam tidur (y), dan tingkat gizi (z) terhadap nilai di sekolah. Sehingga dapat diformulasikan:

$AX = B \rightarrow A^{-1}AX = A^{-1}B \rightarrow X = A^{-1}B$ . X adalah vektor (jam belajar (x), jam tidur (y), tingkat gizi(z))

Nama	Jam belajar	Jam tidur	Gizi	Nilai
D	5	7	8	90
E	2	10	7	70
F	4	9	6	80

$$\begin{array}{ccc} & \xrightarrow{\hspace{1cm}} & \\ \left( \begin{array}{ccc} 5 & 7 & 8 \\ 2 & 10 & 7 \\ 4 & 9 & 6 \end{array} \right) & \left[ \begin{array}{c} x \\ y \\ z \end{array} \right] & = \left( \begin{array}{c} 90 \\ 70 \\ 80 \end{array} \right) \\ & \underbrace{\hspace{1cm}}_{A} & X = B \end{array}$$

$$X = A^{-1} \cdot B = \left( \begin{array}{ccc} 0.038 & -0.380 & 0.392 \\ -0.203 & 0.025 & 0.241 \\ 0.278 & 0.215 & -0.456 \end{array} \right) \cdot \left( \begin{array}{c} 90 \\ 70 \\ 80 \end{array} \right) = \left( \begin{array}{c} 8.228 \\ 2.785 \\ 3.671 \end{array} \right)$$

→ pengaruh jam belajar  
→ pengaruh jam tidur  
→ pengaruh tingkat gizi

# LATIHAN Tim 1

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 0 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix}$$

1. Hitung AB, BA, BC, CB, AB(C), dan A(BC)
2. Jawab pertanyaan berikut:
  - a. Dimensi dari AB
  - b. Dimensi dari BA
  - c. Apakah BC = CB?
  - d. Apakah (AB)C = A(BC)?

# LATIHAN Tim 2

$$A = \begin{bmatrix} 0 & 0 \\ 4 & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix}$$

1. Hitung  $3xA$ ,  $Ax3$ , invers  $A$ , dan invers  $B$
2. Jawab pertanyaan berikut:
  - a. Apakah  $3xA = Ax3$
  - b. Apakah  $A$  dan  $B$  punya invers?
  - c. Apakah  $AA^{-1} = I$ ? apakah  $B^{-1}B = I$ ?

# ISTIRAHAT & TANYA JAWAB

# Jawaban Tim 1

Jawabannya:

- a.  $3 \times 2$ , matriks  $m \times n$  dikali  $n \times p$  hasilnya  $m \times p$
- b. Tidak bisa dikalikan, karena jumlah kolom di B tidak sama dengan jumlah baris A
- c. Tidak sama, jadi perkalian matriks tidak komutatif ( $BC \neq CB$ )
- d. Sama, sifat assosiatif  $(AB)C = A(BC)$

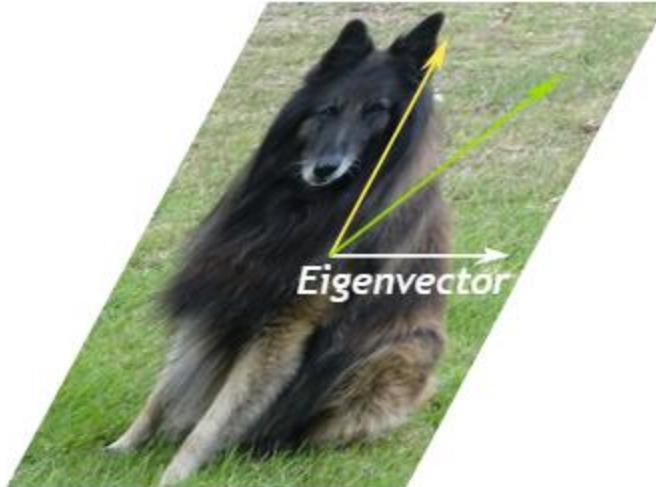
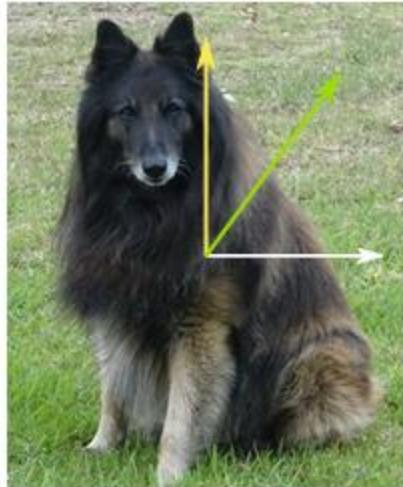
# Jawaban Tim 2

Jawabannya:

- a. Sama, perkalian matriks dengan skalar bersifat komutatif  
 $kA = Ak$
- b. A tidak punya invers karena determinannya 0, B punya invers karena determinannya tidak 0
- c. A tidak punya invers, untuk B jika hitungan inversnya benar maka  $BB^{-1} = B^{-1}B = I$

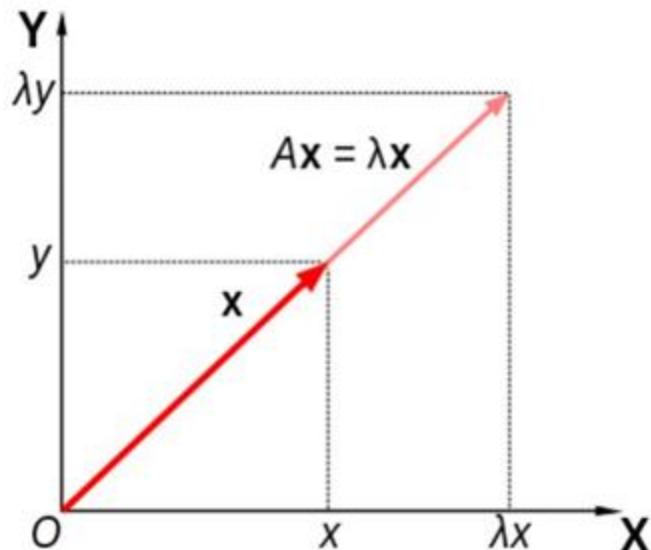
# Aljabar Linier: Nilai & Vektor eigen

- Eigen berasal dari bahasa Jerman yang berarti karakteristik. Vektor eigen adalah vektor yang menjadi karakteristik sebuah matriks dimana arahnya tidak berubah meski dilakukan transformasi



# Aljabar Linier: Nilai & Vektor eigen

- Sementara nilai eigen adalah bilangan yang berasosiasi dengan panjang vektor yang berubah setelah ditransformasi oleh matriks
- Pada gambar berikut,  $A$  adalah matriks persegi,  $x$  adalah vektor eigen, dan  $\lambda$  adalah nilai eigen



Dari gambar disamping kita bisa menulis:

$$Ax = \lambda x$$

$$Ax - \lambda x = 0$$

$$Ax - \lambda Ix = 0$$

$$(A - \lambda I) x = 0$$

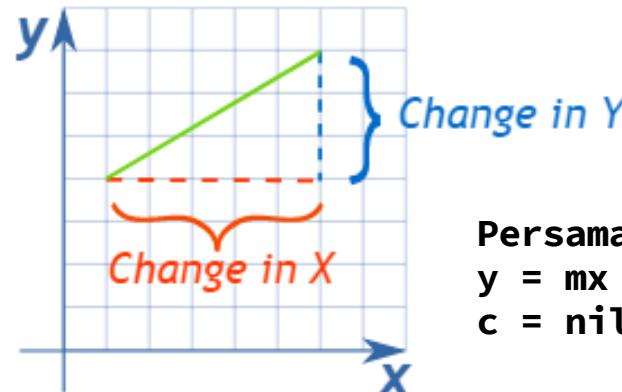
Jika  $x$  bukan vektor 0, maka kita bisa menemukan nilai eigen dengan menghitung determinant matriks  $(A - \lambda I)$ . Lalu mensubtitusi nilai eigen tsb ke  $Ax = \lambda x$  untuk menemukan vektor eigennya

# Kalkulus: fungsi & gradient

- Fungsi adalah aturan yang memetakan antara 2 himpunan
- Gradient adalah besaran perubahan atau kemiringan

$$\text{Gradient} = \frac{\text{Change in Y}}{\text{Change in X}}$$

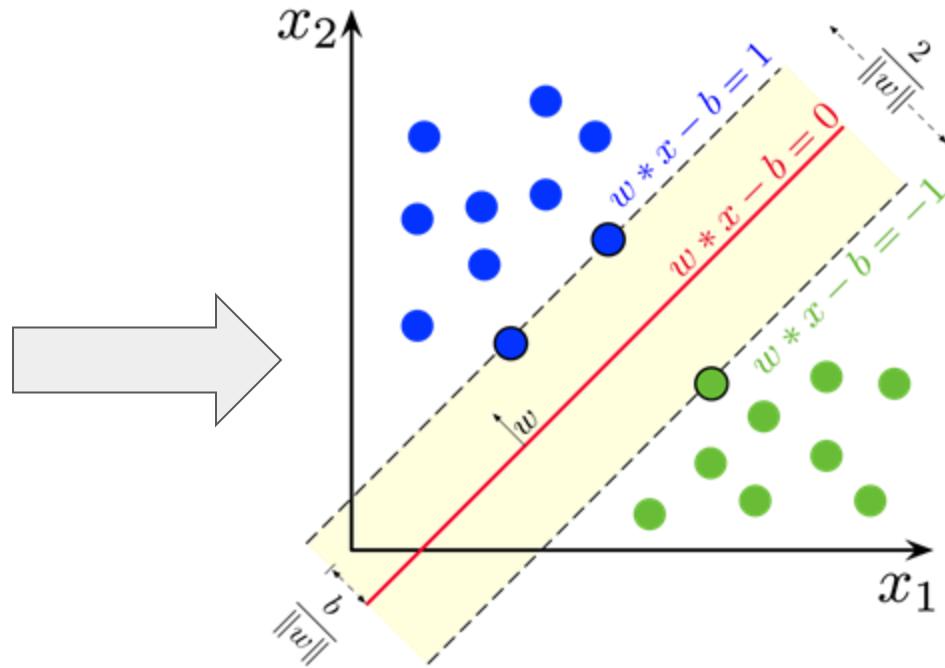
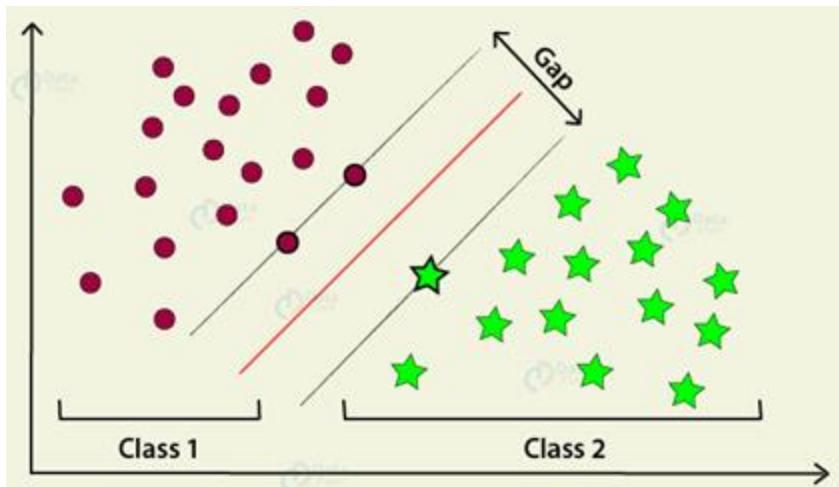
$$m = \frac{Y_1 - Y_2}{X_1 - X_2}$$



Persamaan garis lurus:  
 $y = mx + c$   
 $c = \text{nilai } y \text{ saat } x=0$

# Contoh penggunaan gradient

- Contoh penerapan di algoritma Support Vector Machine (SVM)
- Menentukan garis pembatas antar 2 kelas

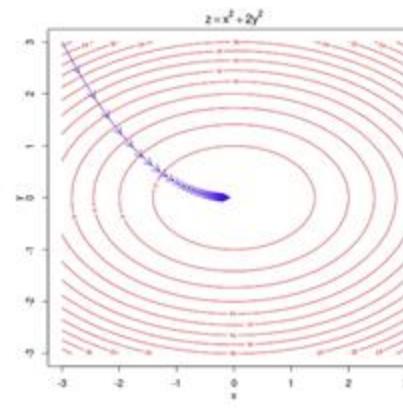
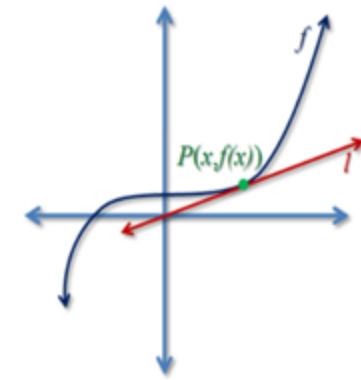


# Kalkulus: turunan & optimisasi

- Turunan (differensial) merupakan pengembangan dari gradient. Turunan dari sebuah fungsi didefinisikan sebagai berikut

$$\frac{df}{dx} := \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

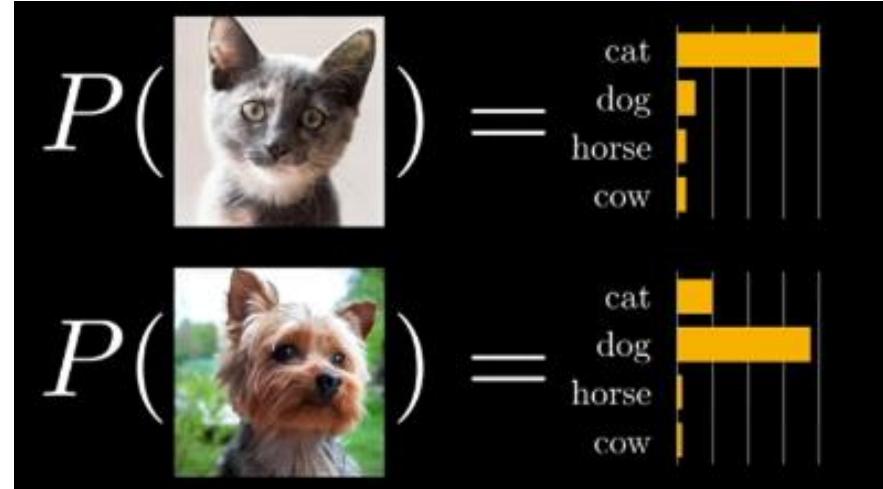
- Turunan sering digunakan untuk mencari nilai maksimum atau minimum
- Titik maksimum/minimum ditemukan dengan menemukan  $x$  yang memenuhi  $f'(x) = 0$
- Nilai maksimum/minimum digunakan untuk memperkecil error atau memaksimalkan jarak 2 kelas yang berbeda.



# Peluang: konsep dasar

Dalam Artificial Intelligence, kita perlu:

1. Menghitung peluang suatu prediksi
2. Menghitung peluang error atau kegagalan
3. Memahami distribusi data
4. Memperkirakan pengaruh suatu variabel terhadap hasil prediksi
5. Memperkirakan hal-hal lain yang berpotensi mempengaruhi performance program
6. dsb



## Konsep dasar:

- Ruang sample (S): himpunan dari semua kemungkinan dari suatu percobaan
- Peluang suatu kejadian A adalah banyaknya himpunan A dibagi banyaknya himpunan S.  $P(A) = n(A) / n(S)$

# Peluang Bersyarat

- 2 kejadian A dan B dikatakan **saling lepas** jika  $A \cap B = \emptyset$ , misalnya kejadian munculnya angka ganjil dan genap saat melempar 1 dadu. Pada kejadian ini  $P(A \cap B) = 0$ .
- Jika kejadian A tidak mempengaruhi kejadian B (dan sebaliknya) maka A dan B dikatakan **saling bebas**. Peluang munculnya kejadian A dan B =  $P(A \cap B) = P(A) \times P(B)$ . Misalnya kejadian terjadinya kecelakaan dan hujan
- Peluang kejadian A atau B =  $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- Peluang terjadi suatu kejadian H bila diketahui bahwa kejadian X telah terjadi disebut **peluang bersyarat** yang dikenal dalam **teorema bayes**

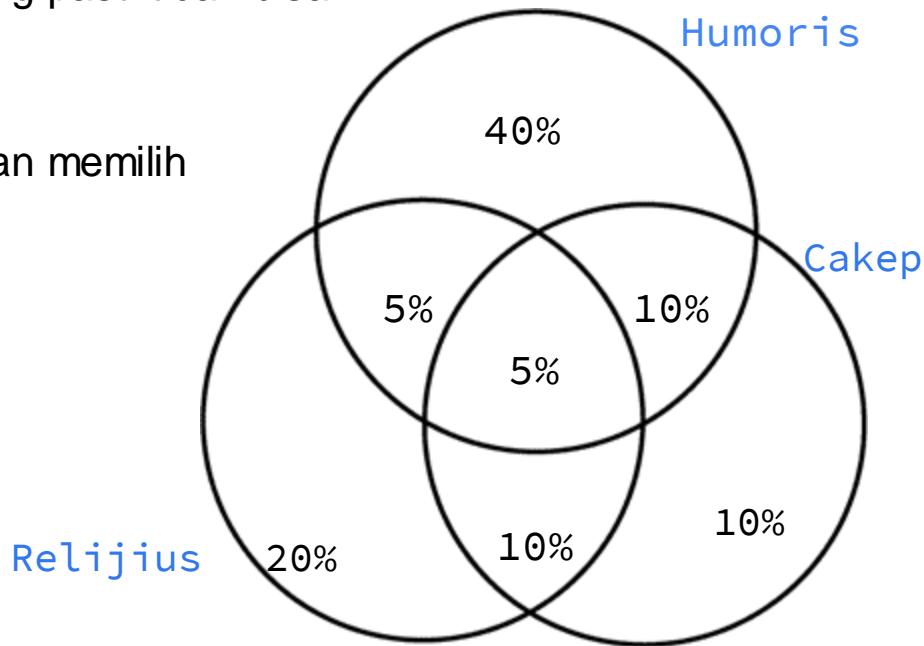
$$P(H|X) = \frac{P(H \cap X)}{P(X)} = \frac{P(X|H)P(H)}{P(X)}$$

# Tebak-tebakan matematika

1. Manakah yang lebih baik dalam hidup ini? Menjadi skalar atau vektor?
2. Jika  $e^x$ ,  $1/x$ , dan  $x^2$  diibaratkan peluang gubernur petahana yang akan maju dalam pilkada tahun ini, manakah yang pasti tidak bisa diturunkan dari jabatannya?
3.  $d(\text{sapi}) / d(\text{daging}) = ?$
4. Bayangkan anda pemimpin partai yang akan memilih kandidat untuk maju ke pilkada daerah A.  
Setelah disurvei:

Mana kandidat yang lebih potensial?

- a. Kandidat humoris + religius?
- b. Kandidat humoris + cakep
- c. Kandidat religius + cakep



# Tebak-tebakan matematika

1. Vektor, karena hidup itu harus punya nilai dan juga arah
2.  $e^x$ , karena  $d(e^x) = e^x$ , diturunkan berapa kali pun tetap  $e^x$
3.  $d(\text{sapi}) / d(\text{daging}) = ?$  Bisa rendang, sop, sate, dll
  
4. Mana kandidat yang lebih potensial?

a. Kandidat yang humoris dan relijius?

$$P(\text{humoris}) = 0.4 + 0.05 + 0.05 + 0.1 = 0.6$$

$$P(\text{relijius}) = 0.2 + 0.05 + 0.05 + 0.1 = 0.4$$

$$P(h \cap r) = 0.05 + 0.05 = 0.1$$

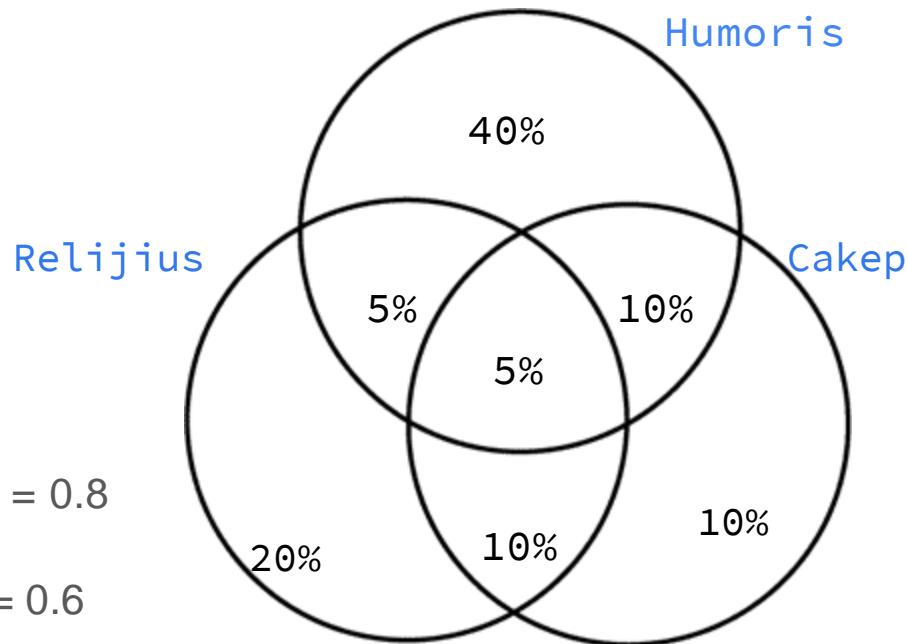
$$P(h \cup r) = 0.6 + 0.4 - 0.1 = 0.9$$

b. Kandidat humoris dan cakep

$$P(h \cup c) = P(h) + P(c) - P(h \cap c) = 0.6 + 0.35 - 0.15 = 0.8$$

c. Kandidat relijius dan cakep

$$P(r \cup h) = P(r) + P(h) - P(r \cap h) = 0.4 + 0.35 - 0.15 = 0.6$$



## Video lainnya seputar matematika di Artificial Intelligence

1. FBI menggunakan aljabar untuk mengolah data fingerprint:

<https://youtu.be/fRjFwTbJfes>

2. Jarak Euclidean vs Manhattan distance :

[https://youtu.be/Usngvpiv\\_LI](https://youtu.be/Usngvpiv_LI)

3. Visualisasi Nilai Eigen dan Vektor Eigen :

<https://youtu.be/PFDu9oVAE-g>

## Operasi Matematika & NumPy

# NumPy



Library Python untuk membantu operasi matematika

Image Source: <https://numpy.org/>

# Operasi Matematika & NumPy

## 1) Contoh Penggunaan

- Meng-import library NumPy:

```
import numpy as np
```

- Membuat 1D array (vektor)

```
a = np.array([3, 1, 5, 6])
```



$$\begin{pmatrix} 3 \\ 1 \\ 5 \\ 6 \end{pmatrix}$$

begitupun 3D (tensor)

- Membuat 2D array (matriks)

```
A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```



$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

# Operasi Matematika & NumPy

## 2) Indexing & Slicing (Pemisahan)

Index array NumPy dimulai dari **0**

- Array 1D (vektor)

`a = np.array([3, 1, 5, 6])`



$$\begin{pmatrix} 3 \\ 1 \\ 5 \\ 6 \end{pmatrix}$$



`a[0] = 3`  
`a[1] = 1`  
`a[2] = 5`  
`a[3] = 6`

banyaknya entri → **a.size** : 4

dimensi array → **a.ndim** : 1 (menunjukkan vektor)

ukuran array → **a.shape** : (4,)

# Operasi Matematika & NumPy

## 2) Indexing & Slicing (Pemisahan)

Index array NumPy dimulai dari **0**

- Array 1D (vektor)

`a = np.array([3, 1, 5, 6])`



$$\begin{pmatrix} 3 \\ 1 \\ 5 \\ 6 \end{pmatrix}$$



`a[0] = 3`  
`a[1] = 1`  
`a[2] = 5`  
`a[3] = 6`

banyaknya entri → **a.size** : 4

dimensi array → **a.ndim** : 1 (menunjukkan vektor)

ukuran array → **a.shape** : (4,)

# Operasi Matematika & NumPy

## 2) Indexing & Slicing (Pemisahan)

- Array 1D (vektor)

a = np.array([3, 1, 5, 6])



$$\begin{pmatrix} 3 \\ 1 \\ 5 \\ 6 \end{pmatrix}$$



a[0] = 3  
a[1] = 1  
a[2] = 5  
a[3] = 6

### Slicing:

a[1:] : array([1, 5, 6])

a[1:3] : array([1, 5])

a[:3] : array([3, 1, 5])

# Operasi Matematika & NumPy

## 2) Indexing & Slicing (Pemisahan)

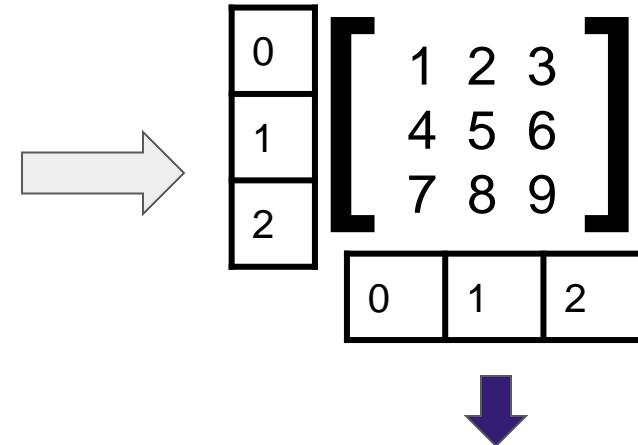
- Array 2D (matriks)

`A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`

banyaknya entri → `A.size` : 9

dimensi array → `A.ndim` : 2 (menunjukkan matriks)

ukuran array → `A.shape` : (3,3)



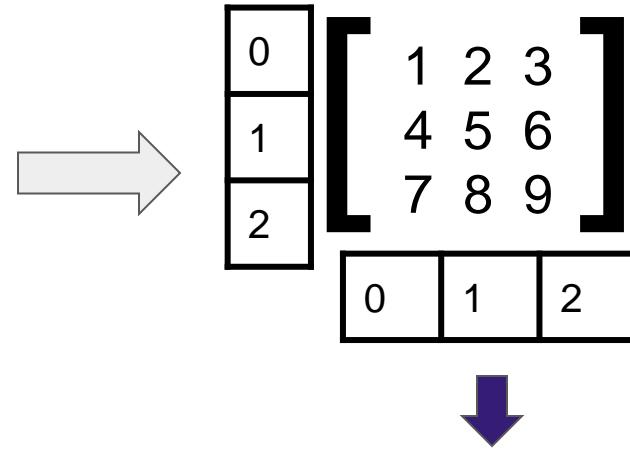
$A[0][2] = 3$   
 $A[0, 2] = 3$   
 $A[2][0] = 7$   
 $A[2, 0] = 7$

# Operasi Matematika & NumPy

## 2) Indexing & Slicing (Pemisahan)

- Array 2D (matriks)

`A = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])`



### Slicing:

`A[1]` : array([4,5,6])

`A[1, 1:]` : array([5, 6])

`A[:2]` : array([[1, 2, 3], [4, 5, 6]])

`A[:2, 1]` : array([2, 5])

$A[0][2] = 3$   
 $A[0, 2] = 3$   
 $A[2][0] = 7$   
 $A[2, 0] = 7$

# Operasi Matematika & NumPy

## 3) Operasi Matematika



### A) Penjumlahan/Pengurangan Array & Dot Product Vektor

Misal:

`A = np.array([[1, 2], [7,8]])`

`B = np.array([[3, 5], [1,6]])`

`2*A` : `array([[ 2, 4], [14, 16]])`

`2+A` : `array([[ 3, 4], [9, 10]])`

`A+B` : `array([[ 4, 7], [ 8, 14]])`

`A-B` : `array([[-2, -3], [ 6, 2]])`

Misalkan :

$$\begin{matrix} C \\ \left[ \begin{matrix} 1 \\ 2 \end{matrix} \right] \end{matrix} \quad \begin{matrix} D \\ \left[ \begin{matrix} 3 \\ 5 \end{matrix} \right] \end{matrix} \quad C \bullet D = \\ 1 \times 3 + 2 \times 5 = 13$$

**Perkalian dot product vektor:**  
`np.dot(C, D)`

# Operasi Matematika & NumPy

## 3) Operasi Matematika



### B) Perkalian Matriks

Perkalian antar entri

$$\begin{bmatrix} 1 & 2 \\ 7 & 8 \end{bmatrix} * \begin{bmatrix} 3 & 5 \\ 1 & 6 \end{bmatrix} =$$

A                    B

$$\begin{bmatrix} 1 \times 3 & 2 \times 5 \\ 7 \times 1 & 8 \times 6 \end{bmatrix} = \begin{bmatrix} 3 & 10 \\ 7 & 48 \end{bmatrix}$$

A\*B: array([[ 3, 10], [ 7, 48]])

Perkalian matriks A & B

$$AB = \begin{bmatrix} (1 \times 3) + (2 \times 1) & (1 \times 5) + (2 \times 6) \\ (7 \times 3) + (8 \times 1) & (7 \times 5) + (8 \times 6) \end{bmatrix}$$

$$= \begin{bmatrix} 5 & 17 \\ 29 & 83 \end{bmatrix}$$

`np.matmul(A, B):`  
`array([[ 5, 17], [29, 83]])`

### C) Operasi Matematika Lainnya

- mean → np.mean(A)
- median → np.median(A)
- nilai maksimum → np.max(A)
- sorting → np.sort(A)
- transpose → np.transpose(A)
- determinan → np.linalg.det(A)
- invers → np.linalg.inv(A)
- eigen value & vector → np.linalg.eig(A)
- generate matriks yg entrinya 0 semua ukuran pxq → np.zeros(p,q)
- generate matriks identitas ukuran pxp → np.identity(p)

# References

- “Mathematics for Machine Learning”, by M. P. Deisenroth, A. A. Faisal, and C. S. Ong (Cambridge University Press). <https://mml-book.com>.
- “Kalkulus” by Dale Varberg, Edwin Purcell, and Steven E. Rigdon
- “Aljabar Linier Elementer” by Anton and Rorres
- “Pengantar Statistika” by Ronald E Walpole
- “Statistika Tanpa Stres” by Sarini A. dan Taufik E.S.

# Let's Code!



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

- Orbit Future Academy
- @OrbitFutureAcademyInd
- OrbitFutureAcademy
- Orbit Future Academy

# AI Mastery Course



Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

## Modul 2

Introduction to everything

### Bagian

Data analytics with python-  
Advance statistics





# Learning Objectives

- Di akhir modul ini, Anda akan dapat:
  - Memahami teori peluang
  - Memahami peluang bersyarat dan teorema bayes
  - Menjelajahi aturan empiris dan teorema chevbeys
  - Menjelajahi skor z, uji z
  - Memahami ide logis dari uji t, uji t satu sampel dan uji t berpasangan
  - Menjelajahi ANOVA dan uji chi square



# Agenda

01

## STATUS INFERENSIAL

- Pengujian hipotesis
- Error dan selang kepercayaan
- Distribusi normal

02

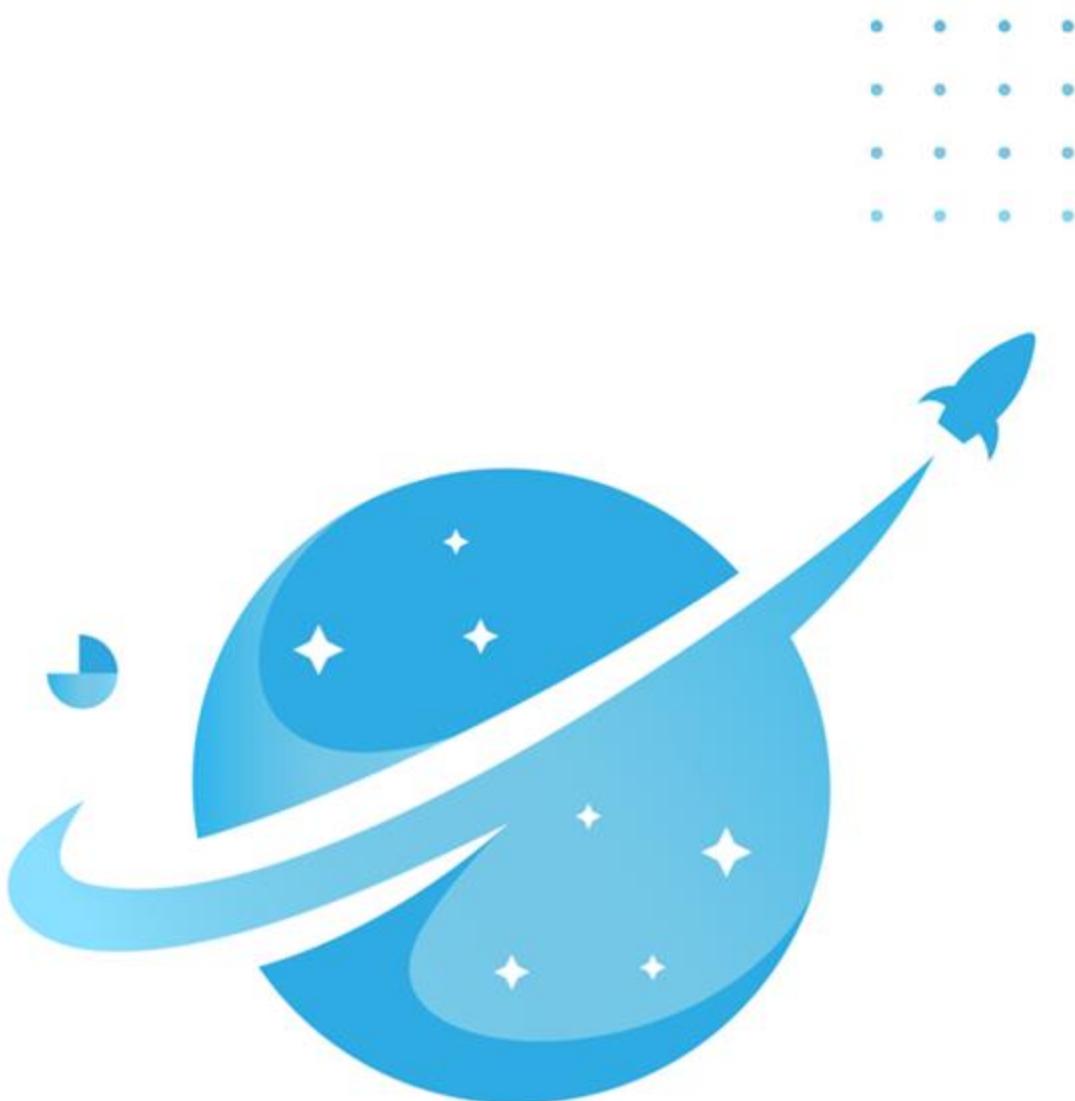
## UJI HIPOTESIS

- uji T
- Tes Chi Square
- ANOVA

03

## KESIMPULAN

- Ulangan
- Ringkasan



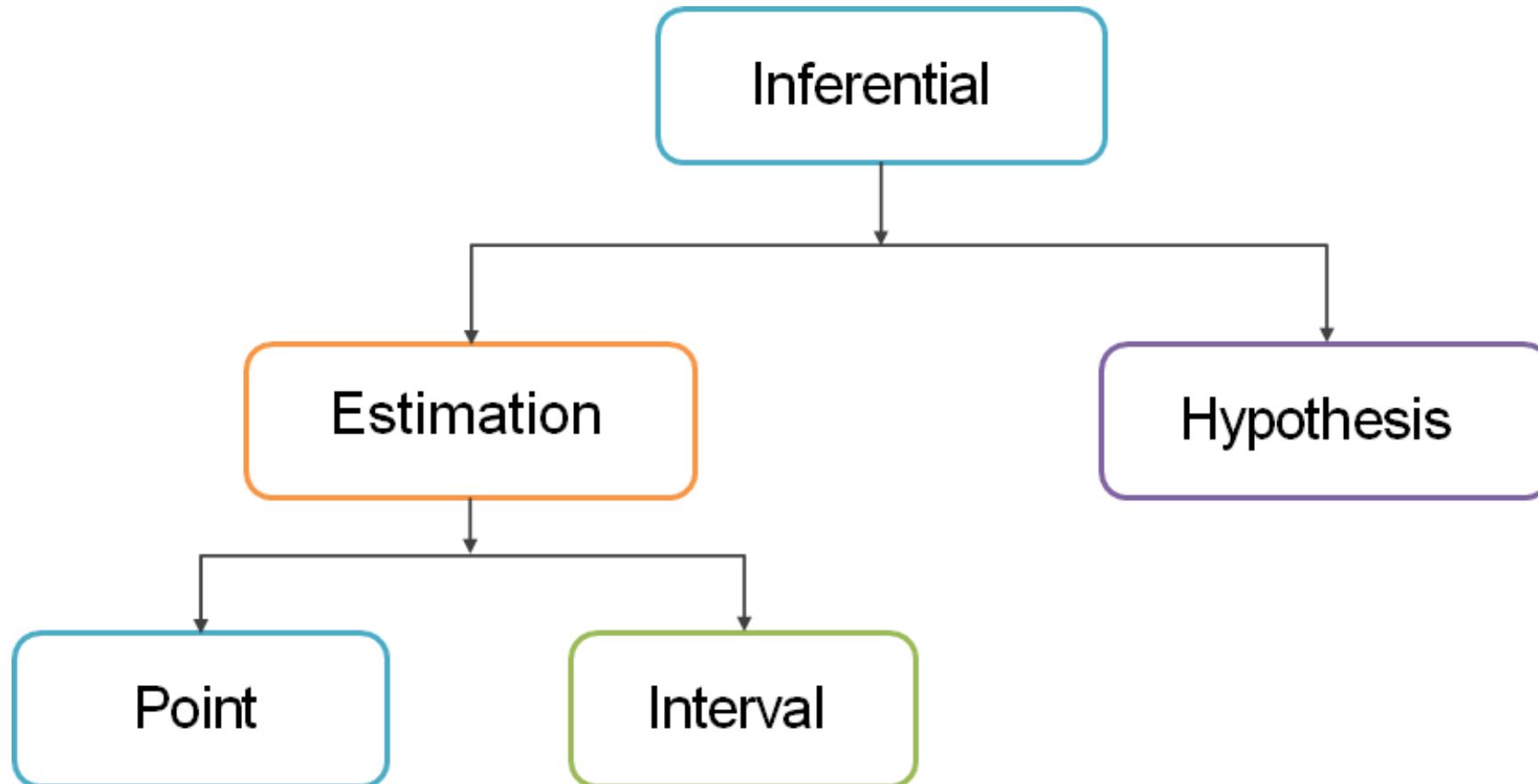
01

## STATISTIK INFERENSIAL

- Pengujian hipotesis
- Kesalahan Tipe I & II
- Distribusi normal

# Statistik Inferensial

Statistik Inferensial terdiri dari pengujian hipotesis dan estimasi



# Hipotesis

- Hipotesis berasal dari bahasa Yunani, yaitu *Hypo* dan *Thesis*. *Hypo* berarti lemah, kurang, atau di bawah, sedangkan *Thesis* berarti teori atau pernyataan yang disajikan dengan bukti.
- Hipotesis adalah pernyataan yang masih bernilai lemah dan perlu dibuktikan

$H_0$  : Hipotesis Nol



Hipotesis nol sering kali merupakan klaim awal yang didasarkan pada analisis sebelumnya atau pengetahuan khusus.

$H_a$  : Hipotesis Alternatif



Hipotesis alternatif adalah pernyataan yang digunakan peneliti untuk membuktikan suatu kebenaran berdasarkan data sampel

# Kriteria Konstruksi Hipotesis

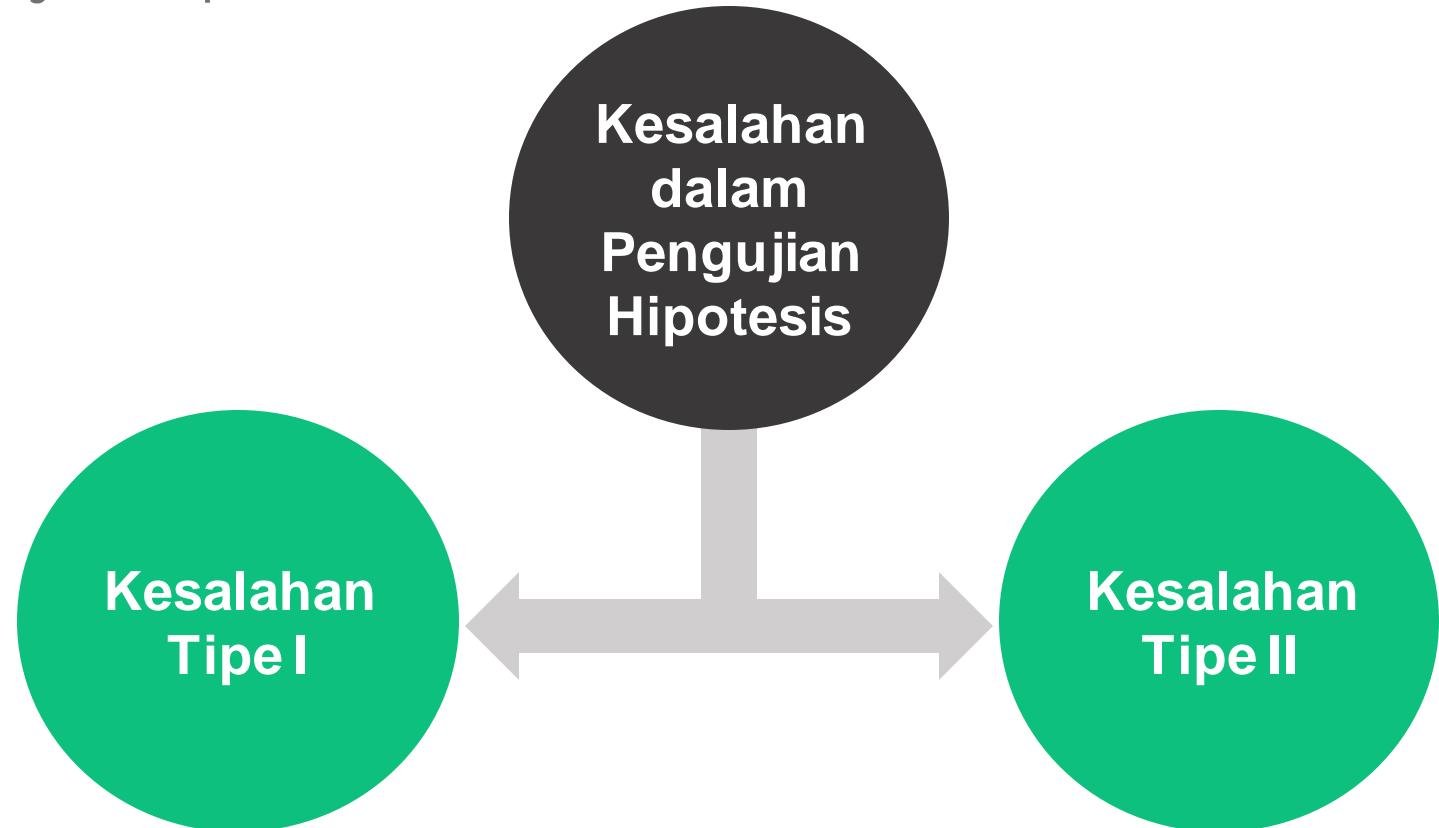


Sebuah hipotesis harus memenuhi kriteria berikut:

- Harus dapat diuji secara empiris, apakah itu benar atau salah.
- Harus spesifik.
- Pernyataan dalam hipotesis tidak boleh bertentangan.
- Harus menentukan variabel di mana hubungan itu akan dibangun.
- 1 hipotesis menggambarkan 1 masalah saja.

# Kesalahan dalam Pengujian Hipotesis

- Dua jenis pengujian hipotesis:



# Skenario Kesalahan dalam Pengujian Hipotesis



Tabel berikut merangkum skenario kesalahan dalam pengujian hipotesis.

Keputusan	Ho benar	Ha benar
Tolak Ho	Error Tipe I	Keputusan benar
Terima Ho	Keputusan benar	Error Tipe II

# Error Tipe I dan Selang Kepercayaan



Artificial Intelligence  
Mastery Program

- Peluang terjadinya Error Tipe I disebut juga level signifikansi ( $\alpha$ )
- Lawan dari Peluang Error tipe I adalah Confidence Interval (CI) atau Selang Kepercayaan
- Jadi, jika selang kepercayaan 95%, maka  $\alpha = 5\%$
- Bagaimana pemilihan  $\alpha$  yang tepat? Tergantung kasus, tapi umumnya 5%
- Semakin tinggi resiko --> alpha semakin kecil

Misalnya di kasus efek obat, karena obat menyangkut nyawa seseorang, maka  $\alpha$  maksimal 1%

- Semakin tinggi subjektifitas suatu kasus --> alpha semakin besar
- Misalnya di kasus sosial,  $\alpha$  boleh hingga 10%

# P-value dan keputusan hipotesis

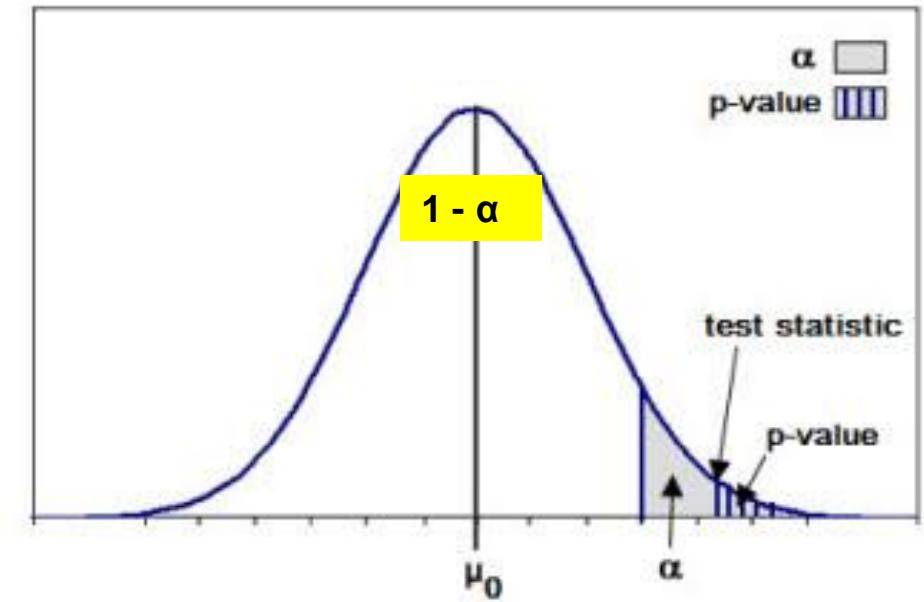
P-value adalah suatu nilai signifikansi untuk menguji hipotesis.

Pengambilan keputusan ini :

- Jika  $p\text{-value} \leq \alpha$ , maka  $H_0$  ditolak
- Jika  $p\text{-value} > \alpha$ , maka  $H_0$  diterima

Misalkan  $\alpha=0.05$ , artinya kamu **mentolerir adanya peluang terjadinya error tipe I sebesar 5%**.

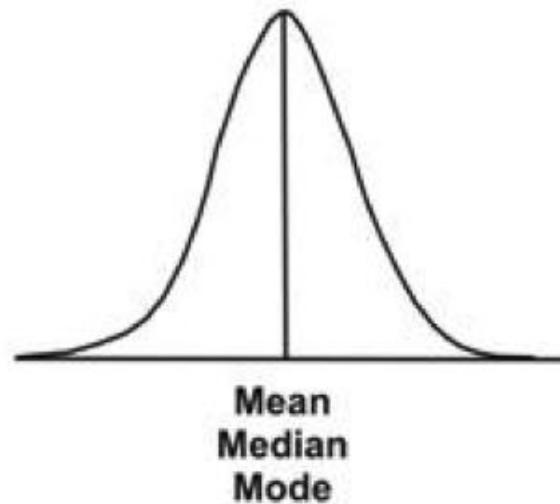
Saat misalnya kamu dapati nilai  $p\text{-value} = 0.03$ , maka kamu menolak  $H_0$  dengan **menerima adanya kemungkinan kesalahan membuat keputusan sebesar 3%**



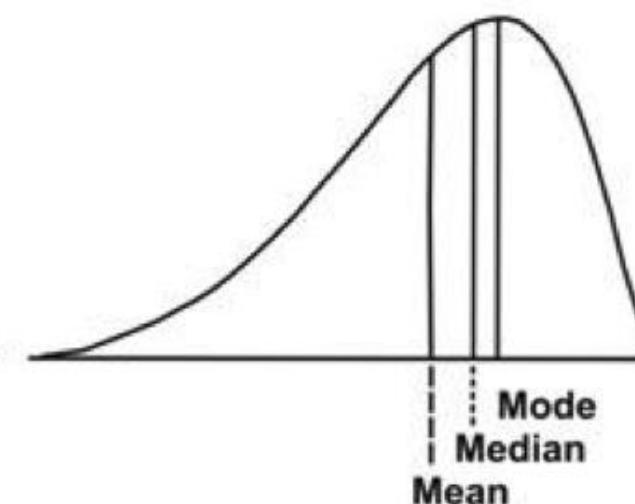
# Distribusi normal

- Distribusi Normal adalah distribusi data dimana nilai modus, mean, dan mediannya mendekati nilai yang sama

Berdistribusi normal



Tidak berdistribusi normal

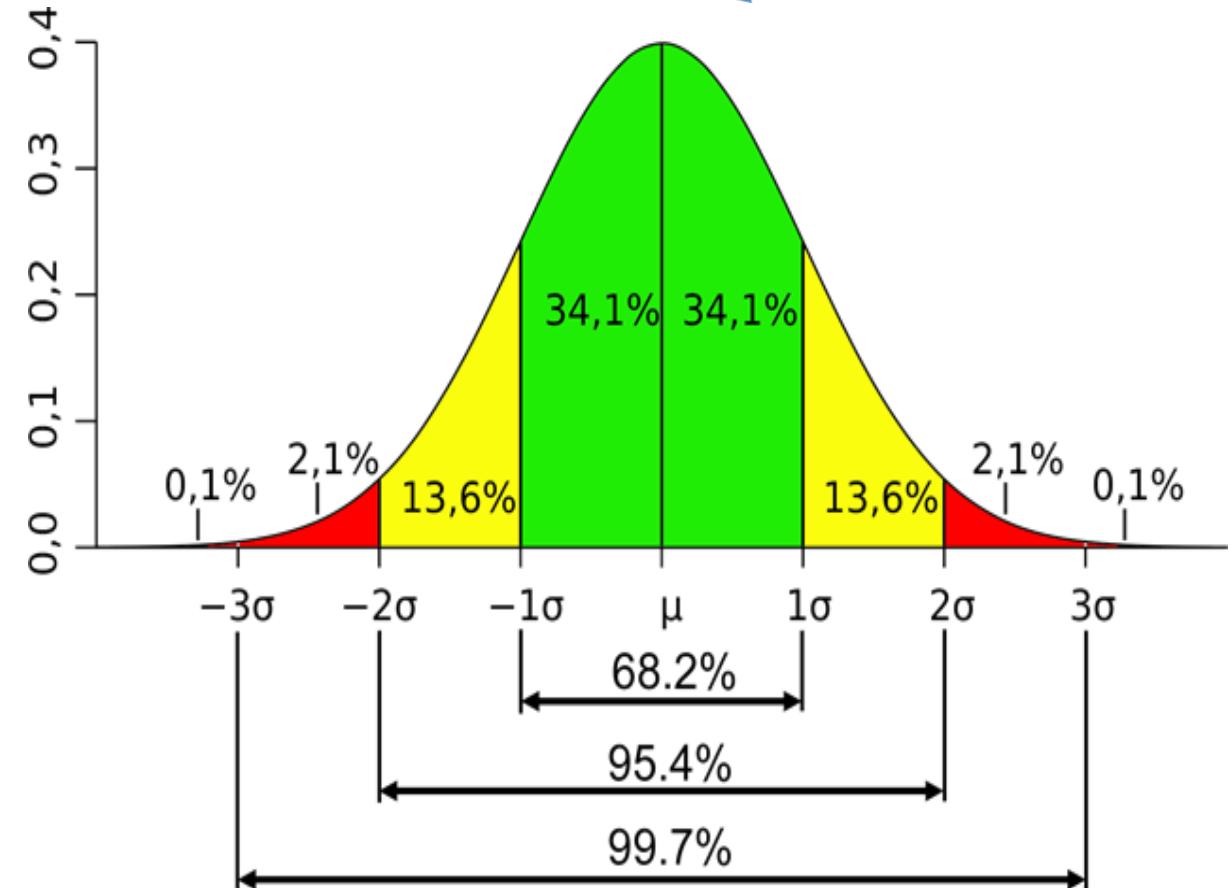


# Sifat-sifat Distribusi Normal

Sifat-sifat distribusi normal adalah sebagai berikut:

- Kurva normal simetris terhadap mean
- Mean berada di tengah dan membagi area menjadi dua (bentuk histogram seperti lonceng)
- 68% data tersebar merata di sekitar mean  $\pm$  standar deviasinya,

95% tersebar di mean  $\pm 2^*\text{st. deviasi}$ , dan 99% tersebar di mean  $\pm 3^*\text{standar deviasi}$



# Mengapa perlu tahu distribusi normal?

- Beberapa uji statistik mengasumsikan data berdistribusi normal
- Cara menguji normalitas data dengan visualisasi :

Cek apakah histogram membentuk lonceng dengan mean pas di tengah atau tidak

- Cara menguji normalitas data dengan uji Kolmogorov-Smirnov :  
Di python, uji ini ada di library scipy :

```
from scipy import stats  
z_score = stats.zscore(data)  
stats.kstest(z_score, "norm")
```

$H_0$  : data berdistribusi normal

p-value > alpha (misalnya 0.05) maka  $H_0$  diterima, data berdistribusi normal





02

## UJI HIPOTESIS

- uji T
- Tes Chi Square
- ANOVA

# UJI BEDA



- Uji beda (comparative analysis) adalah uji statistik untuk menguji pengaruh suatu perlakuan atau perbedaan dari beberapa populasi
- Contoh : pengaruh wfh terhadap kesehatan mental karyawan. Maka kita menganalisis perbedaan kesehatan mental karyawan saat wfo dan wfh
- Contoh lainnya, analisa pemerataan pendidikan di Indonesia. Maka kita membandingkan nilai UN siswa SMA dari tiap provinsi, lalu menganalisa apakah perbedaannya signifikan atau tidak
- Jika data berdistribusi normal, maka statistik yang digunakan adalah parametrik, jika tidak normal maka pilih statistik nonparametrik

# Statistik Parametrik – Uji-T



- Asumsi penggunaan Uji-T adalah data harus berdistribusi normal
- Uji-t digunakan untuk membandingkan 2 kelompok, terdapat 2 jenis kasus:

## 1. Sampel Berhubungan (Paired Sample T-Test)

Yaitu datanya bersumber dari sampel yang sama, misalnya perbedaan kepuasan mahasiswa saat belajar online dengan ms teams dan zoom (apakah tools belajar online mempengaruhi kepuasan mahasiswa?)

## 2. Sampel independent (Independent Sample T-Test)

Yaitu datanya bersumber dari sampel yang berbeda, misalnya analisis perbedaan kemampuan matematika siswa sekolah swasta dan negeri

# Statistik Parametrik – ANOVA



- Analisis of Variance (ANOVA) digunakan untuk membandingkan lebih dari 2 kelompok
- Asumsi dari ANOVA adalah data harus berdistribusi normal dan variansnya harus homogen
- Pengujian homogen ini bisa dilakukan dengan uji Bartlet atau Levene. Kedua uji ini tersedia di library scipy

$$s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}$$

Keterangan:  
 $s^2$ : Varian       $s$  : Standar deviasi

$$s = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

# Statistik Nonparametrik



Artificial Intelligence  
Mastery Program

- Jika asumsi pada statistik parametrik tidak terpenuhi bukan berarti kita tidak bisa menganalisa data, kita bisa menggunakan statistik nonparametrik :

	Data	Asumsi	2 kelompok berhubungan	2 kelompok independen	> 2 kelompok
Parametrik	Numerik	Normalitas	Paired Sample T-test	Independent Sample T-test	ANOVA
Non-Parametrik	Numerik	-	Wilcoxon	Mann-Whitney	Kruskal Wallis, Friedman
	Ordinal	-	Wilcoxon	Mann-Whitney	Kruskal Wallis, Friedman
	Nominal	-	Mc Nemar	Chi-Square	Chi-Square



03

## KESIMPULAN

- Ulangan
- Ringkasan

# Ringkasan

- Uji Hipotesis
- P-value
- Selang kepercayaan
- $H_0$  dan  $H_a$
- Distribusi Normal
- Uji-T
- ANOVA
- Statistik nonparametrik

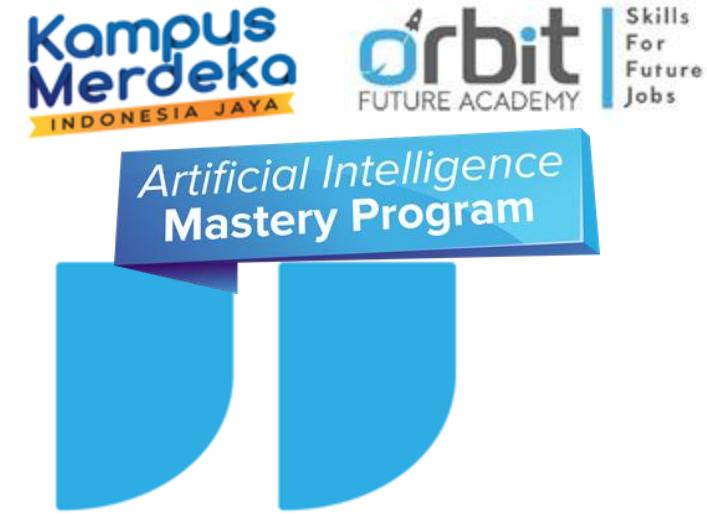


# Review

## Pertanyaan

Jika seseorang ingin mengetahui perbedaan kondisi kesehatan mental mahasiswa sebelum dan saat pandemi, yang manakah uji yang sesuai?

- A. ANOVA – uji f
- B. uji T sampel independent
- C. uji T sampel berhubungan
- D. Tes chi kuadrat



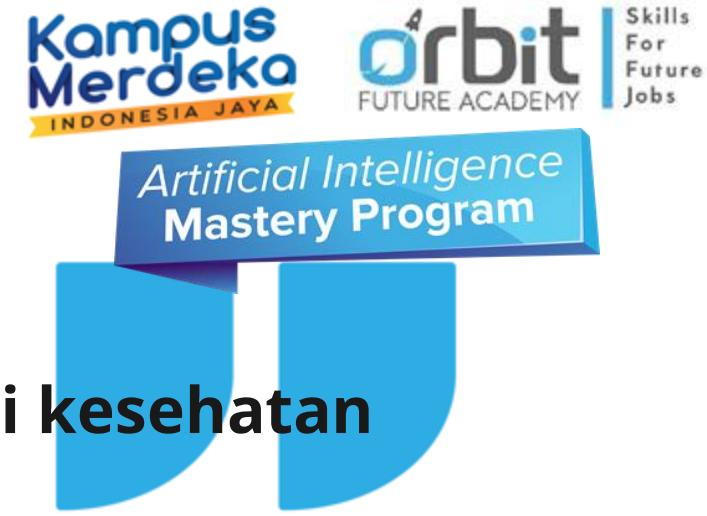
# Review

## Pertanyaan

Jika seseorang ingin mengetahui perbedaan kondisi kesehatan mental mahasiswa sebelum dan saat pandemik, yang manakah uji yang sesuai?

- A. ANOVA – uji f
- B. uji T sampel independent
- C. uji T sampel berhubungan
- D. Tes chi kuadrat

Jawaban : C, karena walaupun datanya ada 2 (sebelum dan saat pandemi) tapi bersumber dari sampel yang sama



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# AI Mastery Course



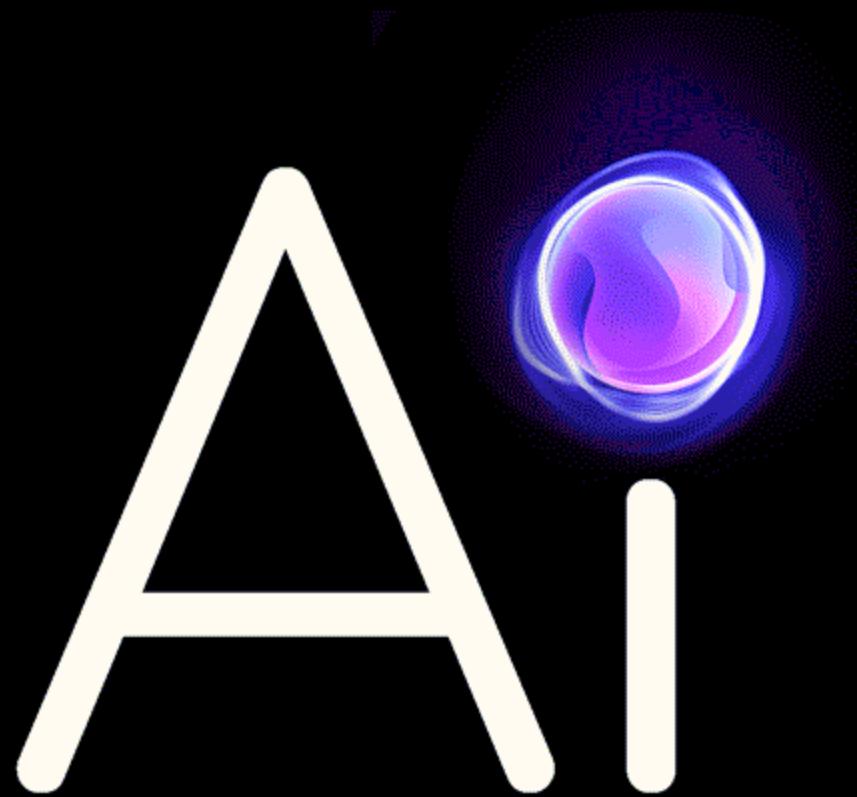
## Module

Artificial Intelligence Fundamental

## Section

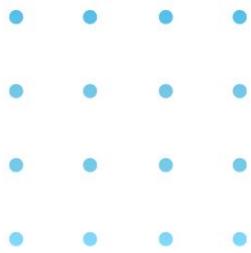
Artificial Intelligence





AI

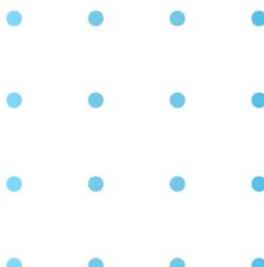




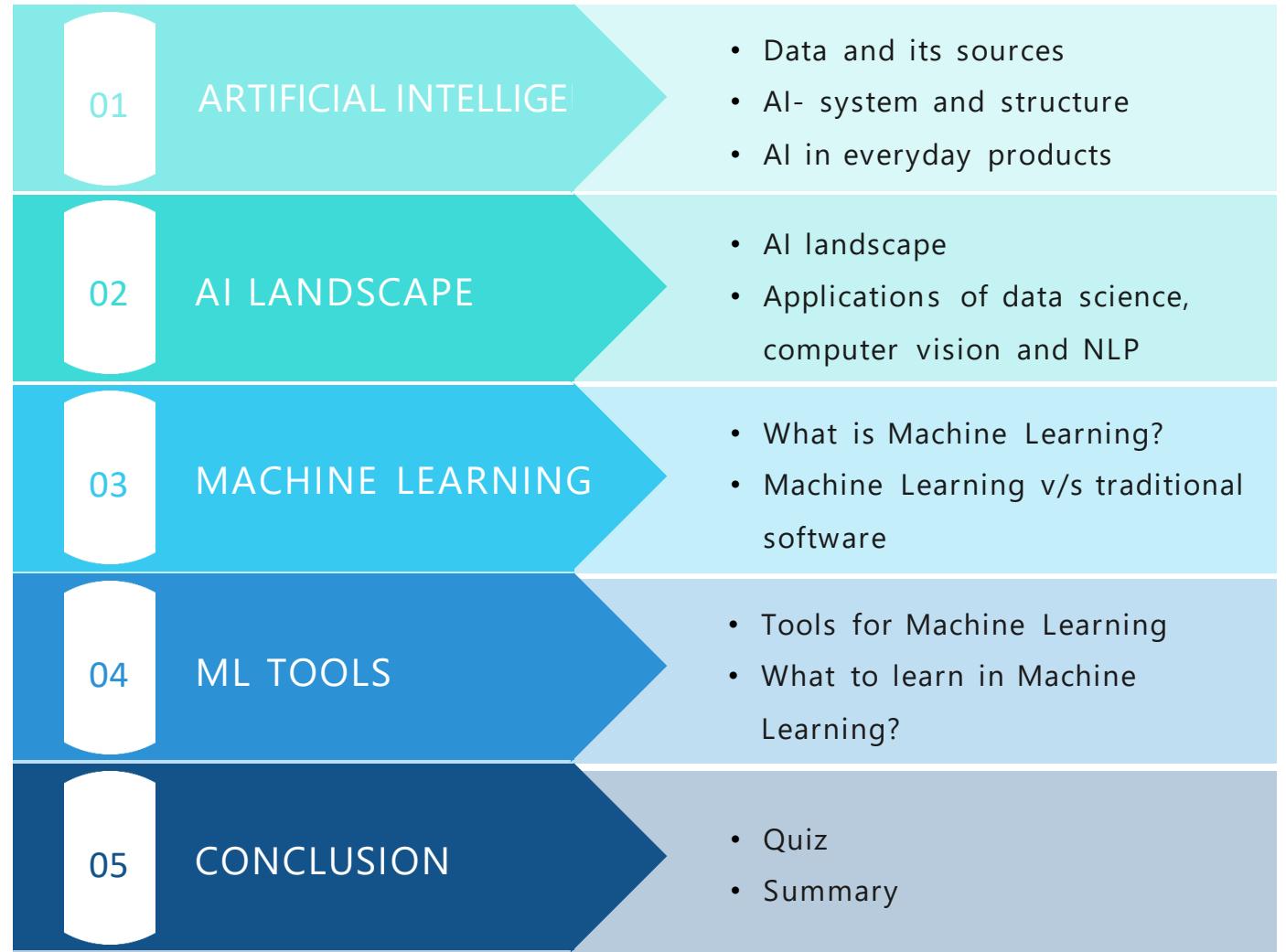
# Learning Objectives

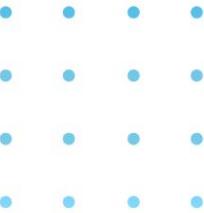
Pada akhir pembelajaran ini, diharapkan Anda untuk mampu:

- Memahami mengenai AI atau kecerdasan buatan dan evolusinya
- Memahami hubungan antara AI, Machine Learning and Deep Learning
- Dapat menghubungkan bidang lain seperti Data Science, Computer Vision and NLP with AI
- Dapat memahami konsep dibalik machine learning
- Dapat memahami proses standar dari machine learning



# Agenda



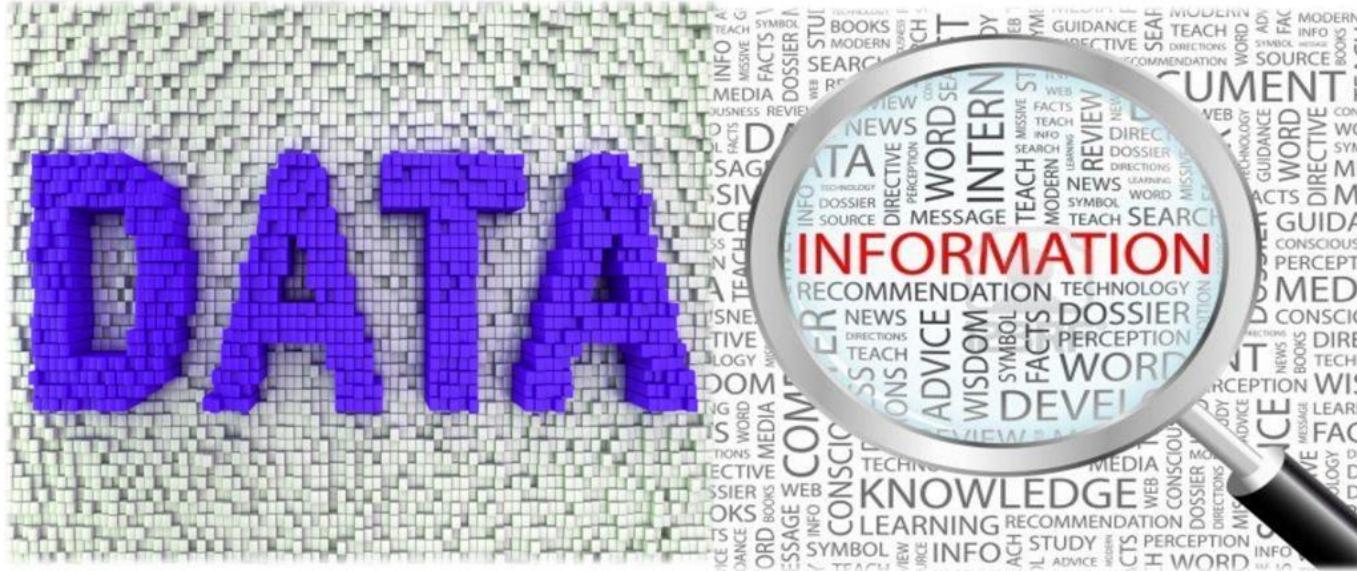


## 01 ARTIFICIAL INTELLIGENCE

- Data and its sources
- AI- system and structure
- AI in everyday products



# What is Data?



Structured

- Business Data, Excel, CSV

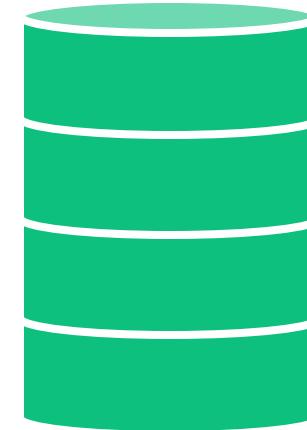
Unstructured

- Text, Images and Speech

# What is Data?

## Data

- Kumpulan fakta-fakta yang ada disekitar kita
- Bentuk jamak dari kata 'datum'
- Kumpulan fakta dan statistik dalam bentuk mentah atau tidak terorganisir
- Kumpulan fakta dan statistik yang digunakan untuk referensi dan analisis



## Digital data

- Informasi yang diproses atau disimpan oleh komputer dalam bentuk dokumen teks, gambar, audio, video, dll.



# Sources of Data

## COMMON SOURCES:

Sensors/meters and activity records from electronic devices

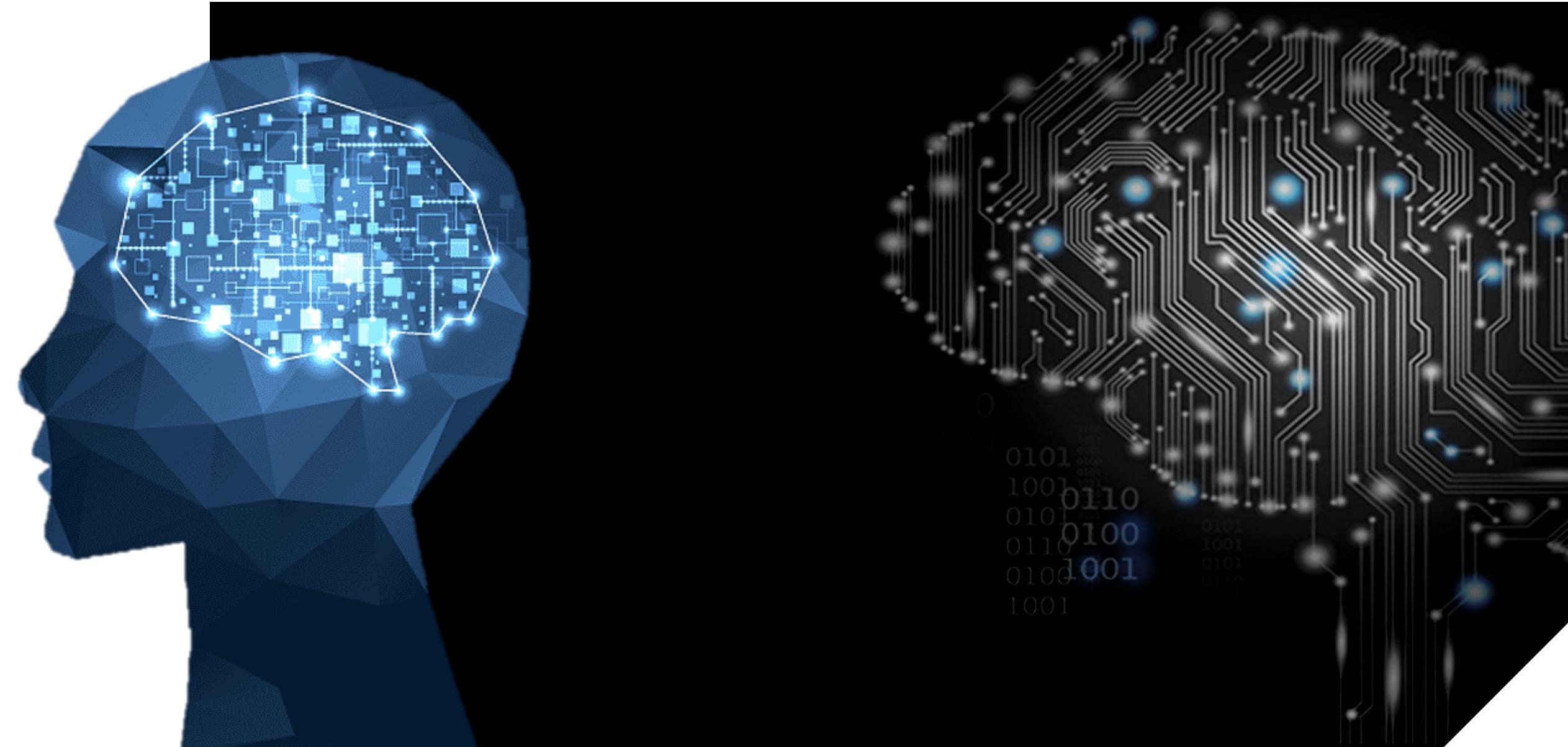
Social interactions

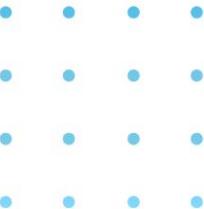
Business transactions

Electronic files

Broadcastings

# What is AI?





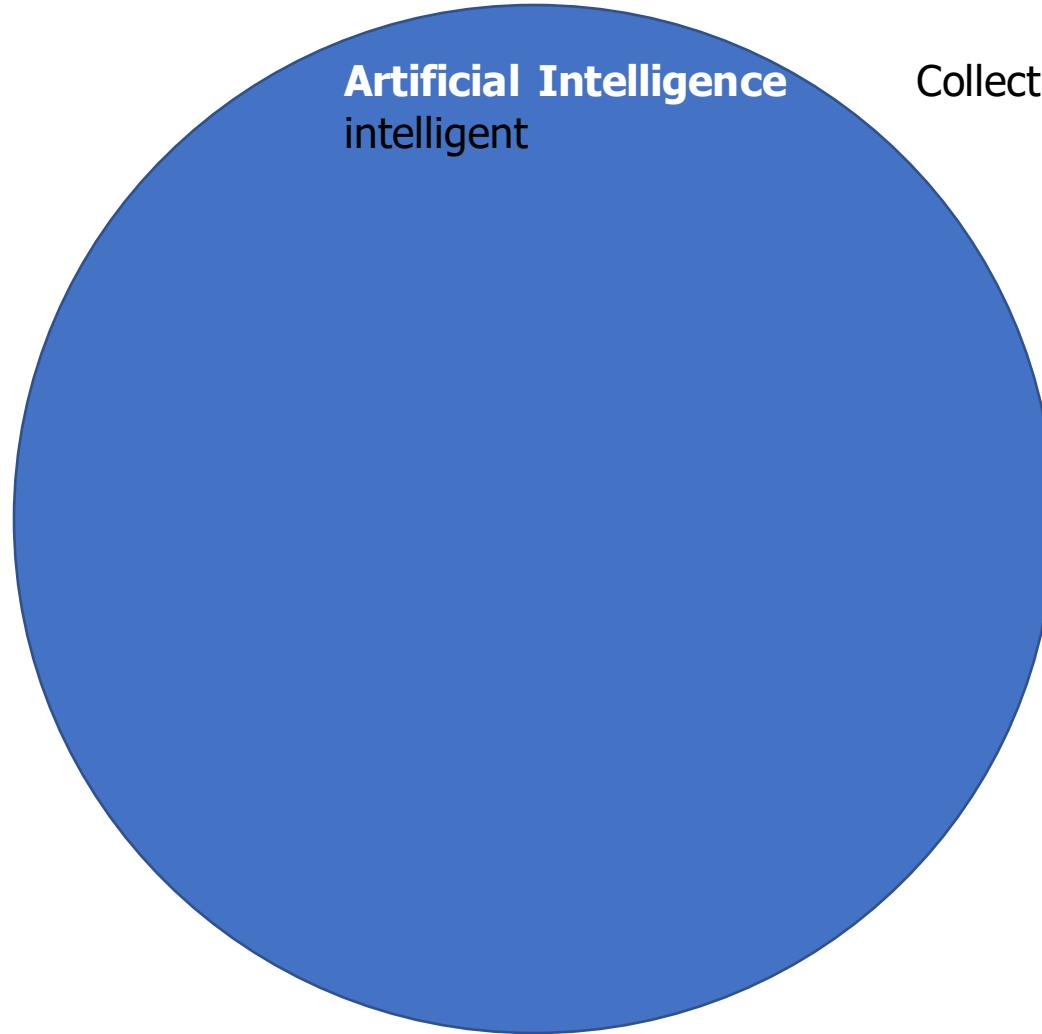
02

## AI LANDSCAPE

- AI landscape
- Applications of data science,  
computer vision and NLP

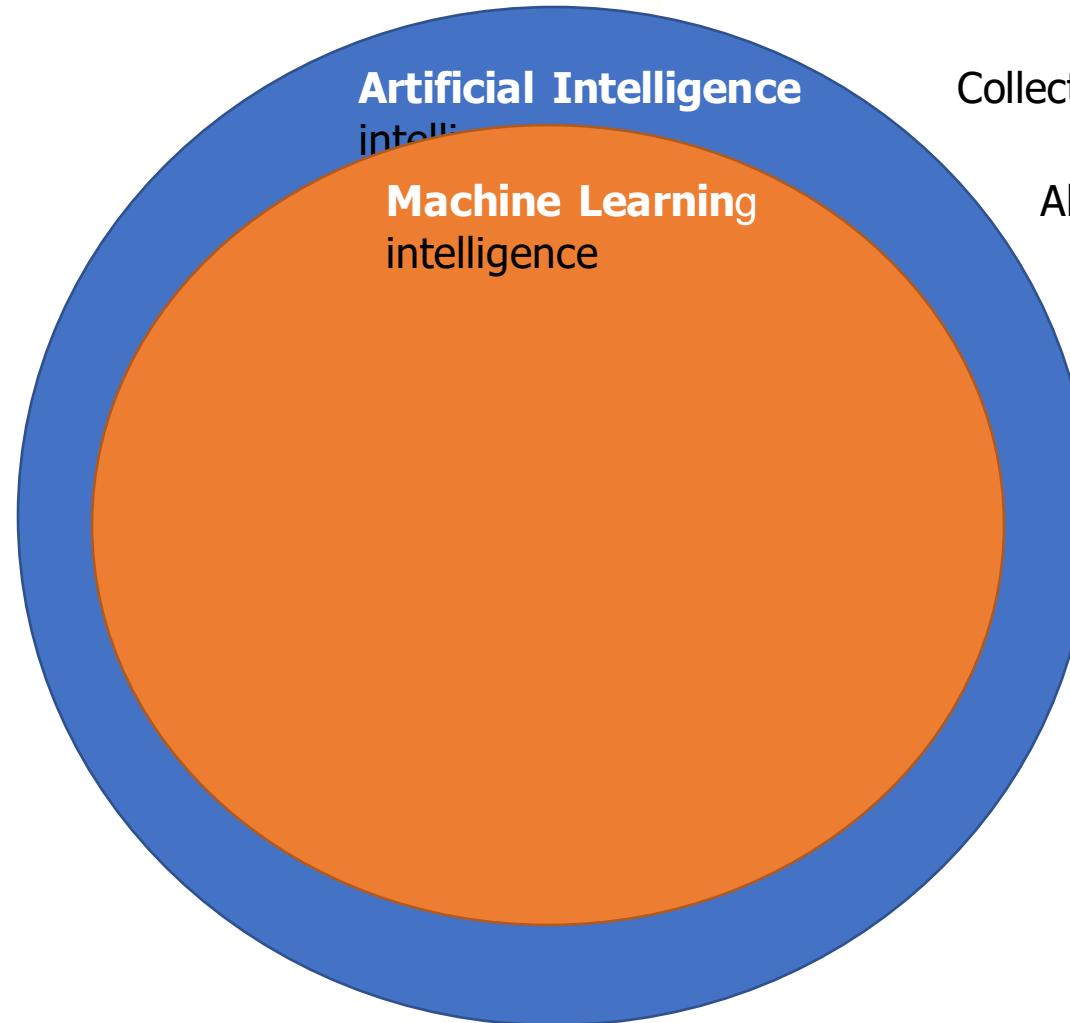


# AI Landscape



Collection of algorithms used to build

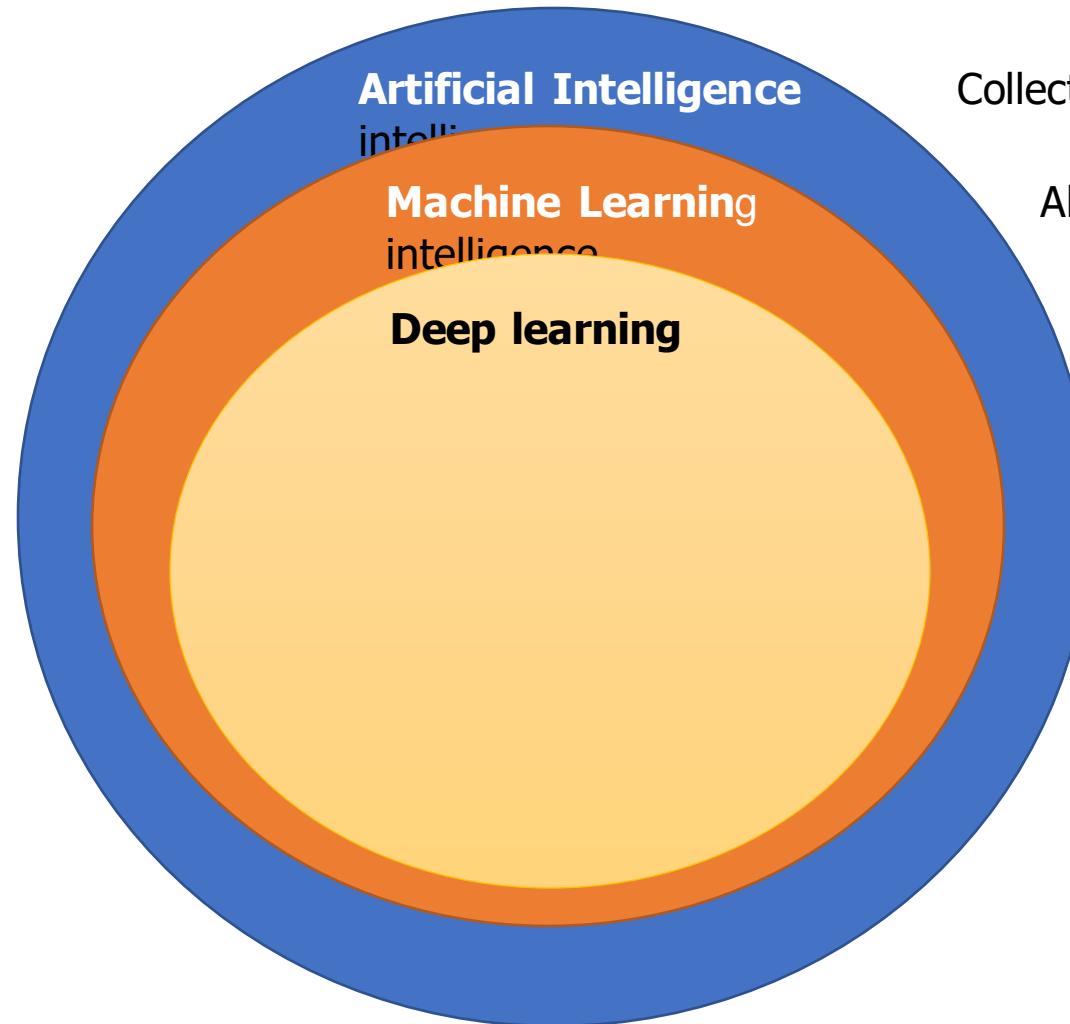
# AI Landscape



Collection of algorithms used to build

Algorithms which use DATA to build

# AI Landscape

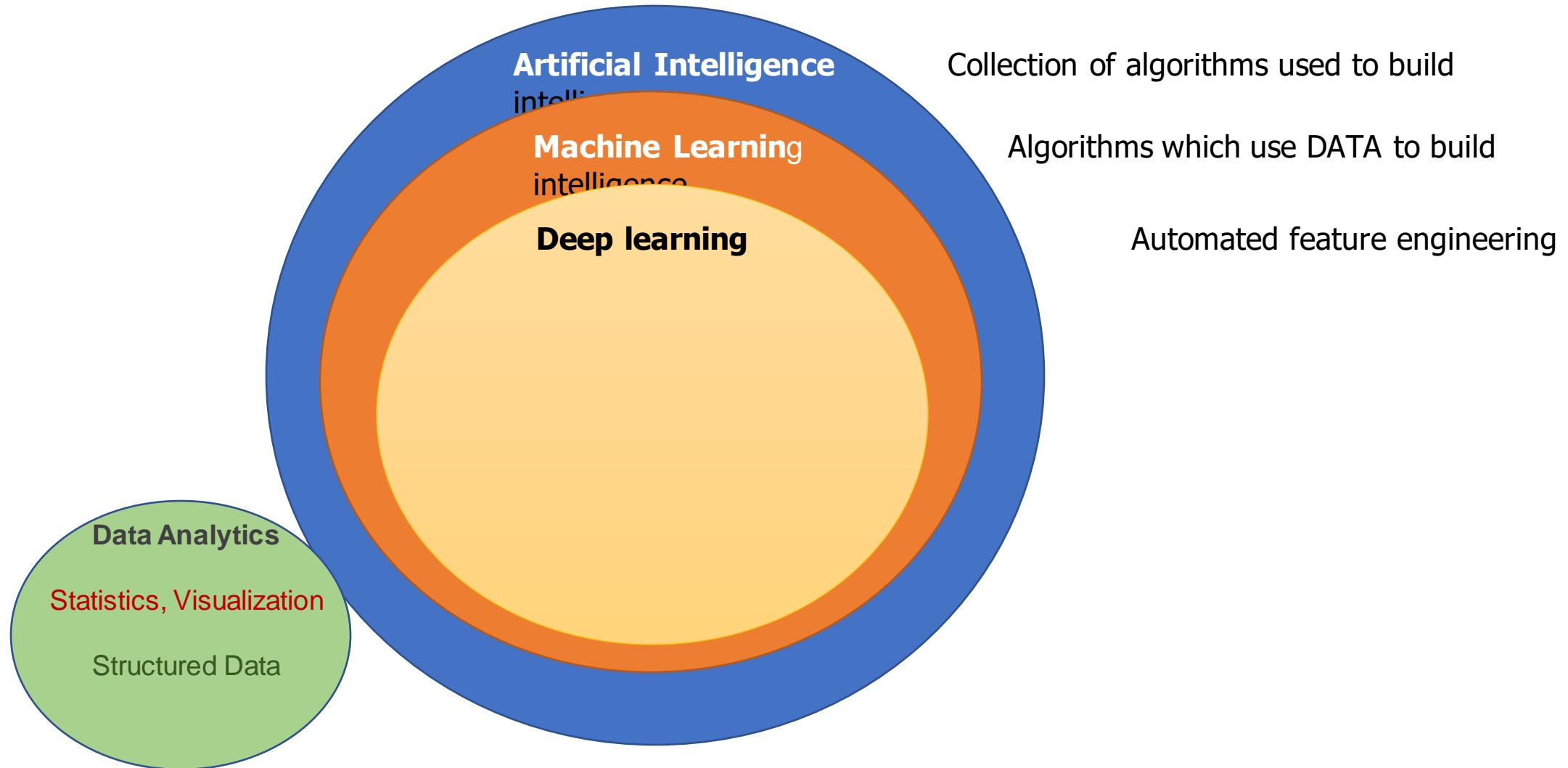


Collection of algorithms used to build

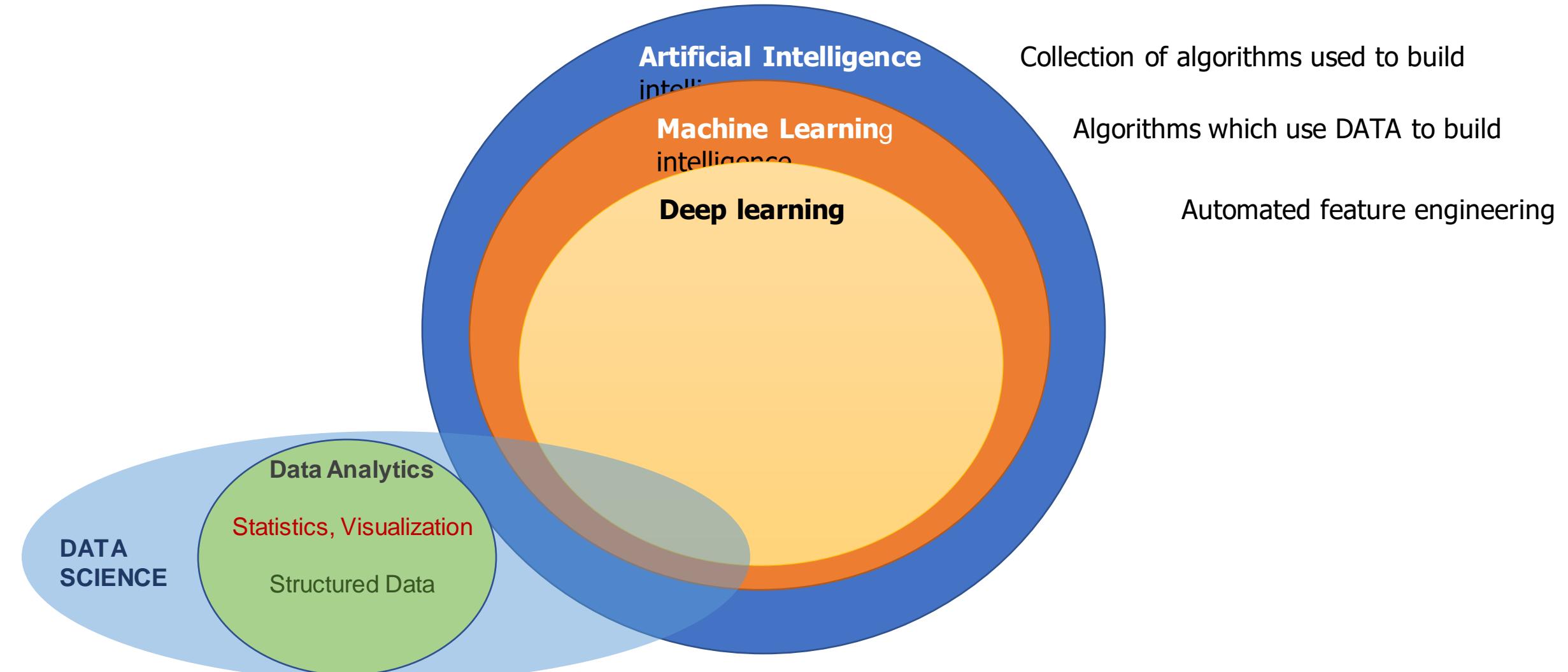
Algorithms which use DATA to build

Automated feature engineering

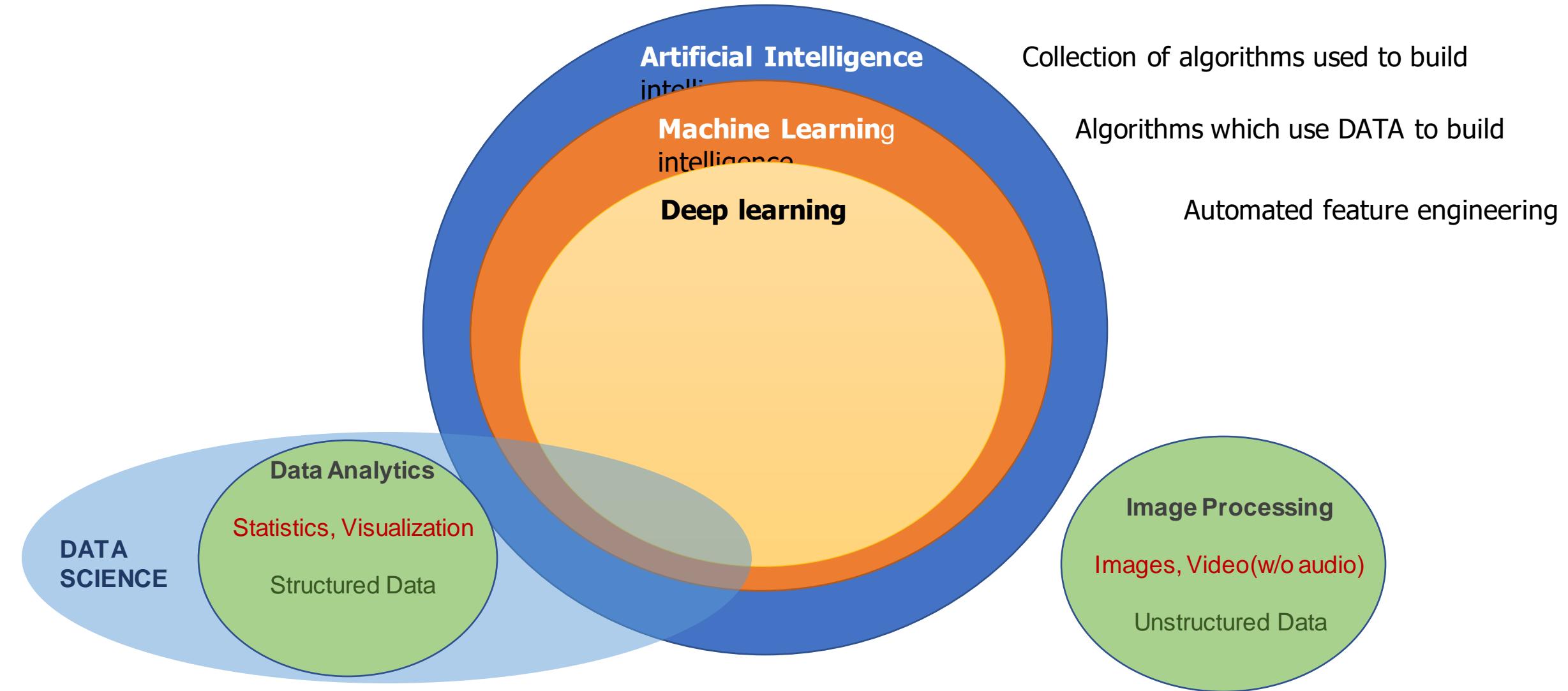
# AI Landscape



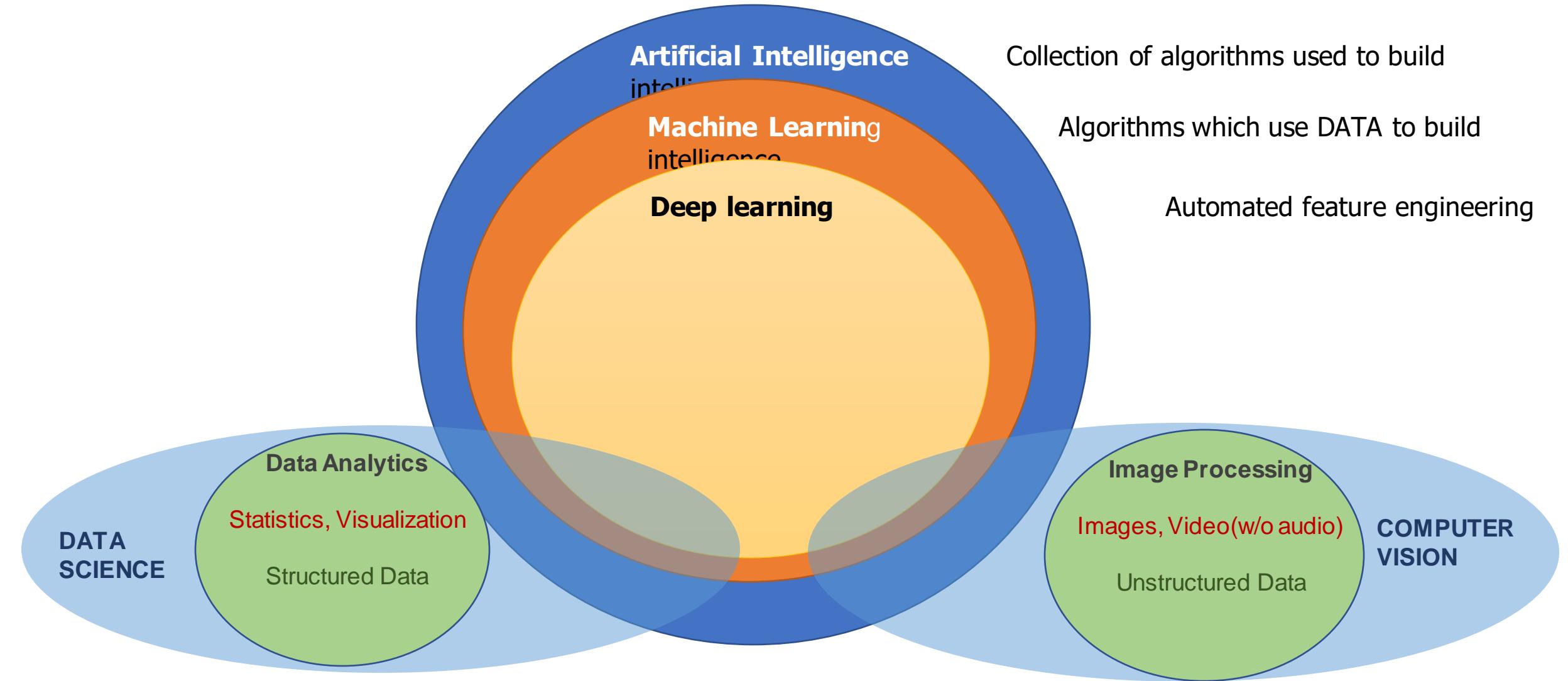
# AI Landscape



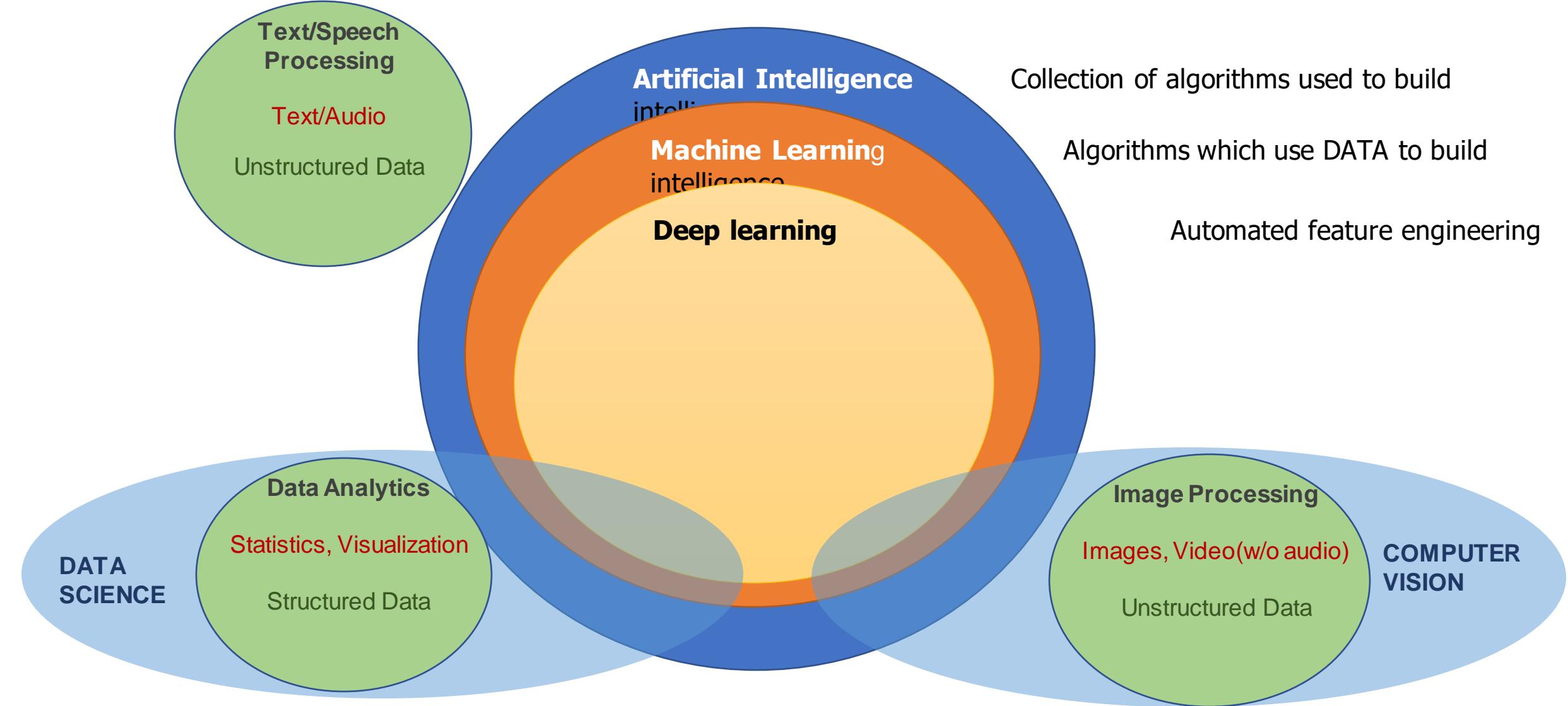
# AI Landscape



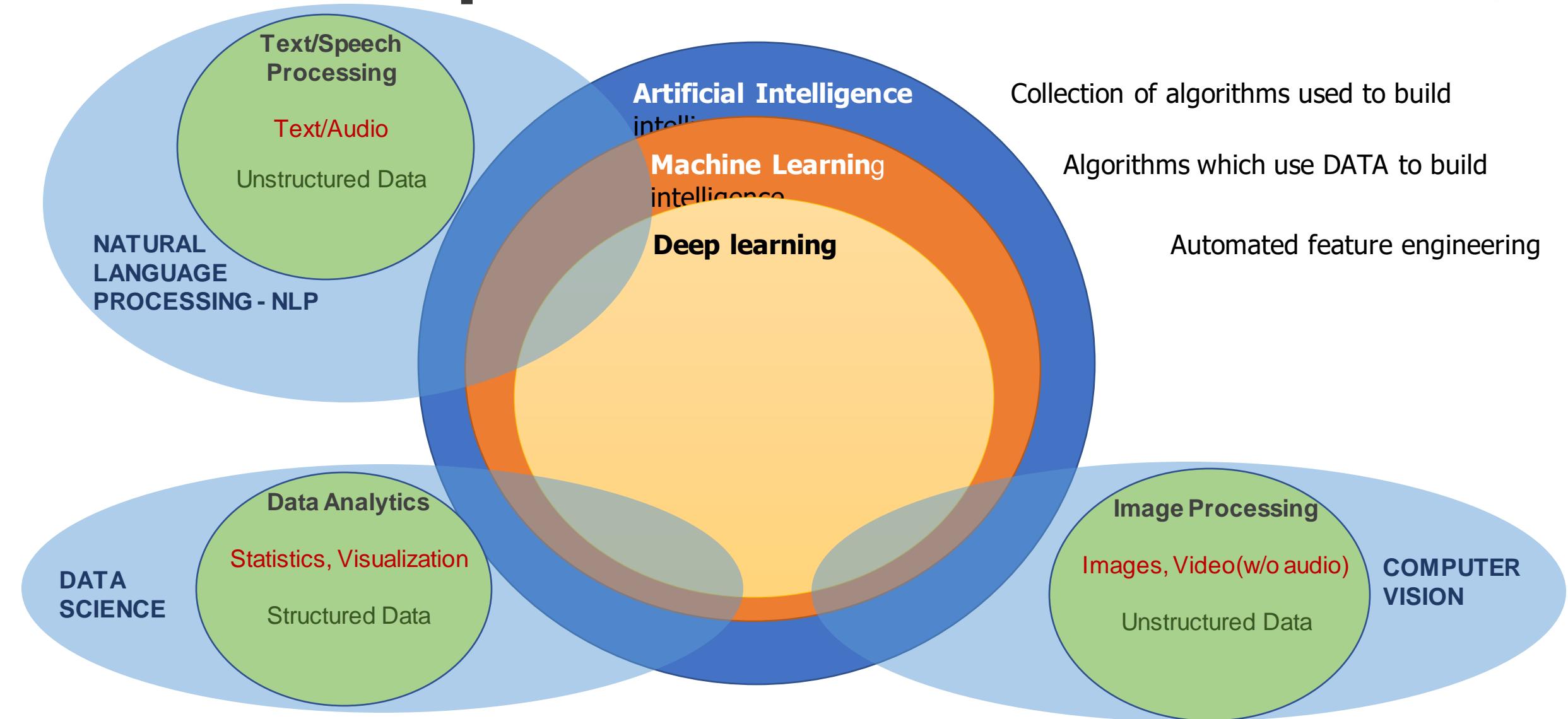
# AI Landscape



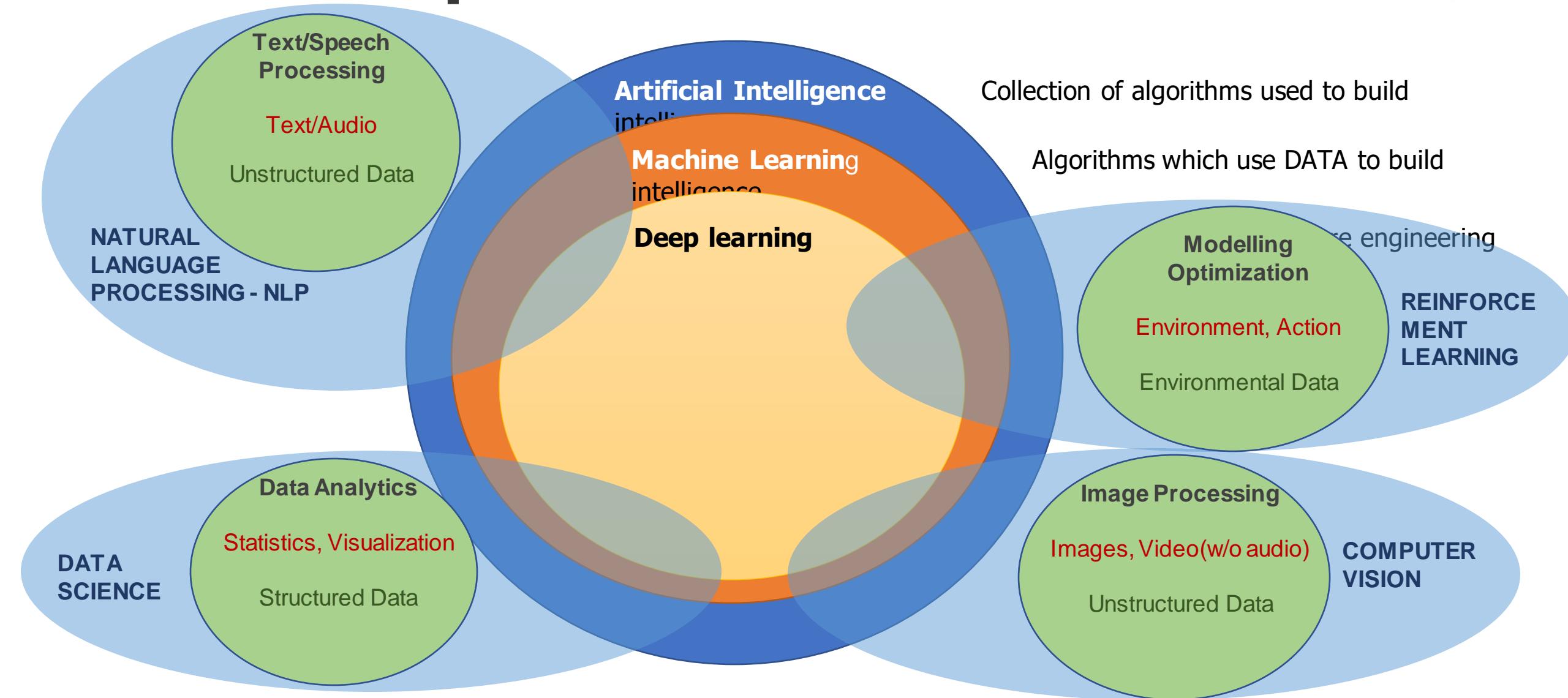
# AI Landscape

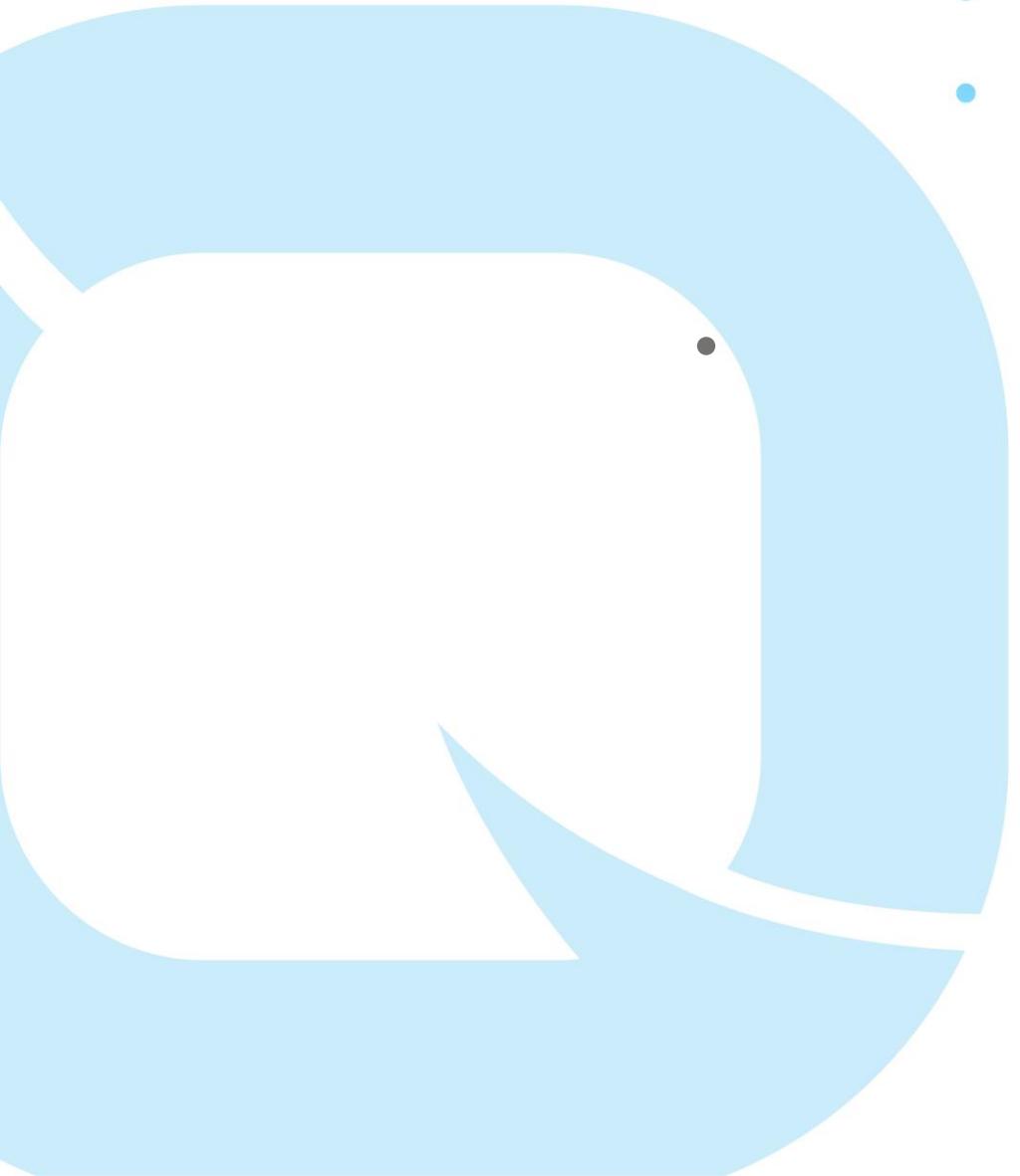


# AI Landscape



# AI Landscape





• • • •  
• • • •  
• • • •  
• • • •

Data Science

Computer Vision

Natural Language  
Processing

# Artificial Intelligence

Data Science

- Predicting Stock prices, housing prices or any other item prices based on historical data
- Predicting whether customer will buy a product or not, customer will churn or not
- Classifying the customers in different known groups
- Risk predictions for financial transactions.
- Fraud Detection from transactional data
- Segmentation of customers, stocks and server logs
- Predicting patient readmission into hospital
- Detecting anomalies in access management, data control
- Building product recommendation systems

Computer Vision

Natural Language  
Processing

# Artificial Intelligence

Data Science

Computer Vision

Natural Language  
Processing

- Face Recognition, Emotion Recognition
- Optical Character Recognition
- Document verification, authentication
- Object Detection and Classification from images
- Identifying forgery in the images
- Vehicle number plate, type recognition
- Self Driving Cars – lane detection, traffic sign classification, Behavioural Cloning
- Motion Estimation from videos
- Image restoration, colouring and pattern transfer
- Action Prediction

# Artificial Intelligence

Data Science

- Text/document classification
- Social Media Text mining and Analysis
- Speech to Text and Text to Speech conversion
- Caption generation
- Machine Translation

Computer Vision

- Sentiment analysis from text
- Chatbots
- Speaker recognition
- Personal Assistant, Sentence Correction
- Text Generation, Similarity Matching, Topic Modelling

Natural Language  
Processing

# Artificial Intelligence

## Reinforcement Learning

- Playing game (Okay this is a joke, but serious!)
- Self Driving Car
- Artificial Sun Environmental control
- Warehouse Robot control
- Education
- Social Media
- Marketing and Ads

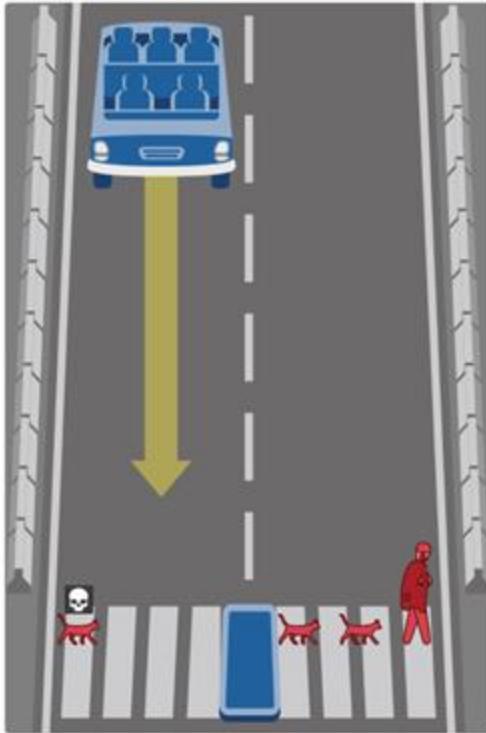
# Game AI



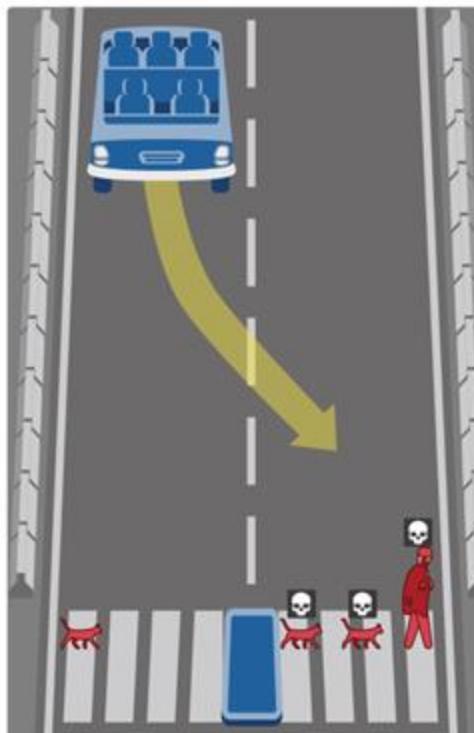
# Game AI: Moral Machine

What should the self-driving car do?

1 / 13



Show Description



Show Description



03

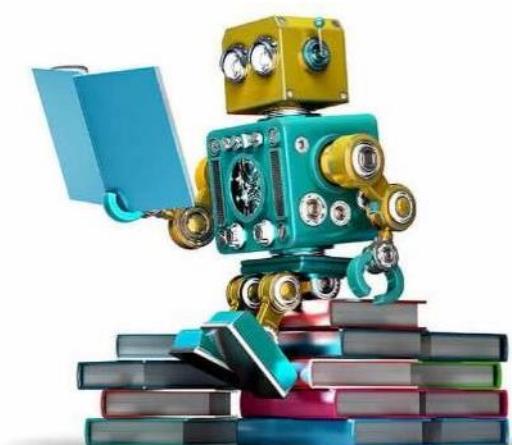
## MACHINE LEARNING

- What is Machine Learning?
- Machine Learning v/s traditional software

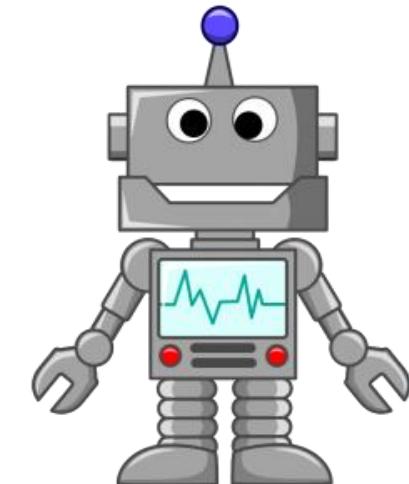
Humans learn from experiences



Machine Learning – making machines learn from experiences aka data.



Machines follow instructions



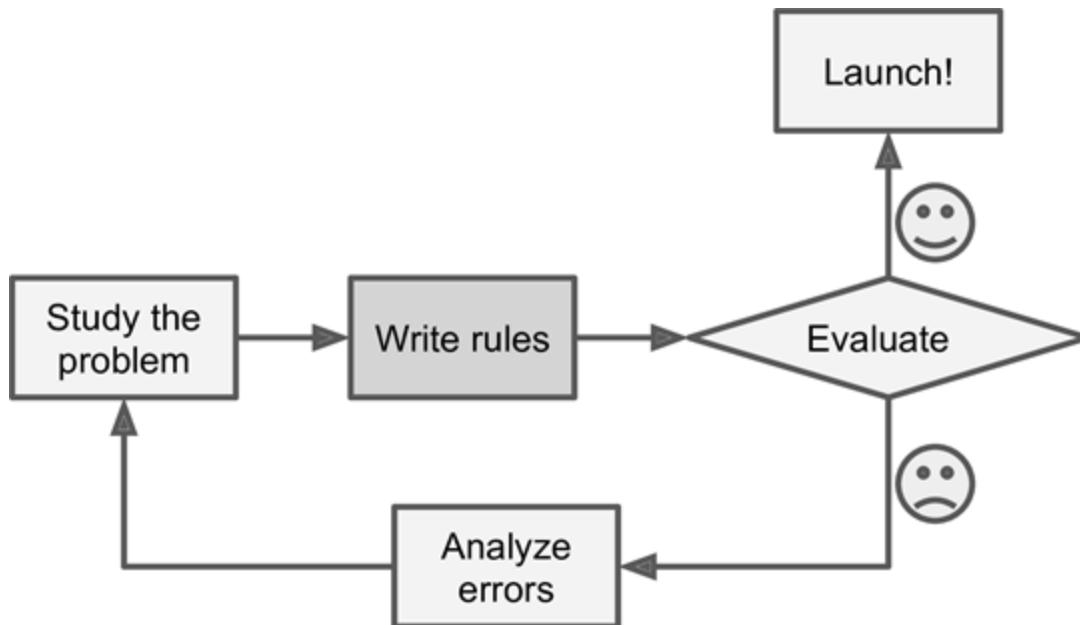
# What is Machine Learning?



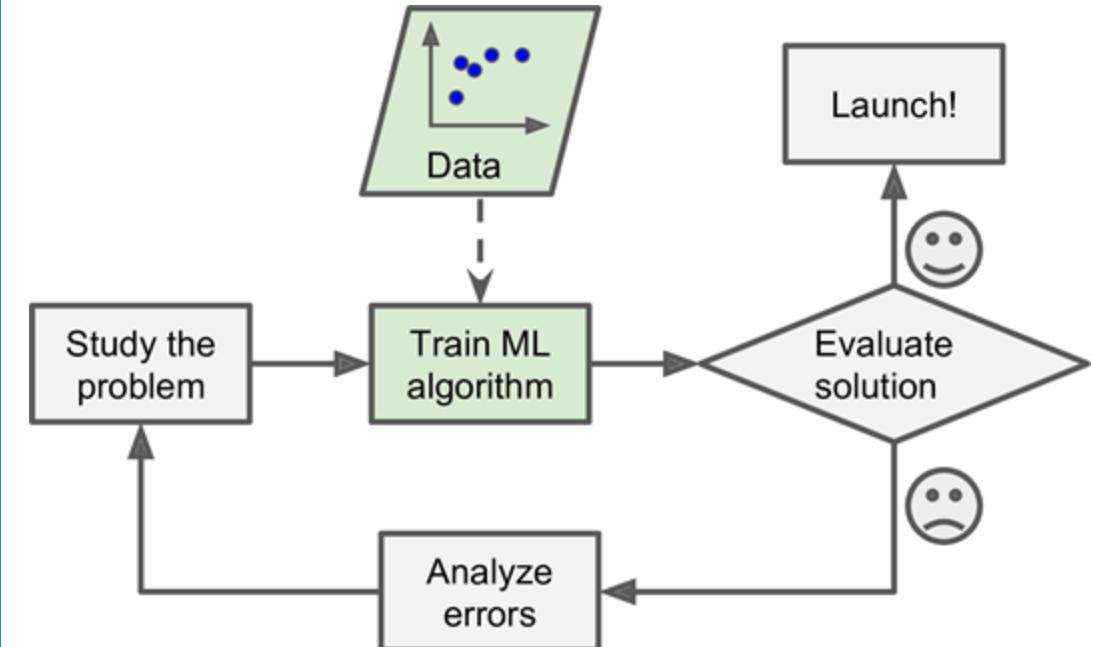
*Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.*

# Why Machine Learning?

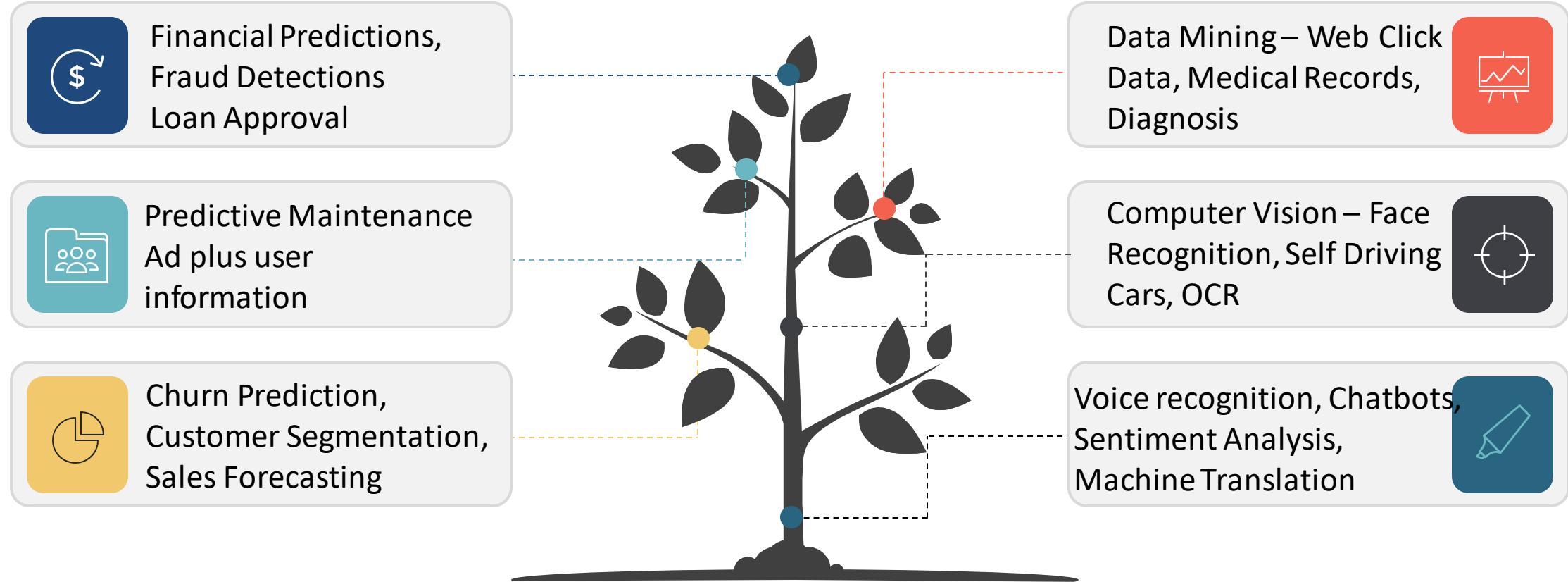
## TRADITIONAL APPROACH



## MACHINE LEARNING APPROACH

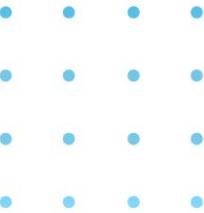


# Applications of Machine Learning



# Applications of Machine Learning

Manufacturing	Retail	Financial Services
<ul style="list-style-type: none"><li>❖ Predictive maintenance or condition monitoring</li><li>❖ Warranty reserve estimation</li><li>❖ Propensity to buy</li><li>❖ Demand forecasting</li><li>❖ Process optimization</li></ul>	<ul style="list-style-type: none"><li>❖ Predictive inventory planning</li><li>❖ Recommendation engines</li><li>❖ Upsell and cross-channel marketing</li><li>❖ Market segmentation and targeting</li><li>❖ Customer ROI and lifetime value</li></ul>	<ul style="list-style-type: none"><li>❖ Risk Analytics and Regulations</li><li>❖ Customer Segmentation</li><li>❖ Cross-selling and up-selling</li><li>❖ Sales and marketing campaign management</li><li>❖ Credit worthiness evaluation</li></ul>
Travel and Hospitality	Health Care and Life Sciences	Energy, Feedstock and Utility
<ul style="list-style-type: none"><li>❖ Aircraft scheduling</li><li>❖ Dynamic pricing</li><li>❖ Social media — consumer feedback and interaction analysis</li><li>❖ Customer complaint resolution</li><li>❖ Traffic patterns &amp; congestion management</li></ul>	<ul style="list-style-type: none"><li>❖ Alerts and diagnostics from real-time patient data</li><li>❖ Disease identification and risk stratification</li><li>❖ Patient triage optimization</li><li>❖ Proactive health management</li></ul>	<ul style="list-style-type: none"><li>❖ Power usage analytics</li><li>❖ Seismic data processing</li><li>❖ Carbon emissions and trading</li><li>❖ Customer-specific pricing</li><li>❖ Smart grid management</li><li>❖ Energy demand and supply optimization</li></ul>



04

## ML TOOLS

- Tools for Machine Learning
- What to learn in Machine Learning?



# Programming Languages

Python

R

# Machine Learning Cloud Platforms -



## **Microsoft Azure ML Studio**

<https://azure.microsoft.com/en-us/services/machine-learning-studio/>

## **Amazon Machine Learning**

<https://aws.amazon.com/ml/>

## **SAP Leonardo Machine Learning**

<https://www.sap.com/india/products/leonardo/machine-learning.html>

## **Google ML Platform**

<https://cloud.google.com/products/machine-learning/>

## **IBM Machine Learning**

<https://www.ibm.com/analytics/data-science/machine-learning>

# What to learn in Machine Learning?

**Programming and Tools**  
Python/R, spark etc.

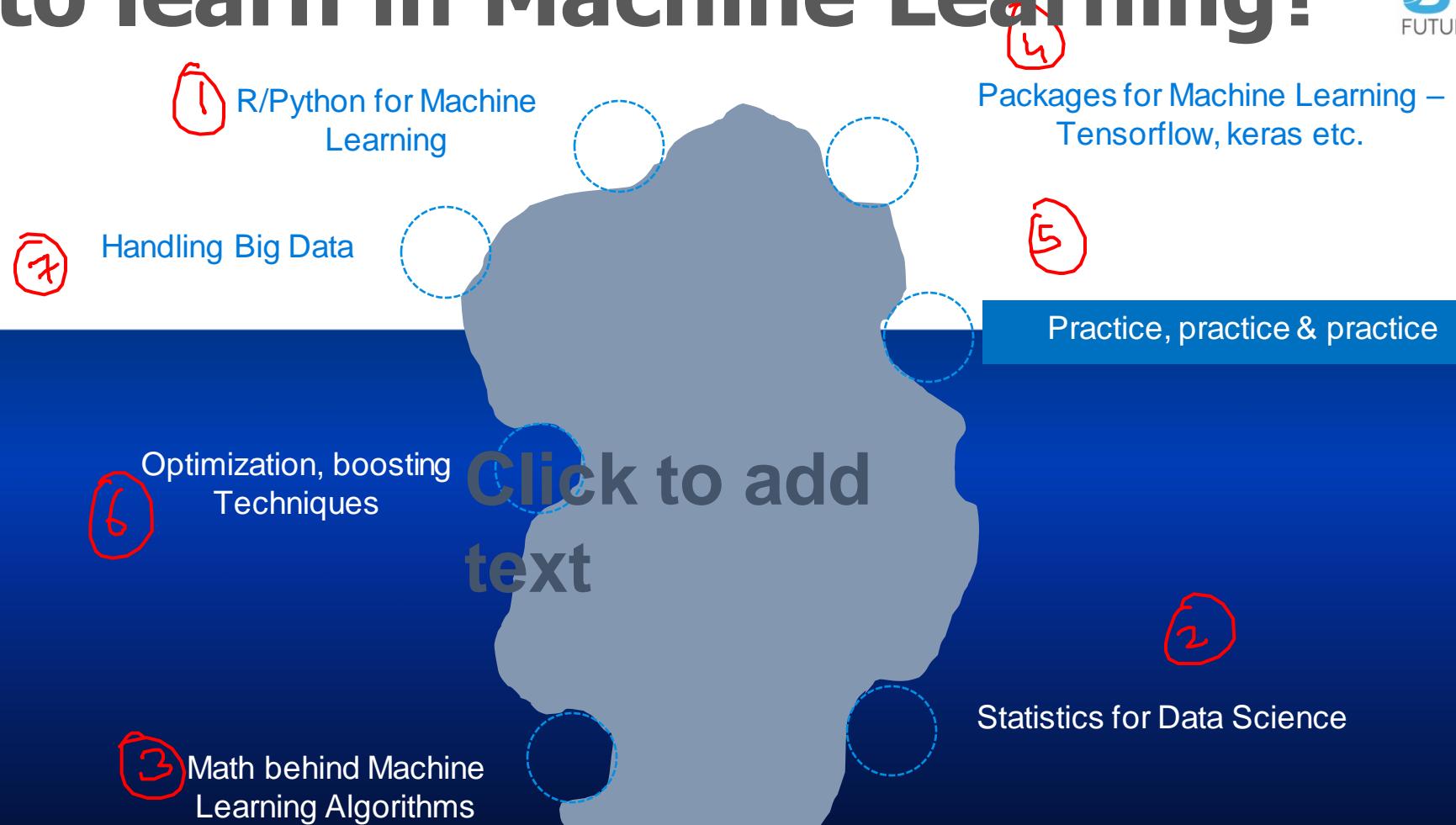
**30%**

**The Math behind Machine Learning**

Probabilistic Theory,  
Statistics and Linear  
Algebra

**70%**

# What to learn in Machine Learning?





05

## CONCLUSION

- Quiz
- Summary



# Quiz

## Question

Which of the following is not an application of ML?

- A. Descriptive analysis of sales and profit
- B. Predicting future sales
- C. Estimating customer's willingness to buy a product based on past data
- D. Developing emotion recognition solution using image data



# Quiz

## Question

Which of the following is not an application of ML?

- A. Descriptive analysis of sales and profit
- B. Predicting future sales
- C. Estimating customer's willingness to buy a product based on past data
- D. Developing emotion recognition solution using image data



Answer: A



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# AI Mastery Course



Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

## Modul 2

Introduction to everything

### Bagian

Data analytics with python- Basic  
statistics





# Learning Objectives

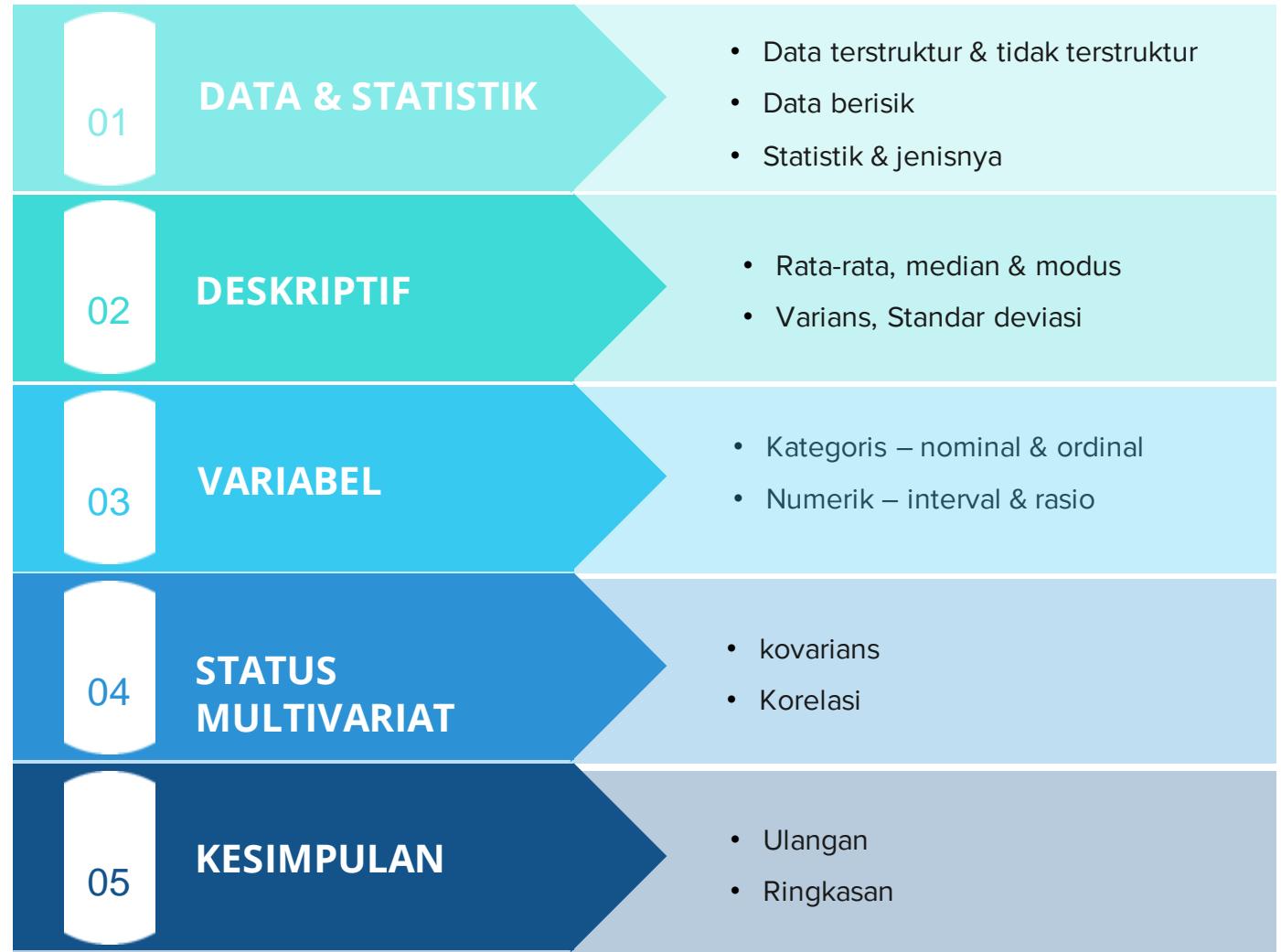
Di akhir modul ini, Anda akan dapat:

- Sebutkan macam-macam data.
- Menjelaskan konsep dasar statistika beserta klasifikasinya: statistik deskriptif dan inferensial.
- Memahami ukuran skala: skala nominal, ordinal, interval, dan rasio.
- Jelaskan ukuran tendensi sentral: mean, median & modus.
- Hitung ukuran variabilitas: jangkauan, kuartil dan jangkauan antarkuartil, varians dan standar deviasi.
- Jelaskan ukuran-ukuran bentuk: kemiringan dan kurtosis.
- Memahami pengenalan dan definisi penting dari Teori Probabilitas.
- Jelaskan kemungkinan bersyarat.
- Memahami konsep Kovarians dan Korelasi.





# Agenda





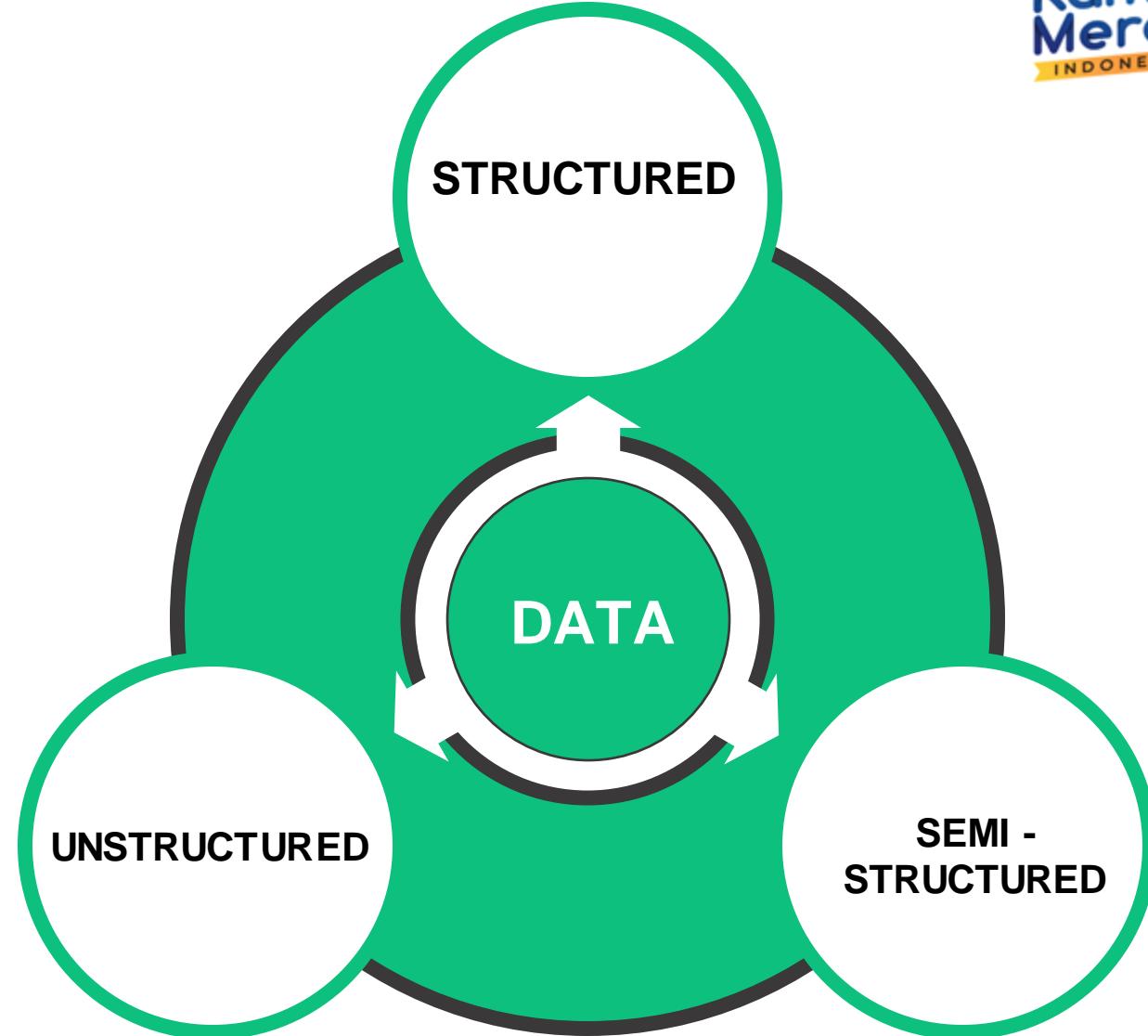
01

## DATA & STATISTIK

- Data terstruktur & tidak terstruktur
- Noisy Data
- Statistik & jenisnya



# Jenis Data

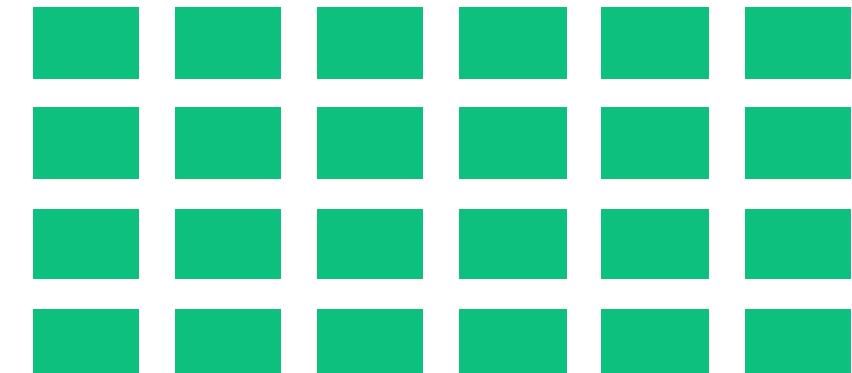


# Data Terstruktur

Data Terstruktur:

- Sangat terorganisir dan mudah dicari dengan kueri atau algoritme.
- Dapat dengan cepat dikonsolidasikan menjadi fakta.
- Mengikuti skema yang telah ditentukan sebelumnya.
- Biasanya berada di bidang tetap.
- Contoh tipikal adalah Relational Database Management System (RDBMS).
- Skema ditentukan sebelum konten dibuat dan data diisi.

## DATA TERSTRUKTUR



Apa yang Anda temukan di DB  
(khas)

# Organisasi Data Terstruktur

Diagram dan tabel berikut menggambarkan data terstruktur.



Artificial Intelligence  
Mastery Program

ORANG		
Indo		
Nama		
Usia		

PUBLIKASI		
Indo		
Nama		

Model Skema  
(diagram ER, DDL)

Tabel: Orang		
Indo	Nama	Usia
1	Rolf	28
2	Steffi	22

1	1
1	2
2	2

Tabel: Publikasi	
Indo	Nama
1	Penandaan Sosial
2	Klasifikasi

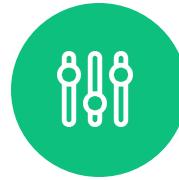
Tabel data

Contoh tabel dalam sistem database relasional

# Contoh Data Terstruktur

Beberapa contoh data terstruktur:

Tanggal



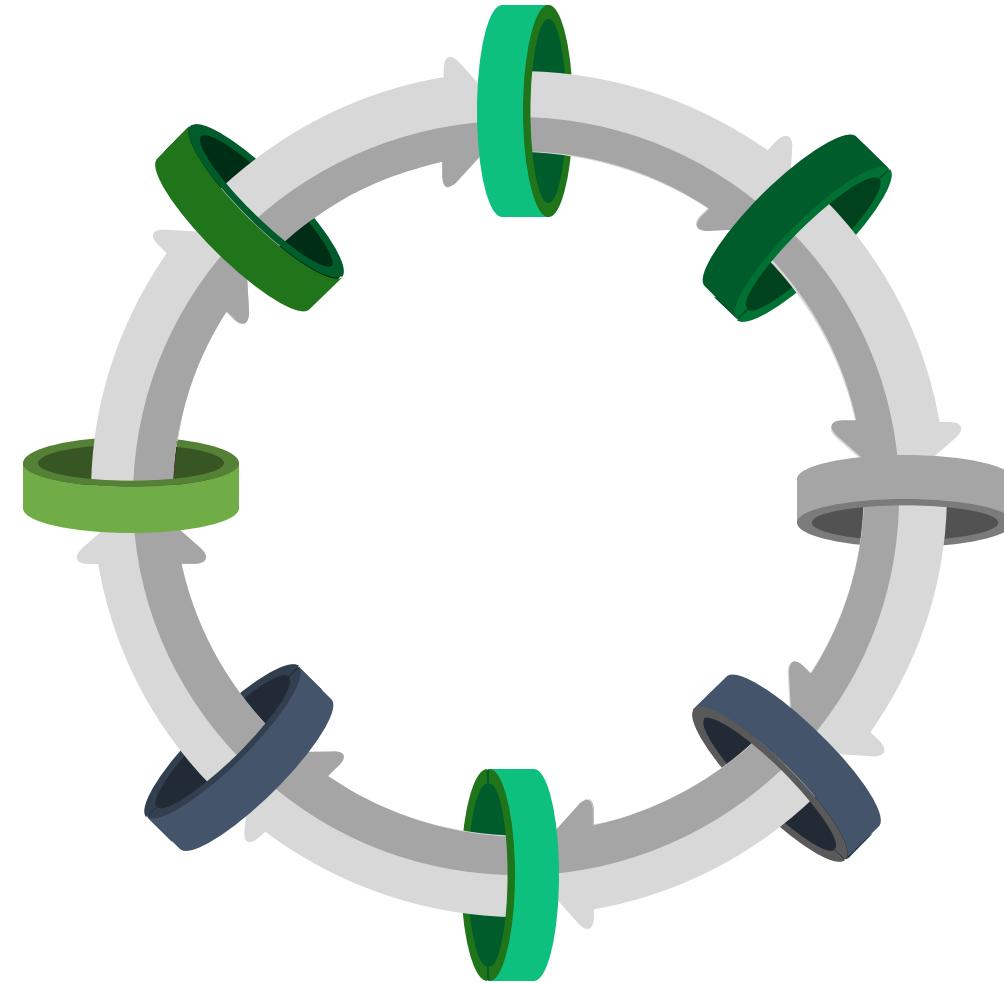
Nomor telepon



Keamanan sosial angka



Nomor Kartu Kredit



# Aplikasi Data Terstruktur

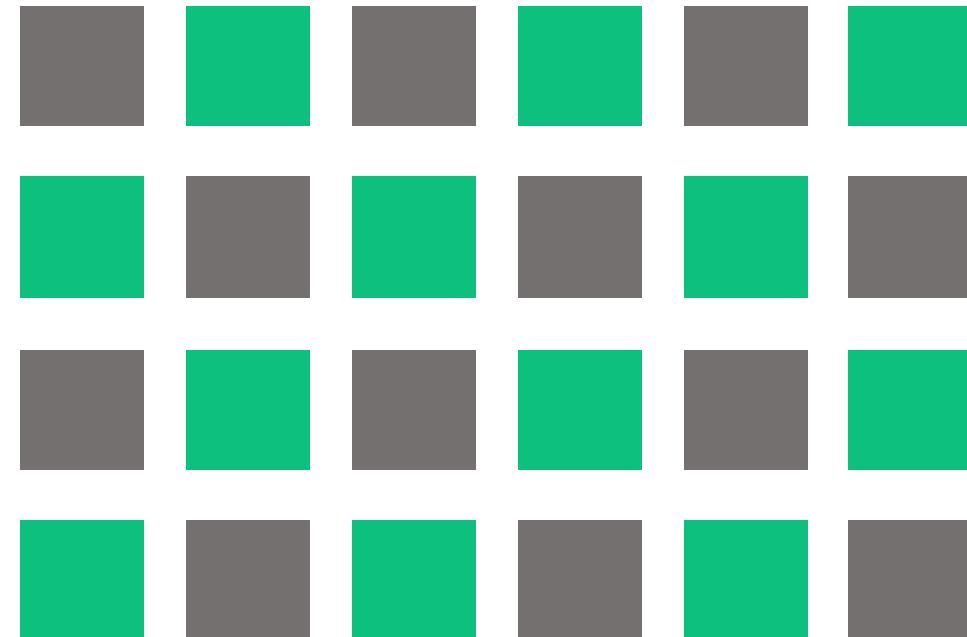
Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For Future Jobs

Artificial Intelligence  
Mastery Program

- Aplikasi database relasional dengan Data Terstruktur:
- Sistem reservasi maskapai penerbangan
- Kontrol inventaris
- Transaksi penjualan
- aktivitas ATM

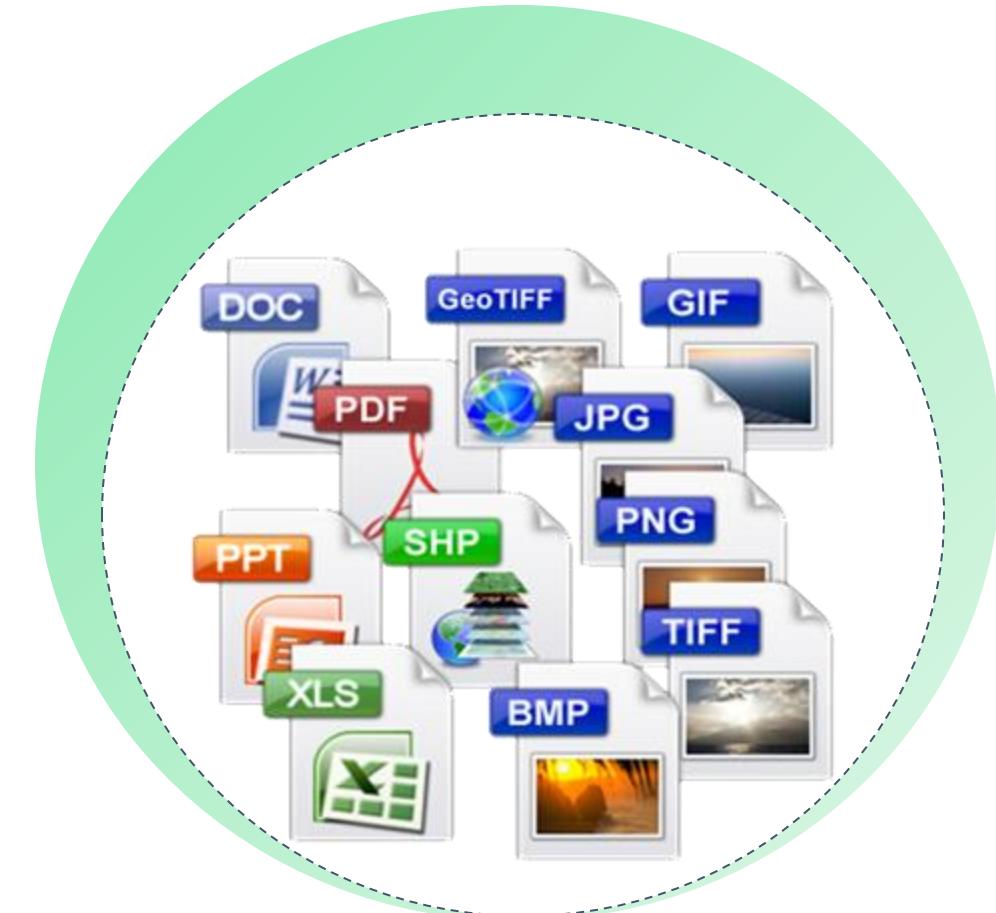
**DATA TERSTRUKTUR**



# Apa itu Data Tidak Terstruktur?

Data Tidak Terstruktur:

- Tidak ada struktur yang dapat diidentifikasi untuk jenis data ini.
- Memiliki struktur internal, tetapi tidak memiliki skema yang telah ditentukan sebelumnya.
- Tidak dapat disimpan dalam baris dan kolom seperti database relasional.
- Tidak ada model data tetap, tidak terorganisir
- Memerlukan lebih banyak ruang penyimpanan daripada data terstruktur.



# Contoh Data Tidak Terstruktur

Data Tidak Terstruktur yang dibuat oleh manusia mencakup hal-hal berikut:

## DIBUAT MANUSIA

- Email
- File teks
- Media sosial
- Situs web
- Media
- Data seluler
- Komunikasi
- Aplikasi bisnis

Data Tidak Terstruktur yang dihasilkan mesin mencakup hal-hal berikut:

## DIHASILKAN MESIN

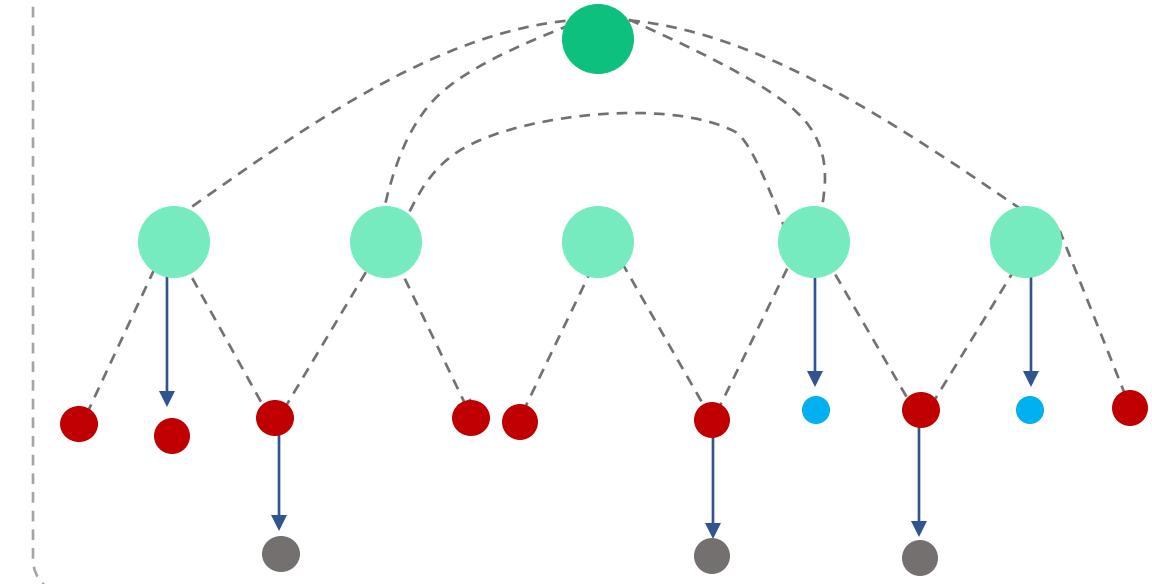
- Gambar satelit
- Data ilmiah
- Pengawasan digital
- data sensor
- Data dari termostat, mesin penjual otomatis & robot

# Apa itu Data Semi Terstruktur?

Data semi terstruktur:

- Sering dijelaskan sebagai tanpa skema atau menggambarkan diri sendiri.
- Berisi tag semantik, tetapi tidak sesuai dengan standar atau struktur database relasional biasa.
- Tidak ada deskripsi terpisah tentang jenis atau struktur data.
- Tidak memerlukan definisi skema, definisi tersebut bukan tidak mungkin, tetapi bersifat opsional.
- Data dapat memiliki atribut yang berbeda, dan atribut baru dapat ditambahkan kapan saja.

## DATA SEMI-STRUKTUR



# Contoh Data Semi Terstruktur

Contoh Data Semi-terstruktur meliputi:

- Bahasa markup XML - seperangkat aturan pengkodean dokumen yang menentukan format yang dapat dibaca manusia dan mesin.
- Open Standard JSON - format pertukaran data ringan, teks biasa, berdasarkan subset dari bahasa pemrograman JavaScript.
- NoSQL - beberapa database noSQL berisi data semi-terstruktur.

# Aplikasi Data Semi-terstruktur

Aplikasi yang berisi data semi terstruktur antara lain:

- Infrastruktur Data Besar.
- Aplikasi web
- LinkedIn
- Tenaga penjualan
- Rekomendasi pembaca di Amazon

# Noisy Data



## Noisy Data:

- Berisi sejumlah informasi yang tidak terlalu dibutuhkan, menyimpang, atau tidak diketahui kebenarannya
- Sebuah anomali dan dianggap sebagai kesalahan. Menurut Zhu et al. (2004), noise dalam data berdampak negatif pada hasil data mining.
- Beberapa sumber Noisy Data:
  - a) Random Noise, misalnya suara kebisingan yang terekam saat meeting online
  - b) Human Error, misalnya kesalahan input data
  - c) Outlier, suatu data yang sangat berbeda dari data umumnya

# Pengantar Statistika

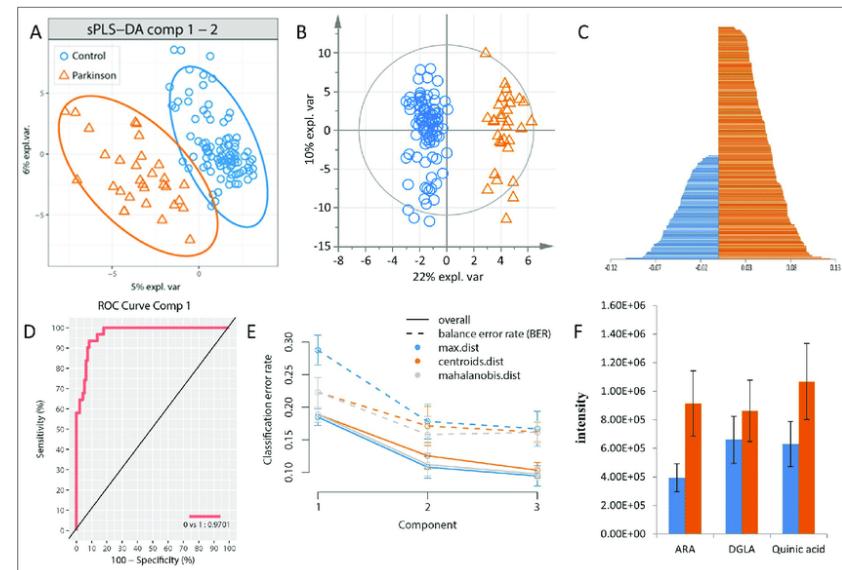
Statistika:

- **Ilmu** yang berkaitan dengan pengumpulan, penataan, penyajian, analisis, dan interpretasi data menjadi informasi untuk membantu pengambilan keputusan yang efektif.
- Disiplin ilmu yang menggunakan sampel data untuk mendukung klaim tentang populasi.

# Pengantar Statistika

Statistik:

- **Pengukuran** dari data, seperti mean, median, modus, dll.
- **Penyajian** data statistik dapat berbentuk tabel, grafik, diagram, deretan angka dan visualisasi angka



# Beberapa istilah penting



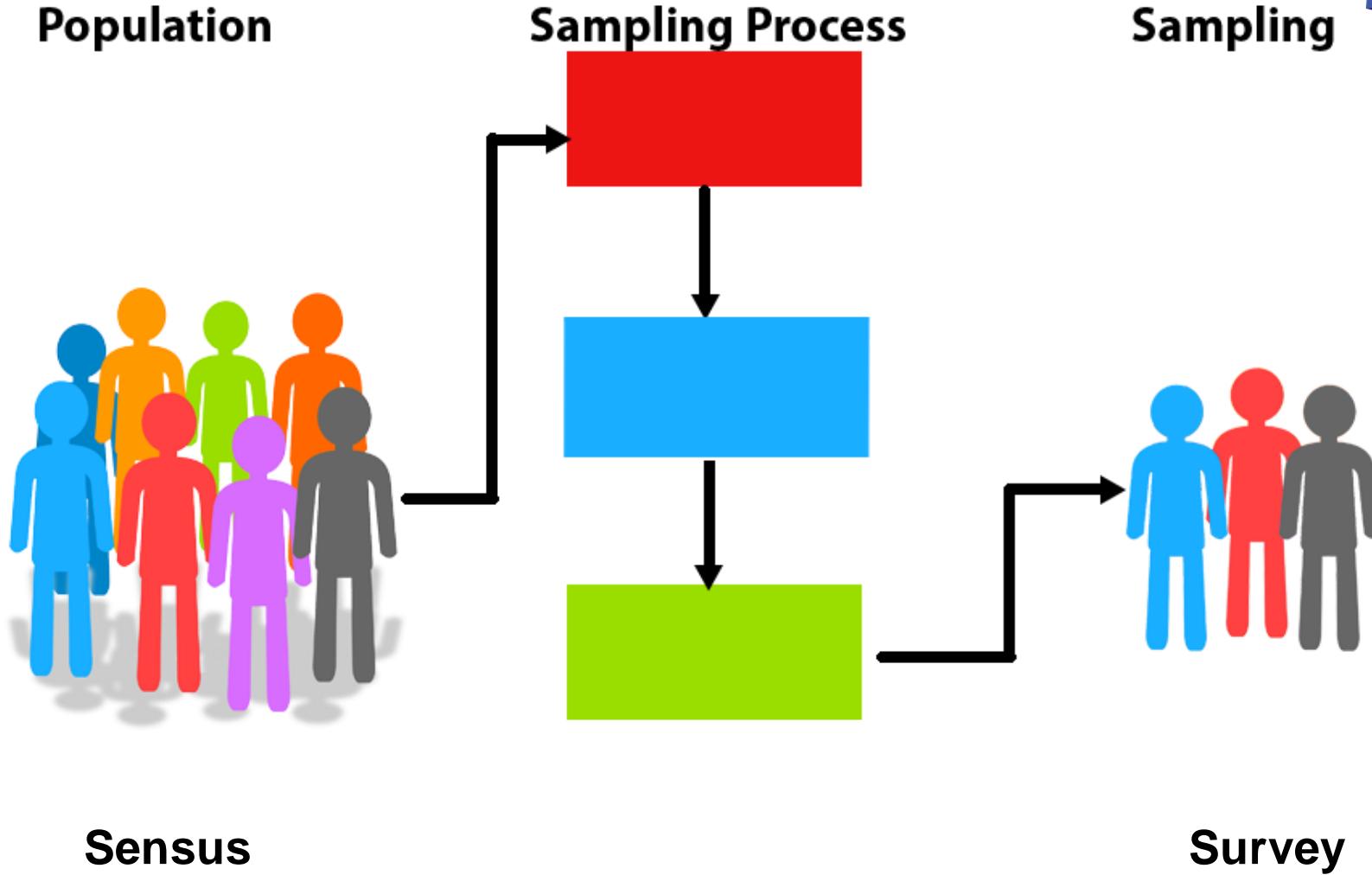
- Populasi: Himpunan objek yang akan dianalisa.  
Misalnya : Semua Mobil, Semua karyawan Microsoft
- Sensus: Ketika peneliti mengumpulkan data dari seluruh populasi.  
Misalnya Sensus Penduduk AS diambil setiap 10 tahun
- Sampel : Bagian dari populasi yang dianggap sudah cukup mewakili populasi.  
Misalnya : Dalam eksperimen mobil diambil sampel 1 mobil dari tiap brand
- Survey : Ketika peneliti mengumpulkan data dari Sampel saja

# Beberapa istilah penting

Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program



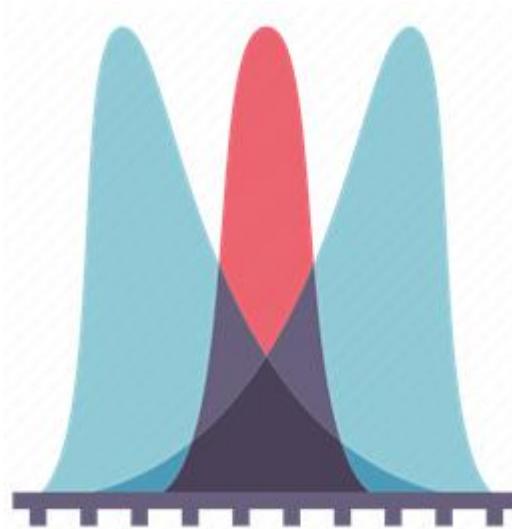
# Pembagian Statistik

## Deskriptif



**Mendeskripsikan** dan  
menganalisis **sampel** data  
tanpa melakukan proses  
penarikan kesimpulan

## Inferensial



**Menganalisa sampel**  
untuk mengestimasi dan  
membuat **kesimpulan**  
**mengenai populasi**

# Statistik deskriptif

- Bertujuan untuk mendapatkan gambaran karakteristik dari sampel.
- Meringkas dan melihat representasi dari data menggunakan grafik, bagan, dan tabel.
- Beberapa metodenya:
  - Ukuran Pemusatan
  - Ukuran penyebaran
  - Visualisasi Data
  - Eksplorasi hubungan data

# Statistik Inferensial

- Bertujuan untuk menguji suatu hipotesis dan menarik kesimpulan tentang suatu populasi berdasarkan sampel.
- Hasil analisis digeneralisasikan dari sampel ke populasi yang lebih besar.
- Yang berhubungan dengan metode ini:
  - Uji Hipotesis
  - P-value
  - Selang kepercayaan

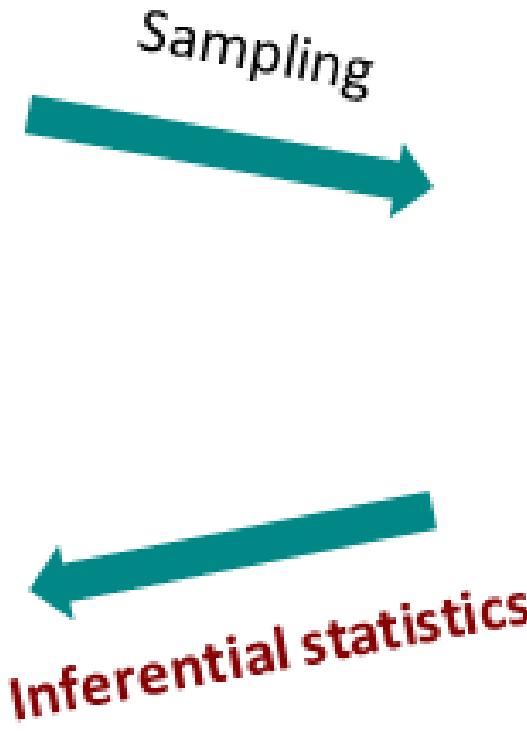
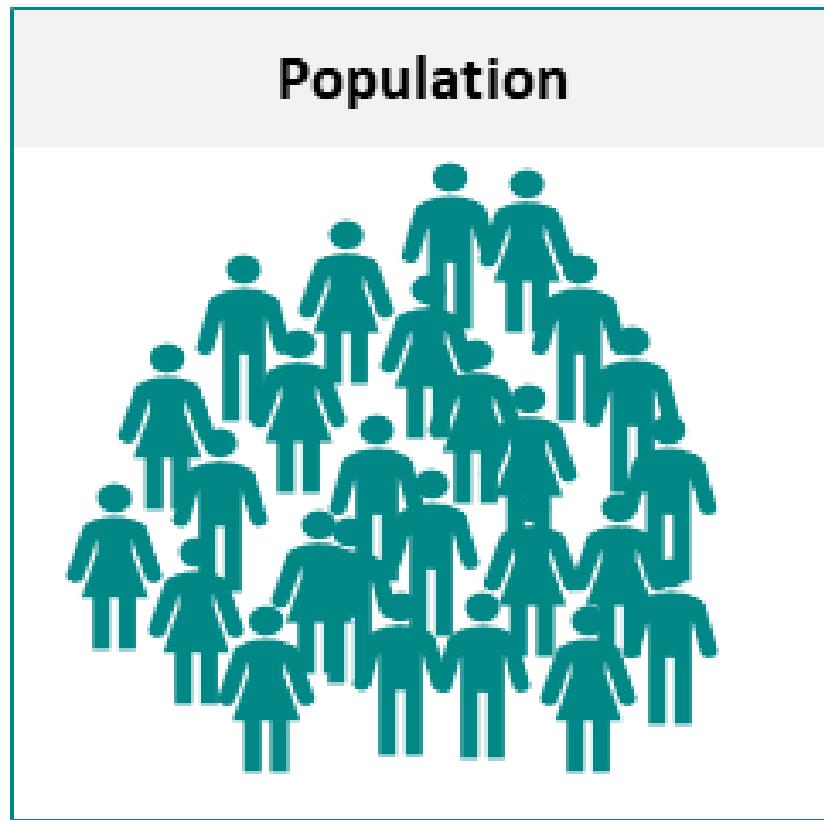
# Inferential Statistics Vs Descriptive Statistics

Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY

Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program



**Descriptive statistics**



02

## STATISTIK DESKRIPTIF

- Rata-rata, median & modus
- Varians, Standar deviasi
- Kemiringan & Kurtosis



## Ukuran Pemusatan

- Ukuran pemusatan adalah metode yang umum digunakan untuk mengidentifikasi pusat kumpulan data.
- Digunakan untuk menggambarkan kumpulan data dengan mengidentifikasi posisi sentral dalam kumpulan data tersebut.

Ukuran pemusatan:

- Mean
- median
- Mode

# Mean

Mean :

- Juga disebut sebagai nilai rata-rata dalam kumpulan data.
- Didefinisikan sebagai jumlah semua nilai dalam kumpulan data, dibagi dengan jumlah total nilai dalam kumpulan data.
- Mudah terpengaruh *outlier*
- Contoh: Misalkan  $x_1, x_2, x_3, \dots, x_n$  menjadi nilai dalam kumpulan data, dan ada 'n' jumlah nilai dalam kumpulan data:

Mean =

$$\frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad \text{or} \quad \frac{\sum x}{n}$$

# Median



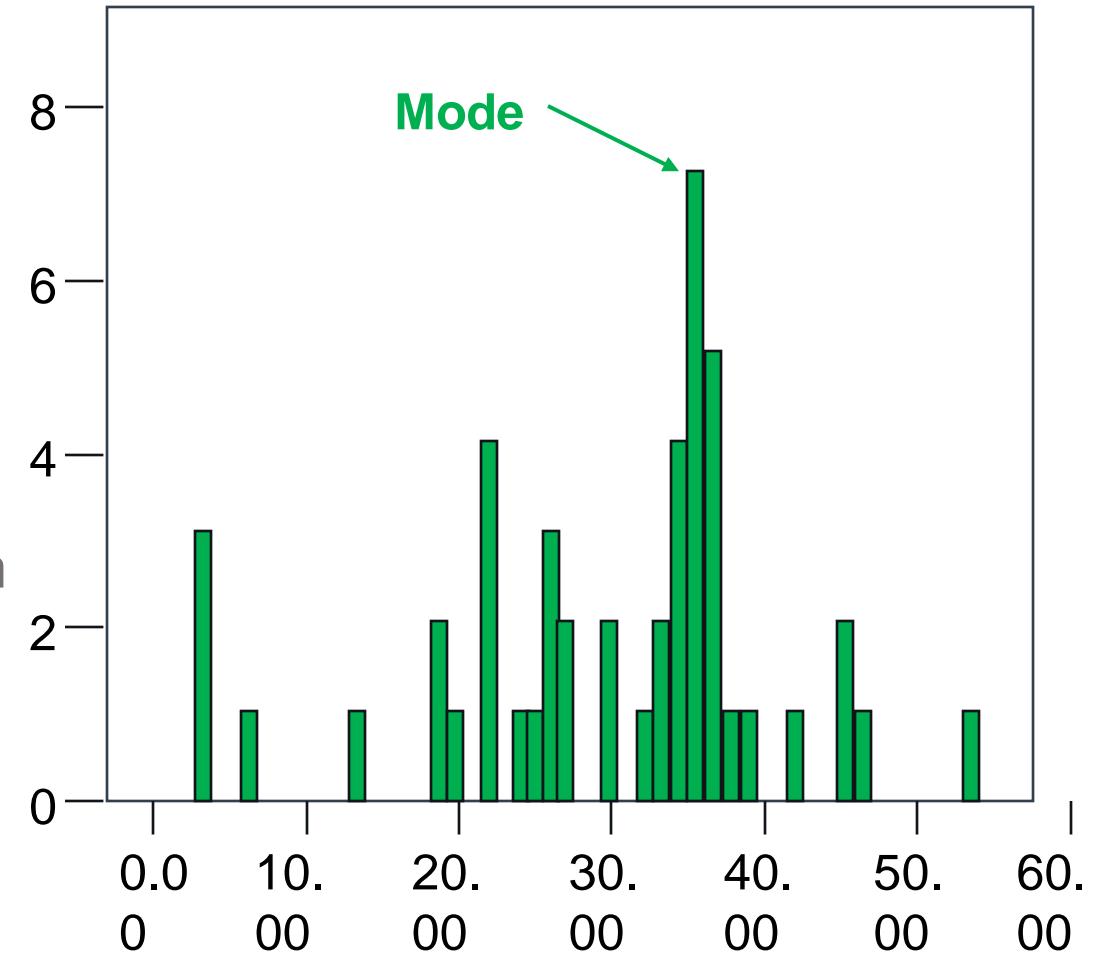
## median:

- Nilai tengah dalam suatu kumpulan data, setelah disusun menurut urutan besarnya.
- Nilai diurutkan terlebih dahulu dari yang terkecil sampai yang terbesar.
- Median adalah nilai tengah dari kumpulan data jika jumlah nilai adalah bilangan ganjil; rata-rata dari dua nilai tengah, jika jumlah nilai adalah bilangan genap.
- Median = Nilai ke- $[(n+1)/2]$ , jika ada bilangan ganjil.
- Median = rata-rata dari nilai ke- $[n/2]$  dan  $[(n/2)+1]$ , jika ada bilangan genap.

# Mode

Mode:

- Nilai yang paling sering muncul dalam kumpulan data, yaitu nilai sampel acak yang terjadi dengan frekuensi terbesar.
- Digunakan secara kualitatif.
- Berlaku untuk semua tingkat pengukuran data, yaitu skala nominal, ordinal, interval dan rasio.
- Bimodal - jika nilai paling sering muncul, dua mode.
- Multimodal - Kumpulan data dengan lebih dari dua mode.



# Latihan



Misalkan kita punya 2 data berikut:

Data A : 1, 1, 2, 2, 3, 3, 4, 4, 5, 50

Data B : 1, 1, 2, 2, 3, 3, 4, 4, 5, 5

Coba tentukan mean dan median kedua data!

# Latihan

Misalkan kita punya 2 data berikut:

Data A : 1, 1, 2, 2, 3, 3, 4, 4, 5, 50

Data B : 1, 1, 2, 2, 3, 3, 4, 4, 5, 5



**Jawaban :**

Data A dan B sama-sama memiliki median 3, tetapi mean data A 7.5, sementara mean data B 3.

Terlihat **mean dari A kurang mampu merepresentasikan data A** yang didominasi angka 1-5, ini karena **A memiliki outlier** dan **mean mudah terpengaruh oleh outlier**. Sementara median tidak terpengaruh sehingga mampu memberi kita gambaran (representasi) dari data keseluruhannya.

Biasanya, jika **mean >> median**, maka **outliernya nilai ekstrim besar** (seperti contoh diatas), sementara jika **mean << median**, maka **outliernya nilai ekstrim kecil**

# Ukuran Penyebaran

- Digunakan untuk mengetahui bagaimana data tersebar di sekitar mean
- Digunakan dalam data kuantitatif, karena variabel dapat disusun dalam urutan yang logis, dengan nilai rendah dan tinggi.

Langkah-langkah penyebaran:

- Range
- Kuartil dan Inter-quartile range
- Varians
- Standar deviasi

# Range

- Ukuran penyebaran yang paling sederhana.
- Jarak antara nilai terkecil dan nilai terbesar dalam sebuah dataset.
- Dipengaruhi oleh outlier, karena varians mungkin terlalu rendah atau terlalu tinggi karena outlier.

# Kuartil dan Inter-quartile range

- Bagilah kumpulan data yang diurutkan menjadi empat bagian yang sama, dan lihat nilai titik di antara kuarter:
- Kuartil bawah (Q1) adalah titik antara 25% nilai terendah dan 75% nilai tertinggi. Ini juga disebut persentil ke-25.
- Kuartil kedua (Q2) adalah bagian tengah dari kumpulan data. Ini juga disebut persentil ke-50, atau median.
- Kuartil atas (Q3) adalah titik antara nilai 75% terendah dan 25% tertinggi. Ini juga disebut persentil ke-75.
- Rentang interkuartil (IQR) adalah perbedaan antara kuartil atas (Q3) dan bawah (Q1) dan menggambarkan 50% nilai tengah ketika diurutkan dari terendah ke tertinggi.

# Varians

Perbedaan:

- Varians adalah ukuran seberapa menyebar set data.
- Dihitung sebagai deviasi kuadrat rata-rata setiap angka dari rata-rata kumpulan data.
- Rumus untuk menghitung varians ( $S^2$ ) dari kumpulan data adalah:

$$S^2 = \frac{\sum (x - \bar{x})^2}{n}$$

- (di mana x adalah nilai individual,  $\bar{x}$  adalah rata-rata dari kumpulan sampel, n adalah jumlah nilai dalam distribusi.)

# Standar Deviasi (SD)



SD:

- Jarak rata-rata semua nilai dari rata-rata keseluruhan.
- Akar kuadrat dari varians.
- SD(s) dari kumpulan data sampel dapat dihitung menggunakan rumus:

$$S = \sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

- (di mana  $x$  adalah nilai individua,  $\bar{x}$  adalah rata-rata dari kumpulan sampel,  $n$  adalah jumlah nilai dalam distribusi.)

# Latihan



Misalkan ada data nilai di 2 kelas :

C : 60, 70, 70, 70, 80, 80, 80, 80, 90, 100

D : 75, 75, 75, 75, 80, 80, 80, 80, 80, 80

Coba hitung mean, range, dan varians, dan standar deviasinya!

# Latihan

C : 60, 70, 70, 70, 80, 80, 80, 80, 90, 100

D : 75, 75, 75, 75, 80, 80, 80, 80, 80, 80

Mean dari kedua kelas sama-sama 78. Namun Range C 40, dan Range D 5. Varians C 128.9 (std=11.4) dan Varians D 6.7 (std=2.6).

Meski rata-ratanya sama, tapi dari range dan varians kita mengetahui bahwa nilai di kelas C cenderung beragam sementara di D hampir seragam. Biasanya data paling banyak **tersebar di mean ± standar deviasi**. Sehingga, tanpa melihat datanya 1 per 1 kamu bisa memperkirakan nilai kelas C paling banyak tersebar di 66 hingga 89 sementara nilai kelas D tersebar di 75 hingga 80.



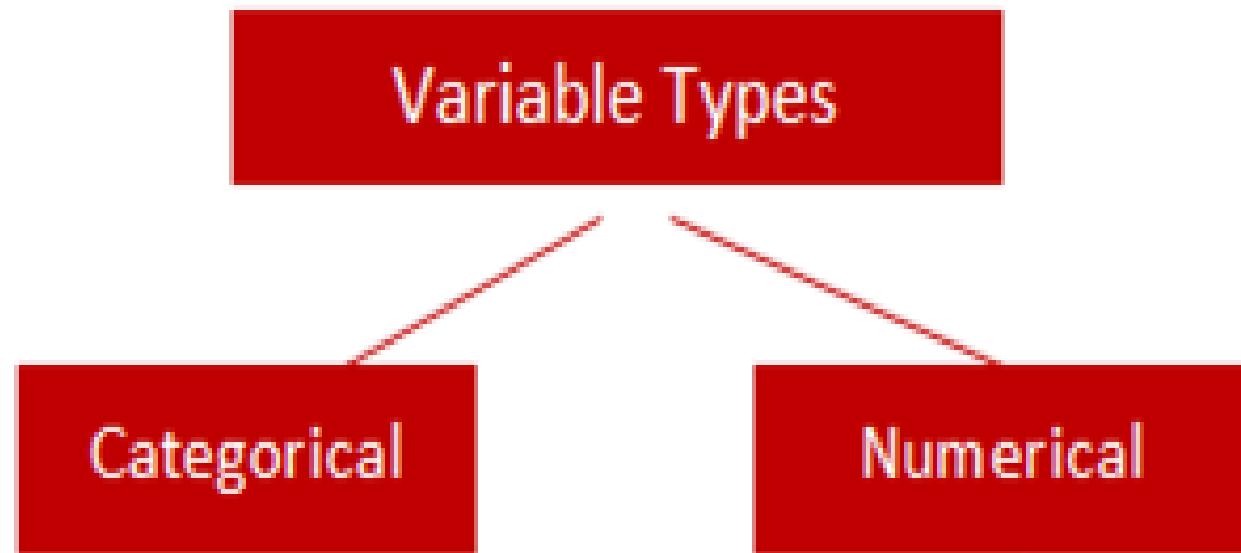
03

## Skala Pengukuran Data

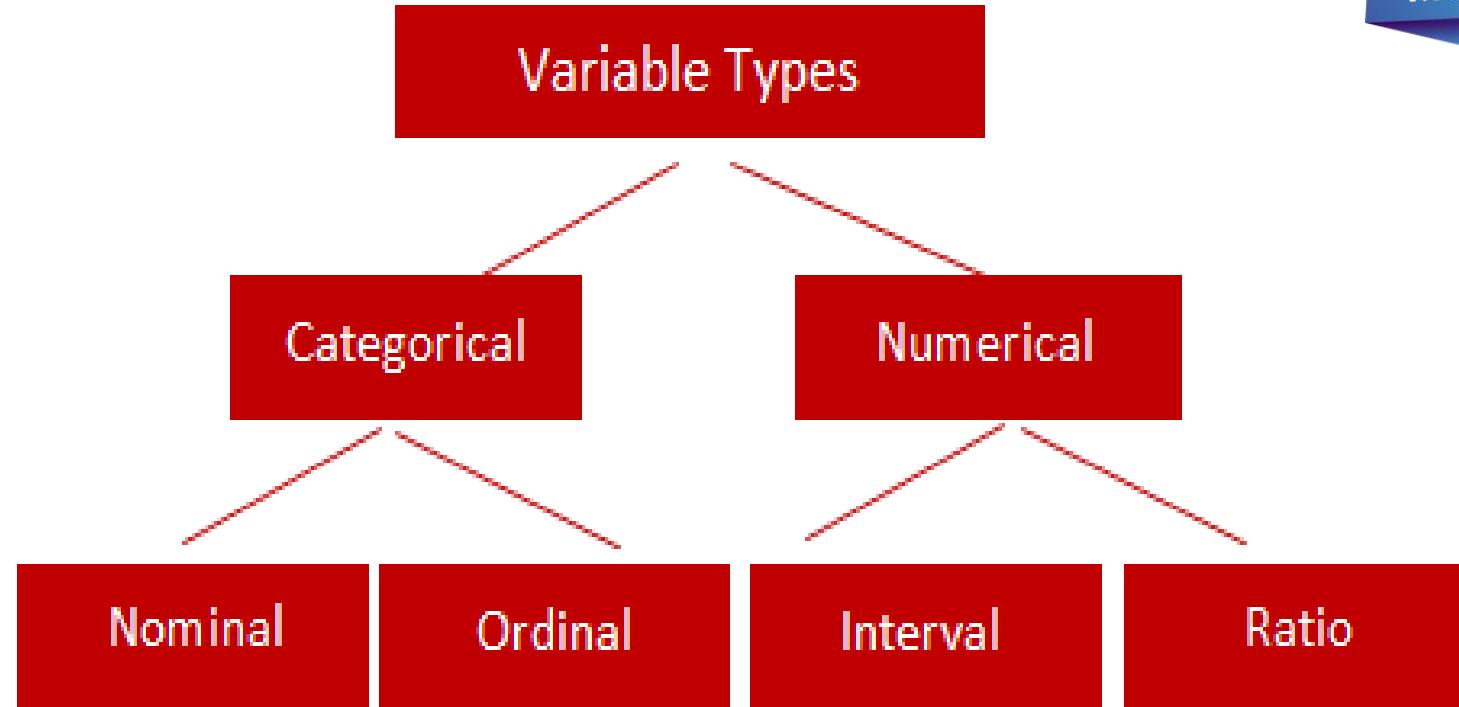
- Kategoris – nominal & ordinal
- Numerik – interval & rasio



# Jenis variabel / skala pengukuran



# Jenis variabel / skala pengukuran



Mengapa penting dipelajari? Karena perbedaan skala pengukuran juga menentukan pemilihan statistik yang tepat untuk analisa datanya.

# Skala Nominal



## Kategorik tanpa urutan

- Data berupa kategorik, namun tidak berurutan
- Misalnya Data jenis kelamin, misalnya kita mengkodekan 1 = laki-laki, 2 perempuan, karena  $2 > 1$  bukan berarti perempuan itu diatas laki-laki
- Contoh lainnya adalah data provinsi, data suku, dll
- Pada data ini, statistik yang cocok adalah frekuensi dan modus

# Skala Ordinal



## Kategorik berurutan

- Data bersifat kategorik, namun memiliki makna urutan
- Misalkan kita mengkodekan data tingkat dengan : 4 = PT, 3 = SMA sederajat, 2 = SMP sederajat, 1 = SD sederajat
- $4 > 3 > 2 > 1$  menunjukkan lulusan PT tingkat pendidikannya memang diatas lulusan SMA, lulusan SMA diatas lulusan SMP, begitu seterusnya
- Contoh lainnya adalah data jabatan di perusahaan, data skala kepuasan (Sangat Puas, Puas, Tidak Puas, Sangat Tidak Puas), dll
- Pada data ini, statistik yang cocok adalah frekuensi, modus, dan median

# Skala Interval



## Numerik, 0 tidak mutlak

- Datanya berupa angka
- 0 pada data interval tidak menunjukkan 0 yang sesungguhnya
- Contohnya data suhu,  $0^{\circ}\text{C}$  bukan berarti tidak ada suhunya, 0 disini bukan 0 sebenarnya
- Statistik yang cocok untuk analisa datanya adalah frekuensi, mean, median, modus, varians

# Skala Rasio



## Numerik, 0 mutlak

- Data berupa angka
- 0 disini bernilai sebenarnya (0 mutlak)
- Misalnya data kecepatan, saat kecepatannya 0 artinya benda tersebut tidak bergerak
- Statistik yang cocok untuk analisa datanya adalah frekuensi, mean, median, modus, varians

# Ringkasan

Skala	Penjelasan	Contoh	Statistik deskriptif	grafik
Nominal	data kategorik tanpa urutan	Laki-laki dan perempuan	frekuensi, modus	lebih utama pie chart, bisa bar chart, tidak disarankan line chart
Ordinal	data kategorik dengan urutan	tingkat pendidikan	frekuensi, modus, median	bisa pie chart, lebih utama bar chart, tidak disarankan line chart
Interval	data numerik tanpa 0 mutlak	suhu di beberapa wilayah		tidak disarankan pie chart, untuk bar chart harus diubah jadi interval tertentu dulu (atau biasanya disebut histogram), bisa juga line chart
Rasio	data numerik dengan 0 mutlak	Jumlah orang di suatu ruangan	frekuensi, mean, median, modus, varians	



04

## STATUS MULTIVARIAT

- kovarians
- Korelasi



# kovarians

## kovarians:

- Memberikan ukuran yang menunjukkan sejauh mana dua variabel acak berubah bersama-sama.
- Meski mengukur hubungan antara dua variabel dalam kumpulan data yang diberikan, namun nilai hubungan ini masih bergantung pada varians data, sehingga belum terstandarisasi (data bervarians tinggi kovariansnya bisa tinggi pula, begitu pun sebaliknya).
- Nilainya bisa dari  $-\infty$  hingga  $\infty$  (karena belum terstandarisasi)
- Misalkan  $E(x)$  adalah nilai ekspektasi dari suatu variabel  $x$ , dan  $E(y)$  adalah nilai ekspektasi dari variabel  $y$ , maka kovarians antara  $x$  dan  $y$  diberikan oleh:

$$\text{cov}(x, y) = E[xy] - E[x]E[y]$$

# Sifat Kovarians

Diberikan konstanta 'a' dan variabel acak X, Y, dan Z, sifat-sifat berikut adalah:

$$\text{Cov}(X, X) = \text{Var}(X) \geq 0$$

$$\text{Cov}(X, Y) = \text{Cov}(Y, X)$$

$$\text{Cov}(aX, Y) = a\text{Cov}(X, Y)$$

$$\text{Cov}(X + Y, Z) = \text{Cov}(X, Z) + \text{Cov}(Y, Z).$$

# Korelasi

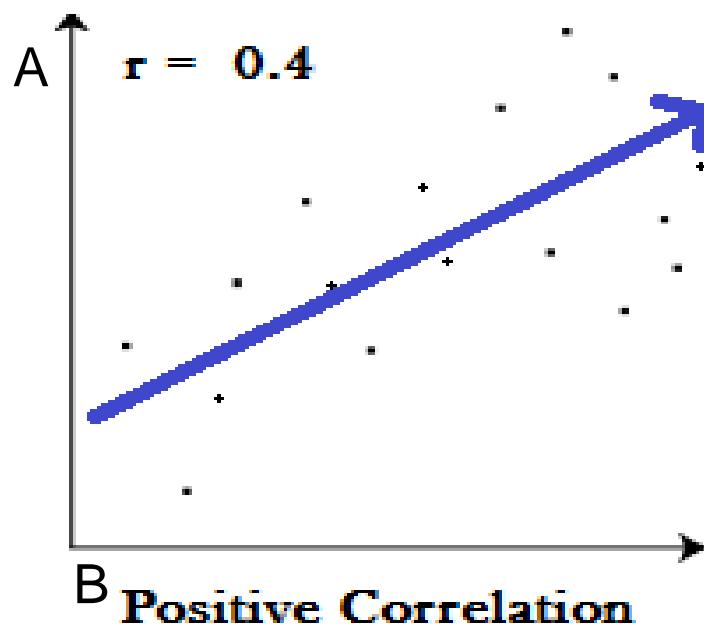
## Korelasi:

- Ukuran yang menunjukkan hubungan antar 2 variabel acak
- Nilainya terletak pada kisaran -1 dan +1. (sudah terstandarisasi sehingga lebih mudah diinterpretasikan daripada kovarians)
- Semakin tinggi nilai koefisien korelasinya (tanpa melihat – atau +) maka hubungan antar 2 variabel semakin erat
- - atau + menunjukkan bagaimana hubungan kedua variabel, misalnya :
  - Hubungan A dan B –0.9, ini menunjukkan hubungan A dan B sangat kuat, namun ketika nilai A semakin besar, nilai B akan semakin kecil
  - Hubungan C dan D 0.2, ini menunjukkan hubungan C dan D cenderung kecil, ketika nilai A semakin besar, nilai B cenderung makin besar pula
- Rumus :  $\text{cor}(x,y) = \text{cov}(x,y)/\sigma_x \sigma_y$

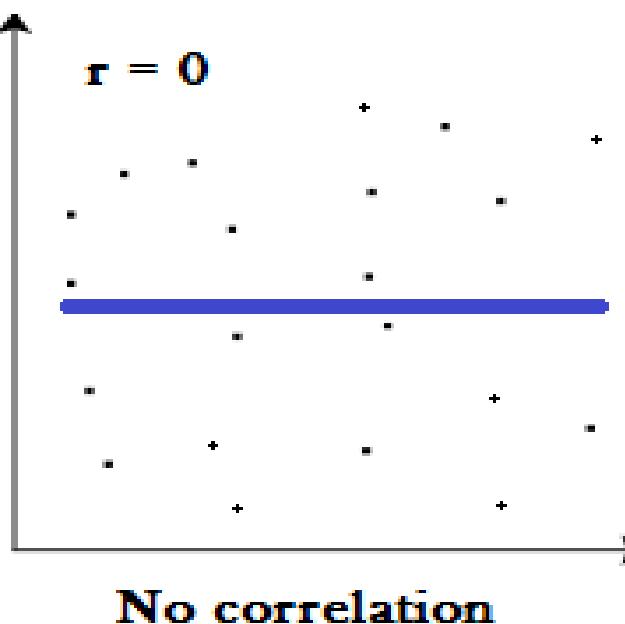
# Koefisien Korelasi

Koefisien korelasi adalah cara untuk memberi nilai pada hubungan.

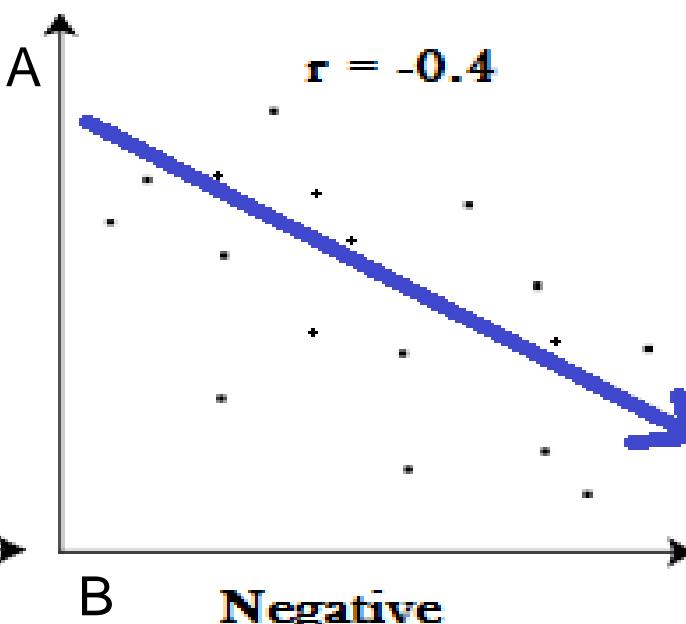
Grafik berikut menunjukkan korelasi -1, 0 dan 1.



Saat B naik, A juga naik



Tidak ada hubungan linier



Saat B naik, A malah turun

# Besar keeratan hubungan



- Menurut Guilford (1956) keeratan hubungan antar kedua variabel (positif/negatif hanya bentuk hubungannya seperti di slide sebelumnya):

ABSOLUTE VALUE OF R	INTERPRETATION
< 0.19	Slight; almost no relationship
0.20–0.39	Low correlation; definite but small relationship
0.40–0.69	Moderate correlation; substantial relationship
0.70–0.89	High correlation; strong relationship
0.90–1.00	Very high correlation; very dependable relationship
≥ 0.30	Practically significant relationship

# Korelasi ≠ Kausalitas



- Punya korelasi yang kuat bukan berarti berlaku hukum kausalitas.
- Jika 2 variabel saling berhubungan, misalnya A berhubungan erat dengan B, bukan berarti A mempengaruhi B.
  - Jika A dan B berkorelasi bisa saja :
    - A mempengaruhi B atau B mempengaruhi A (bisa keduanya, atau hanya salah 1)
    - A mempengaruhi C, lalu C mempengaruhi A, makanya keduanya terlihat berhubungan. Misalnya Cuaca panas mempengaruhi rasa haus, rasa haus mempengaruhi intensitas minum, jadi pengaruh cuaca ke intensitas minum secara tidak langsung.
    - C mempengaruhi A dan B, misalnya musim hujan mempengaruhi intensitas minum kopi dan masak mie (bukan karena habis minum kopi jadi pengen masak mie).
    - Coincidental : kebetulan kejadiannya bersamaan.



05

## KESIMPULAN

- Ulangan
- Ringkasan



# Ringkasan

- Tipe data - Data Terstruktur, Tidak Terstruktur, dan Semi Terstruktur.
- Data Terstruktur, organisasinya, contoh, kelebihan dan keterbatasannya.
- Data Tidak Terstruktur, contoh, kelebihan dan keterbatasan.
- Data semi terstruktur dan contohnya.
- Pengantar statistik beserta klasifikasinya: statistik deskriptif dan inferensial.
- Empat ukuran skala: skala nominal, ordinal, interval dan rasio.
- Ukuran Tendensi Sentral: mean, median dan modus.
- Ukuran variabilitas: jangkauan, kuartil dan jangkauan interkuartil, varians dan standar deviasi.
- Konsep Kovarians dan Korelasi.





# Ulangan

## Pertanyaan

Manakah dari ukuran berikut yang mudah terpengaruh oleh outlier?

- A. Mean
- B. median
- C. Modus
- D. Kecondongan



# Ulangan

## Pertanyaan

Manakah dari ukuran berikut yang mudah terpengaruh oleh outlier?

- A. Mean
- B. median
- C. Modus
- D. Kecondongan

Jawaban : A



# Ulangan

## Pertanyaan

Manakah dari ukuran berikut yang cocok untuk data nominal?

Click to add text

- A. Mean
- B. median
- C. Modus
- D. Kecondongan



# Ulangan

## Pertanyaan

Manakah dari ukuran berikut yang cocok untuk data nominal?

- A. Mean
- B. median
- C. Modus
- D. Kecondongan

Jawaban : C

# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)



Skills  
For  
Future  
Jobs

*Artificial Intelligence  
Mastery Program*

# Module

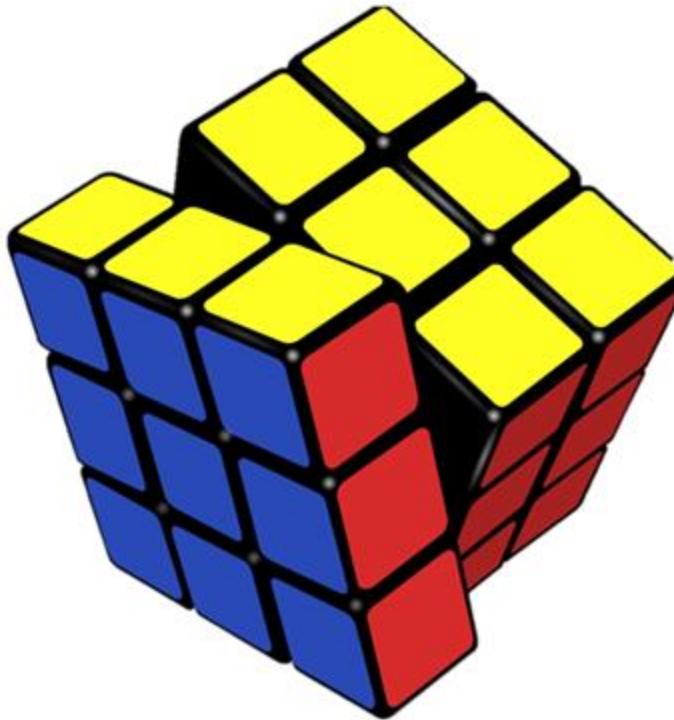
Artificial Intelligence Fundamental

## Section

AI Project Cycle



*“For Every Problem there is a solution”*





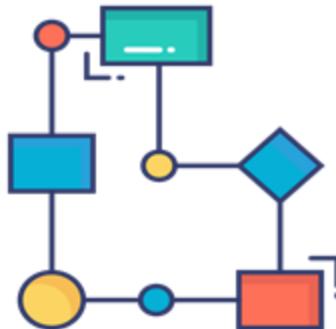
## Session I

**System Thinking, Algo,  
Flow & System Map**



## System Thinking

Merupakan cara dalam memandang sesuatu secara keseluruhan, dimana bagian-bagiannya saling berhubungan. Memandang secara keseluruhan tersebut berarti mempelajari untuk memahami setiap bagian yang terkait dalam suatu sistem.



## Algoritma

Metode efektif diekspresikan sebagai rangkaian terbatas dari instruksi-instruksi yang telah didefinisikan dengan baik untuk menghitung sebuah fungsi.

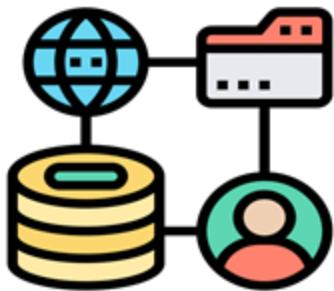


## Flow

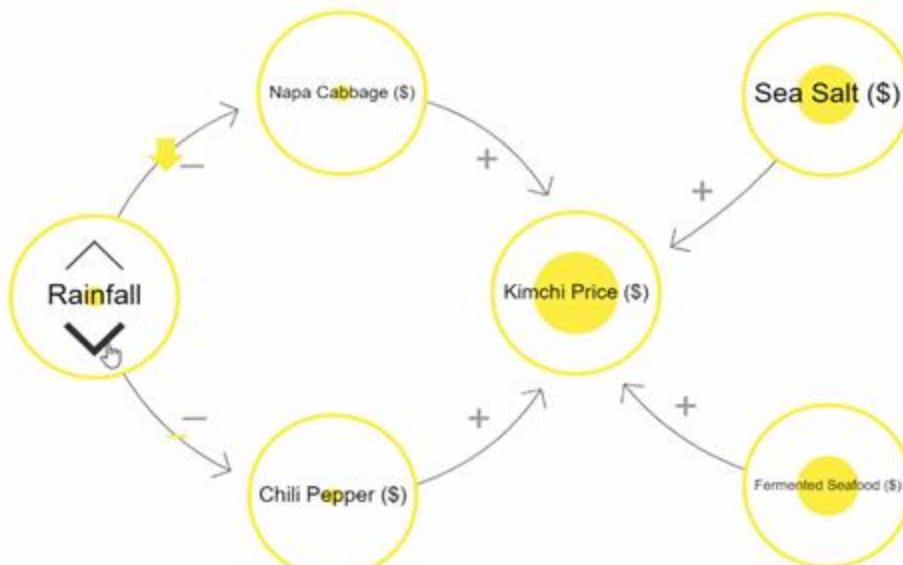
Gambaran arus informasi yang diproses dari input menuju sebuah output tertentu. fokus pada arus informasi, asal dan tujuan data, hingga bagaimana data tersebut disimpan.

Biasanya, digunakan untuk menjelaskan atau menganalisis sebuah sistem informasi.

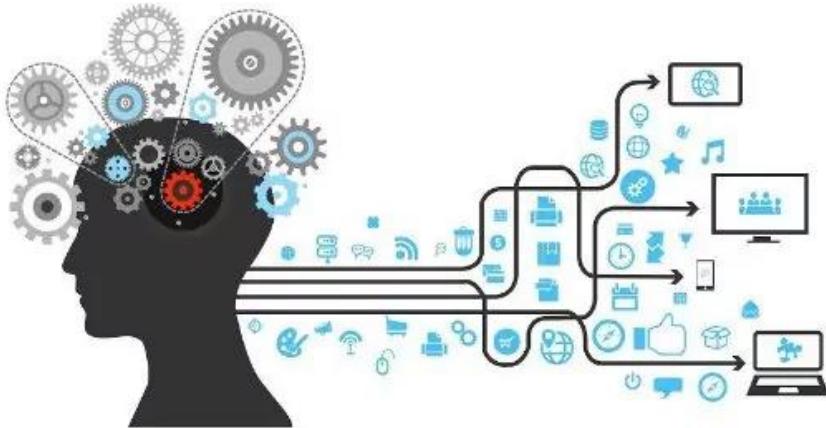
∴



## System Map

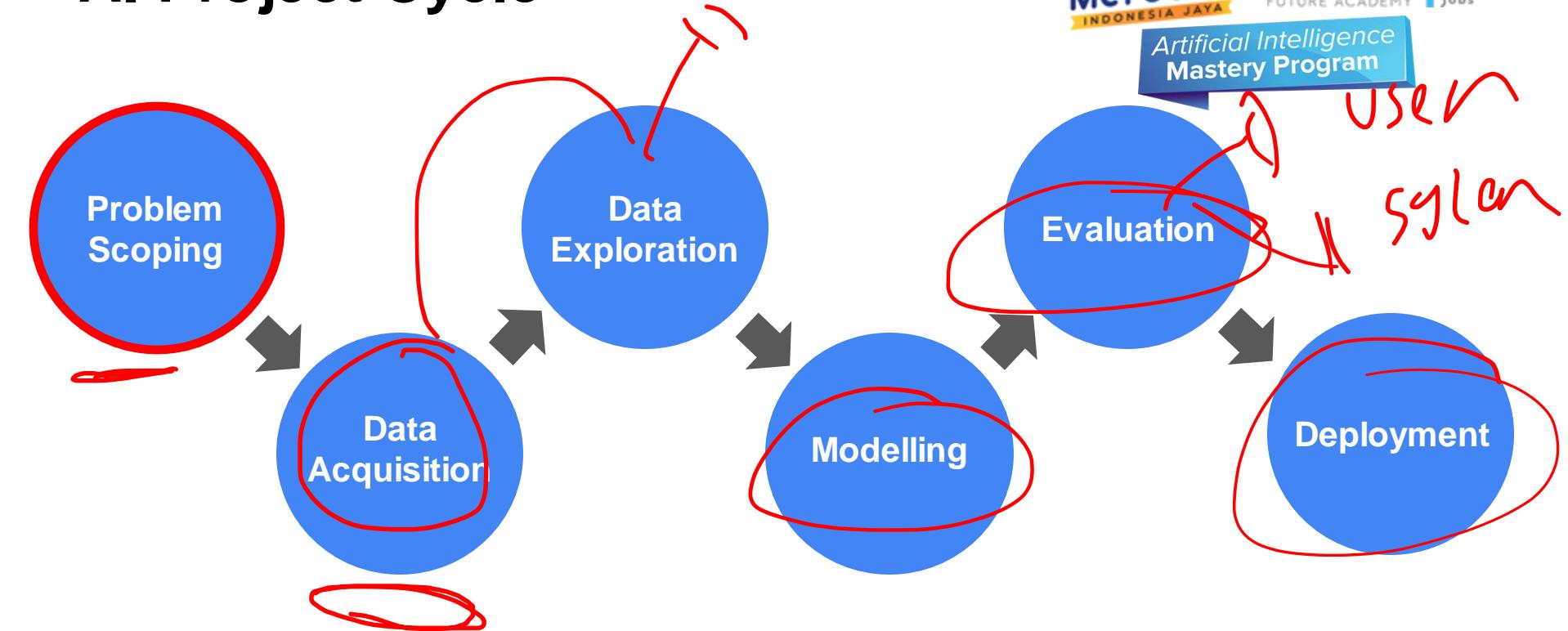


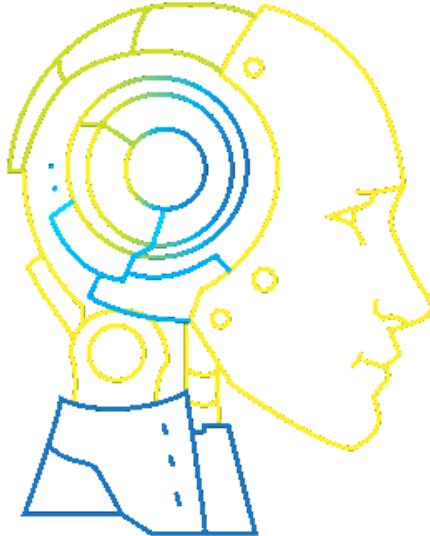
## Artificial Intelligence



## AI Project Cycle

# AI Project Cycle





# What is Problem Scoping?

# What is Problem?

Masalah didefinisikan sebagai sebuah pernyataan yang jelas menggambarkan keadaan awal dari masalah yang harus dipecahkan. Pernyataan tersebut menunjukkan properti masalah seperti tugas yang harus diselesaikan, kinerja sistem yang ada saat ini dan pengalaman dengan sistem saat ini.

# SUSTAINABLE DEVELOPMENT GOALS



P B B  
2016

## What is Problem Scoping?

Setiap kali kita memulai pekerjaan apapun, misalnya pada masalah tertentu selalu terkait dengan pekerjaan atau proses. Sebenarnya kita dikelilingi oleh masalah! Masalah ini bisa kecil atau besar, terkadang kita juga mengabaikannya, terkadang kita membutuhkan solusi mendesak, jika tidak, tentunya pekerjaan kita akan terganggu.

***"The problem scoping refers to the identification of a problem and the vision to solve it."***



# The 4Ws of Problem Scoping

**WHO ?**

**WHAT ?**

**WHERE ?**

**WHY ?**

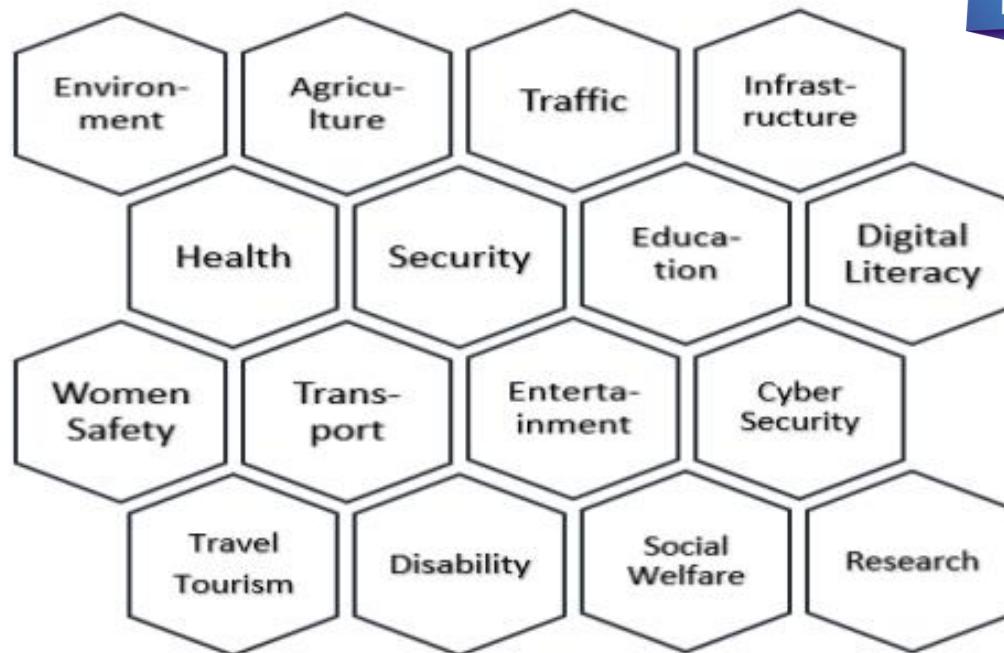
# The 4Ws of Problem Scoping

**Who?** – Mengacu pada siapa yang menghadapi masalah dan siapa pemangku kepentingan dari masalah tersebut.

**What?** – Mengacu pada apa masalahnya dan bagaimana Anda tahu tentang masalahnya.

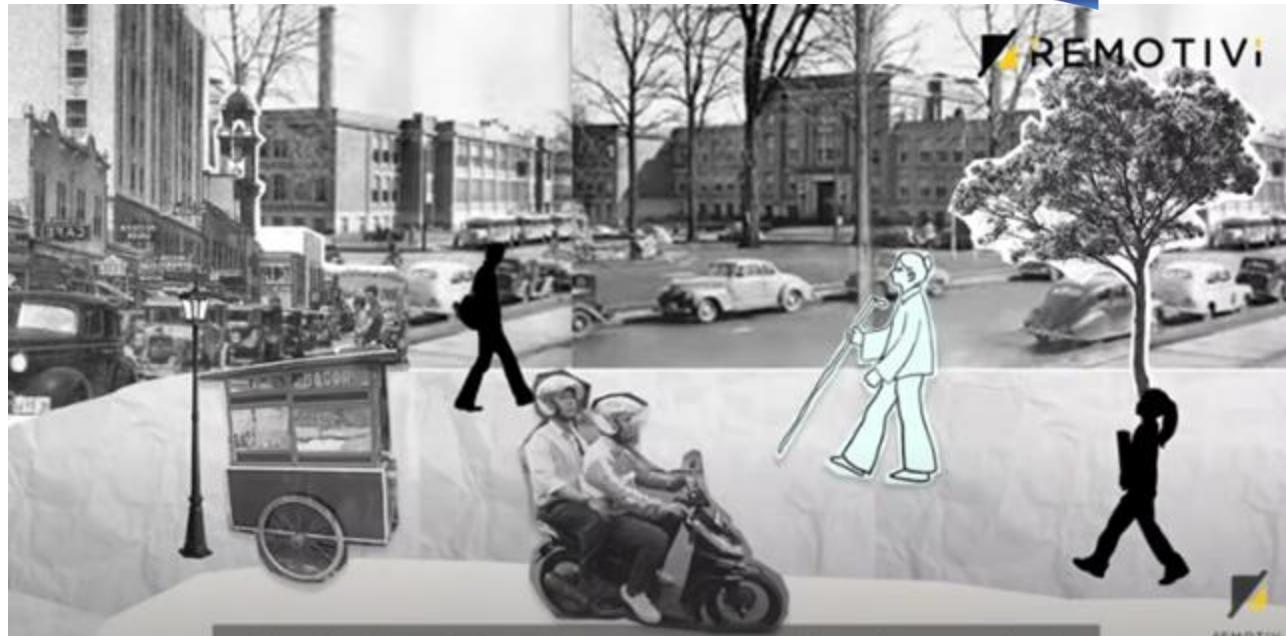
**Where?** – Hal ini terkait dengan konteks atau situasi atau lokasi masalah.

**Why?** – Mengacu pada mengapa kita perlu memecahkan masalah dan apa manfaatnya bagi para pemangku kepentingan setelah menyelesaikan masalah.



# Contoh Kasus

# Keadilan untuk Disabilitas di Indonesia



Disabilitas di media : <https://youtu.be/J1QFbzCirQE>

# The 4Ws of Problem Scoping – WHO?

**Siapa yang memiliki masalah tsb?**

Dalam AI project, “Who” adalah target user atau orang yang akan kita bantu dengan teknologi AI, untuk itu kita perlu merumuskan:

- 1. Siapa saja yang mengalami masalah tsb?**
- 2. Siapa saja yang terlibat dalam masalah tsb?**
- 3. Apa yang kamu ketahui tentang mereka?**

# The 4Ws of Problem Scoping – WHO?



## Siapa yang memiliki masalah tsb?

### 1. Siapa saja yang mengalami masalah tsb?

- Penyandang disabilitas

### 2. Siapa saja yang terlibat dalam masalah tsb?

- Pemangku kebijakan, media, pengelola fasilitas umum, masyarakat

### 3. Apa yang kamu ketahui tentang mereka?

- Disabilitas ingin mendapat kesempatan yang sama tanpa harus dikasihani atau mendapat ekspektasi tertentu
- Pemangku kebijakan, media, pengelola fasilitas umum selama ini belum menganggap disabilitas sebagai issue yang penting dibahas dalam memformulasikan fasilitas/sarana yang diberikan

Klsah inspratif

## The 4Ws of Problem Scoping – **WHAT?**

Apa permasalahannya sebenarnya?

Kita perlu mewawancarai si calon user untuk mengungkapkan pengalaman, tantangan, dan perasaan mereka menghadapi masalah tersebut. Sehingga, kita dapat merumuskan:

- 1. Apa masalah utamanya?**
- 2. Apa faktor-faktor pendukung masalah tsb?**

# The 4Ws of Problem Scoping – **WHAT?**



## Apa permasalahan sebenarnya?

### 1. Masalah utamanya?

- Kurangnya perhatian pemangku kebijakan dan masyarakat untuk mendukung dan memberi kesempatan yang sama pada penyandang disabilitas
- Selama ini disabilitas mendapat gambaran seseorang yang dikasihani atau mendapat ekspektasi tertentu

### 2. Apa faktor-faktor pendukung masalah tsb?

- Adanya riset mengenai stereotip penyandang disabilitas yang menjadi objek yang perlu dikasihani, beban masyarakat, dan bahan hiburan
- Kurangnya fasilitas yang justru lebih dibutuhkan oleh penyandang disabilitas, seperti akses fasilitas pendidikan dan rumah ibadah yang memadai

# The 4Ws of Problem Scoping – WHERE?



Dimana/pada saat apa permasalahan ini muncul?

## 1. Dimana saja si calon “user” mengalami masalah ini?

- Hampir di setiap tempat umum, misalnya di dunia pendidikan, masih jarang universitas ramah terhadap penyandang disabilitas

## 2. Pada kondisi seperti apa si calon “user” mengalami masalah ini?

- Kurangnya kesempatan yang sama, misalnya pada pekerjaan tertentu, terkadang perusahaan langsung menolak disabilitas tanpa melihat kemampuannya



# The 4Ws of Problem Scoping – WHY?

Mengapa kamu yakin masalah ini sangat penting dibahas?

1. Apa nilai utama dari permasalahan ini?
2. Bagaimana kamu bisa memperbaiki (sebagian atau seluruh) dari masalah tsb?

# The 4Ws of Problem Scoping – WHY?

Mengapa kamu yakin masalah ini sangat penting dibahas?

## 1. Mengapa masalah ini penting?

- Berdasarkan data, 1 dari 10 warga Indonesia penyandang disabilitas

## 2. Apa nilai utama dari permasalahan ini?

- Sesuai dengan nilai pancasila kedua dan kelima

## 3. Bagaimana kamu bisa memperbaiki (sebagian atau seluruh) dari masalah tsb?

- Mengembangkan teknologi tertentu, misalnya aplikasi text-to-speech pembaca buku/ebook untuk membantu tunanetra
- Menggencarkan edukasi masyarakat di media, misalnya menentukan jam atau program yang paling baik untuk mengedukasi

1.

# Problem Statement Template

Our	[stakeholders]	Who
has a problem that	[issue, problem, need]	What
when / while	[context, situation].	Where
An ideal solution would	[benefit of solution for them]	Why

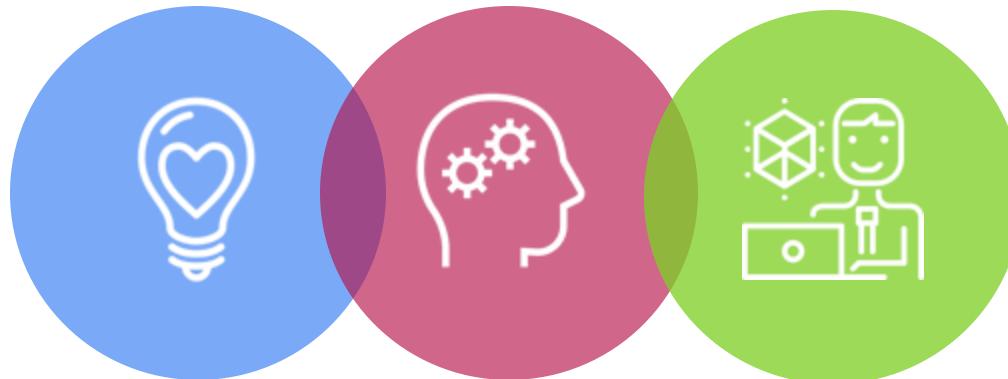
# Problem Statement Template

Subjek :	penyandang disabilitas	Who
Punya masalah berupa :	minimnya sarana dan kesempatan yang sama	What
pada saat:	mengakses fasilitas umum dan sarana pendidikan/karir	Where
Solusi idealnya:	yang dapat membantu mereka beraktivitas layaknya manusia pada umumnya	Why

# Tugas

Cari Masalah Sosial di sekitarmu, lalu buat problem scopingnya,  
Download templatanya disini : <https://bit.ly/Template4WS>

Dengan rumusan masalah yang tepat, kita dapat membuat perencanaan projek yang lebih baik



### STAGE I

Project planning &  
data collection

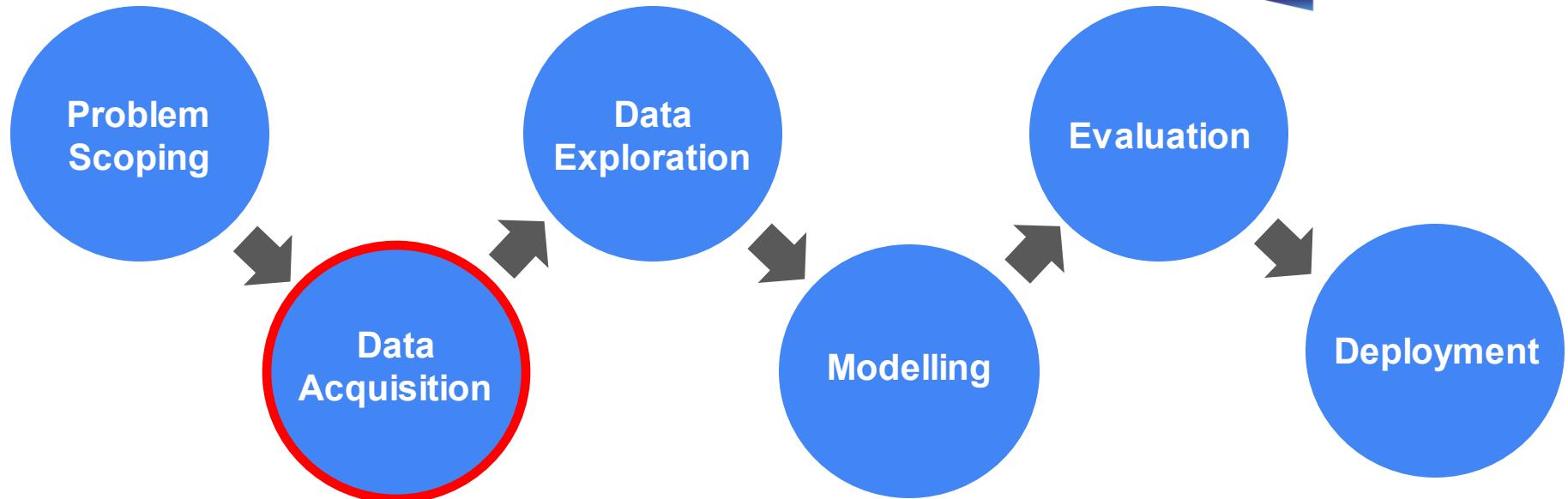
### STAGE II

Design & training  
of the Machine  
Learning (ML)  
model

### STAGE III

Deployment &  
maintenance

# Siklus proyek AI



# 2. Data Acquisition

**Bagaimana kita tahu data apa yang kita butuhkan?**



**Akuisisi Data** adalah serangkaian proses pengukuran, pengumpulan, dan validasi data yang dibutuhkan dalam suatu proyek AI.



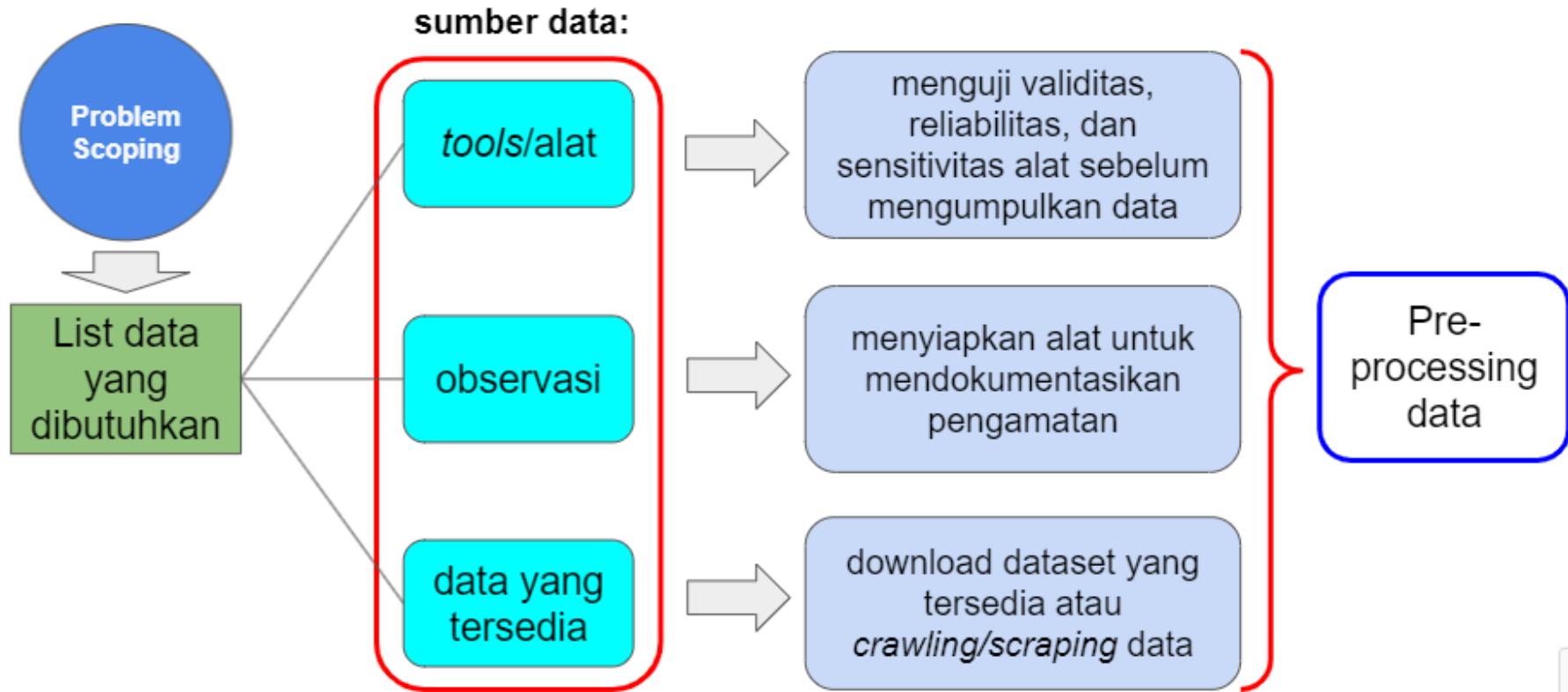
Proses ini bisa dilakukan dengan **menggunakan tools/alat** tertentu seperti sensor, bisa juga melalui **observasi**, atau mengambil **data yang sudah tersedia** di internet atau lembaga

Jika menggunakan alat, pastikan:

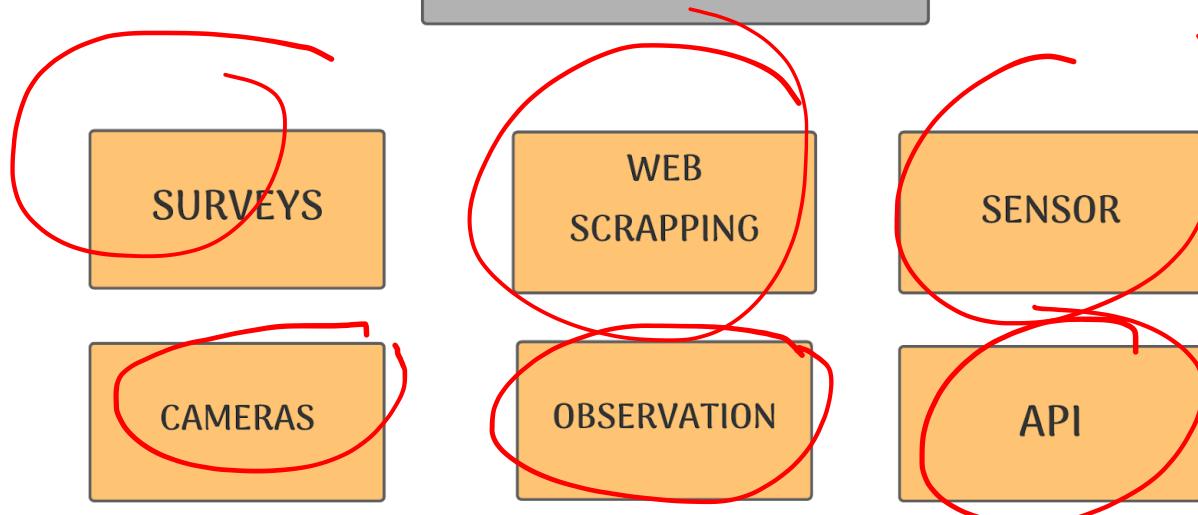
1. **Validitas** → alat mampu mengukur/ mengidentifikasi data yang dibutuhkan
2. **Reliabilitas** → kekonsistensian alat, menghindari bias
3. **Sensitivitas** → seberapa sensitif alatnya

~~extremal value/outlier~~

# Proses persiapan



## DATA SOURCES



# Sistem Akuisisi Data



Artificial Intelligence  
Mastery Program

Secara umum, ada 3 sistem Akuisisi data:

## 1. Manual

Peneliti mengumpulkan dan menginput data secara manual

## 2. Membuat pipeline untuk scraping atau membaca data

Peneliti membuat program komputer untuk mengambil atau membaca data

## 3. Automated Data Acquisition

Menggunakan device tertentu yang dapat mengumpulkan data dari sensor untuk diinput ke komputer

# Contoh Automated Data Acquisition System

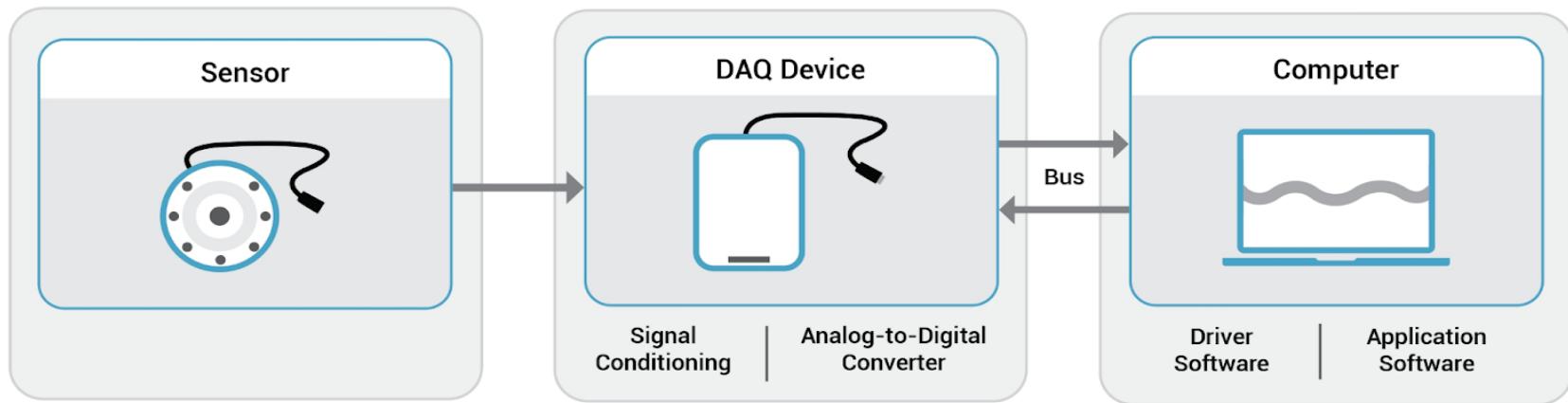
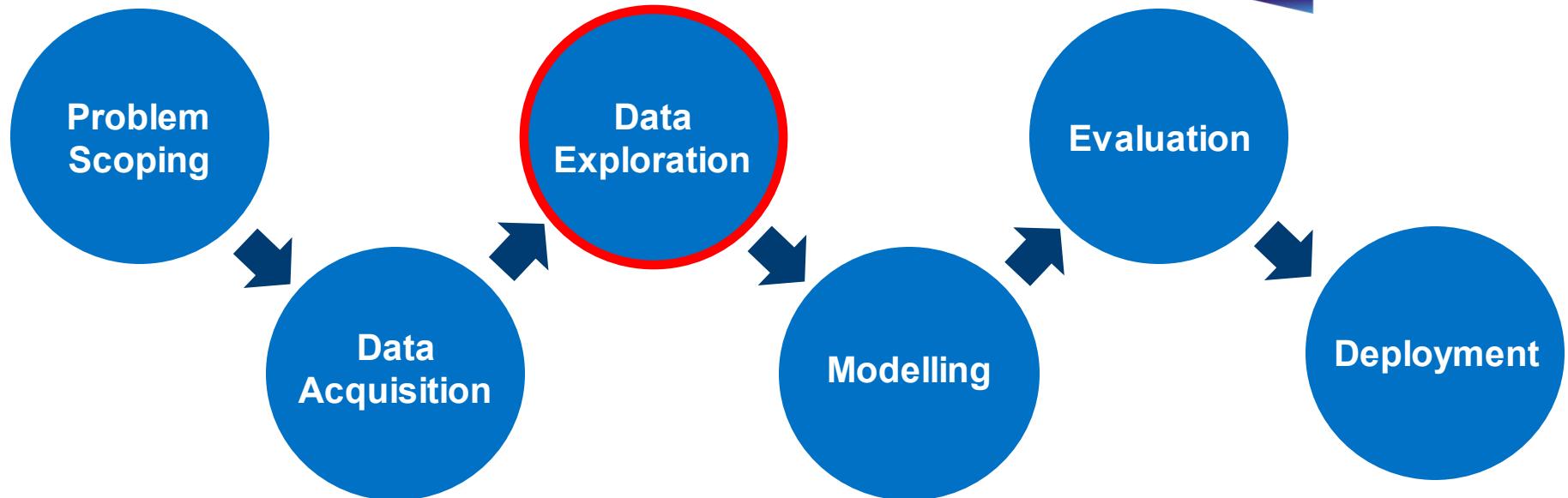


Image Resource: <https://assets.omega.com/resources/DAQ-system-infographic.png>

# AI Project Cycle



# 3. Data Exploration

Mengapa kita perlu mengeksplorasi dan  
memvisualisasi data?

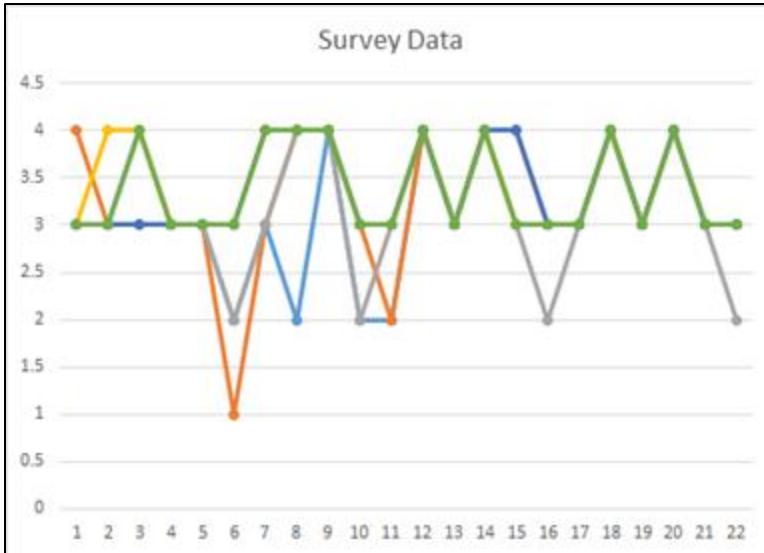
# Mengapa harus eksplorasi data?

- Kita ingin segera memahami tren, hubungan, dan pola yang terkandung dalam data
- Membantu kita menentukan strategi model mana yang akan digunakan pada stage selanjutnya

# Mengapa mengeksplorasi data melalui visualisasi?

Representasi visual lebih mudah untuk dipahami dan dikomunikasikan kepada orang lain

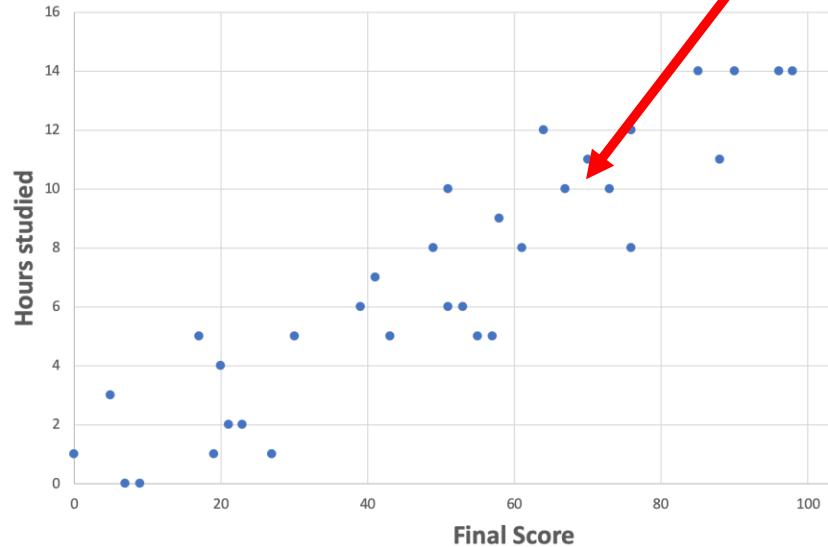
	A	B	C	D	E	F	G
1	1 = Strongly disagree; 4 = Strongly agree						
2	learnin materials g and objecti resources (e.g., were videos,	the ideas, knowledge and skills learnt from the	The presentatio n was clear.	The questions were adequately addressed.	The facilitator was skillful at facilitating the participants' learning.	The duration of programme was sufficient to meet the objectives of the programme.	
3	3	4	3	3	3	3	3 Just Right
4	3	3	3	4	3	3	3 Just Right
5	4	3	4	4	3	3	4 Too Short
6	3	3	3	3	3	3	3 Just Right
7	3	3	3	3	3	3	3 Just Right
8	2	1	2	3	3	3	3 Too Short
9	3	3	3	4	4	4	4 Just Right
10	2	4	4	4	4	4	4 Just Right
11	4	4	4	4	4	4	4 Just Right
12	2	3	2	3	3	3	3 Just Right
13	2	2	3	3	3	3	3 Just Right
14	4	4	4	4	4	4	4 Just Right
15	3	3	3	3	3	3	3 Just Right
16	4	4	4	4	4	4	4 Just Right
17	3	4	3	3	4	3	3 Just Right
18	3	3	2	3	3	3	3 Just Right
19	3	3	3	3	3	3	3 Just Right
20	4	4	4	4	4	4	4 Just Right
21	3	3	3	3	3	3	3 Just Right
22	4	4	4	4	4	4	4 Just Right
23	3	3	3	3	3	3	3 Just Right



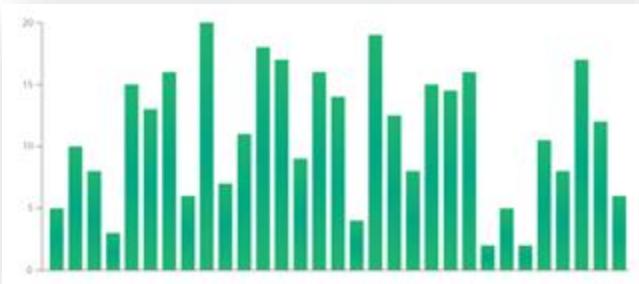
# Contoh visualisasi

Representasi visual membantu kita memahami data

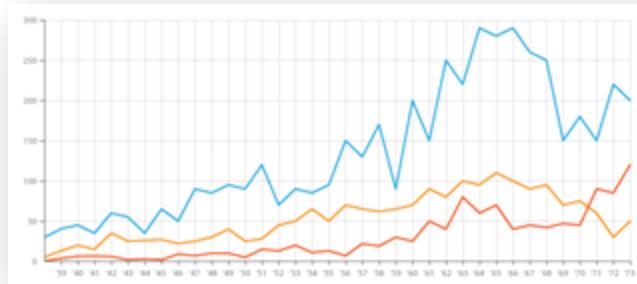
Name	Final Score	Hours Studied
RUBY	55	5
EMILY	19	1
GRACE	51	10
JESSICA	49	8
CHLOE	64	12
SOPHIE	67	10
LILY	5	3
AMELIA	70	11
EVIE	88	11
MIA	53	6
ELLA	58	9
CHARLOTTE	23	2
LUCY	85	14
MEGAN	61	8
ELLIE	73	10
ISABELLE	76	12
ISABELLA	17	5
HANNAH	9	0



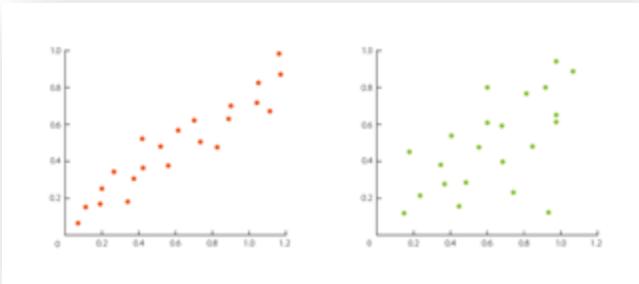
Artificial Intelligence  
Mastery Program



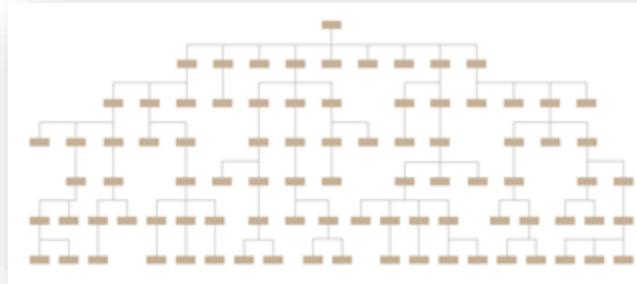
Bar Chart



Line Chart



Scatterplot

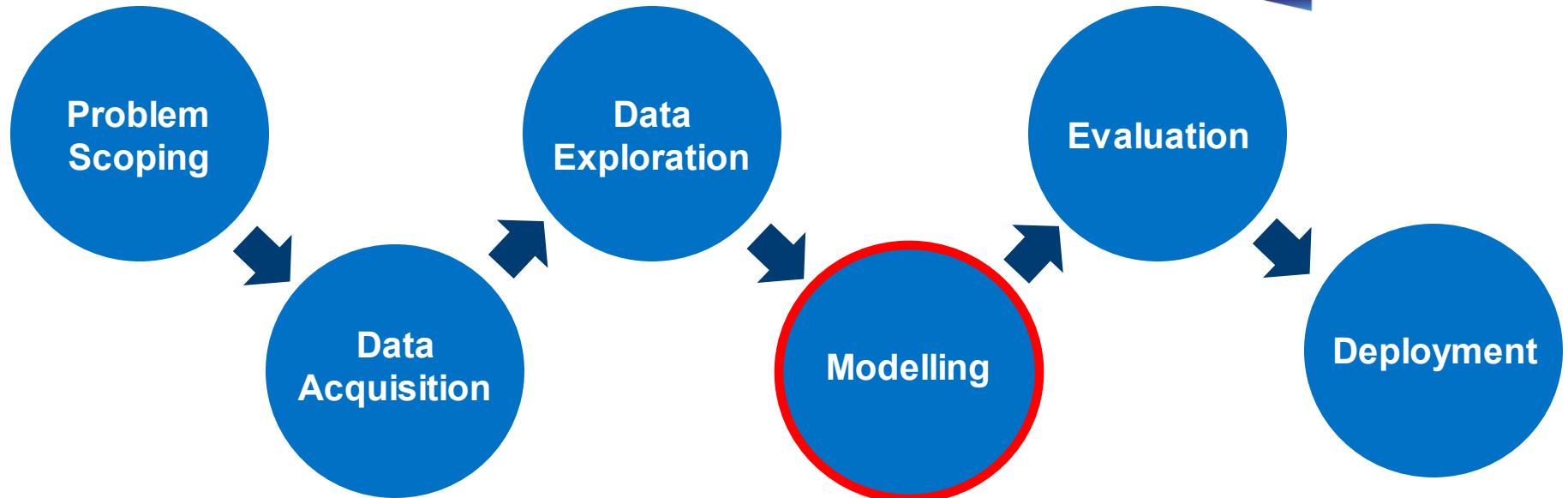


Tree Diagram

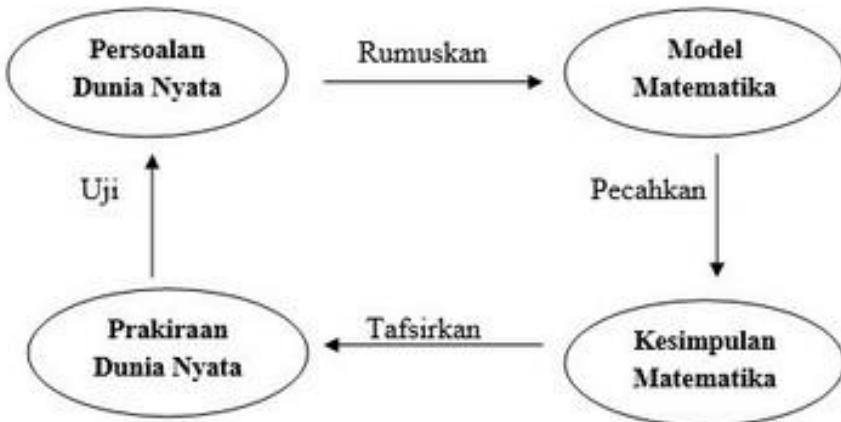
Source:  
[datavizcatalogue.com](http://datavizcatalogue.com)

# 4. Modelling

# AI Project Cycle

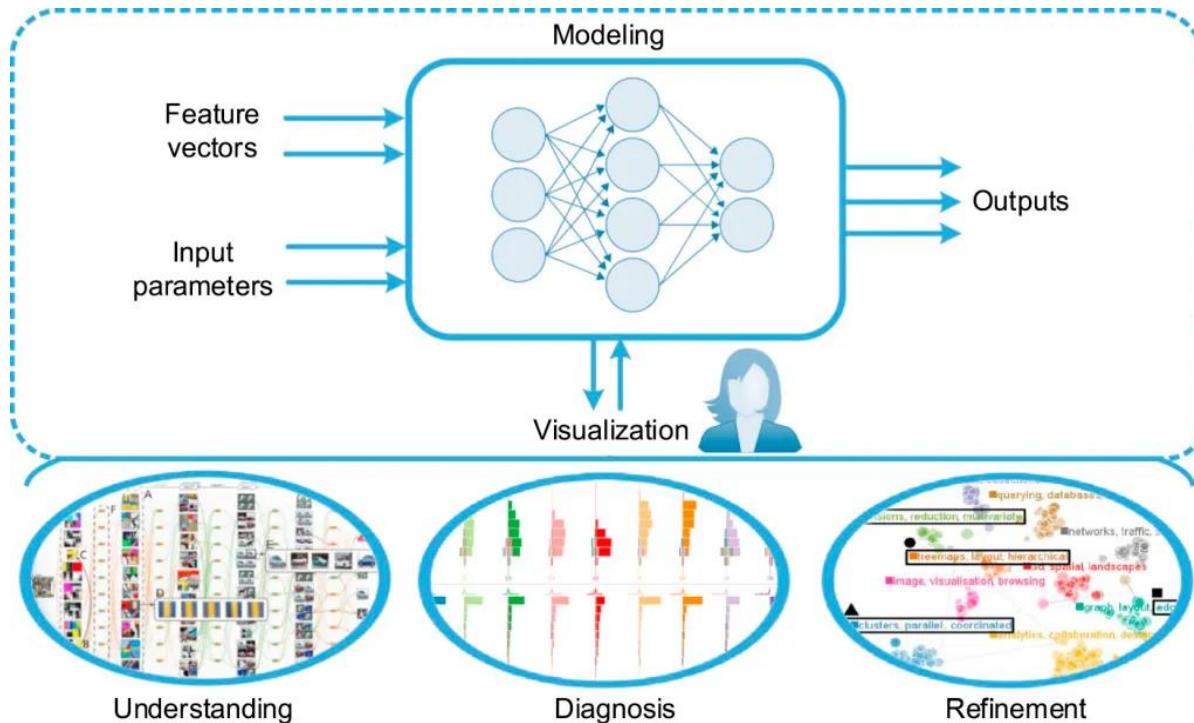


# Modeling



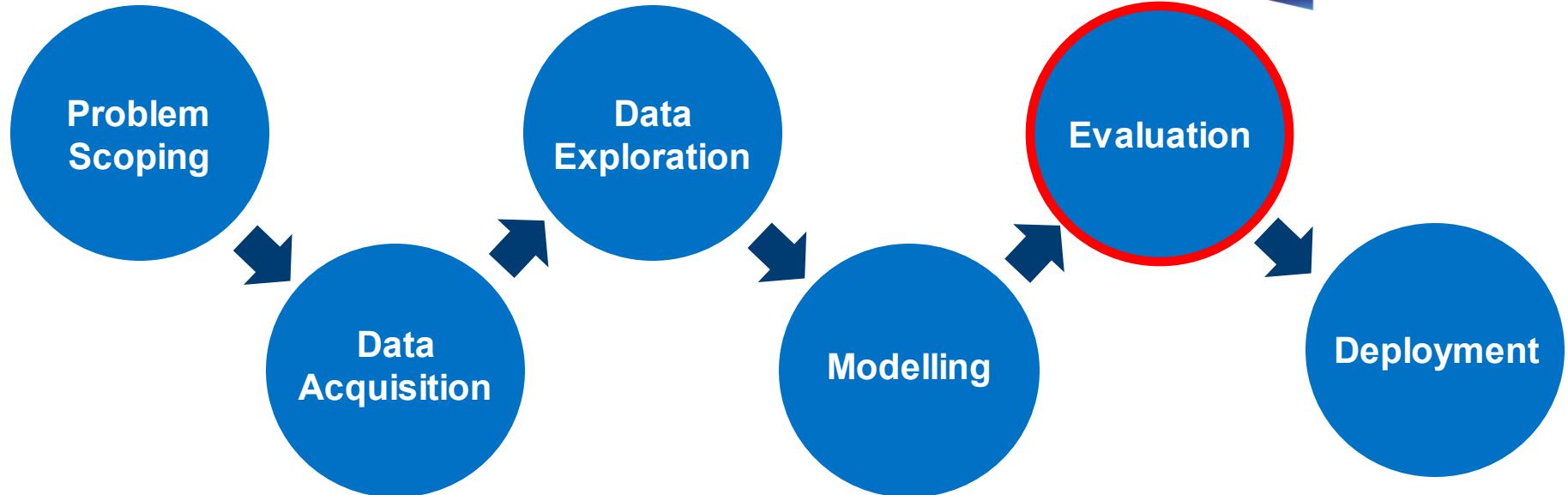
Model dalam istilah teknologi adalah representasi suatu masalah dalam bentuk yang lebih sederhana sehingga lebih jelas dan mudah dikerjakan. Model yang baik cukup mengandung bagian-bagian yang perlu saja.

# Pemodelan Machine learning digunakan untuk memahami pola tertentu di data



sumber gambar : <https://www.sciencedirect.com/science/article/pii/S2468502X17300086>

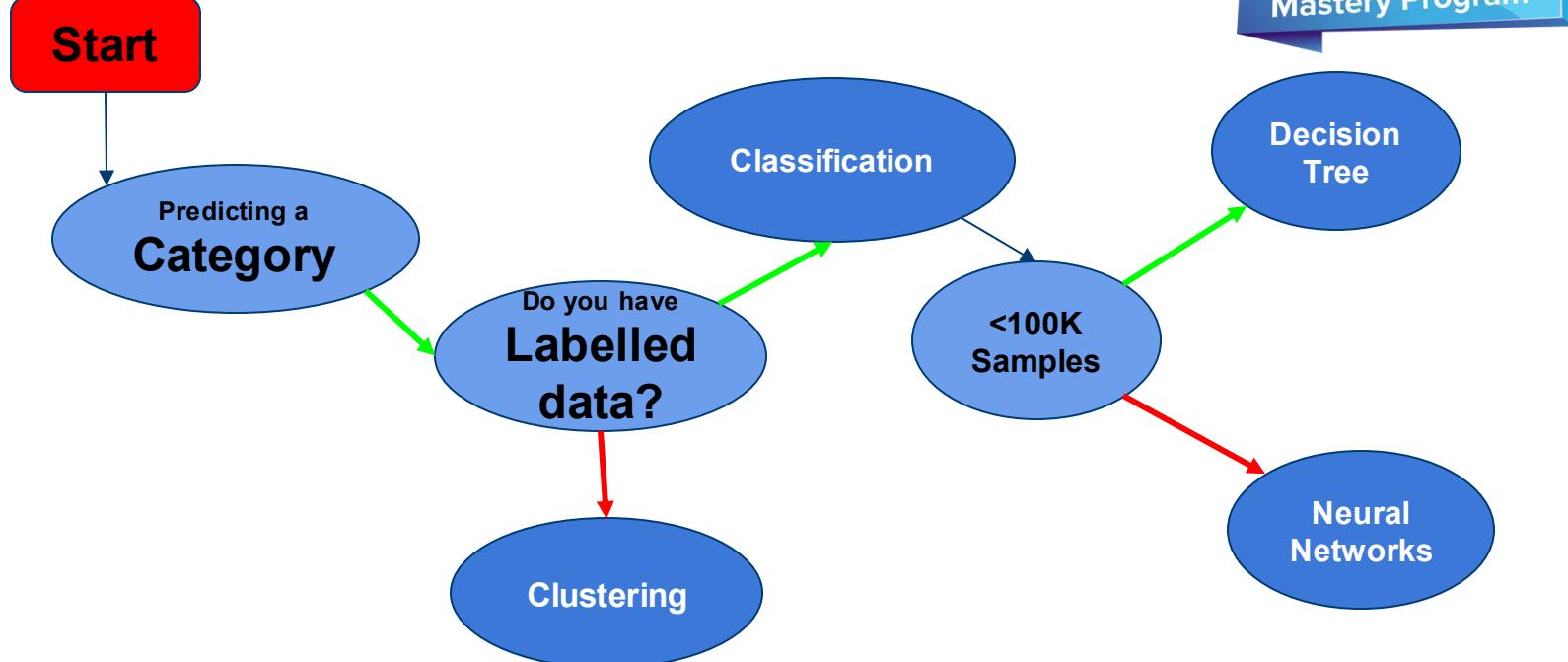
# AI Project Cycle



# 5. EVALUASI MODEL

Bagaimana kita memilih model yang tepat?

# Choosing Suitable Model



# Confusion Matrix

		Realita	
		Yes	No
Prediksi	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

# Confusion Matrix

		Realita	
		Yes	No
Prediksi	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Apakah ada kebakaran di hutan?

Prediksi:  
Yes / No

Realita:  
Yes / No



Apakah ada kebakaran di hutan?

Prediksi : Yes

Realita : Yes

True Positive

Apakah ada kebakaran di hutan?

Prediksi : No

Realita : No

True Negative



Apakah ada kebakaran di hutan?

Prediksi : No

Realita : Yes

False Negative



Artificial Intelligence  
Mastery Program

Apakah ada kebakaran di hutan?

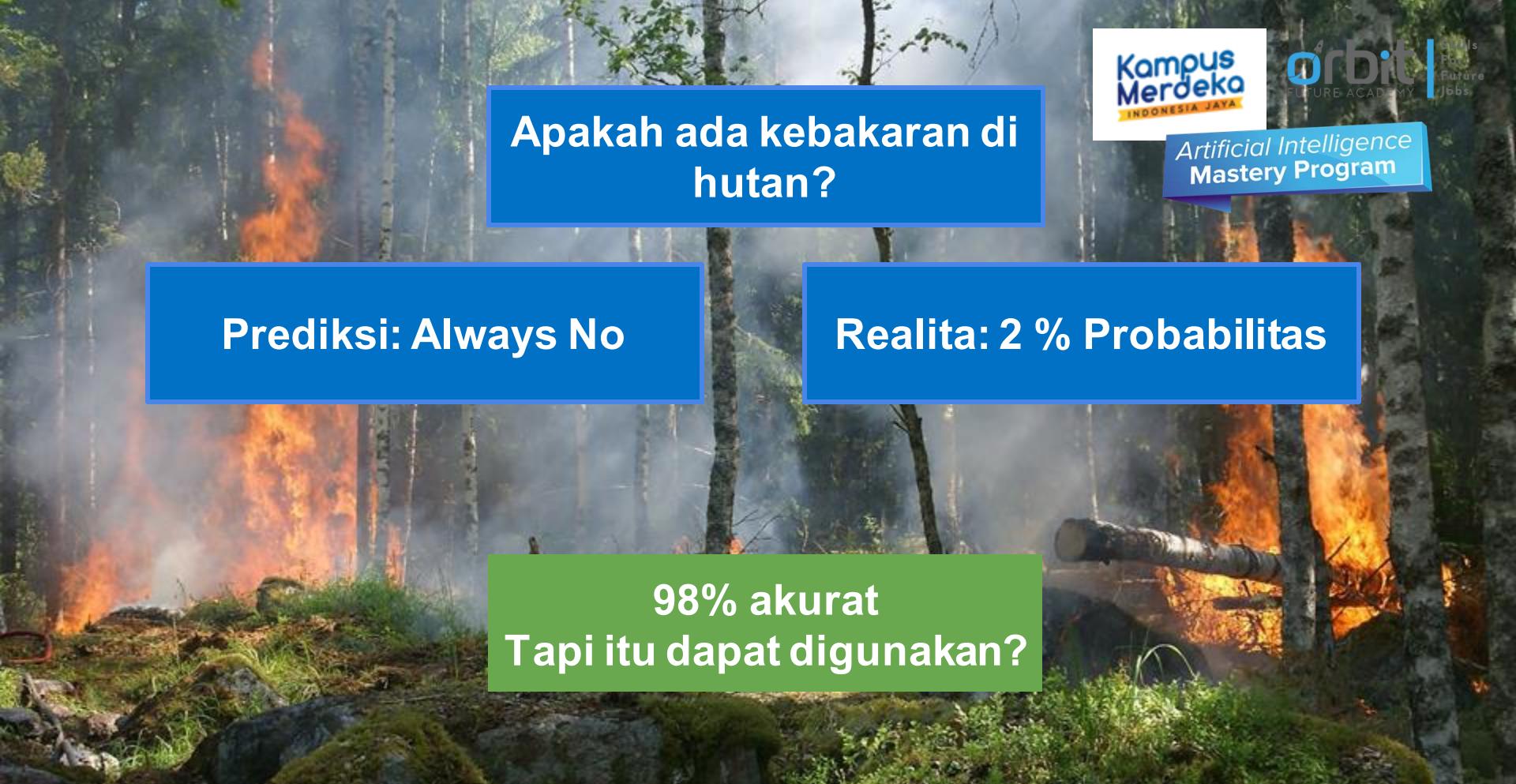
Prediksi : Yes

Realita : No

False Positive

# Akurasi

- Akurasi adalah persentase prediksi yang benar dari semua kemungkinan
- Akurasi =  $(TP + TN) / (TP + FP + TN + FN)$
- Akurasi = (Prediksi Benar) / (Semua Case)
- Apakah akurasi tinggi setara dengan kinerja yang baik?



Apakah ada kebakaran di hutan?

Prediksi: Always No

Realita: 2 % Probabilitas

98% akurat  
Tapi itu dapat digunakan?



Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

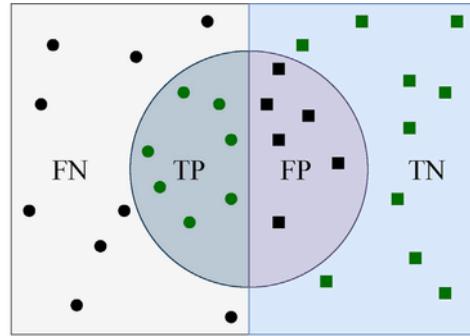
# Presisi

- Presisi adalah persentase dari case yang diprediksi positif (TP + FP) yang ternyata positif (TP)
- Presisi =  $(TP) / (TP + FP)$
- Presisi =  $(\text{True Positive}) / (\text{Total Prediksi Positif})$

# Recall

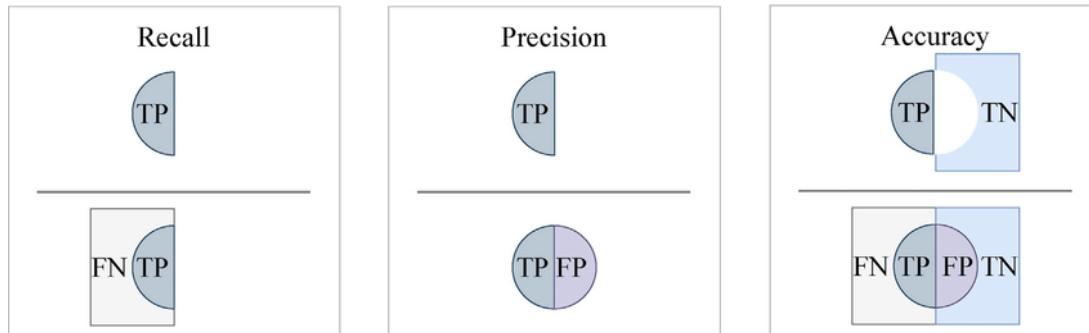
- Recall mengukur pecahan kasus positif (TP + FN) yang diidentifikasi dengan benar (TP)
- $\text{Recall} = (\text{TP}) / (\text{TP} + \text{FN})$
- $\text{Recall} = (\text{True Positive}) / (\text{Total aktual positif})$

# Representasi: Akurasi, Presisi, dan Recall



Sumber:

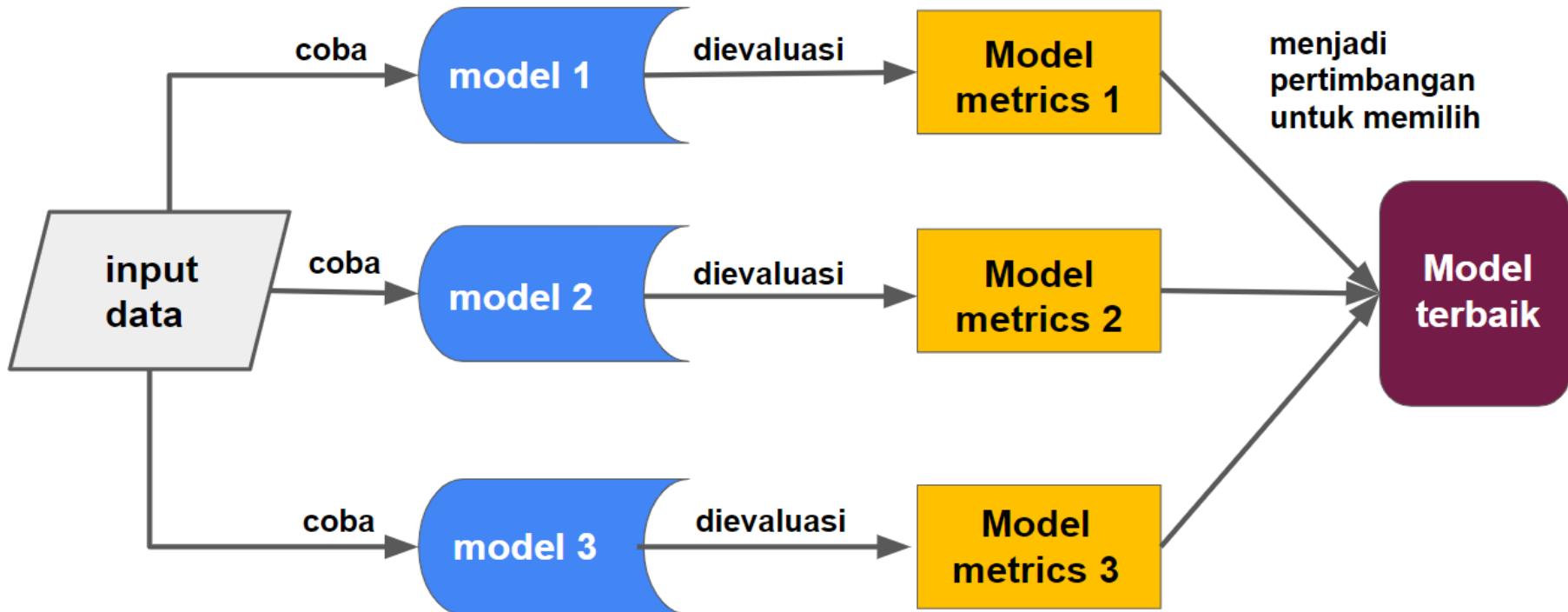
[https://www.researchgate.net/figure/Visualizing-accuracy-recall-aka-sensitivity-and-precision-which-are-the-common\\_fig3\\_346129022](https://www.researchgate.net/figure/Visualizing-accuracy-recall-aka-sensitivity-and-precision-which-are-the-common_fig3_346129022)



Jika terjadi kasus kebakaran hutan, mana yang akan anda pilih? False Negative atau False Positive?

# Model Selection

Berikut adalah contoh proses memilih model terbaik





*Artificial Intelligence  
Mastery Program*

# FGD

Hitung akurasi, presisi dan recall model berikut:

		Realita	
		Yes	No
Prediksi	Yes	2	5
	No	10	150

Manakah metrik yang lebih tepat mengevaluasi jika ini adalah model untuk:

- 1) Pendekripsi covid
- 2) Pendekripsi spam email

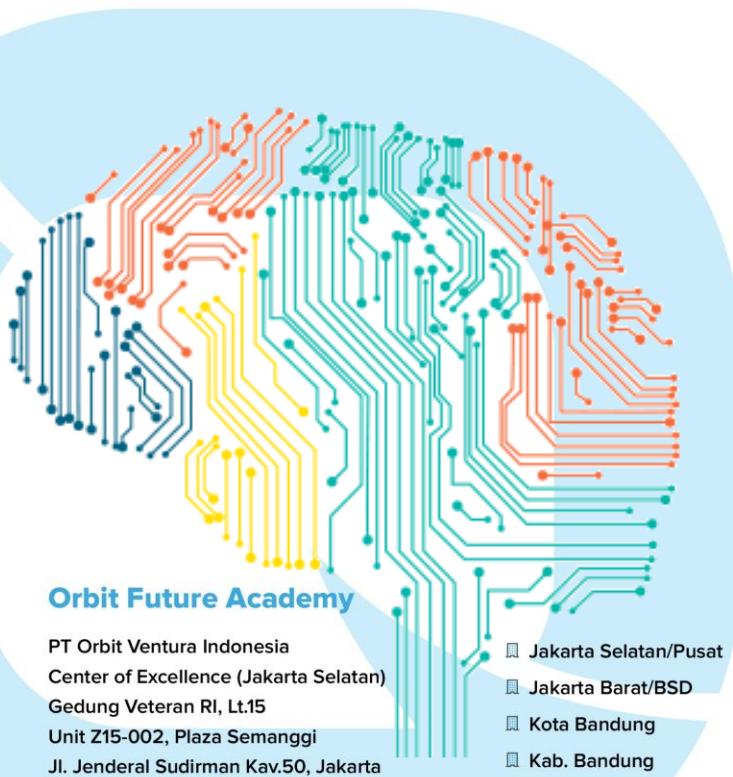


Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY

Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program



## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

- [Orbit Future Academy](#)
- [@OrbitFutureAcademyIn1](#)
- [OrbitFutureAcademy](#)
- [Orbit Future Academy](#)

# TERIMA KASIH

## THANK YOU

# AI Mastery Course



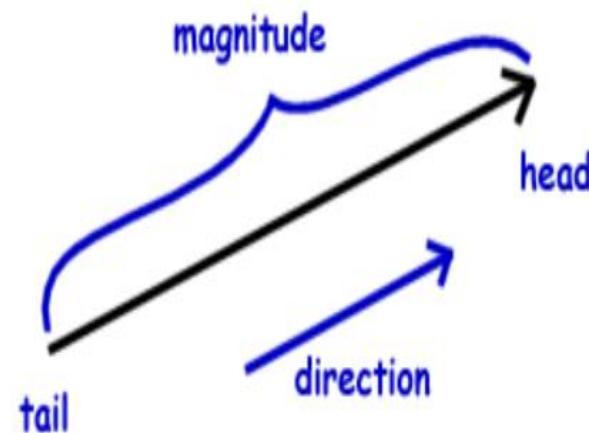
## Section

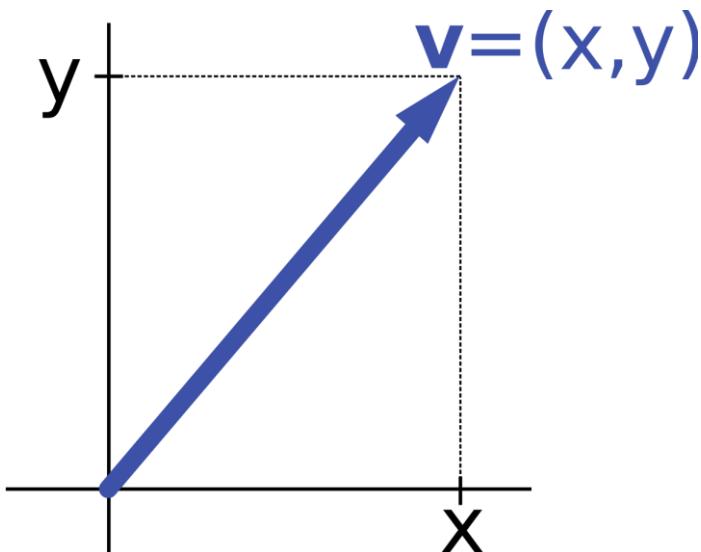
### Recap Maths, Basic & Advanced Statistics



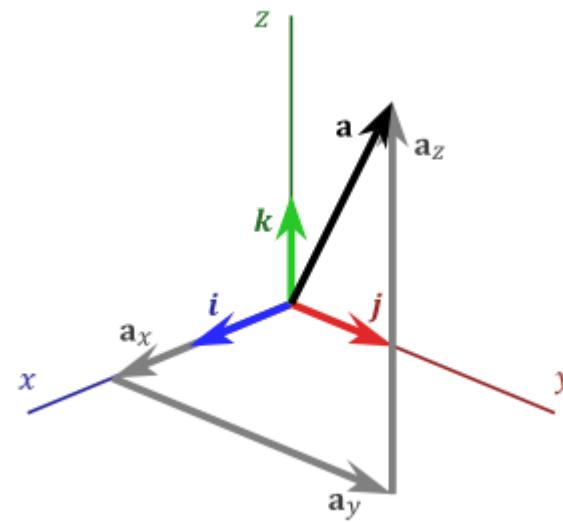


### Visualisasi Vektor





Vector dalam ruang 2D

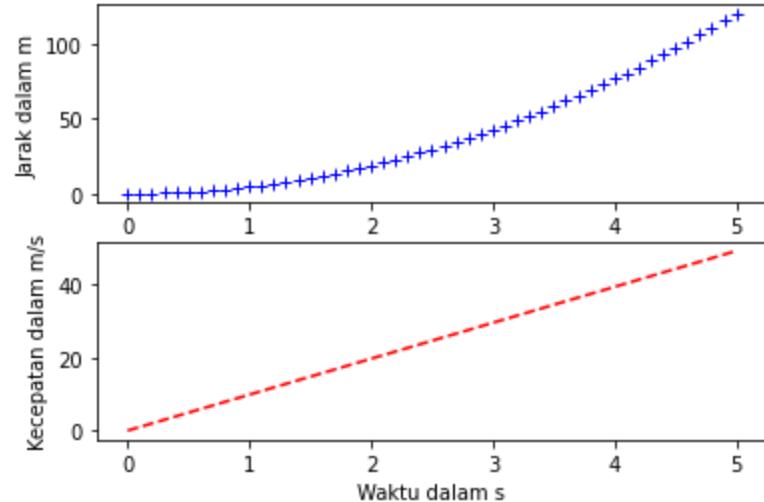
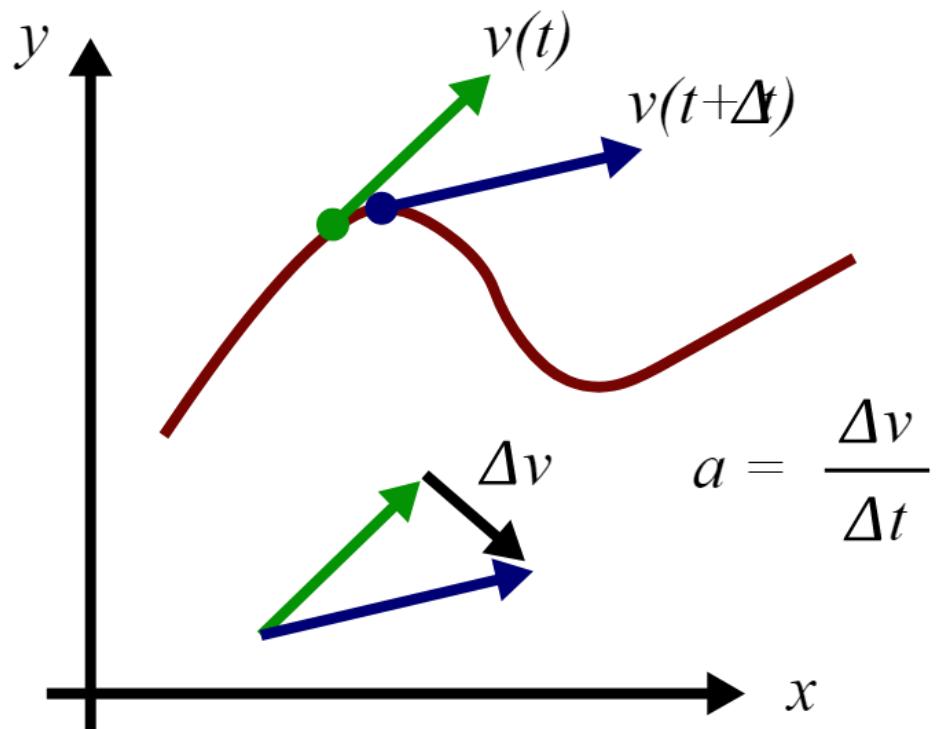


Vector dalam ruang 3D

Let's exploring other dimensions



## Percepatan sebagai vektor



<https://colab.research.google.com/drive/1beVszViybT63eORvebVp25gCqtdQXI3?usp=sharing>

# Basic Statistic with Python cheatsheet

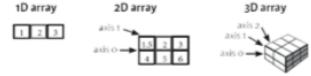
## Numpy

The NumPy library is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays.

Use the following import convention:

```
>>> import numpy as np
```

### NumPy Arrays



## Creating Arrays

```
>>> arr = np.array([1,2,3]) # Create an array of integers
>>> arr = np.array((1,2,3), dtype = float) # Create an array of float values
```

### Initial Placeholders

```
>>> arr = np.empty(10) # Create an array of zeros
>>> arr = np.empty((2,3,4), dtype = int) # Create an array of ones
>>> arr = np.zeros((2,5,6)) # Create an array of mostly spaced values (step value)
>>> arr = np.linspace(1,5,10) # Create an array of evenly spaced values (number of samples)
>>> arr = np.full((2,2),7) # Create a constant array
>>> arr = np.random.randint(2,10) # Create an array with random values
>>> arr = np.empty(10, dtype = float) # Create an empty array
```

## I/O

### Saving & Loading On Disk

```
>>> np.save('my_array', a) # Save array as my_array.npy
>>> np.load('my_array.npy', a) # Load array from my_array.npy
```

### Saving & Loading Text Files

```
>>> np.savetxt('my_file.txt', a) # Save array as my_file.txt
>>> np.loadtxt('my_file.txt', a) # Load array from my_file.txt
```

## Asking For Help

```
>>> np.info(np.ndarray)
```

## Inspecting Your Array

```
>>> a.shape #Array dimensions
>>> a.dtype #Type of array elements
>>> a.ndim #Number of array dimensions
>>> a.size #Number of array elements
>>> a.itemsize #Item size in bytes
>>> a.dtype.name #Name of data type
>>> a.astype(int) #Convert an array to a different type
```

## Data Types

```
>>> np.int64 #Signed 64-bit Integer type
>>> np.float32 #Standard double-precision floating point
>>> np.complex64 #Complex floating point with 32 bits
>>> np.bool_ #Boolean type storing TRUE and FALSE values
>>> np.str_ #String type
>>> np.string_, np.flexible #String type
>>> np.unicode_, np.flexible #Unicode Type
```

## Array Mathematics

### Arithmetic Operations

```
>>> a + b #Addition
>>> a - b #Subtraction
>>> a * b #Multiplication
>>> a / b #Division
>>> a ** b #Exponentiation
>>> a % b #Modulo
>>> a // b #Floor division
>>> a + 1 #Add 1 to every element
>>> a - 1 #Subtract 1 from every element
>>> a * 2 #Double every element
>>> a / 2 #Divide every element by 2
>>> a ** 2 #Square every element
>>> a // 2 #Floor divide every element by 2
```

### Comparison

```
>>> a == b #Element-wise comparison
>>> np.all(a == b) #True if all elements are equal
>>> np.any(a == b) #True if any elements are equal
>>> np.allclose(a, b) #Element-wise equality tolerance check
>>> np.array_equal(a, b) #Element-wise equality
>>> np.array_equiv(a, b) #Element-wise equality
```

### Aggregate Functions

```
>>> a.sum() #Summation
>>> a.mean() #Average value
>>> a.max() #Maximum value
>>> a.argmax() #Index of the maximum value
>>> a.min() #Minimum value
>>> a.argmin() #Index of the minimum value
>>> a.var() #Variance coefficient
>>> a.std() #Standard deviation
```

## Copying Arrays

```
>>> b = a.view() #Create a view of the array with the same data
>>> np.copy(b) #Creates a copy of the array
>>> b = a.copy() #Creates a deep copy of the array
```

## Sorting Arrays

```
>>> a.argsort() #Sort the elements of the arr
>>> a.argsort(0) #Sort the elements of arr's axis 0
```

## Subsetting, Slicing, Indexing

**Selecting**

```
>>> a[2] #Select the element at the 3rd index
>>> a[1,2] #Select the element of row 2 column 0 (equivalent to a[2][0])
>>> a[0,1,2] #Selects all elements of row 0 and column 2
```

**Indexing**

```
>>> a[1:,2:] #Select slices of index 0 to 1 and 2 to 3
>>> a[::2,::2] #Select slices of index 0 to 1 in rows 2 and 4
>>> a[1::2,::2] #Select all items of row 0 (equivalent to a[0::2,::2])
>>> a[1,::2] #Selects slice of (1,..)
>>> a[[1,2],::2] #Selects slice of (1,2,..)
>>> a[1,::2] #Reversed array or a[::2,1,2]
```

**Basic indexing**

```
>>> a[1,2,3] #Select elements from a less than 2
>>> a[1,2,3,4] #Select elements from a less than 4
```

**Advanced indexing**

```
>>> a[[1,2,3],[0,1,2],[3,4,5]] #Select elements (1,0),(0,1),(1,2) and (0,0)
>>> a[[1,2,3],[0,1,2,3,4,5]] #Select elements (1,0),(0,1),(1,2),(0,2),(1,3),(0,3)
>>> a[[1,2,3],[0,1,2,3,4,5],[3,4,5]] #Select elements (1,0,0),(0,1,0),(1,2,0) and (0,0,1)
```

## Array Manipulation

**Transposing Array**

```
>>> a.T #Transpose(b) #Transpose array dimensions
>>> a[1,2].T #Transpose dimension
```

**Changing Array Shape**

```
>>> a.reshape(-1,1) #Reshape the array
>>> a.reshape(1,-1) #Reshape the array
>>> a.reshape(1,2) #Reshape, but don't change data
```

**Adding/Deleting Elements**

```
>>> a.resize((2,3)) #Create a new array with shape (2,3)
>>> a.resize((2,3), 0) #Create a new array with shape (2,3) and fill it with 0
>>> a[0,0] #Delete item from an array
>>> a.delete(0,0) #Delete item from an array
```

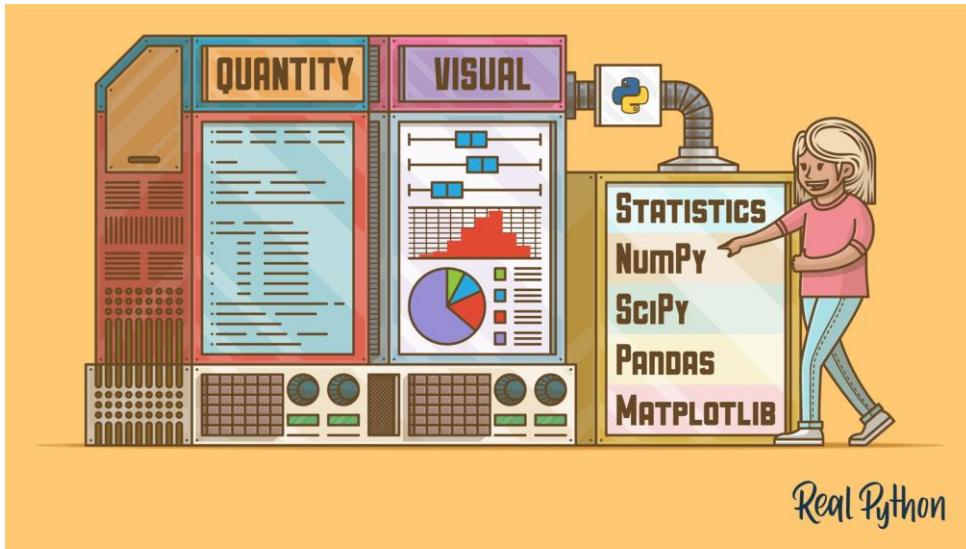
**Combining Arrays**

```
>>> np.concatenate((a,b)) #Concatenate arrays
>>> np.concatenate((a,[2,3,2,3,2])) #Concatenate arrays
>>> np.vstack((a,b)) #Stack arrays vertically (row-wise)
>>> np.hstack((a,b)) #Stack arrays horizontally
>>> np.r_[a,b] #Stack arrays vertically (row-wise)
>>> np.h_[a,b] #Stack arrays horizontally (column-wise)
>>> np.vstack((1,2,3)) #Stack arrays vertically (row-wise)
>>> np.vstack((1,2,3)).T #Stack arrays vertically (column-wise)
>>> np.vstack((1,2,3)).T.T #Stack arrays vertically (row-wise)
>>> np.vstack((1,2,3)).T.T.T #Stack arrays vertically (row-wise)
>>> np.vstack((1,2,3)).T.T.T.T #Stack arrays vertically (row-wise)
```

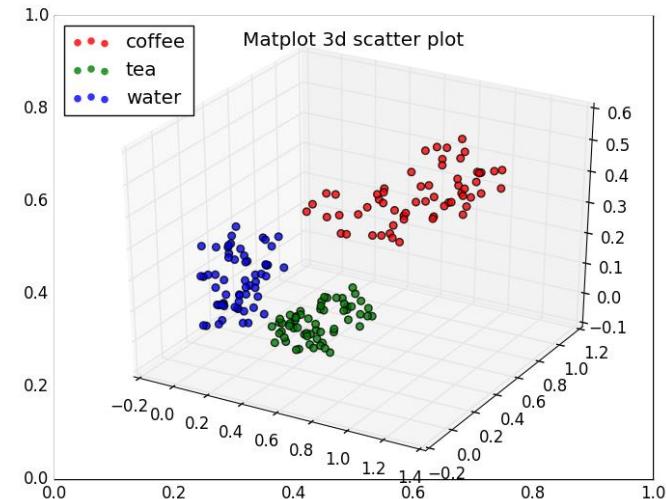
**Slicing Arrays**

```
>>> np.hsplit(a,3) #Split the array horizontally at the 3rd index
>>> np.vsplit(a,3) #Split the array vertically at the 3rd index
>>> np.vstack((1,2,3)) #Stack arrays vertically at the 3rd index
>>> np.vstack((1,2,3)).T #Stack arrays vertically at the 3rd index
```

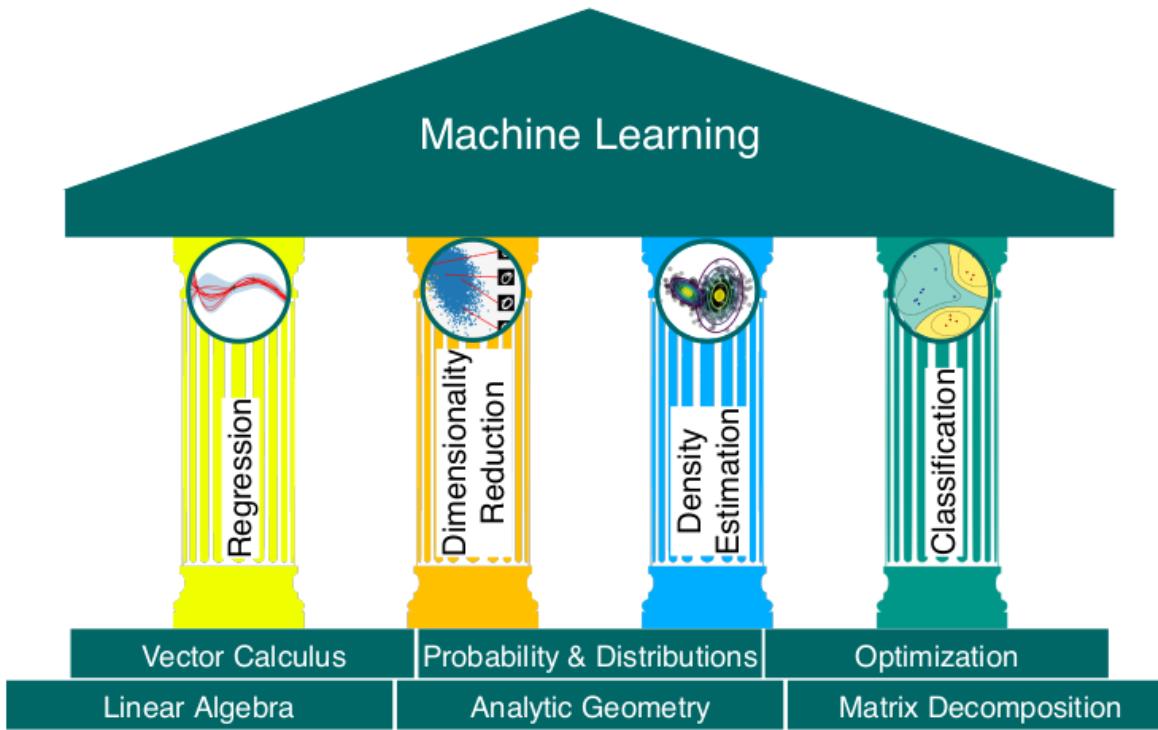




<https://realpython.com/python-statistics/>



Contoh tampilan visual dengan matplotlib





# TERIMA KASIH

## THANK YOU

### Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

### Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

### Social Media

-  Orbit Future Academy
-  @OrbitFutureAcademyIn1
-  OrbitFutureAcademy
-  Orbit Future Academy

# Module

Artificial Intelligence Fundamental

## Section

Machine Learning



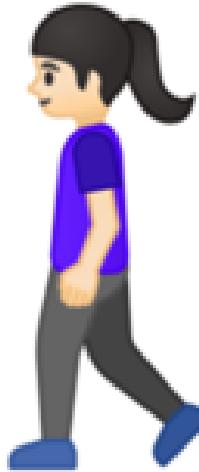
# Fundamental of Machine Learning

Supervised learning

# Paradigma Pemrograman

Traditional Programming & Machine Learning Approach

## Traditional Programming



```
if(speed<4){  
    status=WALKING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



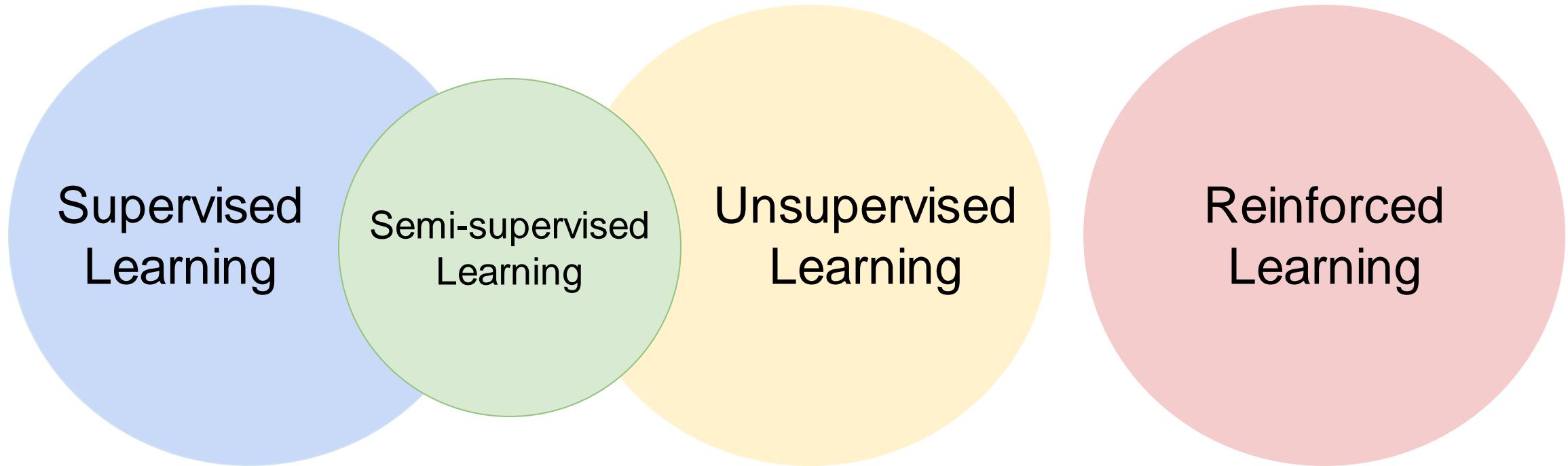
```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```



?

Images Source: <https://developers.google.com/codelabs/tensorflow-1-helloworld#0>

## Machine Learning Approach



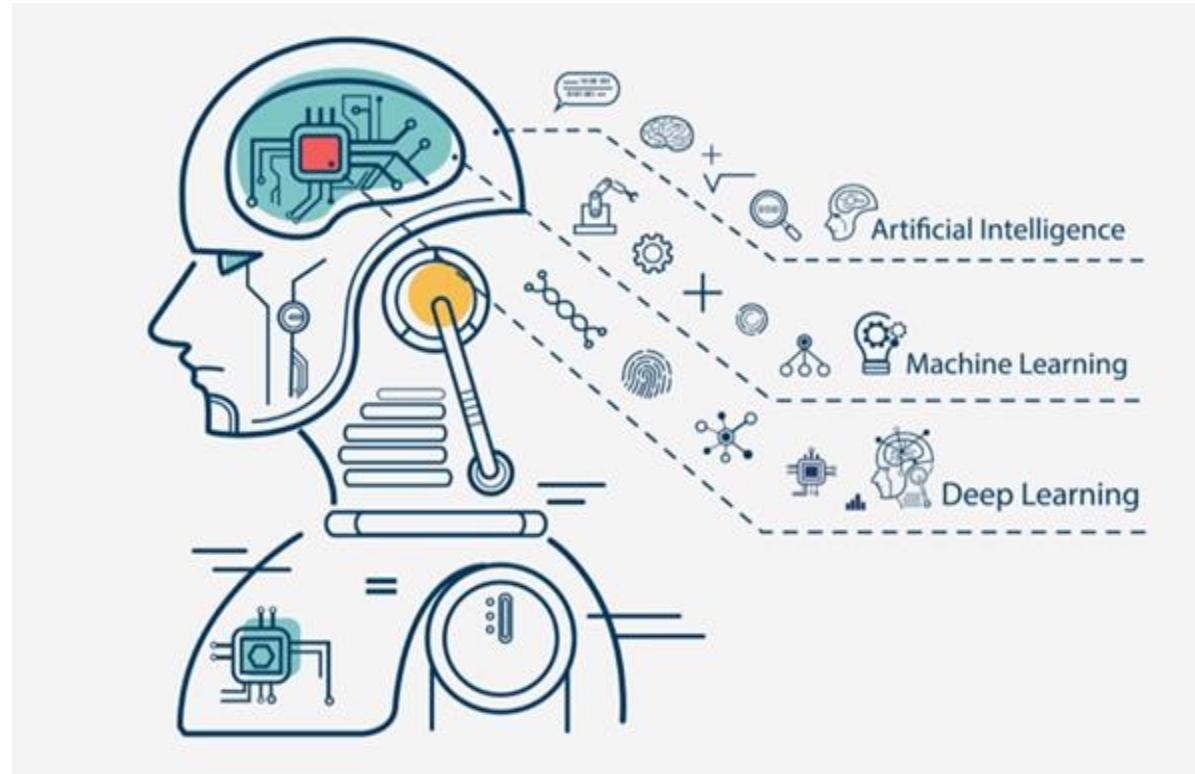
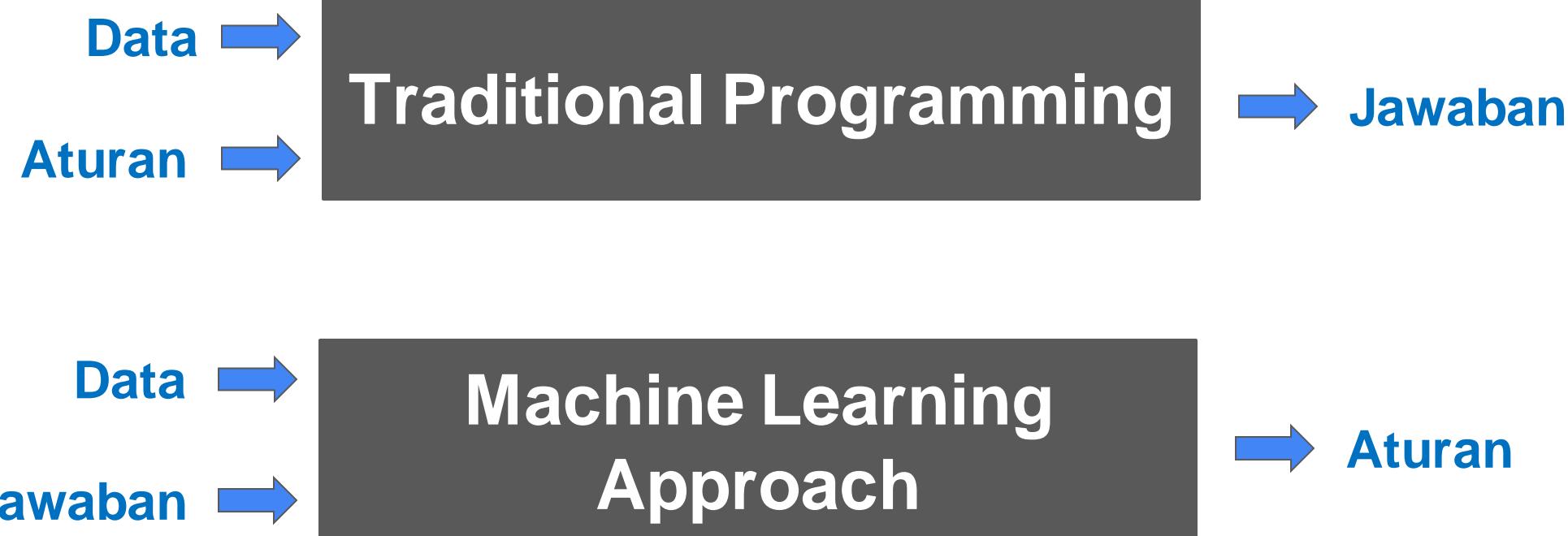


Image Source: Google Image

Bagaimana  
ML Approach  
dapat dibedakan dari  
Traditional Programming?

# Traditional Programming vs Machine Learning Approach



## Traditional Approach vs Modern Approach

Experiment

Data + Role

Process

Answers

Traditional Approach



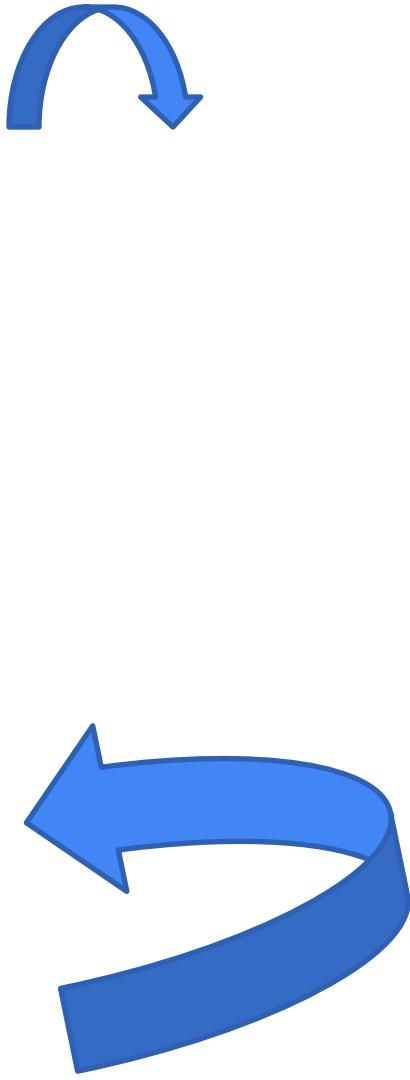
Modern Approach

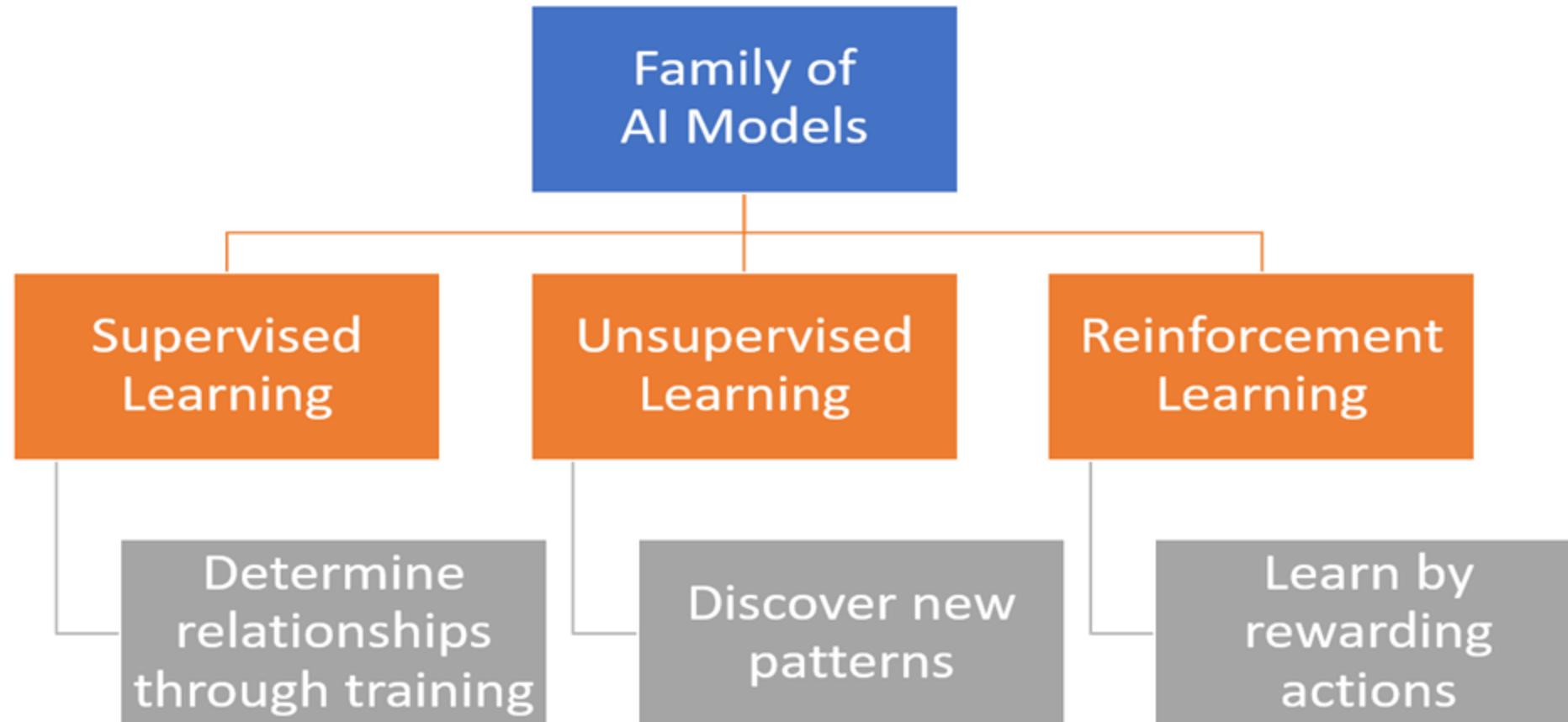
Experiment

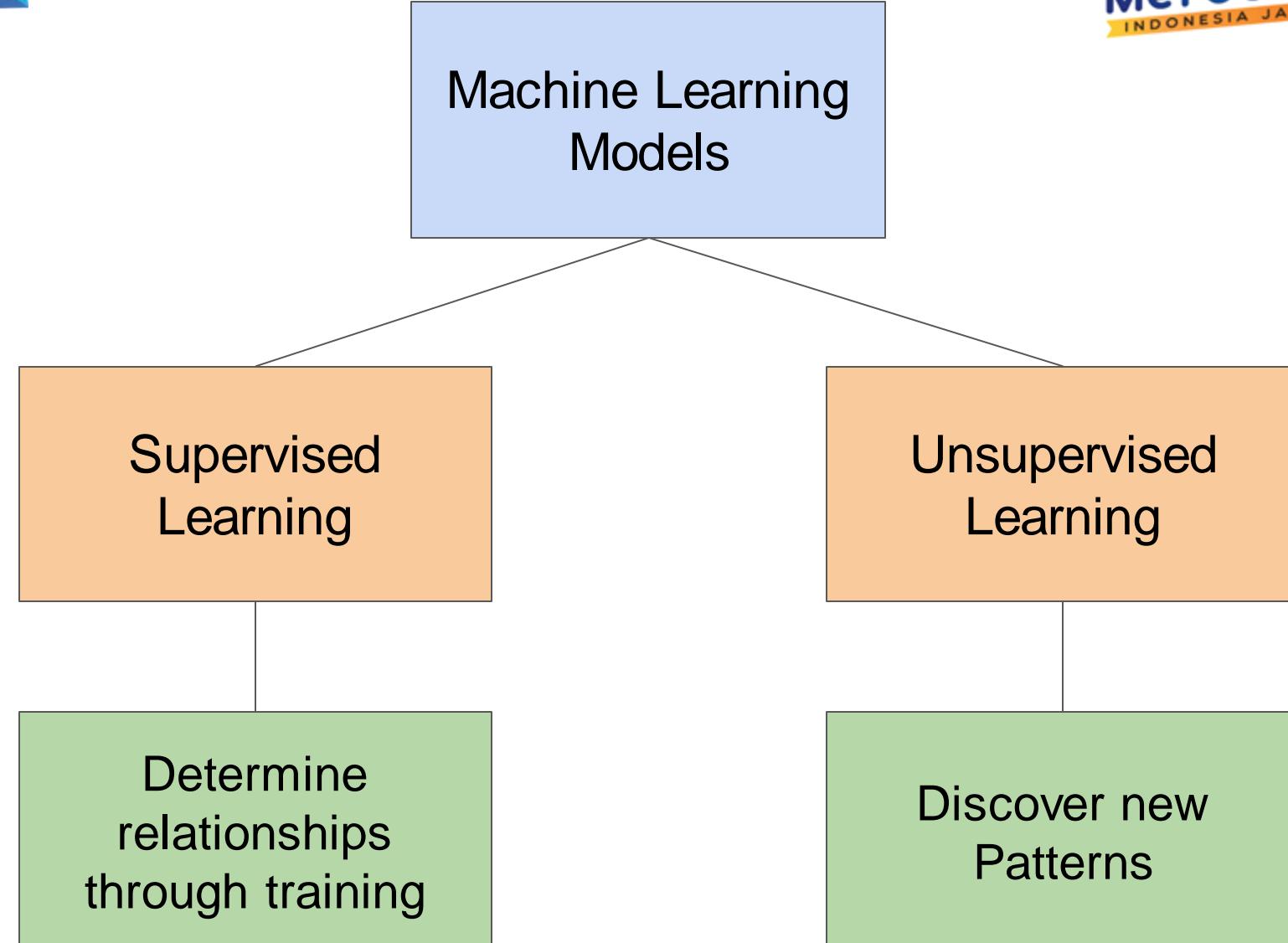
Data + Answers

Process

Role



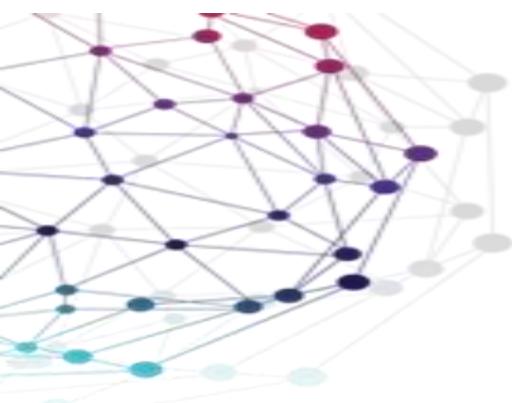




# Supervised Learning vs Unsupervised Learning



# WHAT IS MACHINE LEARNING ?

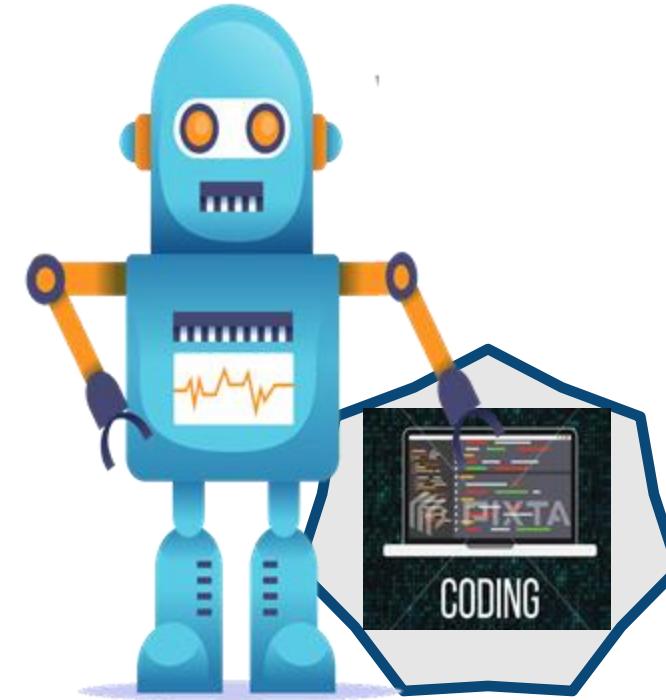


# Human v Machine



**MANUSIA**

Belajar dari pengalaman



**MESIN**

Hanya mengikuti “Instruksi” yang diberikan

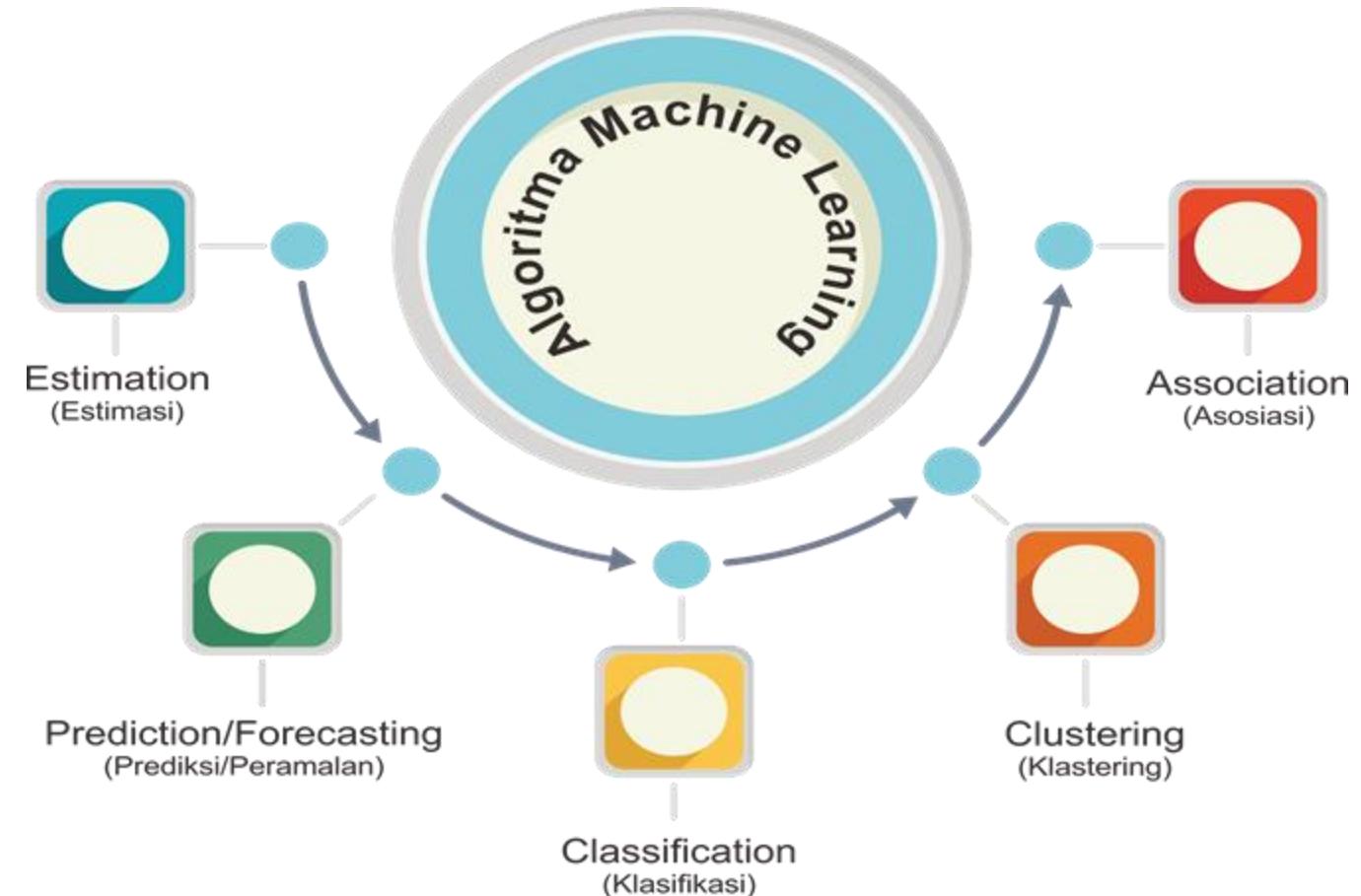
# Segmentasi ML

## Supervised Learning

- Estimation
- Prediction/Forecasting
- Classification

## Unsupervised Learning

- Clustering
- Association



# 7 Langkah Dalam ML



Gathering  
Data



Preparing  
Data



Choosing  
a Model



Training  
Model



Evaluation



Tuning  
Hyperparameter



Prediction

# ML Life Cycle

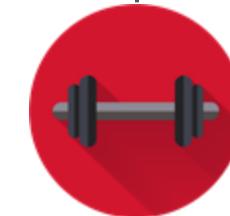
Tentukan Tujuan



Siapkan Data



Latih Model



Integrasikan  
Model



Kumpulkan Data



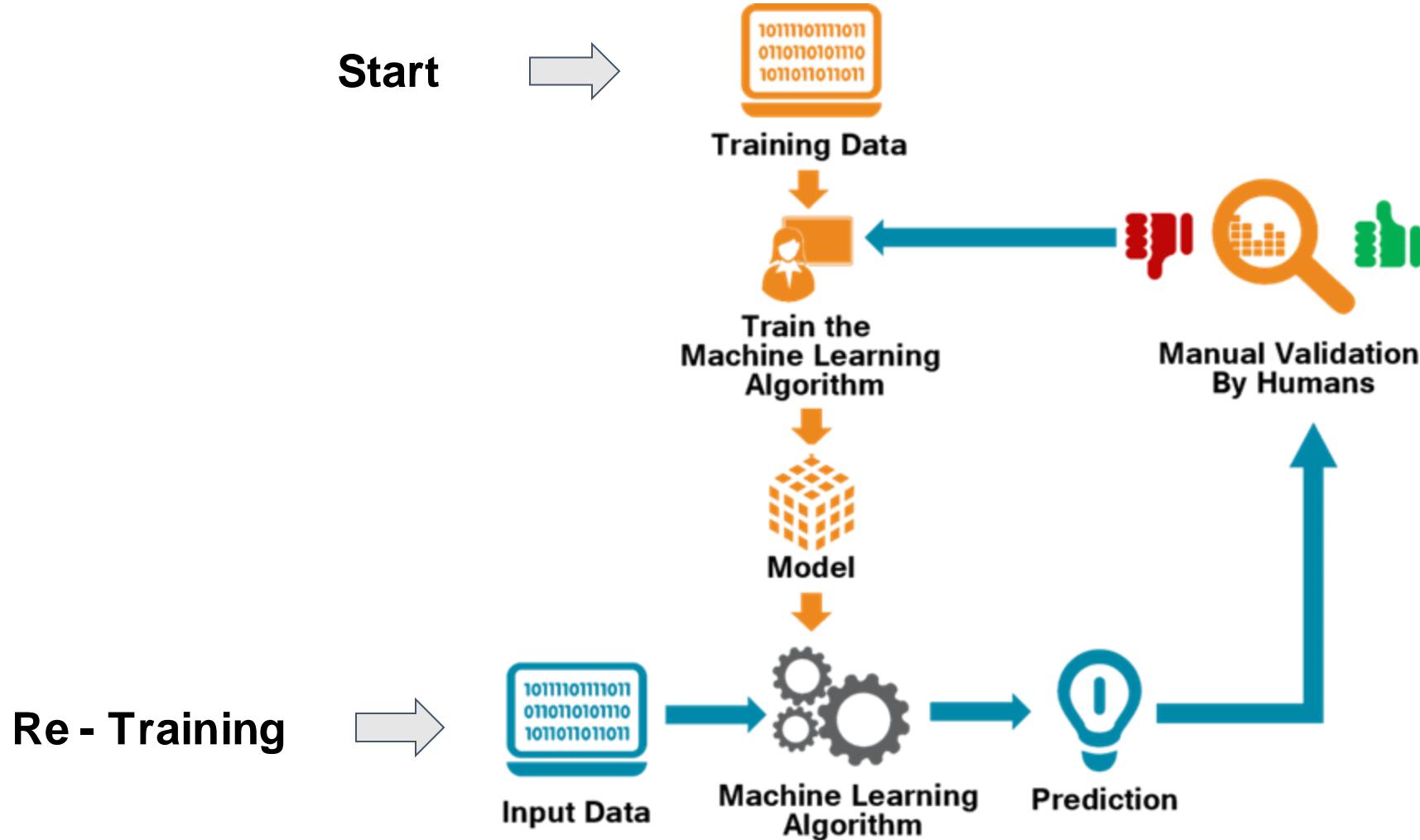
Pilih Algoritma



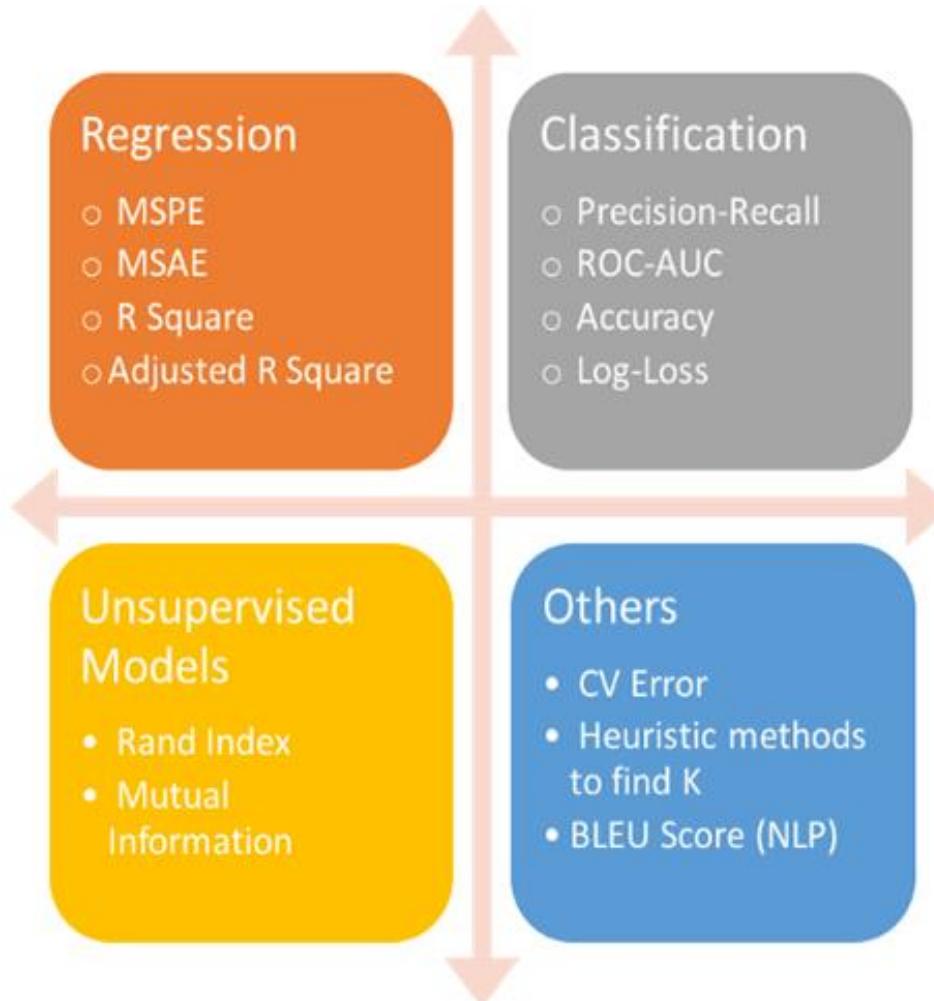
Tes Model



# ML Workflow



# Metrics Evaluation



# Kelebihan dan Kekurangan ML

## Kelebihan

- Mudah dalam mengidentifikasi trend dan pola pada data
- Tidak perlu campur tangan manusia
- Pengembangan dapat dilakukan secara berkelanjutan
- Mampu menangani data multidimensi dan multi-variasi
- Aplikasi/implementasi yang luas

## Kekurangan

- Rentan terhadap kesalahan
- Akuisisi data
- Waktu dan sumber daya
- Hasil bersifat interpretasi (tafsiran)

# Refleksi

# Supervised Learning

**Supervised Learning** umumnya digunakan untuk menemukan pola dalam data masukan yang diberi label sehingga memungkinkan kita menghasilkan data keluaran yang benar secara efektif.

## Ciri-ciri:

- Data training telah diberi label
- Algoritma memprediksi output dari input, contoh algoritma:
  - Klasifikasi (memetakan masukan ke label keluaran)
  - Regresi (memetakan masukan ke keluaran berkelanjutan)

# Supervised Learning Workflow

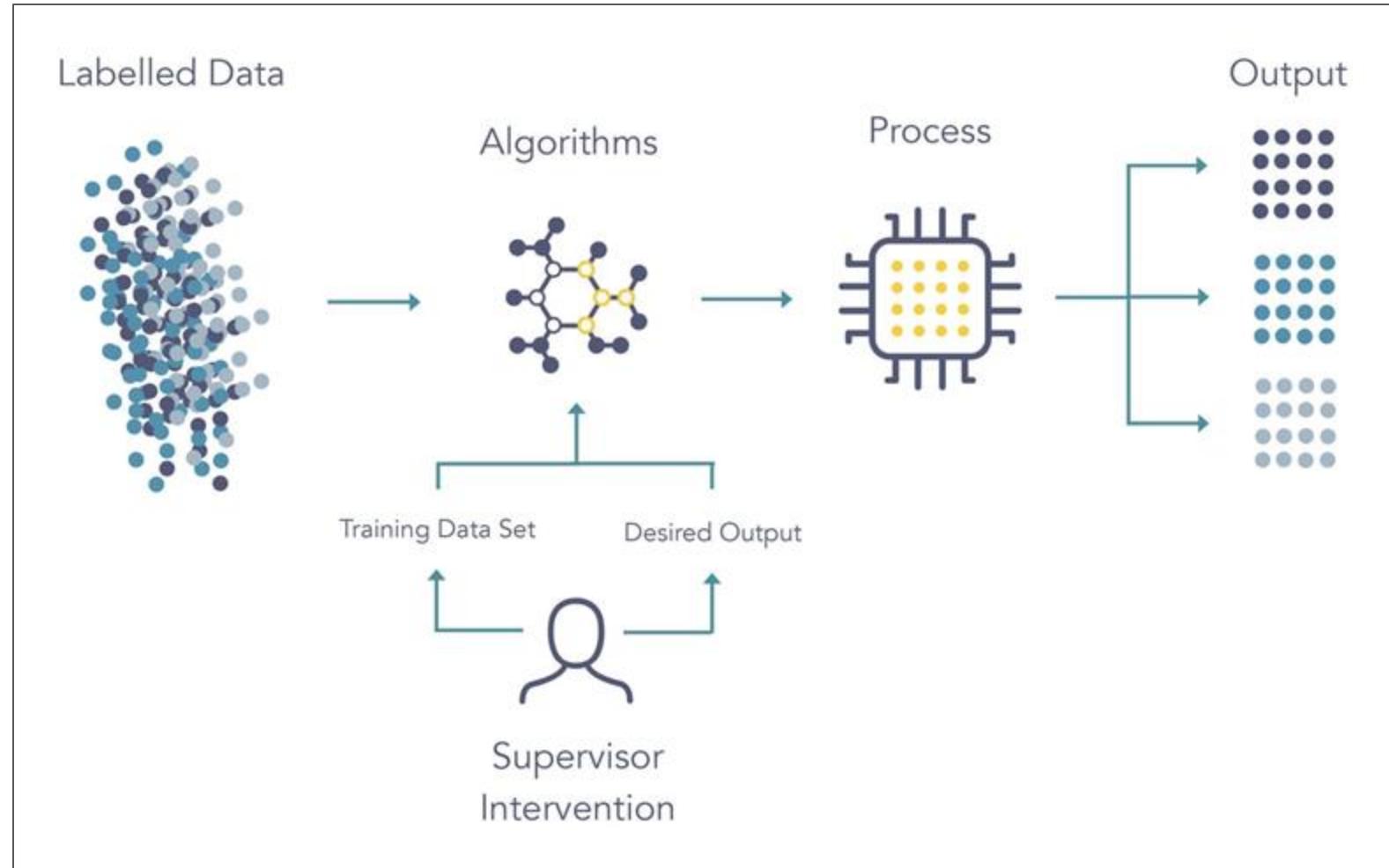
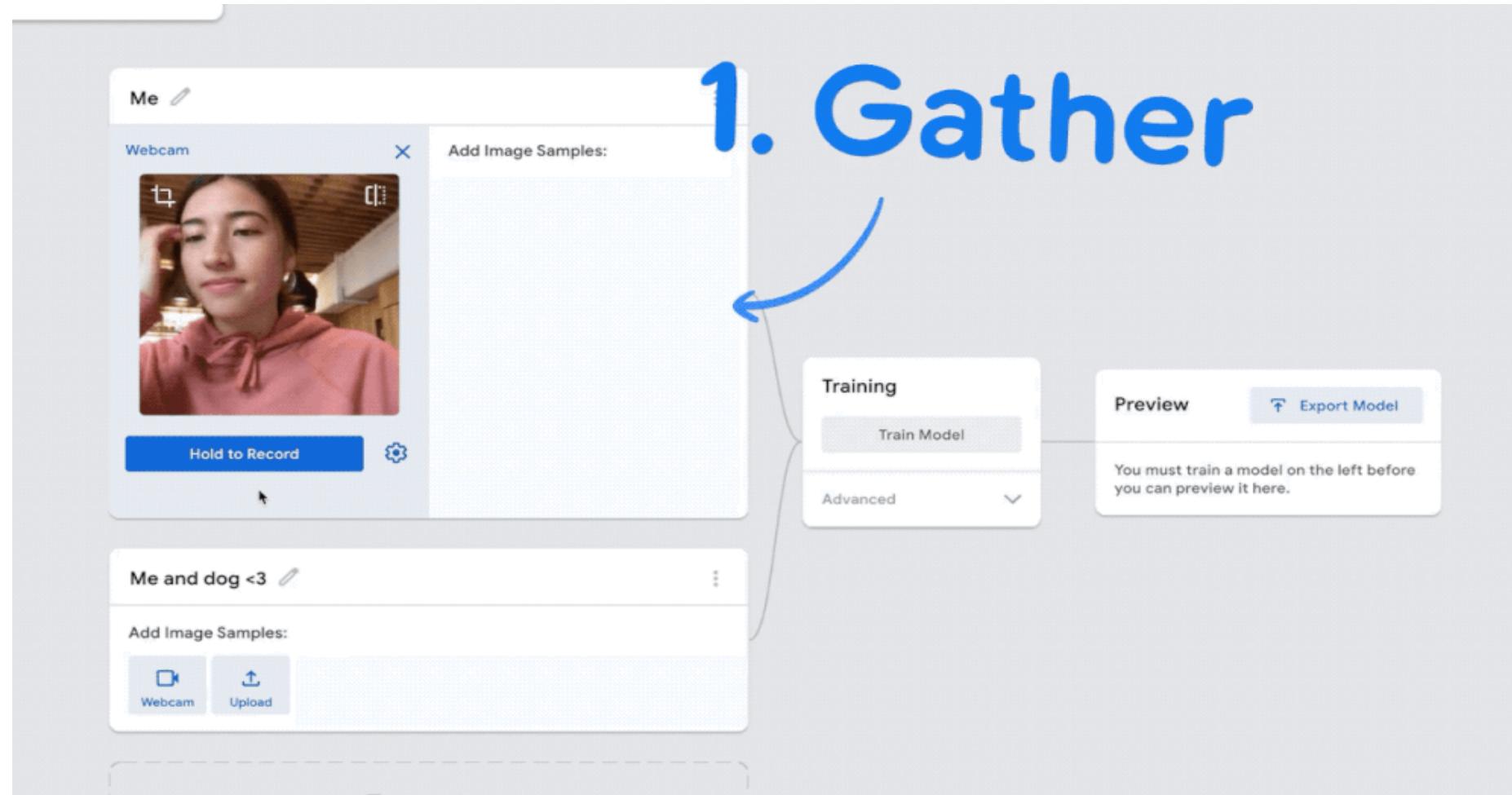


Image Source: <https://www.logpoint.com/sv/blogg-sv/explained-siemply-machine-learning/>

# Supervised Learning Experiment



<https://teachablemachine.withgoogle.com/>

# Unsupervised Learning

**Unsupervised Learning** umumnya digunakan untuk mempelajari struktur karakteristik data kita tanpa menggunakan label yang disediakan secara eksplisit.

## Ciri-ciri:

- Data training **tidak berlabel**
- Algoritma mempelajari struktur karakteristik dari data masukan, contoh algoritma:
  - Clustering (mempelajari hubungan antara fitur individu)
  - Dimensional Reduction (metode untuk mengurangi fitur)

# Unsupervised Learning Workflow

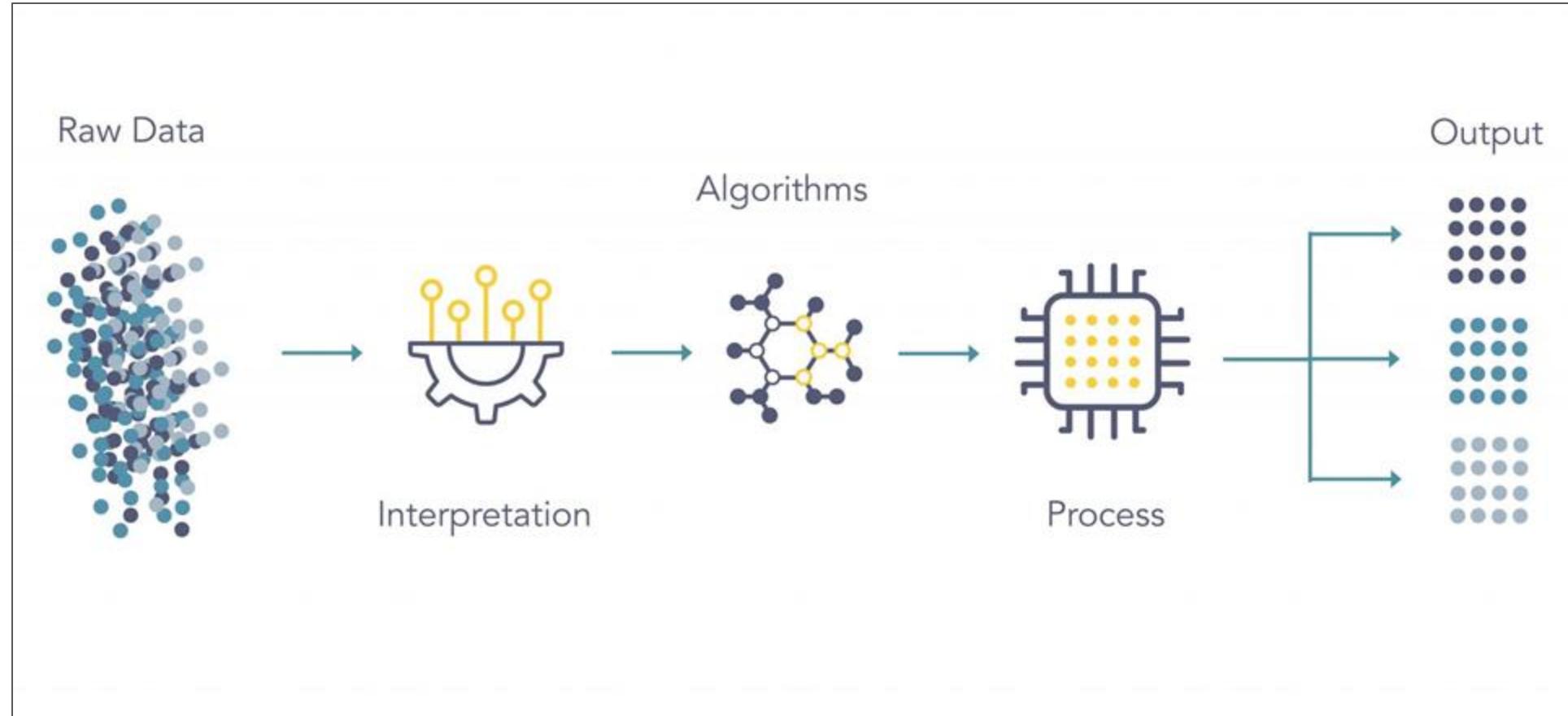
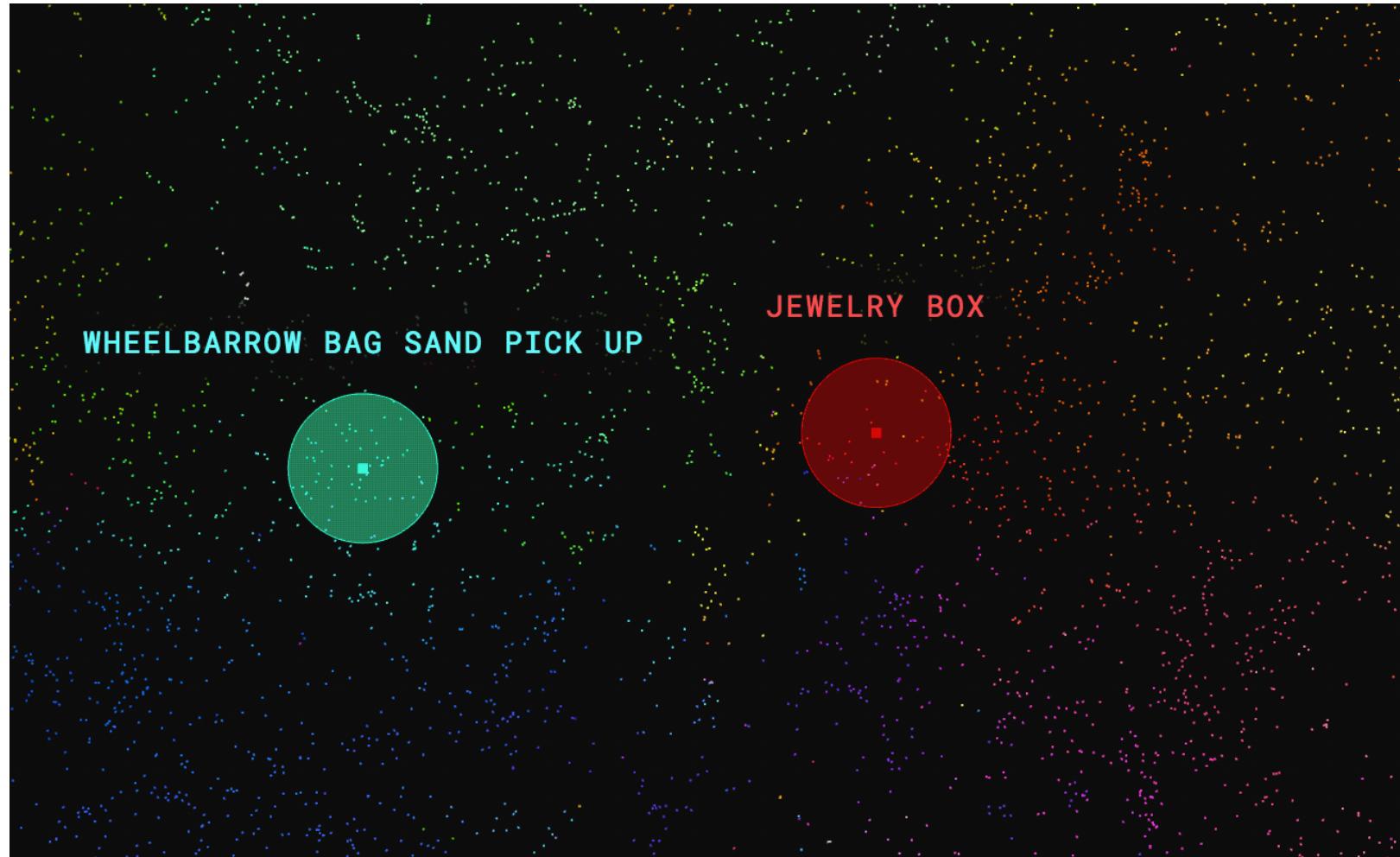


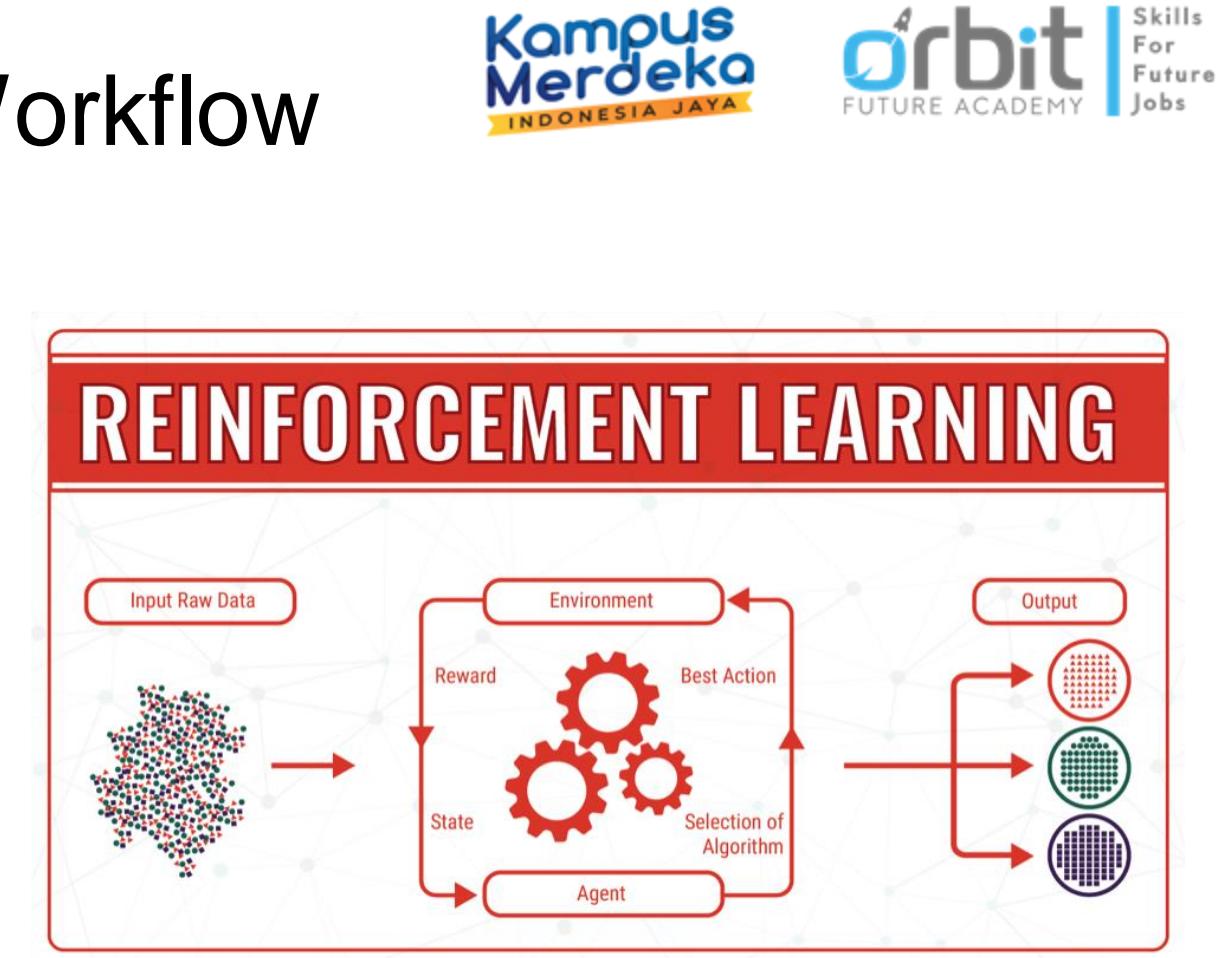
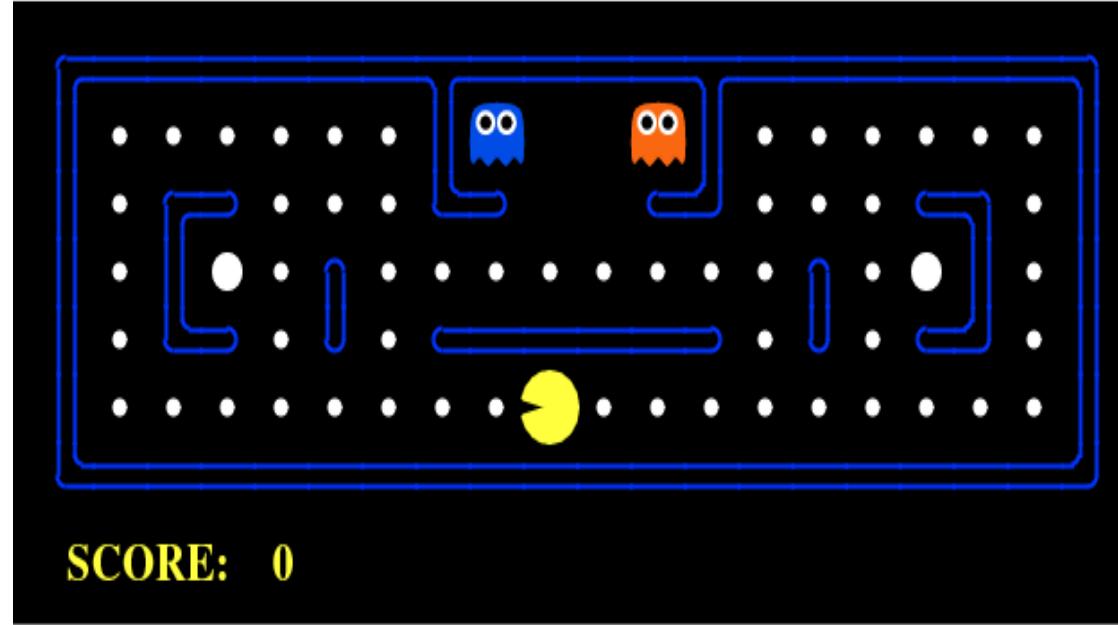
Image Source: <https://www.logpoint.com/sv/blogg-sv/explained-siemply-machine-learning/>

# Unsupervised Learning Experiment

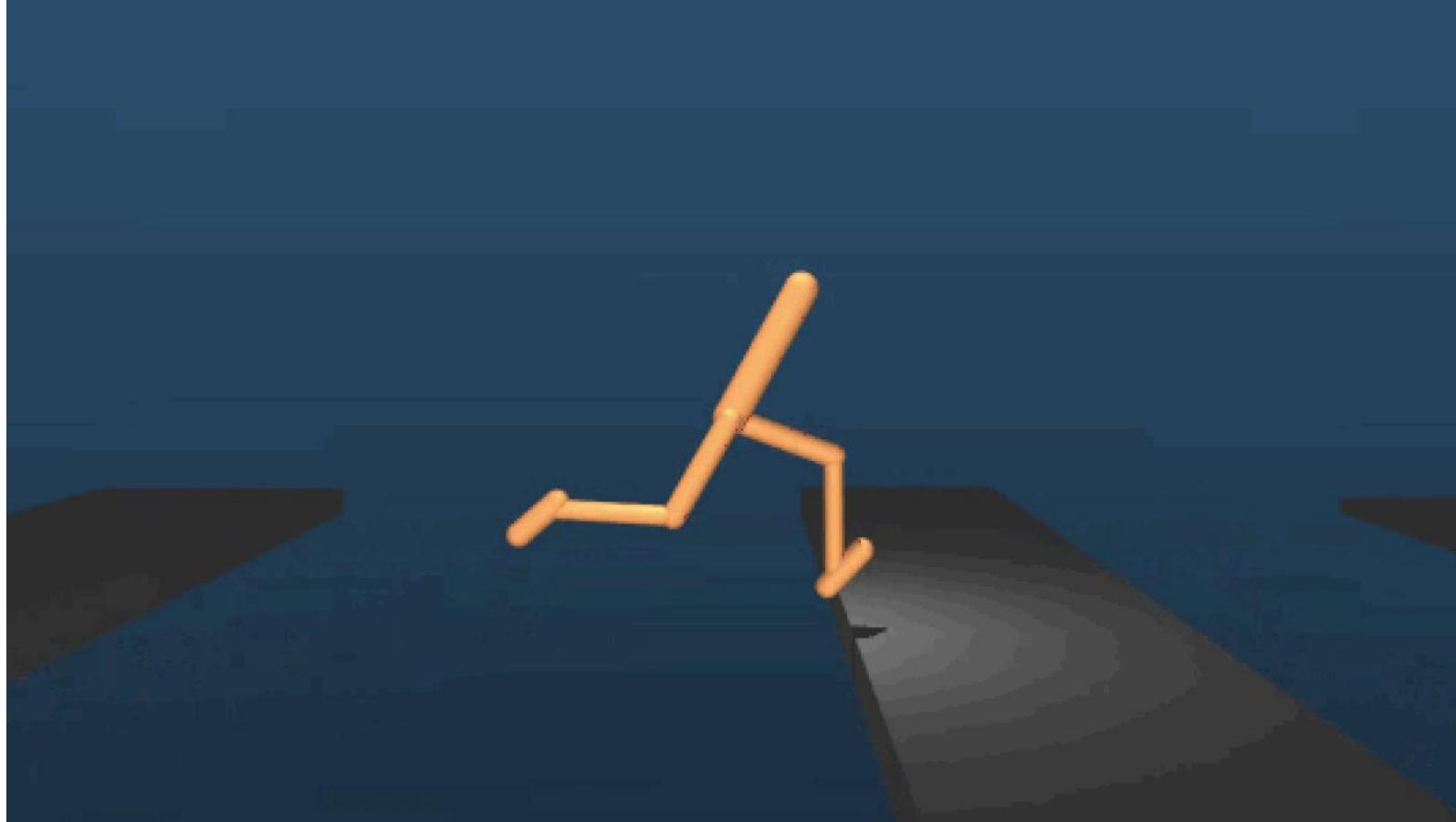


<https://experiments.withgoogle.com/ai/drum-machine/view/>

# Reinforcement Learning Workflow



# Ragdoll; Reinforcement Learning Experiment



<https://keiwan.itch.io/evolution>



## Konsep Machine Learning

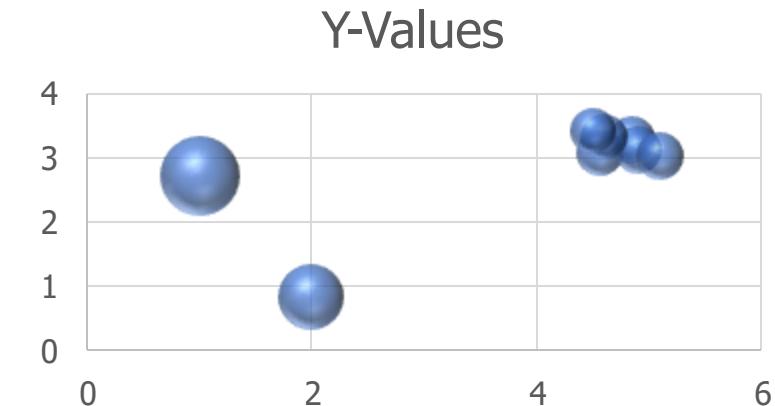
# Supervised Learning

- Kita tahu apa yang sedang coba untuk kita prediksi. Kita menggunakan sejumlah data dan model machine learning kita mampu untuk men generate prediksi yang kita inginkan.
- Contoh: Memprediksi penjualan rumah berapa tahun kedepan. Mengidentifikasi gender berdasarkan foto (computer vision)

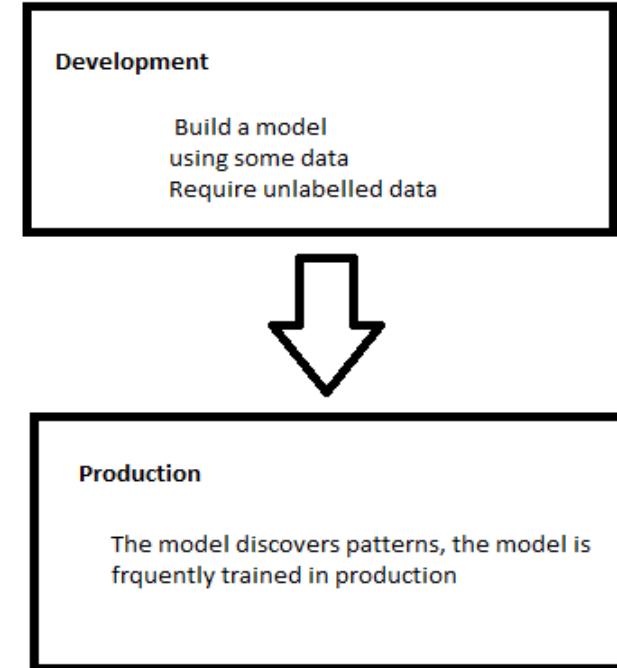
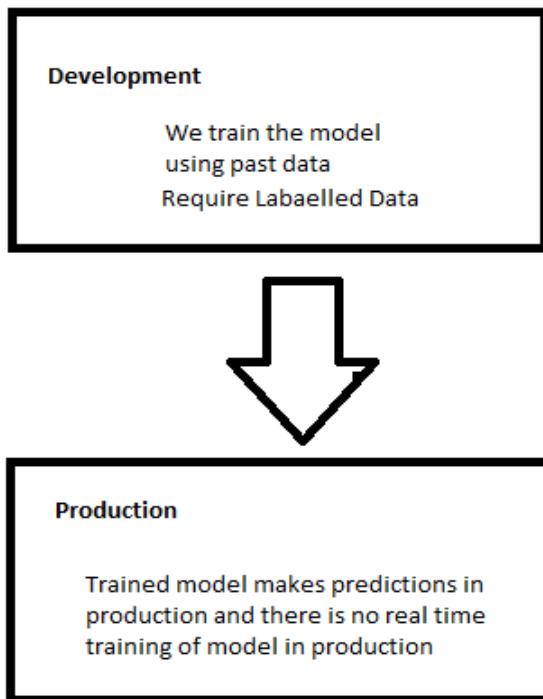


# Unsupervised Learning

- Kita tidak tahu apa yang kita coba prediksi. Kita mencoba untuk mengidentifikasi beberapa pola alami dalam data yang mungkin informatif.
- Contoh: Mengidentifikasi “clusters” kelas berdasarkan data dari mahasiswa program AI Mastery



# Supervised Learning

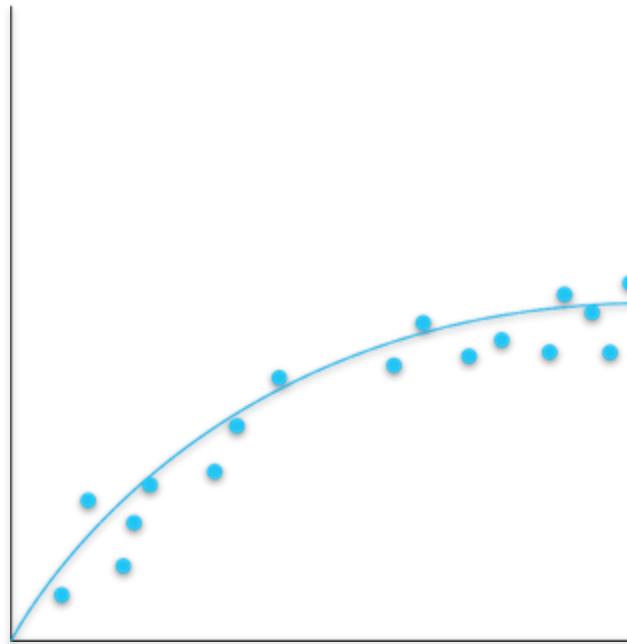


**Supervised  
Learning**

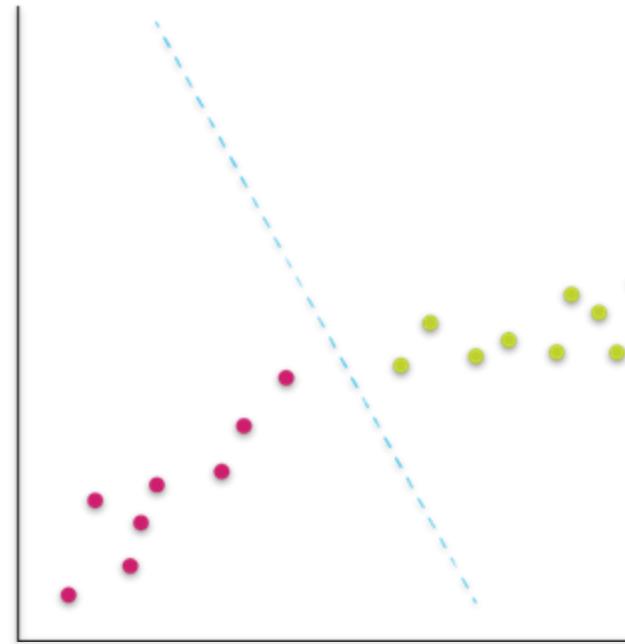
**Unsupervised  
Learning**

# The Common ML Algorithms

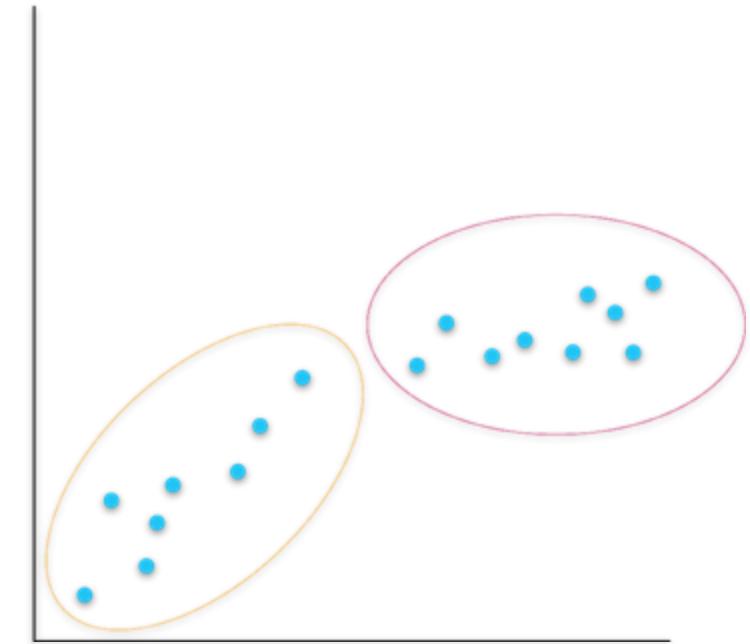
Regression



Classification

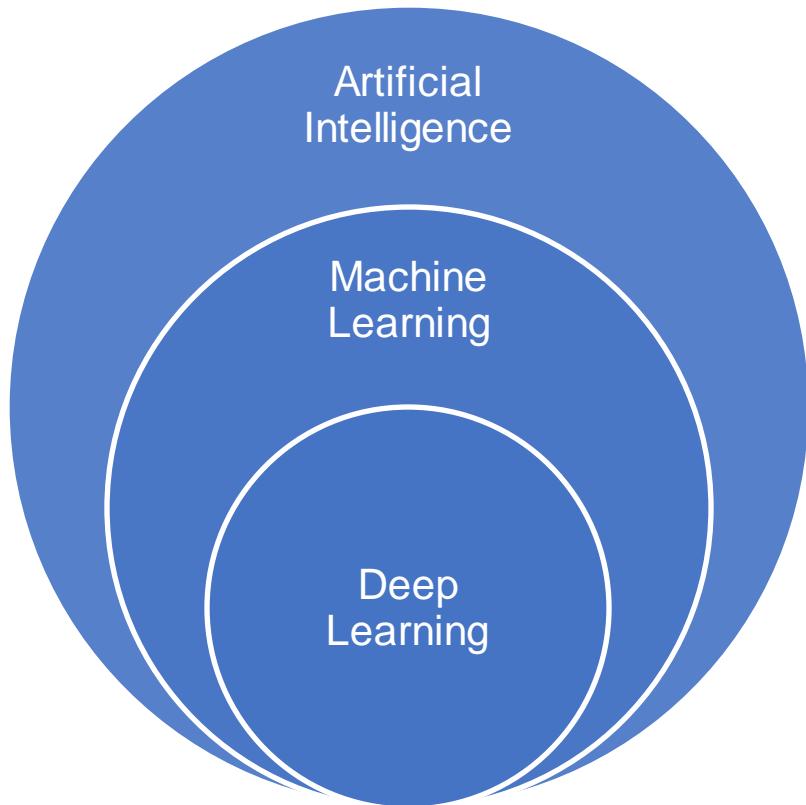


Clustering



Source: <https://www.moogsoft.com/blog/aiops/understanding-machine-learning-aiops-part-2/regression-class-clustering-graph>

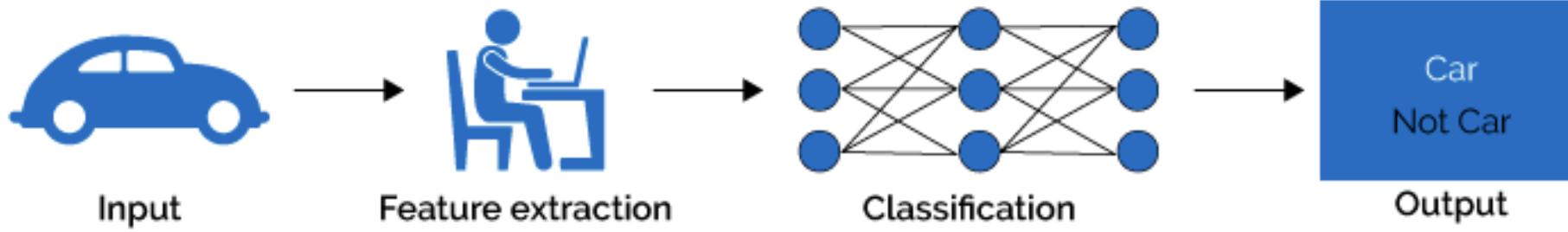
# Review Concept



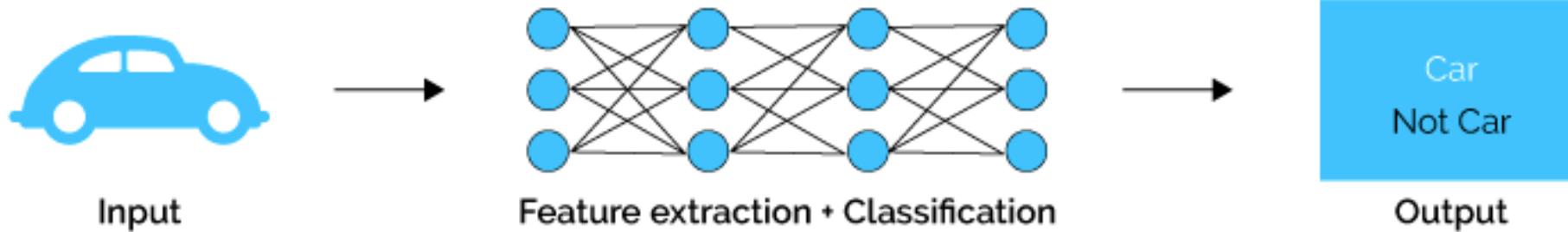
- **AI** - Teknik apa pun yang memungkinkan komputer meniru kecerdasan manusia.
- **Machine Learning** - Bagian dari AI yang memungkinkan mesin untuk mengerjakan tugas dengan lebih baik, dengan pengalaman.
- **Deep Learning** – Bagian dari ML yang memungkinkan software melatih dirinya sendiri untuk melakukan tugas dengan data dalam jumlah besar.

# Machine Learning vs Deep Learning

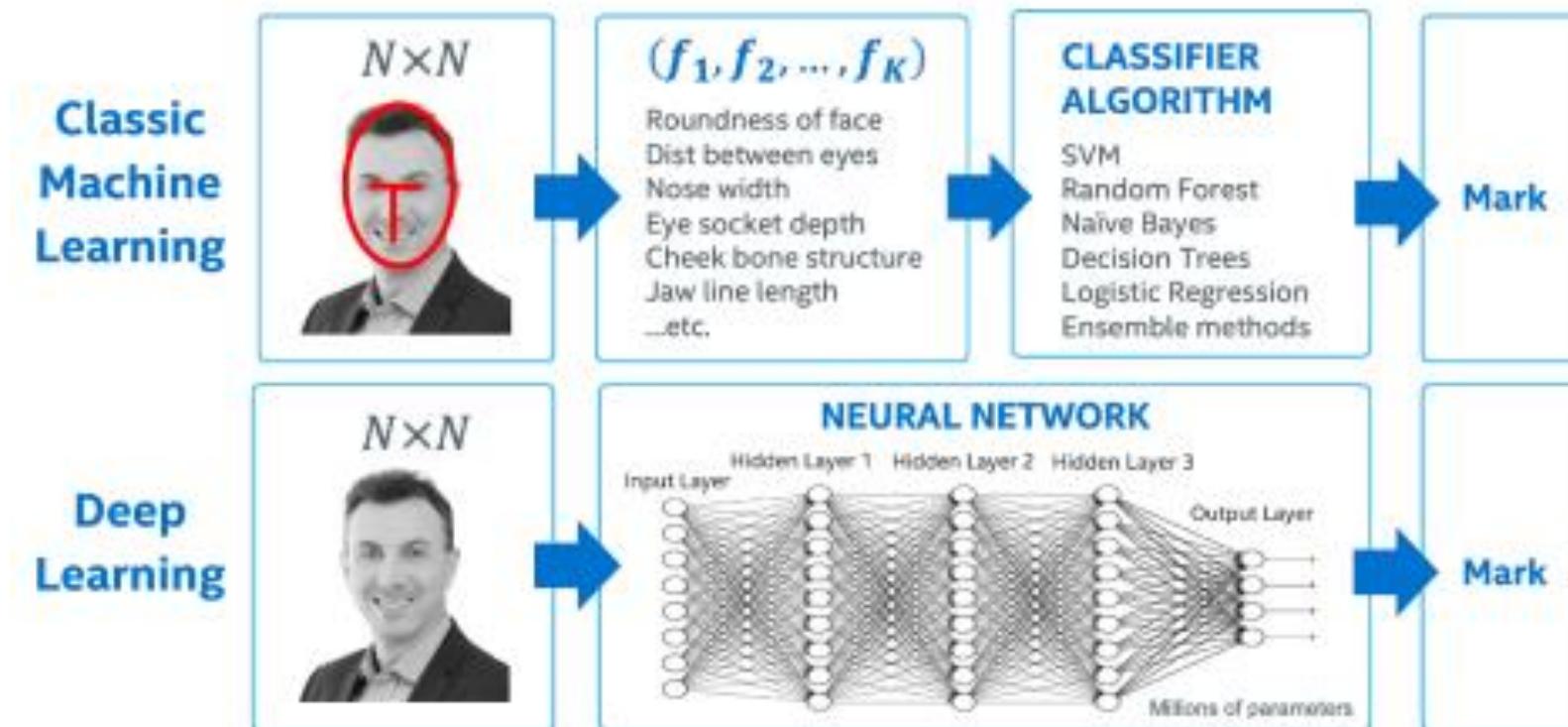
## Machine Learning



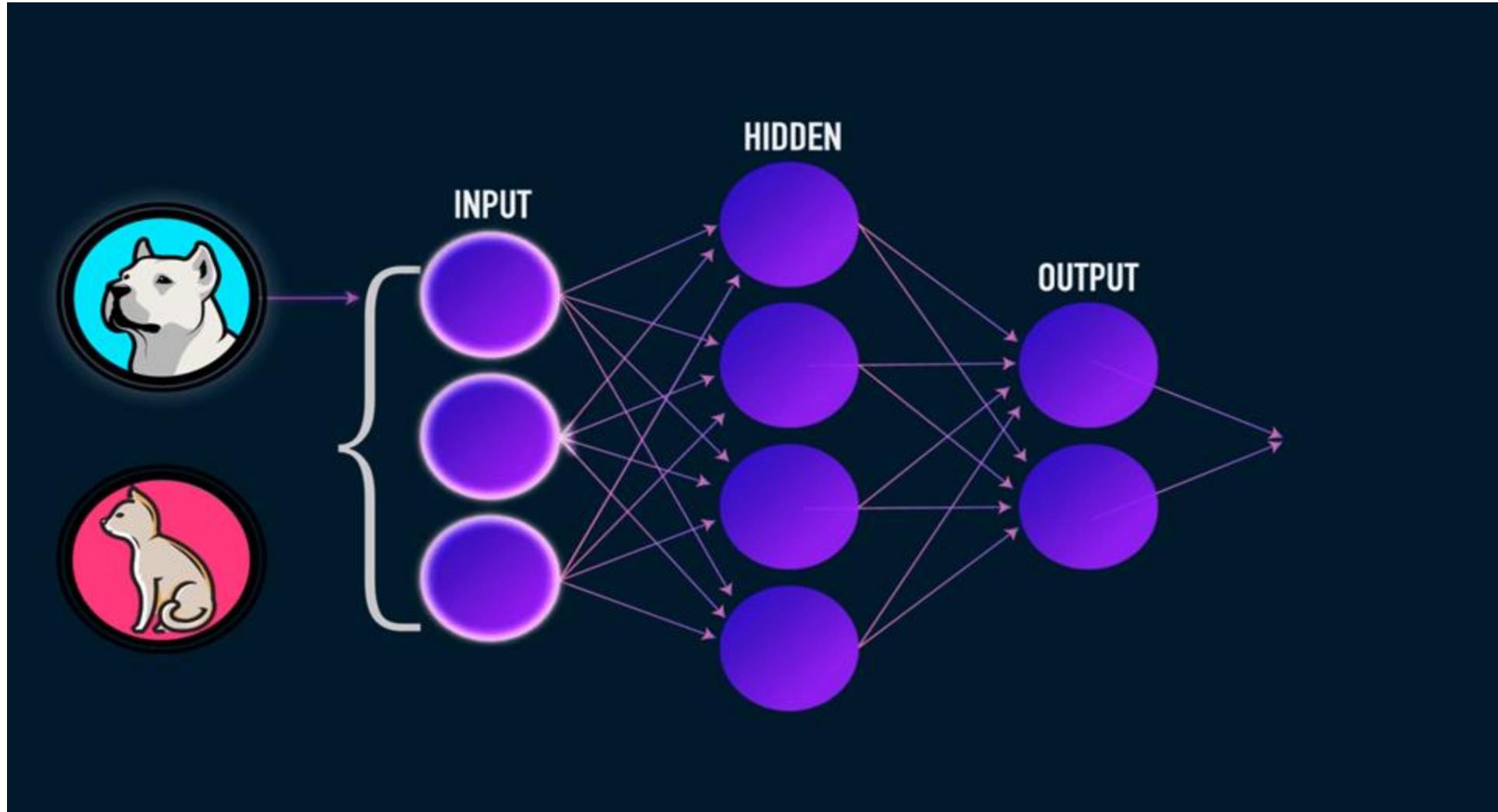
## Deep Learning



# Machine Learning vs Deep Learning



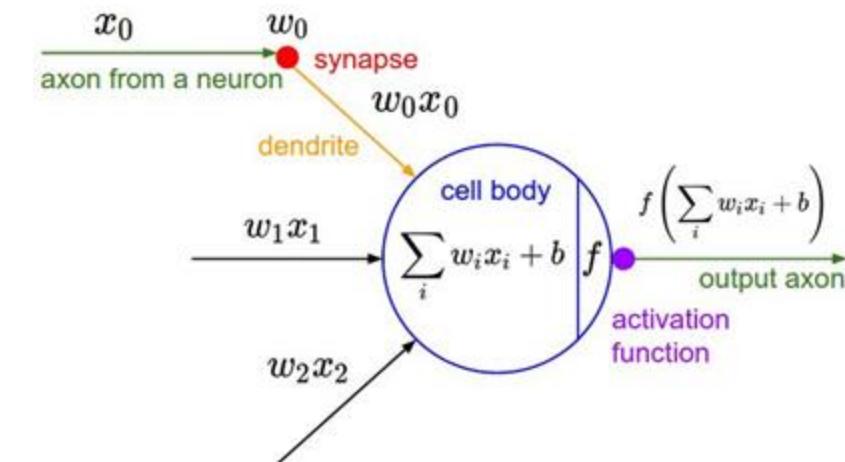
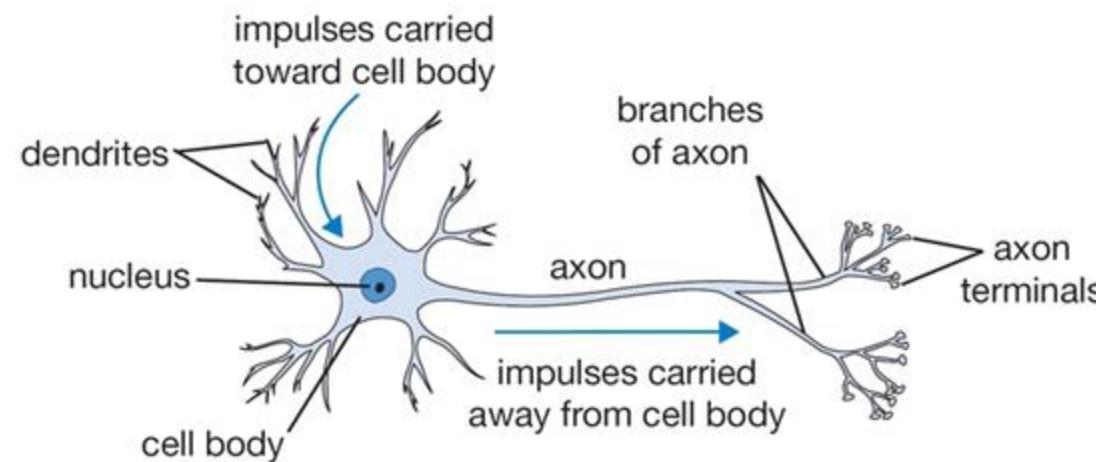
# NN; The Behind of Deep Learning and The Black Box



# Pengantar Artificial Neural Network

# Artificial Neural Networks

**Artificial Neural Network (ANN)** adalah teknologi AI yang meniru dan terinspirasi dari cara kerja *neuron* pada otak manusia.

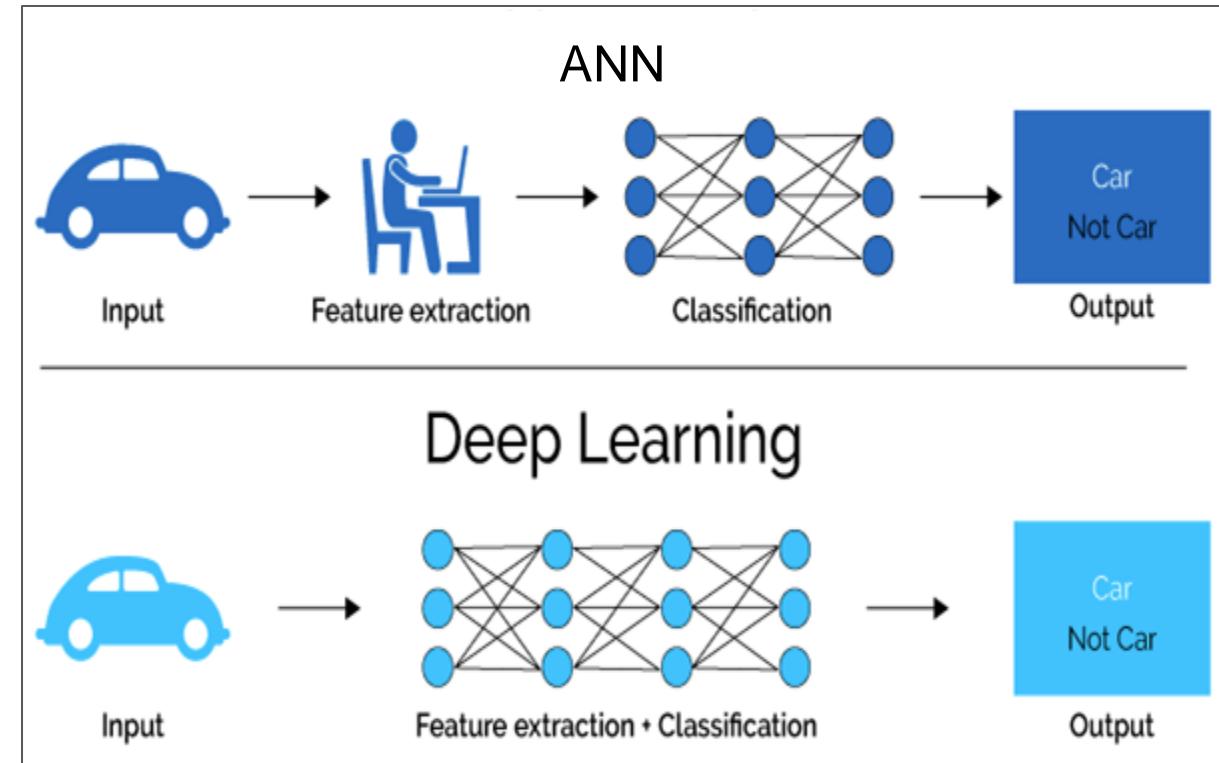


Images Source: Google Image

# Artificial Neural Networks

ANN merupakan cikal bakal dari teknologi *Deep Learning* yang tersusun atas banyak sekali lapisan *neuron* (*perceptron*).

*Deep Learning* mampu melatih dirinya sendiri dalam melakukan tugas dengan data yang besar.



Images Source: Google Image

# Contoh Penerapan Deep Learning

## 1. Face Recognition (pengenalan wajah)

Contoh: smartphone yang mampu mendeteksi wajah *user*

## 2. Speech Recognition (pengenalan ucapan)

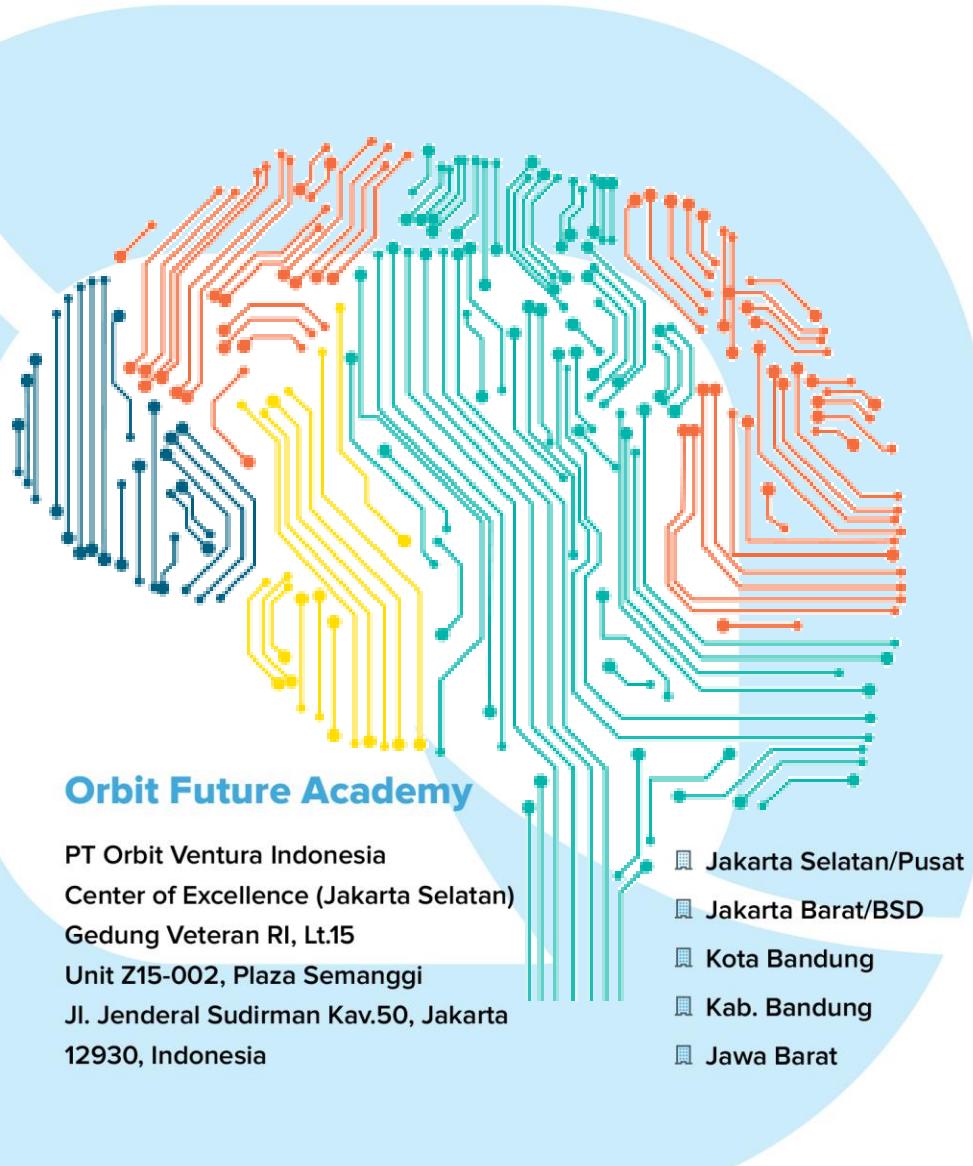
Contoh: memerintahkan smartphone untuk menyetel alarm

## 3. Character Recognition (pengenalan karakter)

Contoh: verifikasi tanda tangan seseorang

# References

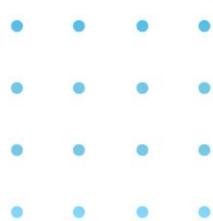
- “Artificial Intelligence and Machine Learning” by Zsolt Nagy
- "Machine Learning and Artificial Intelligence" by Ameet V Joshi



**Orbit Future Academy**

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat



Kampus  
Merdeka  
INDONESIA JAYA

**orbit**  
FUTURE ACADEMY | Skills  
For  
Future  
Jobs

**TERIMA KASIH**  
**THANK YOU**

**Hubungi Kami**

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

**Social Media**

- [Orbit Future Academy](#)
- [@OrbitFutureAcademyIn1](#)
- [OrbitFutureAcademy](#)
- [Orbit Future Academy](#)



# Module

## Artificial Intelligence Fundamental

### Section

#### Deep Learning

# Deep Learning

An introduction

# Why Deep Learning?

- Real applications and others
- But, why now?

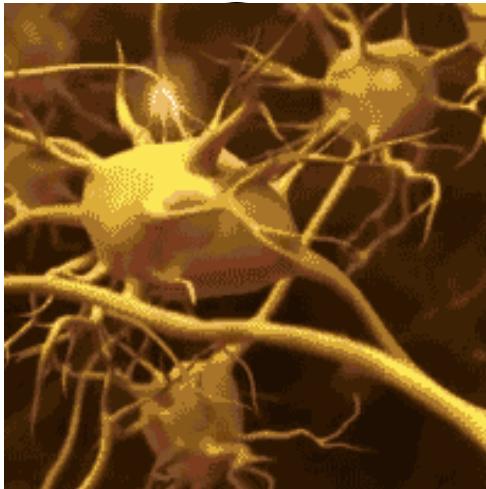


Coba, renungkan bagaimana kesadaran itu? Dan bagaimana struktur saraf pada otak kita bekerja?

- NN dimulai dari kisah bahwa manusia berusaha untuk meniru cara neuron pada manusia bekerja
- Dengan membuat tiruan, yaitu ANN
- Dikemudian hari ANN menjadi cikal bakal DL

Coba, renungkan bagaimana keberadaan kita sebagai manusia dan apa saja yang telah kita perbuat?

- 13.8 miliar tahun yang lalu alam semesta terbentuk
- 4.5 miliar tahun yang lalu bumi mulai tercipta
- 300.000 tahun yang lalu peradaban manusia modern
- Lalu bagaimana dengan “*a brief history of AI and DL*” ?



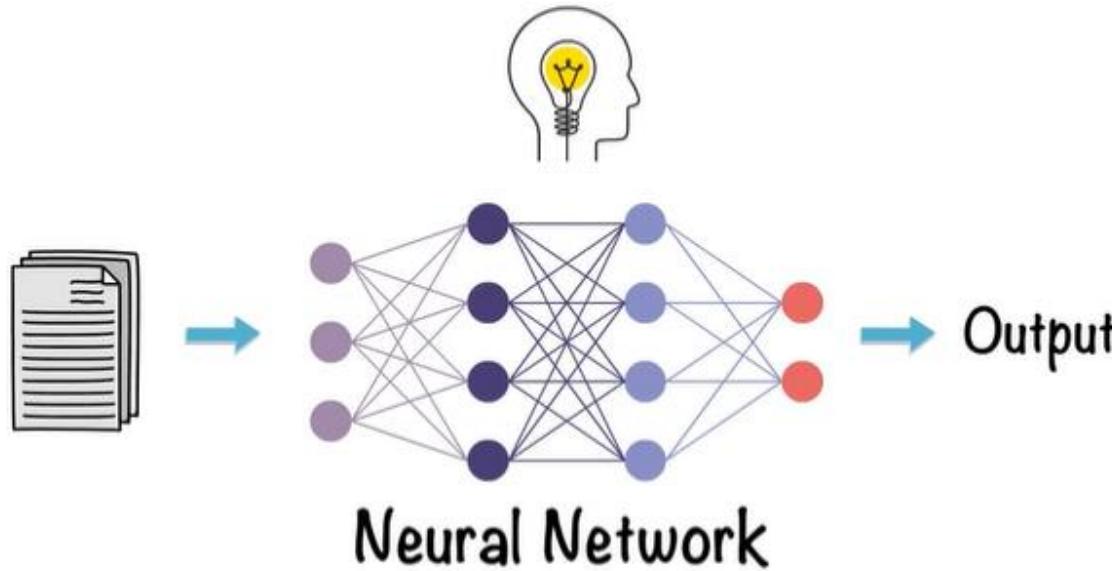
# Session I

## Pengenalan Deep Learning

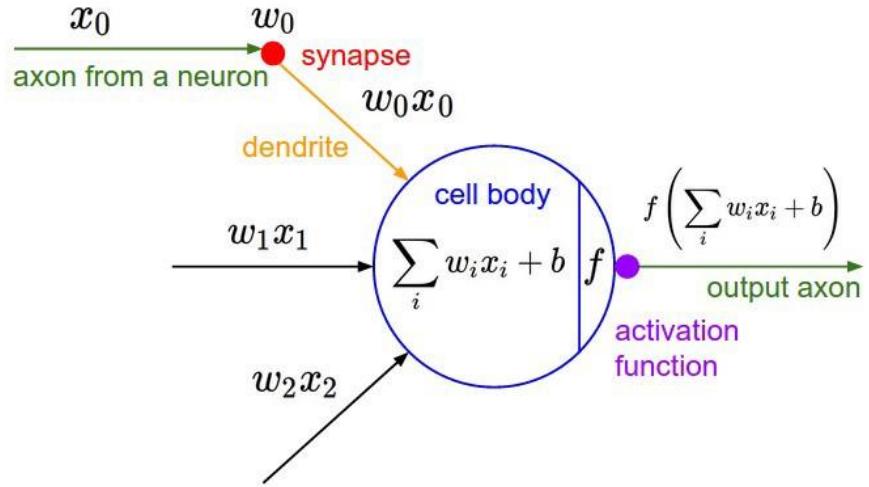
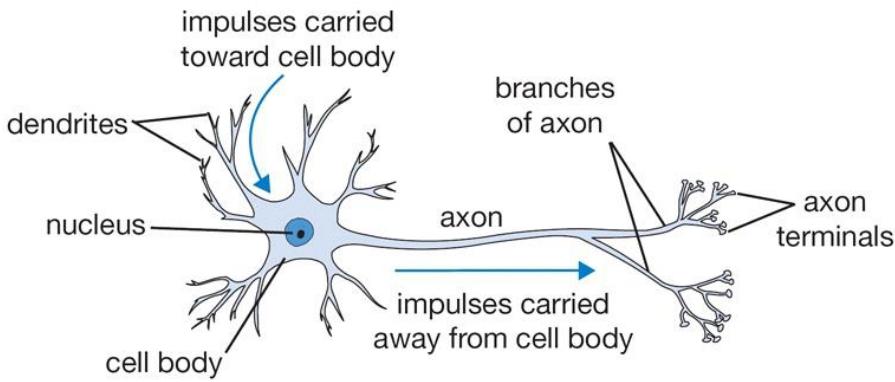
# Neural Network vs Deep Learning

Kampus  
Merdeka  
INDONESIA JAYA

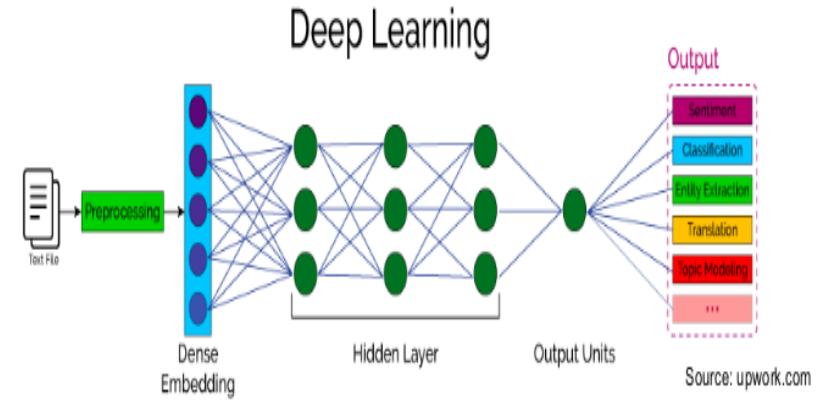
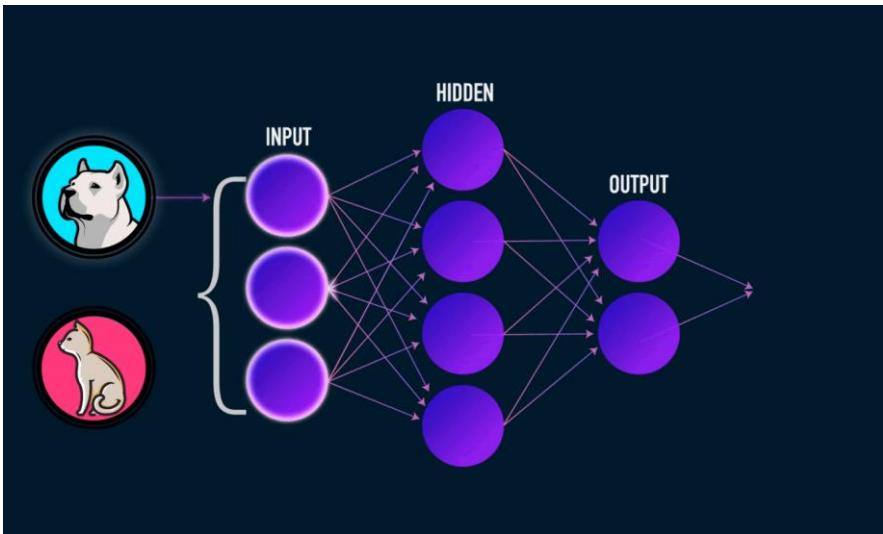
orbit  
FUTURE ACADEMY  
Skills  
For  
Future  
Jobs



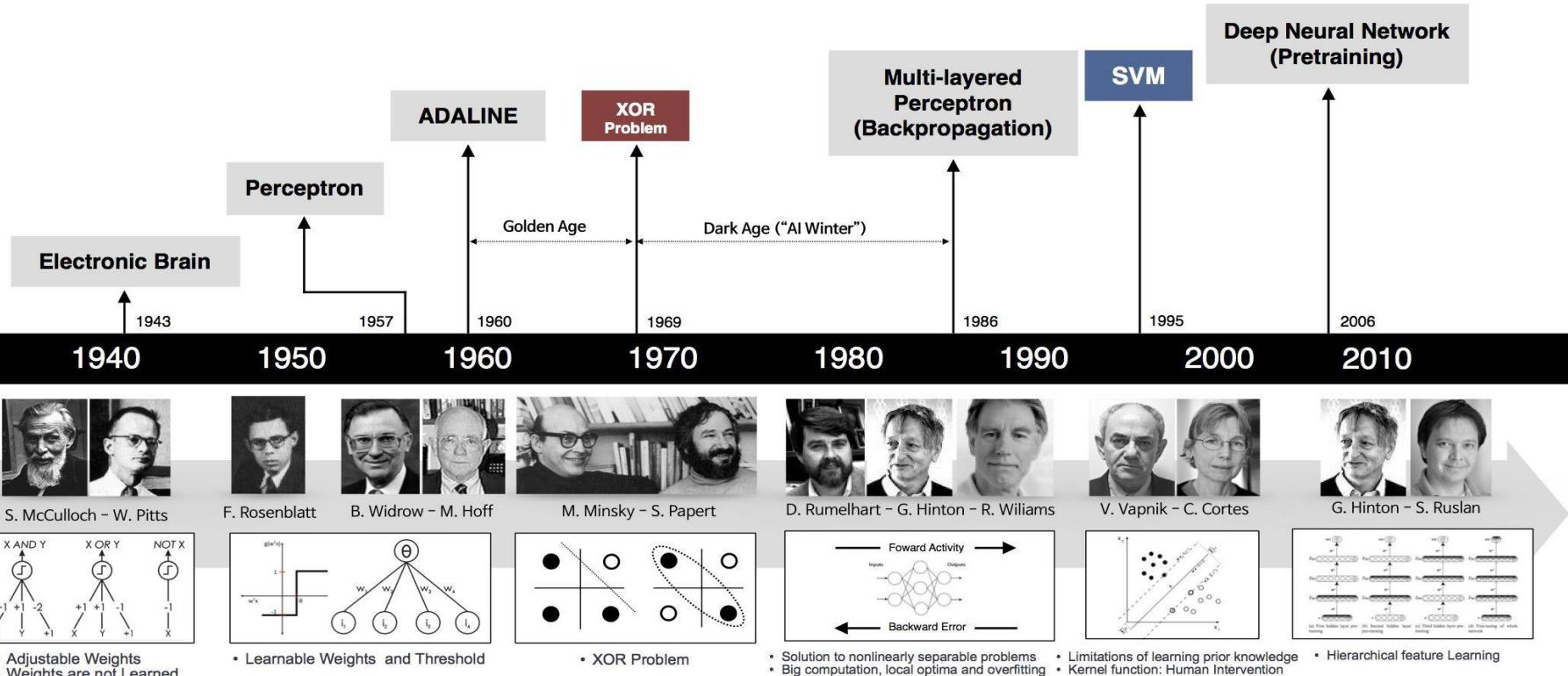
# Neural Network



# Deep Learning



# “Deep” Learning



# Perkembangan & Model DL



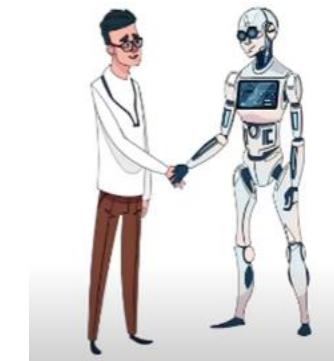
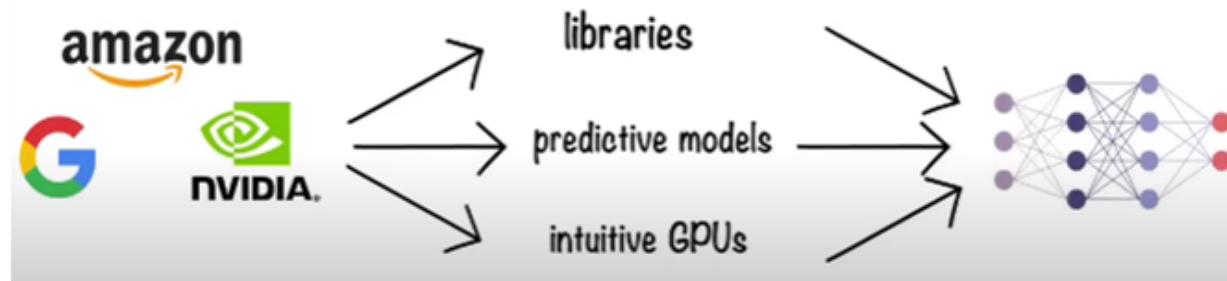
Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For Future Jobs



# Perkembangan Deep Learning

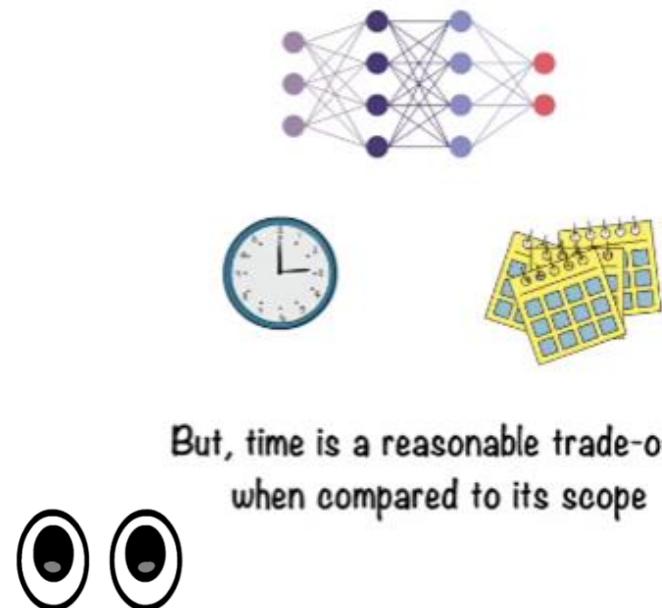
The growth in this field has been foreseen by the big names



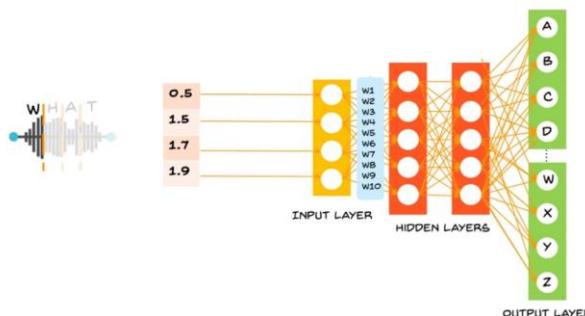
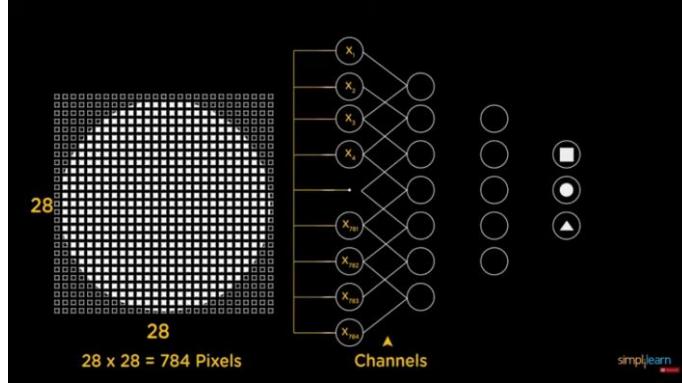
## From hours to weeks or even more!

Bergantung pada faktor-faktor seperti:

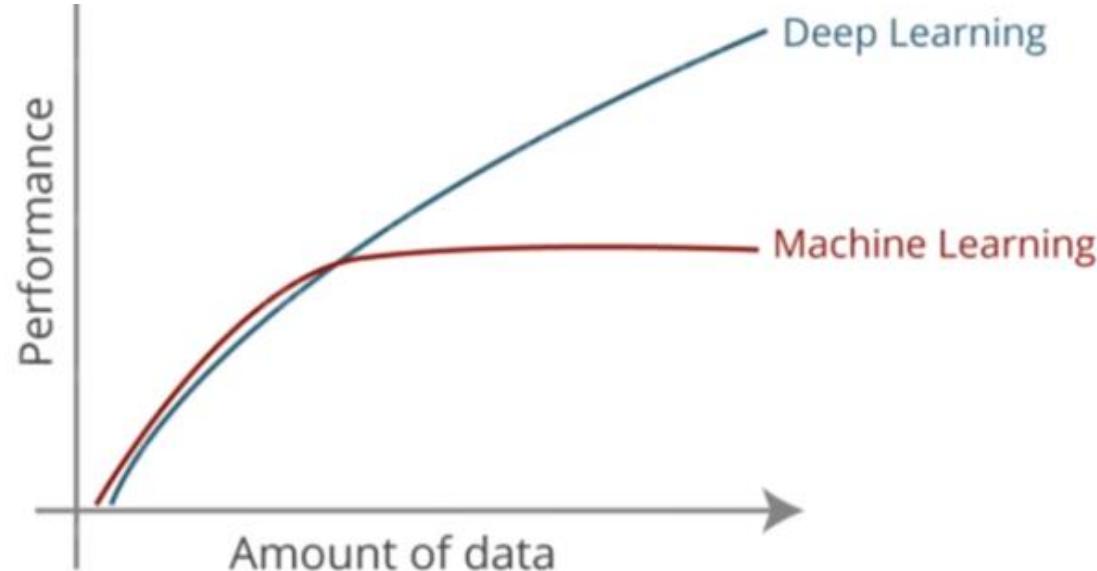
- perangkat keras yang tersedia,
- pengoptimalan,
- jumlah lapisan dalam jaringan saraf,
- arsitektur *neural network*,
- ukuran dataset, dan
- banyak lagi.



# Penggunaan Deep Learning

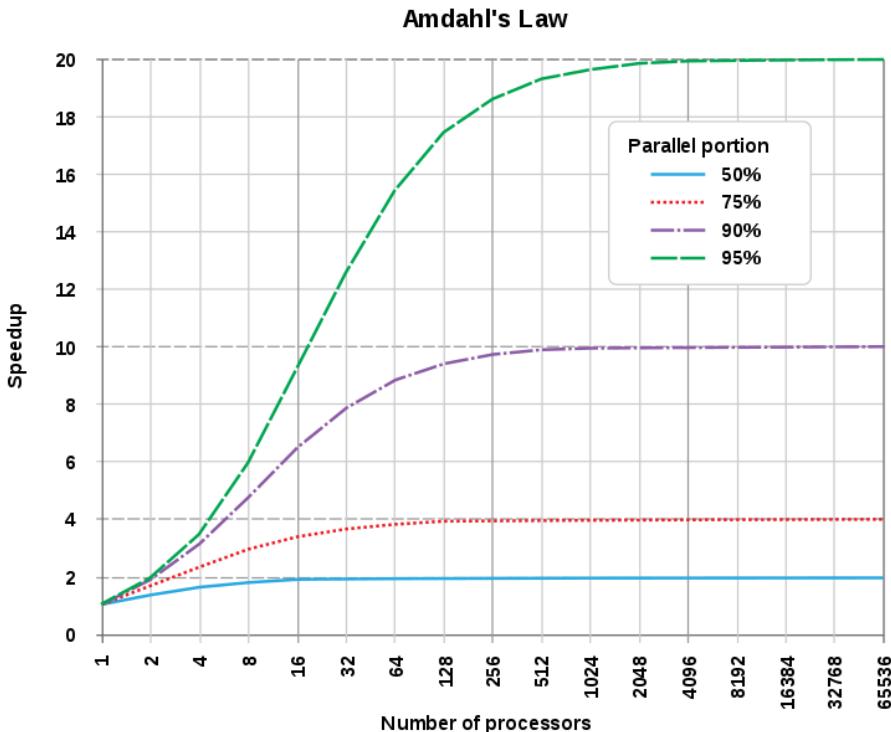


- Ketika mempunyai data tak terstruktur yang sangat besar dan banyak.
- Neural Network dapat mengolah data yang tidak memiliki label.
- Bekerja sangat baik dengan data berbentuk *speech*, *images*, dan *video*.



By: Dr. Andrew Ng

# Amdahl's Law for Parallel Computing Accelerates



Persamaan untuk menghitung *Speed-up parallel computing* menggunakan hukum Amdahl's

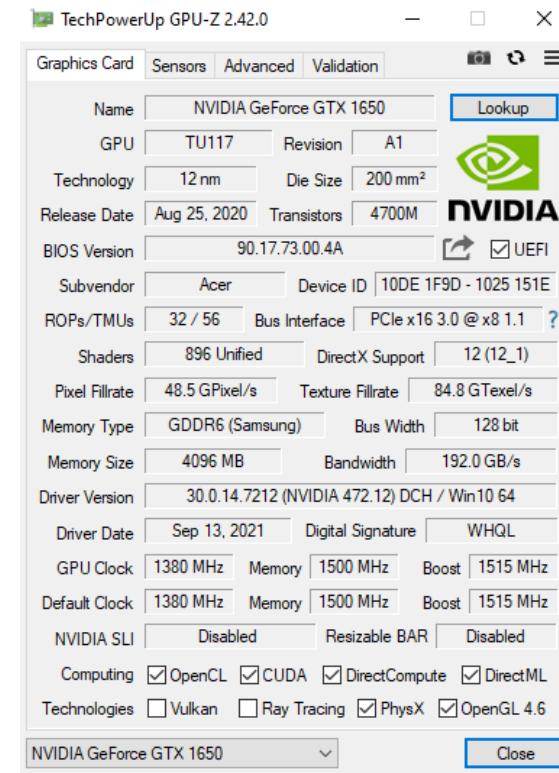
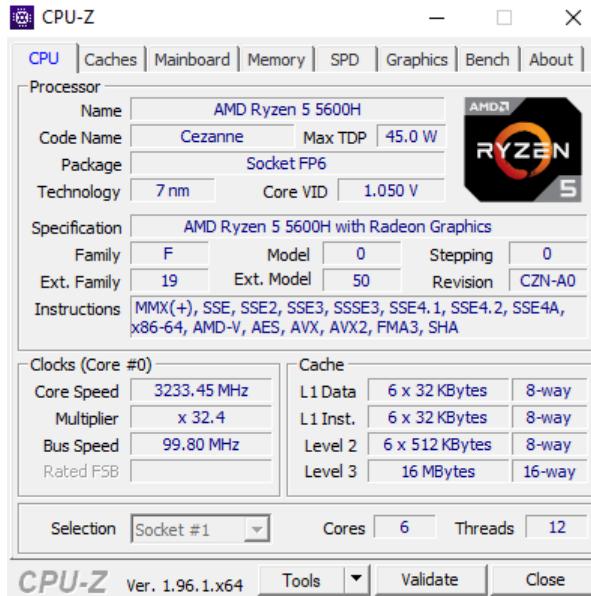
$$\text{Speedup} = \frac{1}{(1-p) + p/N}$$

Formulasi atau hukum ini banyak dipakai dalam bidang komputasi paralel untuk meramalkan peningkatan kecepatan maksimum pemrosesan data (secara teoritis) jika jumlah prosesor di dalam komputer paralel tersebut ditambah.

**GPU**  
**THROUGHPUT TINGGI LATENSI RENDAH**

**CPU**  
**THROUGHPUT RENDAH LATENSI TINGGI**

# Amdahl's Law for Parallel Computing Accelerates



Contoh perbandingan jumlah core:

1. GPU Nvidia GeForce GTX 1650 memiliki 896 CUDA cores
2. CPU AMD Ryzen 5 5600H memiliki 6 cores

# Apakah harus selalu menggunakan Deep Learning?



## Jawabannya adalah **Tidak!**

- Deep Learning sangat boros di komputasional
- Untuk melakukan sebuah *task*, deep learning memproses data yang banyak dengan menggunakan GPU dan memakan waktu cukup lama.
- Untuk permasalahan dengan data yang tidak memerlukan perhitungan complex, penggunaan **Machine learning** dengan komputasi yang lebih ringan **sangat disarankan**
- Deep Learning untuk pekerjaan prediksi dengan jumlah data yang kecil bisa tidak seakurat algoritma Machine Learning. Akurasi yang sama atau lebih tinggi bisa dihasilkan bila deep learningnya dilatih dengan baik.

# Kekurangan Deep Learning



- ❑ Jumlah data yang dibutuhkan dalam pelatihan cukup besar, sehingga membutuhkan akses ke komputer yang berkekuatan tinggi yang dilengkapi dengan GPU atau TPU (High Performance Computing - HPC).
  
- ❑ Deep-neural network sulit menentukan *tuning* terbaik dikarenakan menggunakan teknik **Black Box** yang dimana bobot akan di-update di setiap layernya.

# Machine learning as a service (MLaaS)

Machine learning as a service (MLaaS) adalah definisi umum dari berbagai platform berbasis cloud yang mencakup sebagian besar masalah infrastruktur seperti pra-pemrosesan data, pelatihan model, dan evaluasi model, dengan prediksi lebih lanjut. Hasil prediksi dapat dijembatani dengan REST API.



## CLOUD MACHINE LEARNING SERVICES COMPARISON

	Amazon ML and SageMaker	Microsoft Azure AI Platform	Google AI Platform (Unified)	IBM Watson Machine Learning
Classification	✓	✓	✓	✓
Regression	✓	✓	✓	✓
Clustering	✓	✓	✓	✗
Anomaly detection	✓	✓	✗	✗
Recommendation	✓	✓	✓	✗
Ranking	✓	✓	✗	✗
Data Labeling	✓	✓	✓	✓
MLOps pipeline support	✓	✓	✓	✓
Built-in algorithms	✓	✓	✓	✗
Supported frameworks	TensorFlow, MXNet, Keras, Gluon, PyTorch, Caffe2, Chainer, Torch	TensorFlow, scikit-learn, PyTorch, Microsoft Cognitive Toolkit, Spark ML	TensorFlow, scikit-learn, XGBoost, Keras	TensorFlow, Spark MLlib, scikit-learn, XGBoost, PyTorch, IBM SPSS, PMML

# Deep Learning Framework



theano

PYTORCH



Caffe



Caffe2

**Table 1** Some of the popular deep learning implementation tools

Tools	Platform	Support	Interface
Caffe (Williams and Zipser 1989)	Windows, Linux, Mac OSX	CNN, RNN	Python, C++, Matlab, Cuda
Tensorflow (Salakhutdinov and Hinton 2009)	Windows, Linux, Mac OSX, Android	Almost support all deep learning techniques	Python
Theano (Younes 1999)	Windows, Linux, Mac OSX	Almost support all deep learning techniques	Python, Cuda
Torch (Microsoft 2016)	Windows, Linux, Mac OSX	Almost support all deep learning techniques	Lua
Keras (Delakis and Garcia 2008)	Windows, Linux, Mac OSX	Almost support all deep learning techniques	Cross-platform, Cuda
PyTorch (Xu and Su 2015)	Linux, Mac OSX	Almost support all deep learning techniques	Python, C, Cuda

Recent Advances in Deep Learning Techniques and Its Applications: An Overview  
DOI : [10.1007/978-981-15-6329-4\\_10](https://doi.org/10.1007/978-981-15-6329-4_10)



# Most Popular Framework Deep Learning



TensorFlow

- Versi terbaru di TF saat ini adalah versi 2
- Sudah dikembangkan di bahasa pemrograman JavaScript yang bernama Tensorflow JS



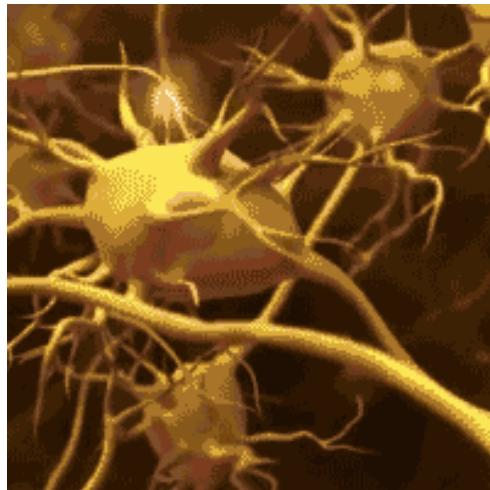
PyTorch

- Mendapatkan Merger dari Framework DL yaitu CAFFE2
- Integrated dengan Framework DL yaitu YOLO V5 (You Only Look Once)



DL4J

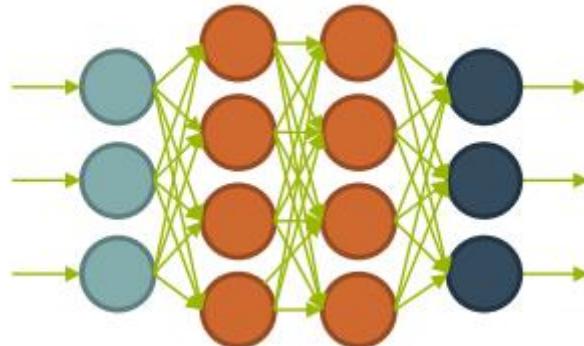
- Bisa diintegrasikan di Hadoop and Apache Spark (Data Engineering)
- Hanya Support dalam bahasa pemrograman JAVA



## Session II

# Artificial Neural Network

## Artificial Neural Networks (ANN) atau Jaringan Syaraf Tiruan (JST)



Adalah paradigma pemrosesan informasi yang terinspirasi dari cara kerja sistem saraf (**otak**) manusia dalam memproses informasi.

ANN terdiri dari sejumlah besar elemen pemrosesan (**neuron**) yang saling berhubungan dan bekerja secara bersama untuk memecahkan masalah tertentu.

# Kelebihan Neural Network



Kemampuannya berguna untuk robotika dan sistem pengenalan pola



Keluaran ANN tidak sepenuhnya dibatasi oleh masukan dan hasil yang diberikan kepada mereka pada awalnya oleh sistem pakar



JST memiliki potensi toleransi kesalahan yang tinggi

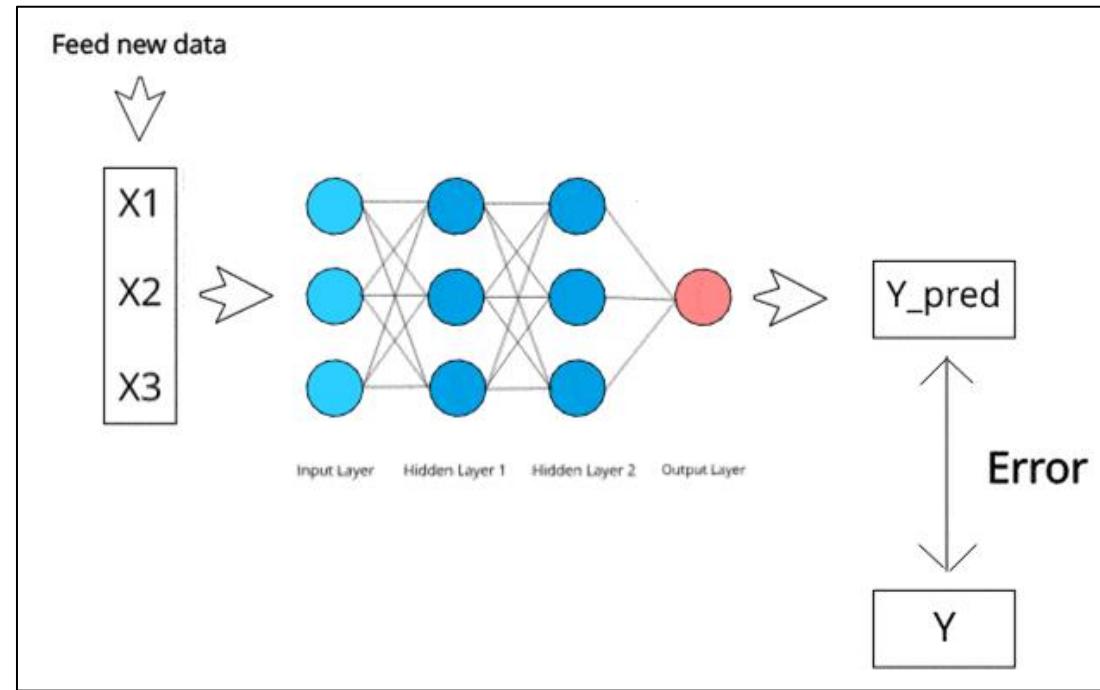


ANN mampu men-debug atau mendiagnosis jaringan sendiri



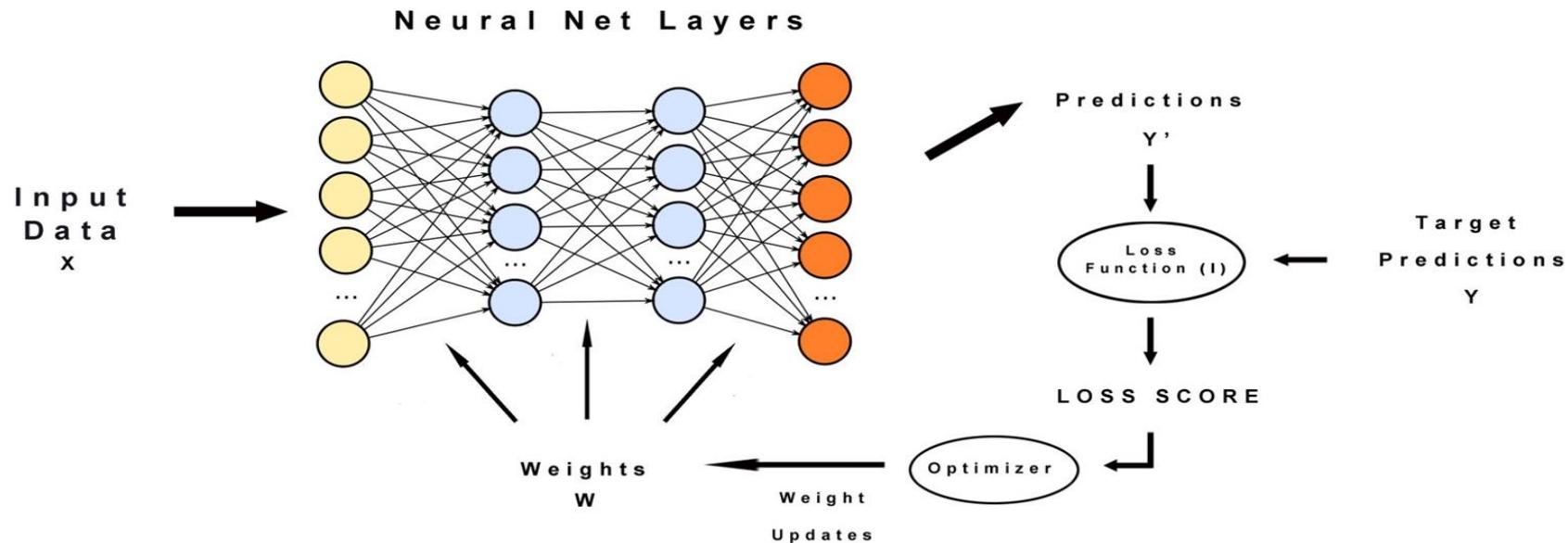
Sistem non-linear yang memiliki kemampuan untuk menemukan jalan pintas untuk mencapai solusi komputasi yang mahal

# Cara Kerja ANN

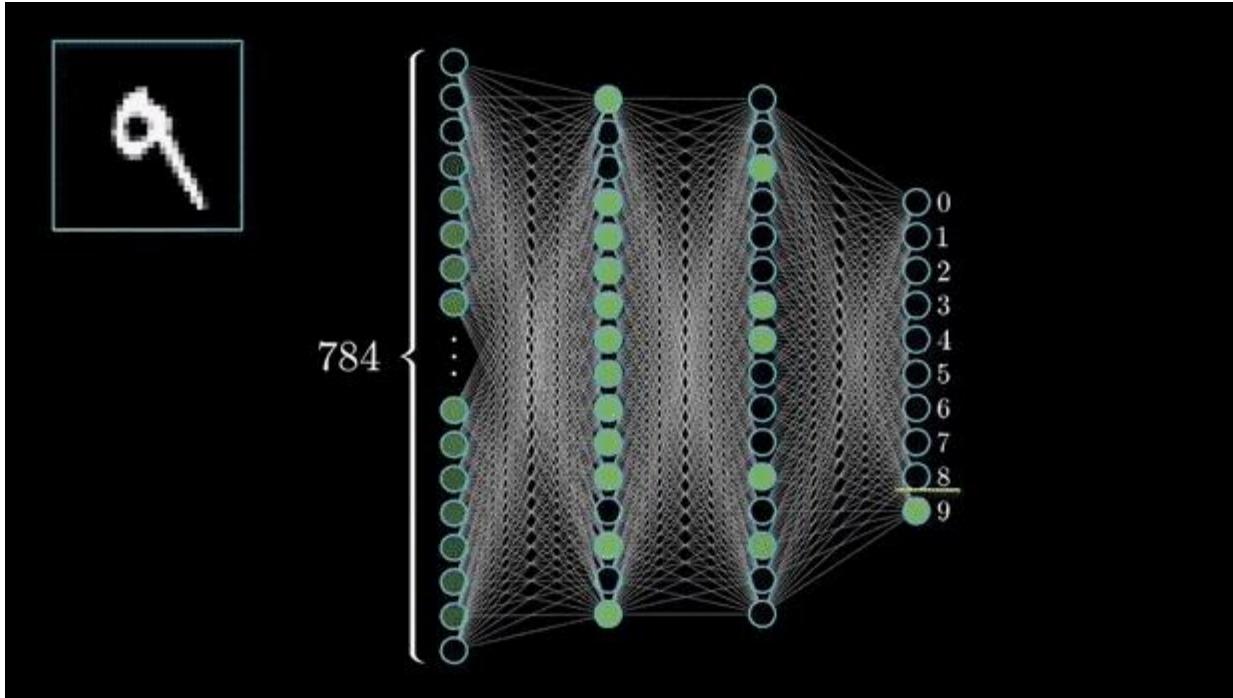


# Cara Kerja ANN

## LOGICAL FLOW OF A NEURAL NETWORK

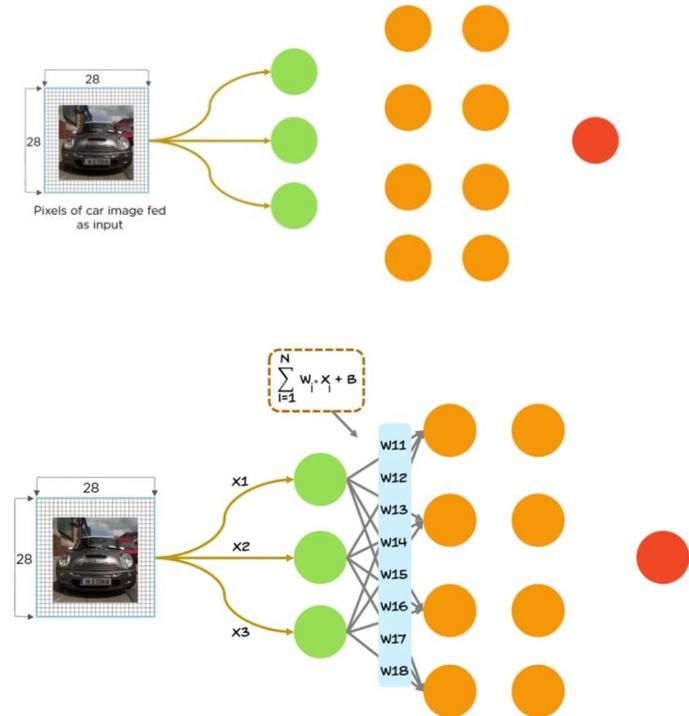
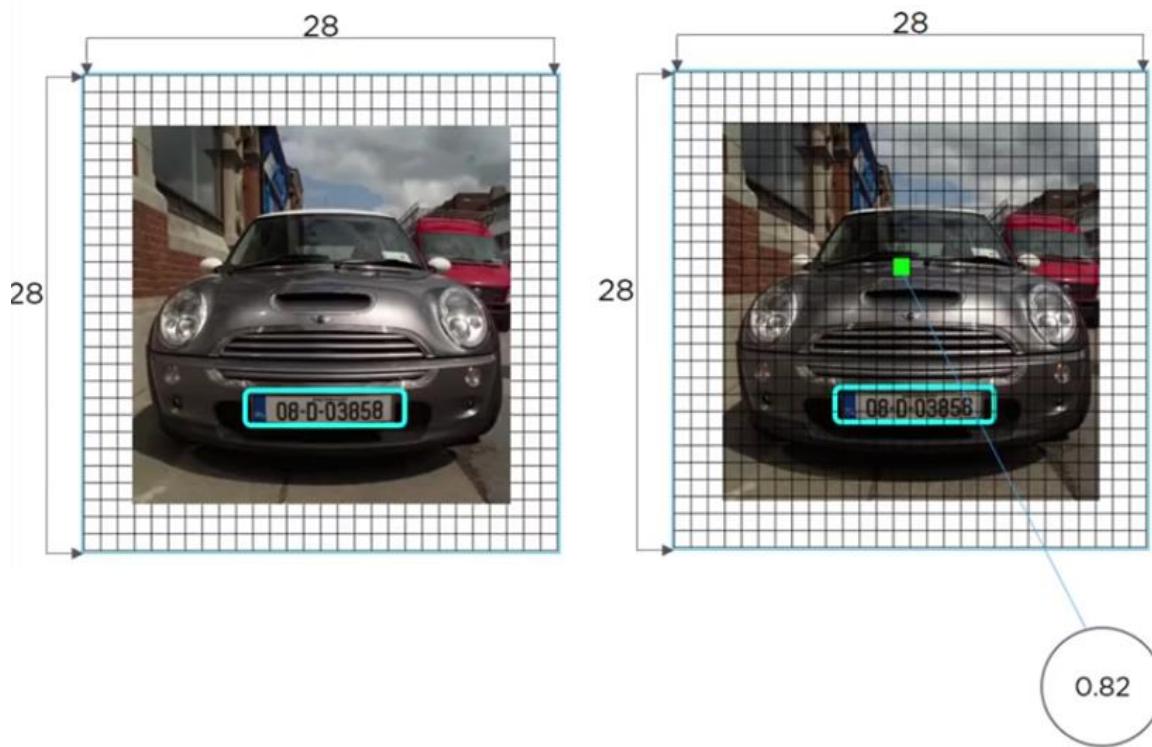


# Cara Kerja ANN



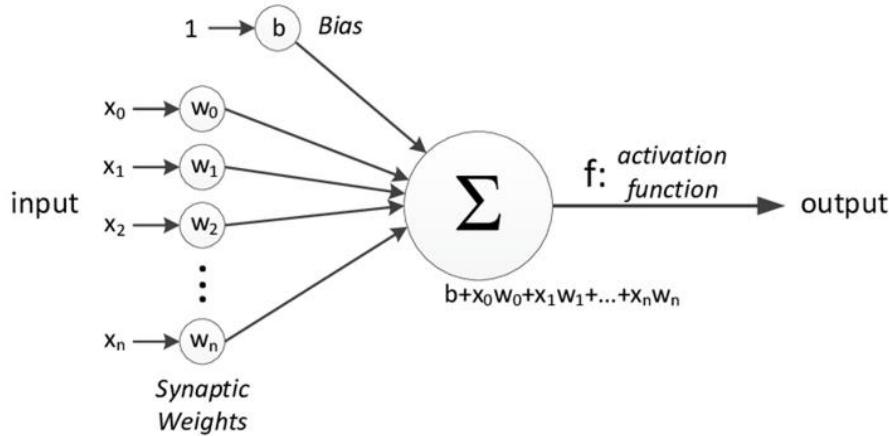
<https://youtu.be/aircAruvnKk>

# Neuron



# Formula Matematis ANN

Neural network satu lapis disebut Perceptron. Perceptron memberikan output tunggal.



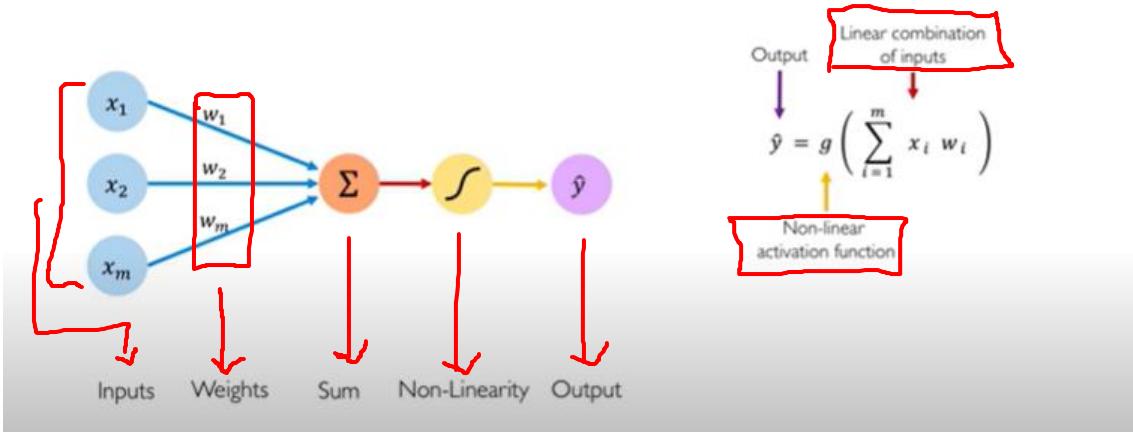
$x_0, x_1, x_2, x_3 \dots x(n)$  mewakili berbagai masukan (variabel bebas). Masing-masing input ini dikalikan dengan bobot koneksi atau sinapsis.

Bobot direpresentasikan sebagai  $w_0, w_1, w_2, w_3 \dots w(n)$ . Bobot menunjukkan kekuatan node tertentu.

$b$  adalah nilai bias. Nilai bias memungkinkan Anda untuk menggeser fungsi aktivasi ke atas atau ke bawah.

$$\begin{aligned}z &= b + \sum_{i=1}^m x_i w_i \\z &= b + x^T w \\a &= f(z)\end{aligned}$$

## The Perceptron: Forward Propagation

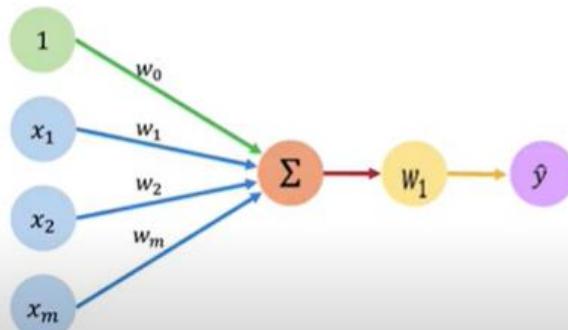


Linear combination of inputs

$$\hat{y} = g \left( \sum_{i=1}^m x_i w_i \right)$$

Non-linear activation function

## The Perceptron: Forward Propagation



Inputs    Weights    Sum    Non-Linearity    Output

Output

Linear combination of inputs

$\hat{y} = g \left( w_0 + \sum_{i=1}^m x_i w_i \right)$

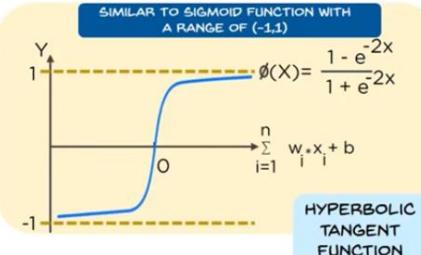
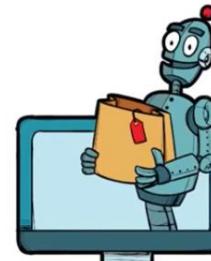
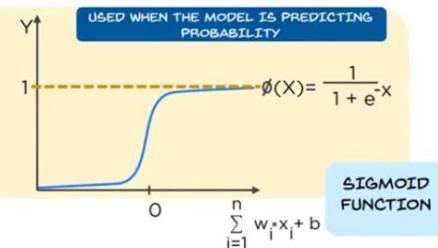
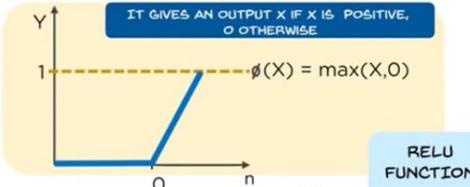
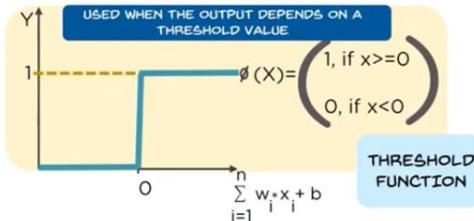
Non-linear activation function

Bias

Diagram illustrating the mathematical formula for the output of a perceptron. The output  $\hat{y}$  is calculated as the result of applying a non-linear activation function  $g$  to the linear combination of inputs and weights. The linear combination is given by the equation  $w_0 + \sum_{i=1}^m x_i w_i$ . The term  $w_0$  is labeled as the bias, and the term  $\sum_{i=1}^m x_i w_i$  is labeled as the linear combination of inputs.

# Activation Function

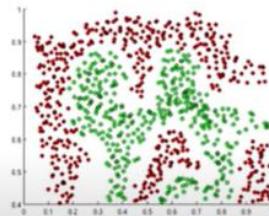
**Activation function** membantu menyelesaikan permasalahan-permasalahan yang bersifat non-trivial dalam suatu neural network dengan cara mengambil sebuah nilai dan melakukan operasi matematika. Fungsi-fungsi matematika yang umum digunakan pada ANN adalah ReLu dan Sigmoid



# Activation Function

## Importance of Activation Functions

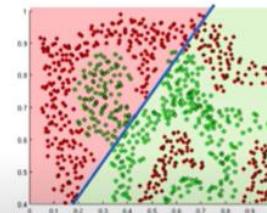
The purpose of activation functions is to **introduce non-linearities** into the network



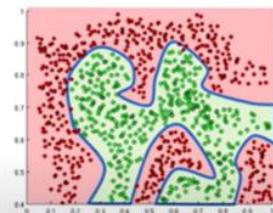
What if we wanted to build a neural network to distinguish green vs red points?

## Importance of Activation Functions

The purpose of activation functions is to **introduce non-linearities** into the network

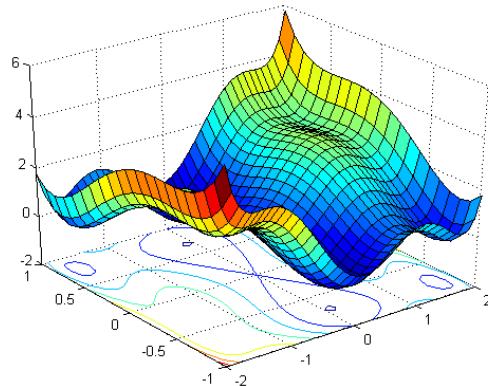


Linear activation functions produce linear decisions no matter the network size



Non-linearities allow us to approximate arbitrarily complex functions

# Loss Function



**Loss function** digunakan untuk mengukur seberapa bagus performa dari *neural network* dalam melakukan prediksi terhadap target.

Loss Function atau Cost Function merupakan fungsi yang menggambarkan **kerugian** yang terkait dengan semua kemungkinan yang dihasilkan oleh model.

Mean Square Error (MSE):

$$\text{error} = \frac{1}{n} \sum_{i=1}^n (\text{target}_i - \text{prediksi}_i)^2$$

# Loss Function

## Loss Optimization

We want to find the network weights that *achieve the lowest loss*

$$\mathbf{W}^* = \operatorname{argmin}_{\mathbf{W}} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x^{(i)}; \mathbf{W}), y^{(i)})$$

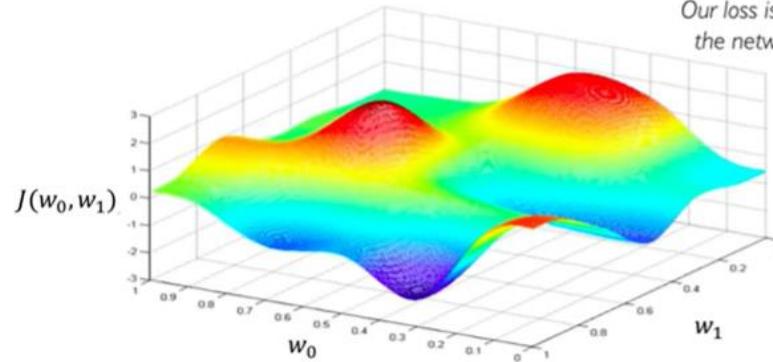
$$\mathbf{W}^* = \operatorname{argmin}_{\mathbf{W}} J(\mathbf{W})$$

↑  
Remember:  
 $\mathbf{W} = \{\mathbf{W}^{(0)}, \mathbf{W}^{(1)}, \dots\}$

## Loss Optimization

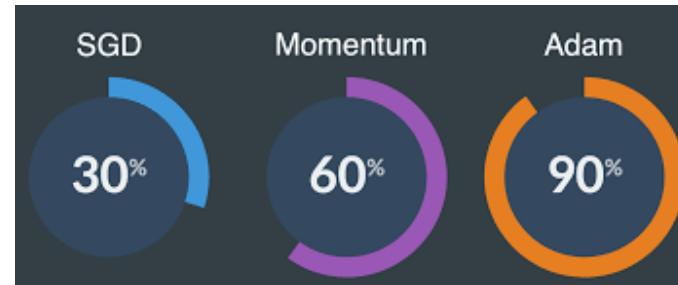
$$\mathbf{W}^* = \operatorname{argmin}_{\mathbf{W}} J(\mathbf{W})$$

Remember:  
Our loss is a function of  
the network weights!



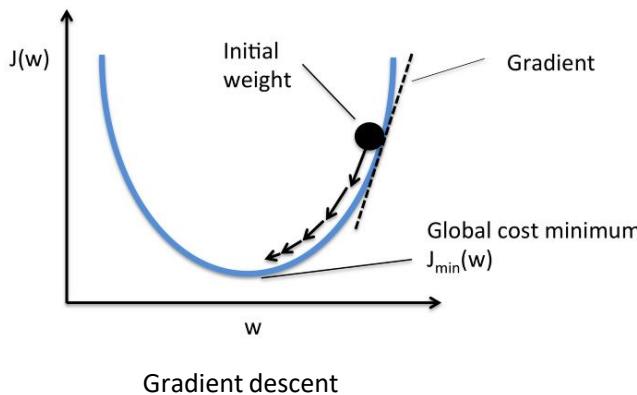
**Optimizer** adalah fungsi pengoptimalan yang dapat digunakan untuk memperbarui *weight network* secara iteratif berdasarkan data training.

Contoh optimizer yang sering digunakan adalah gradient descent, SGD, momentum, dan ADAM (*Adaptive Moment Estimation*) optimizer.



# Back Propagation

- **Back-propagation**—adalah proses memperbarui bobot dari network untuk mereduksi error dalam hasil prediksi
- **Gradient descent**, adalah proses menyesuaikan parameter model untuk turun melalui *loss function*.



Rumus utama untuk memperbaiki suatu bobot  $w$  berdasarkan error  $E$  adalah:

$$w_{\text{new}} = w_{\text{old}} - \alpha \frac{\partial E}{\partial w}$$

Rumus ini juga berlaku untuk memperbaiki nilai bias:

$$b_{\text{new}} = b_{\text{old}} - \alpha \frac{\partial E}{\partial b}$$

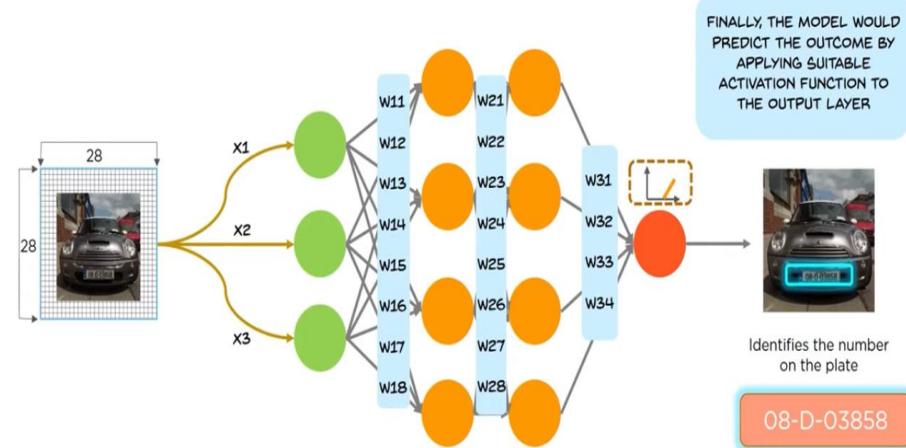
# Back Propagation

## Computing Gradients: Backpropagation



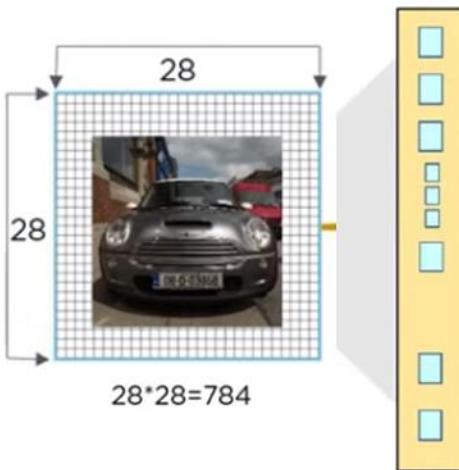
$$\frac{\partial J(\mathbf{W})}{\partial w_1} = \frac{\partial J(\mathbf{W})}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial z_1} * \frac{\partial z_1}{\partial w_1}$$

— — —



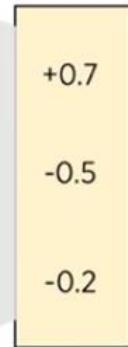
Optical Character Recognition (OCR) is used on the images to convert it into text in order to identify what's written on the plate

# Back Propagation



1<sup>st</sup> iteration  
 $\text{loss}(a) \rightarrow 0.7^2 = 0.49$   
 $\text{loss}(b) \rightarrow 0.5^2 = 0.25$   
 $\text{loss}(c) \rightarrow 0.2^2 = 0.04$

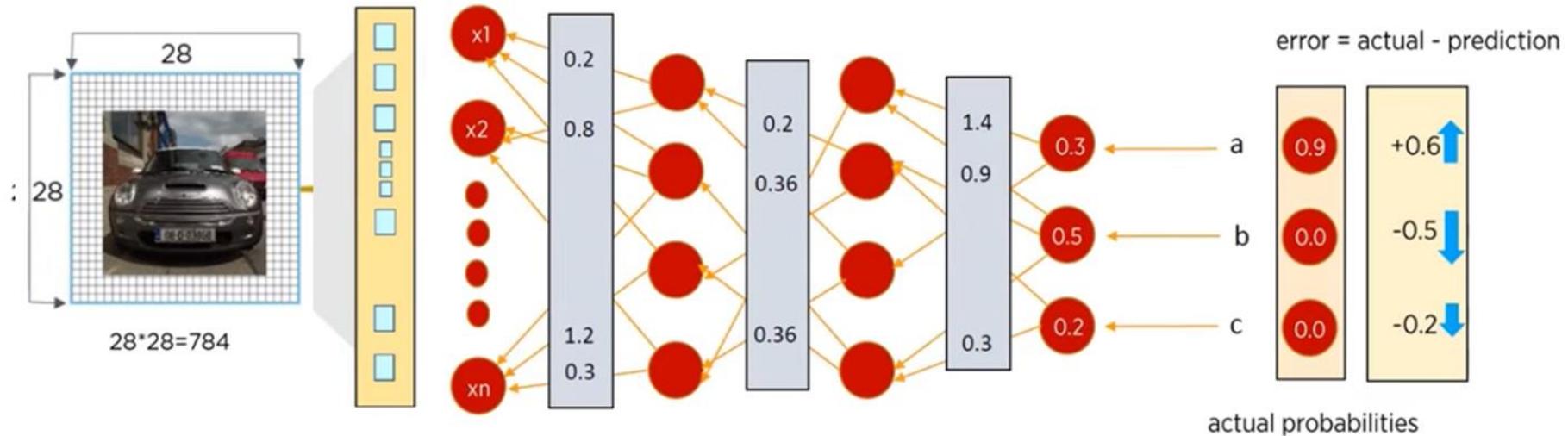
error = actual - prediction



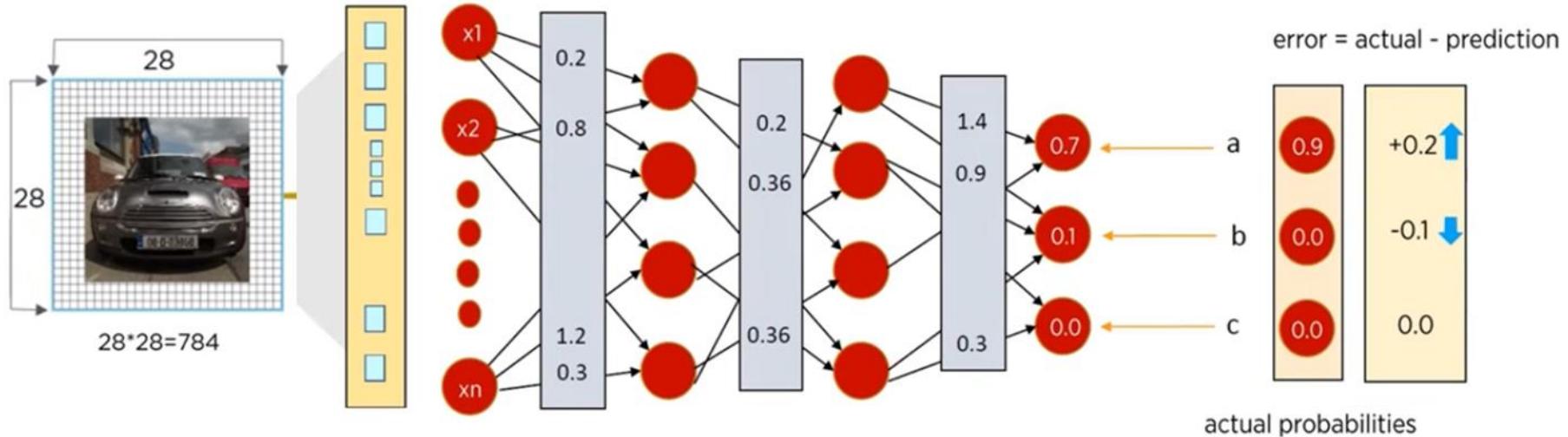
actual probabilities

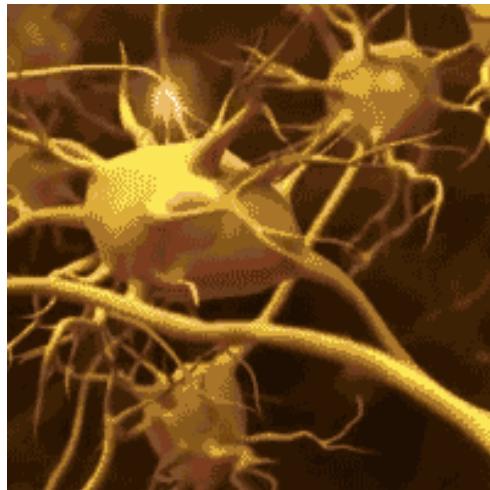
# Back Propagation

## Updating weight (1)



# Back Propagation



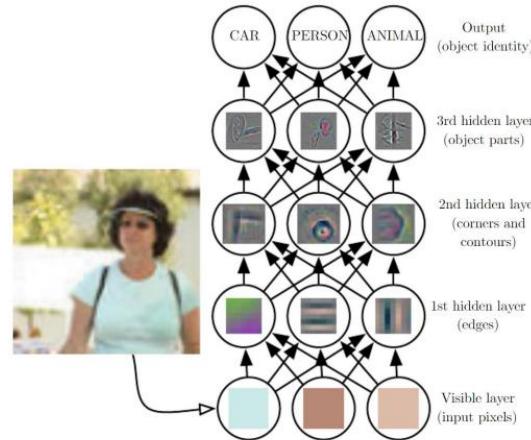
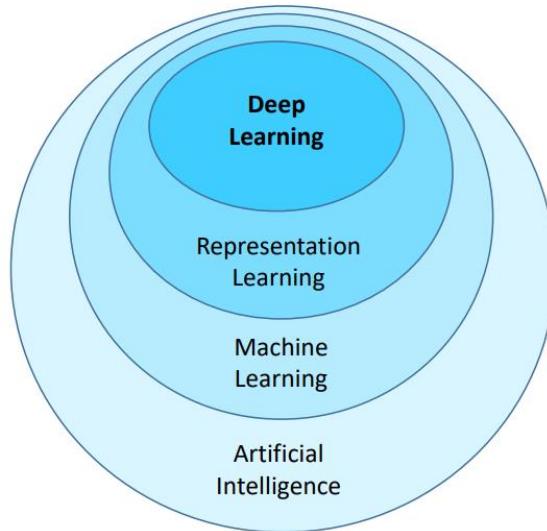


## Session III

### Deep Learning a.k.a Feature Learning

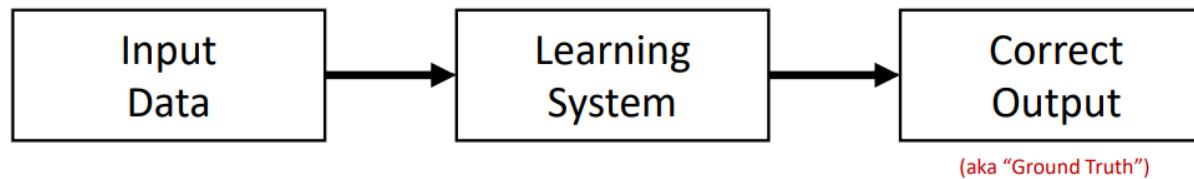
# Deep Learning is Representation Learning

(aka Feature Learning)

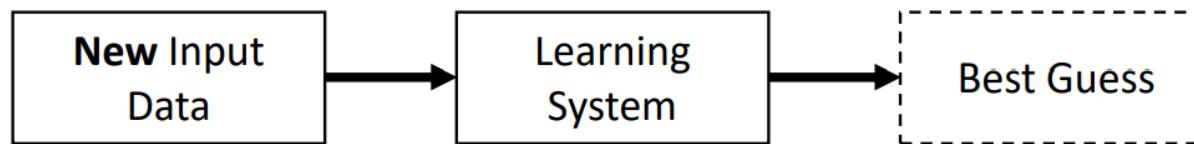


## Deep Learning: Training and Testing

### Training Stage:

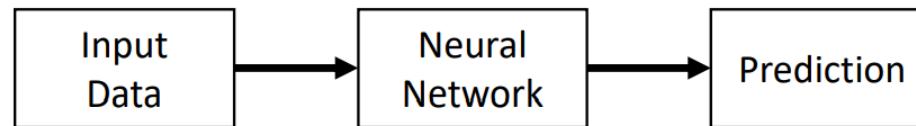


### Testing Stage:

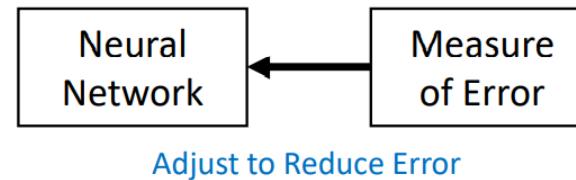


## How Neural Networks Learn: Backpropagation

Forward Pass:



Backward Pass (aka Backpropagation):



## Regression vs Classification



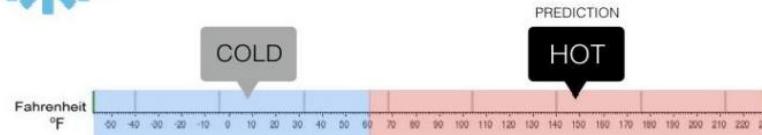
### Regression

What is the temperature going to be tomorrow?



### Classification

Will it be Cold or Hot tomorrow?



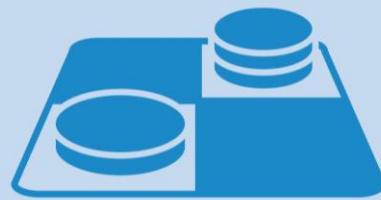
## The Challenge of Deep Learning: Efficient Teaching + Efficient Learning

- Humans can learn from very few examples
- Machines (in most cases) need thousands/millions of examples



## ARTIFICIAL INTELLIGENCE

Artificial Intelligence captures the imagination of the world.



## MACHINE LEARNING

Machine learning starts to gain traction.



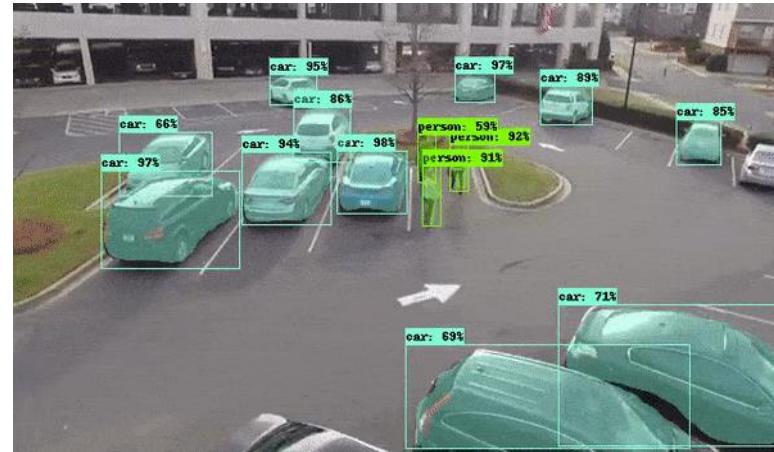
## DEEP LEARNING

Deep learning catapults the industry.

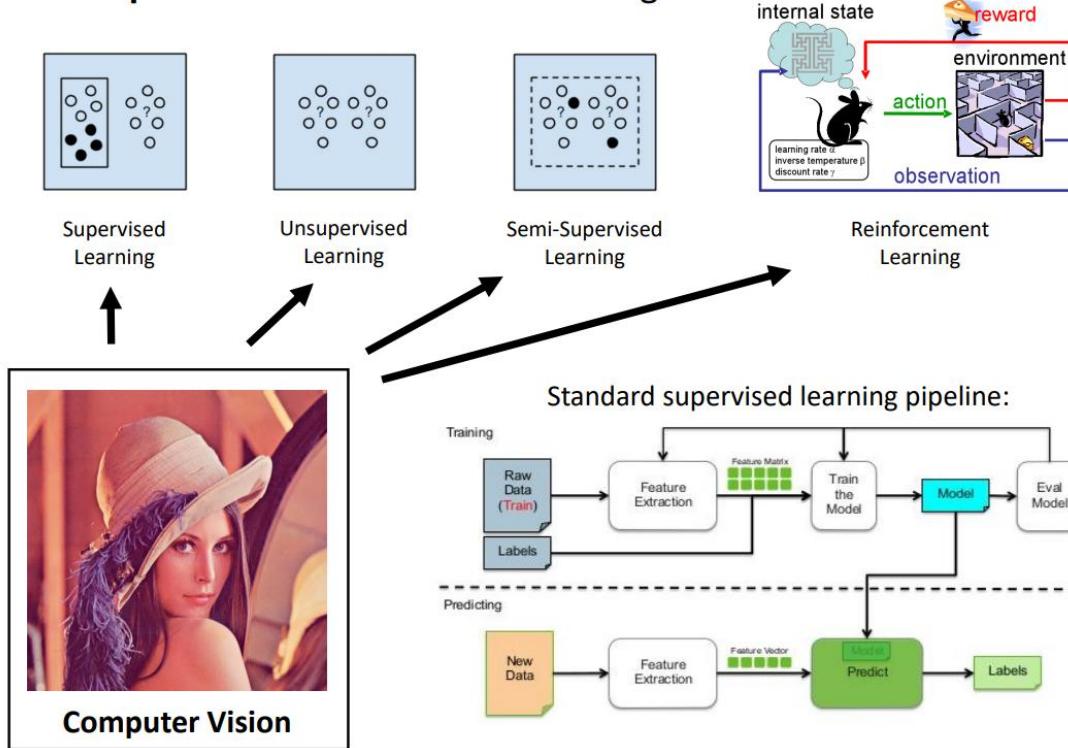


1950 1960 1970 1980 1990 2000 2010 2020 2030 2040

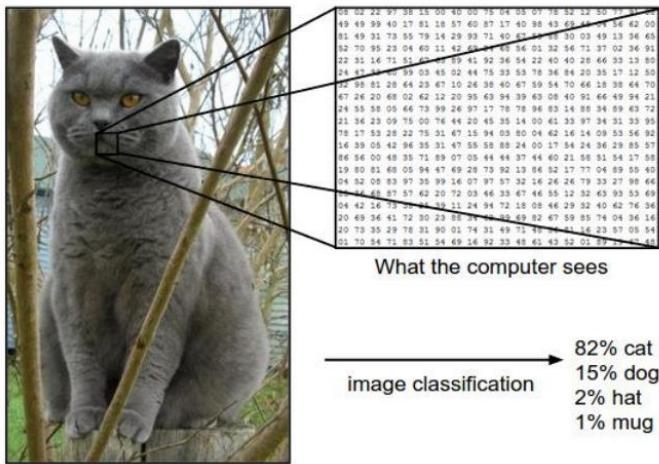
Deep Learning membantu komputer untuk mengenali objek. Tapi, bagaimana computer cara melihat?



## Computer Vision is Machine Learning

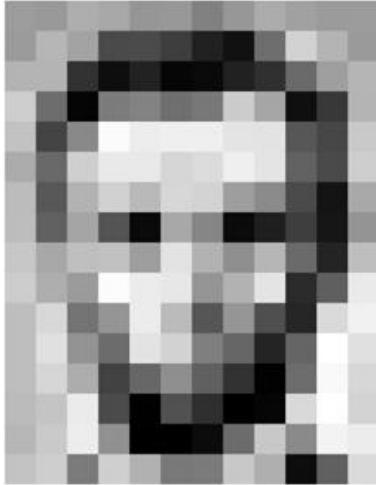


# Images are Numbers



- **Regression:** The output variable takes continuous values
  - **Classification:** The output variable takes class labels
    - Underneath it may still produce continuous values such as probability of belonging to a particular class.

# Image are Numbers

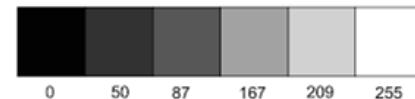


157	153	174	168	160	152	129	151	172	163	165	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	34	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
206	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	88	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	73	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	133	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	93	17	110	210	180	154
180	180	50	14	34	6	10	93	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
206	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	88	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	73	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	133	200	175	13	96	218

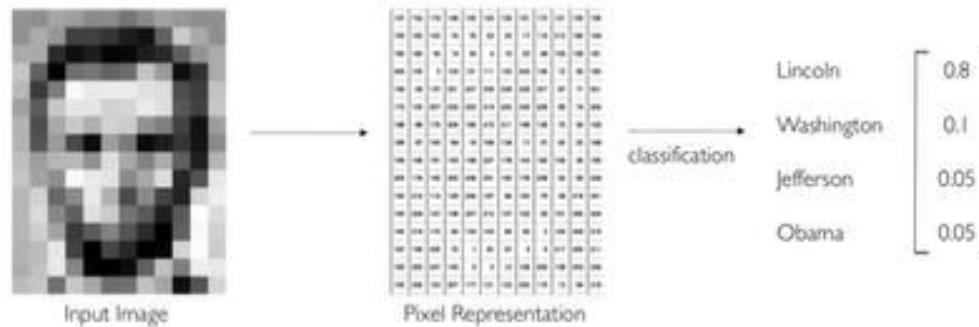
- Why we need math?
- An image is just a matrix of numbers [0,255]!

Source: ai.stanford.edu, mit



Source:  
processing.org

## Tasks in Computer Vision



- **Regression**: output variable takes continuous value
  - **Classification**: output variable takes class label. Can produce probability of belonging to a particular class
- 
- An image is just a matrix of numbers [0,255]!
  - But why must 255?

## Kenapa 255?

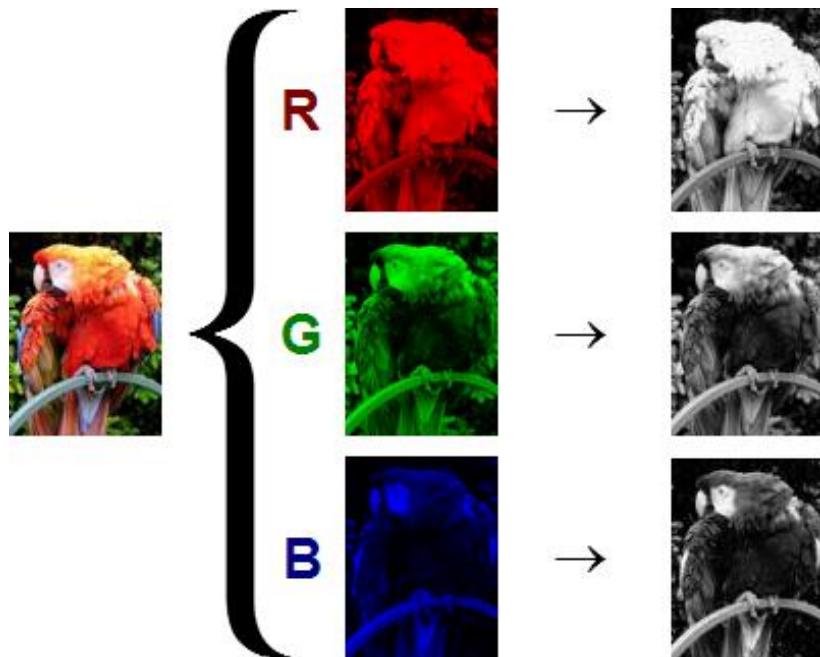
1 bit	
0	1

Number of bits	Different patterns	No. of patterns	No. of patterns
1	0 1	$2^1$	2
2	00 01 10 11	$2^2$	4
3	000 001 010 100 011 101 110 111	$2^3$	8

1 byte = 8 bits

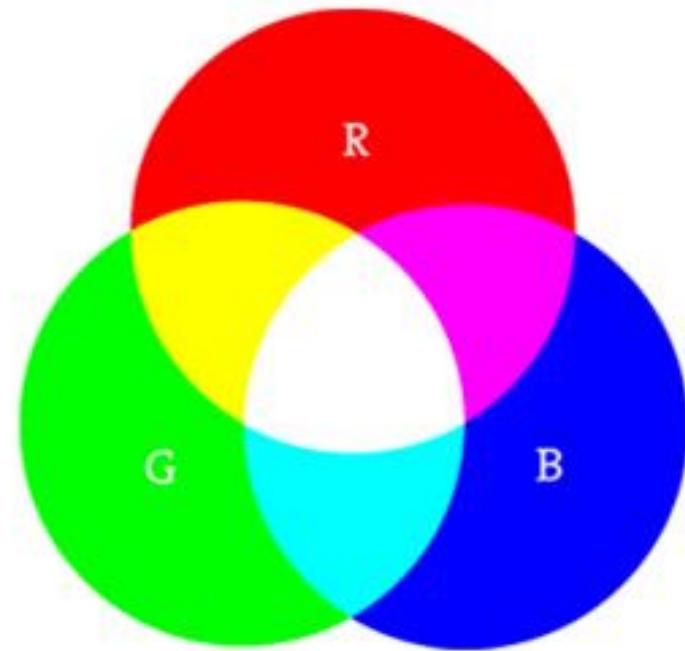
$$2^8 = 256$$

Range: 0 to 255



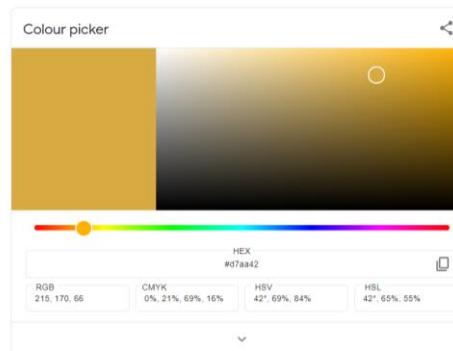
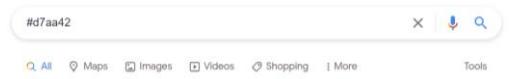
Color channel  
separation

Source: Wikipedia



Source: processing.org

# Let's try by yourself!



[https://www.w3schools.com/colors/colors\\_rgb.asp](https://www.w3schools.com/colors/colors_rgb.asp)

## Is Computer Vision Hard?

Viewpoint variation



Scale variation



Deformation



Occlusion



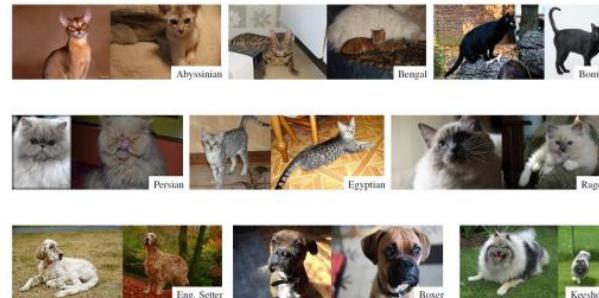
Illumination conditions



Background clutter

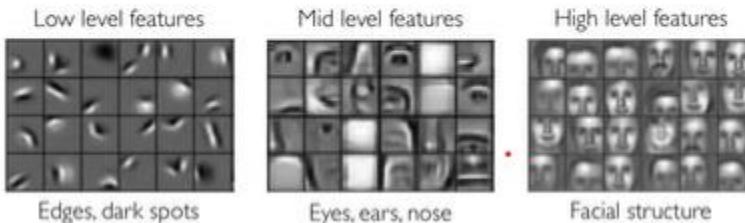


Intra-class variation

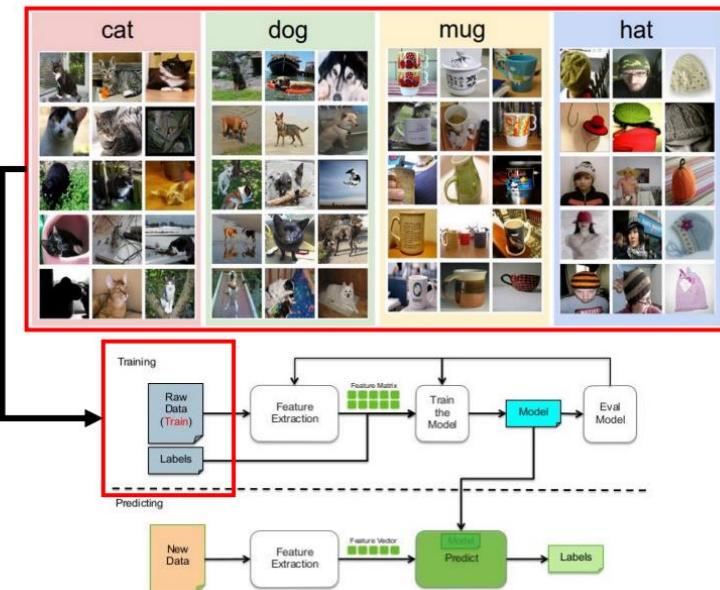


## Learning Feature Representations

Can we learn a **hierarchy of features** directly from the data instead of hand engineering?

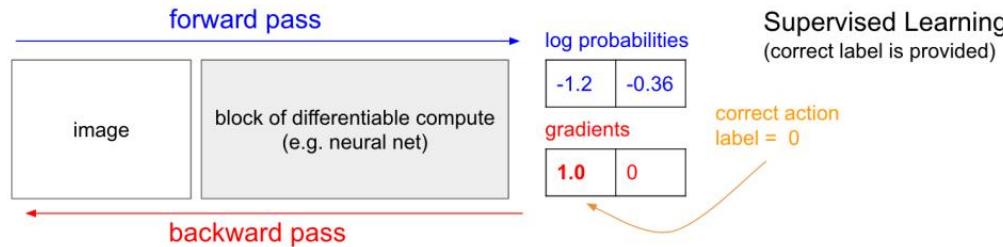


## Image Classification Pipeline



# Think like Machine

*Reminder: “Learning” is Optimization of a Function*

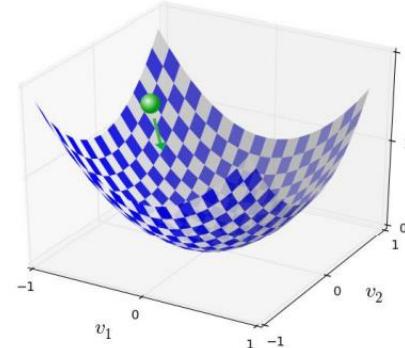


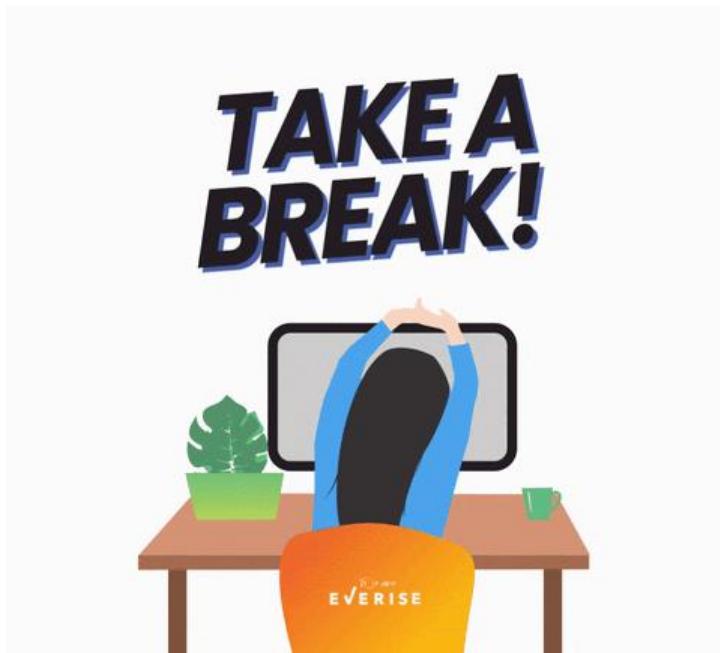
Ground truth for “6”:

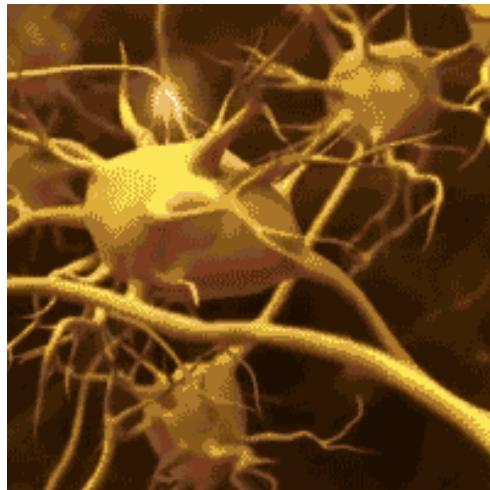
$$y(x) = (0, 0, 0, 0, 0, 0, 1, 0, 0, 0)^T$$

“Loss” function:

$$C(w, b) \equiv \frac{1}{2n} \sum_x \|y(x) - a\|^2$$

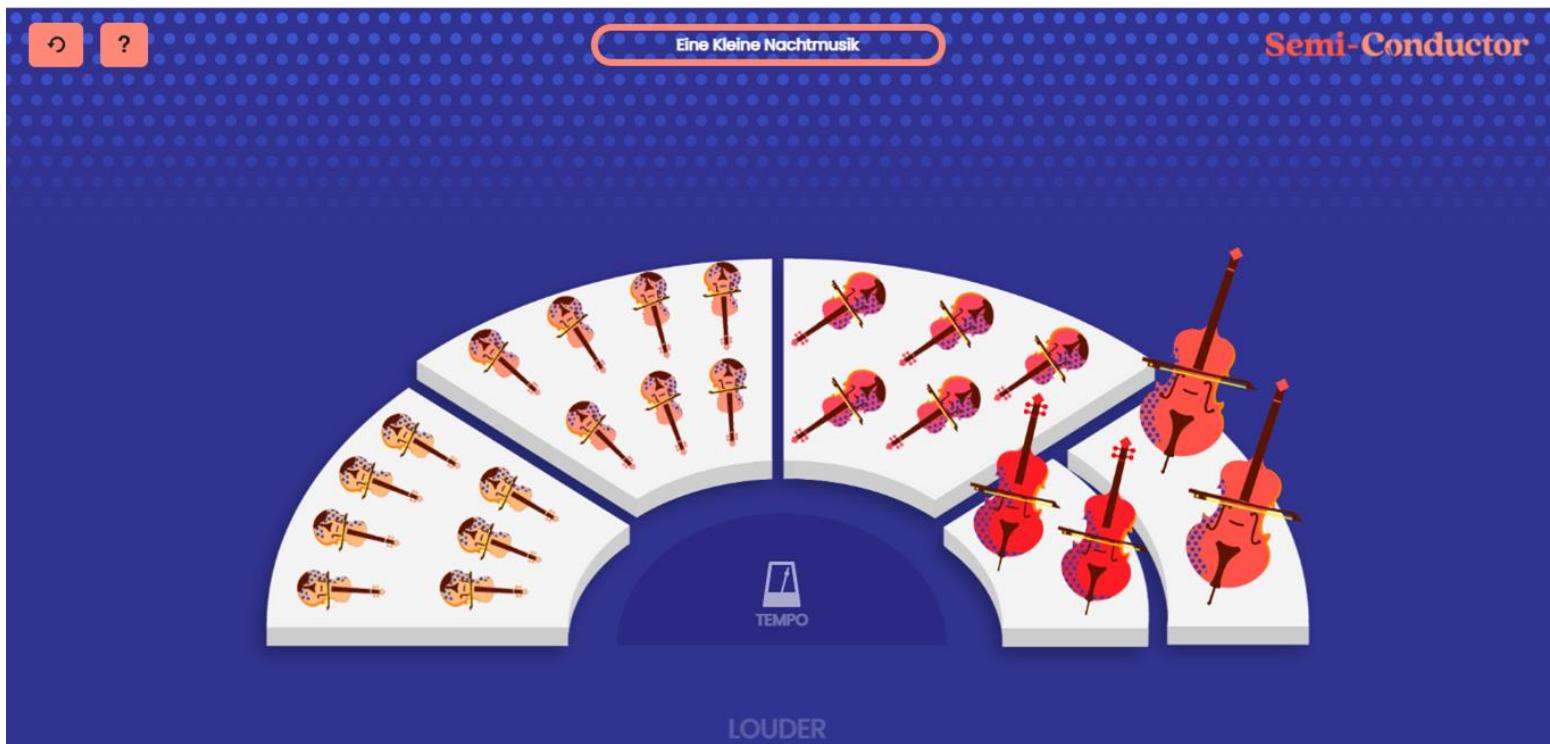






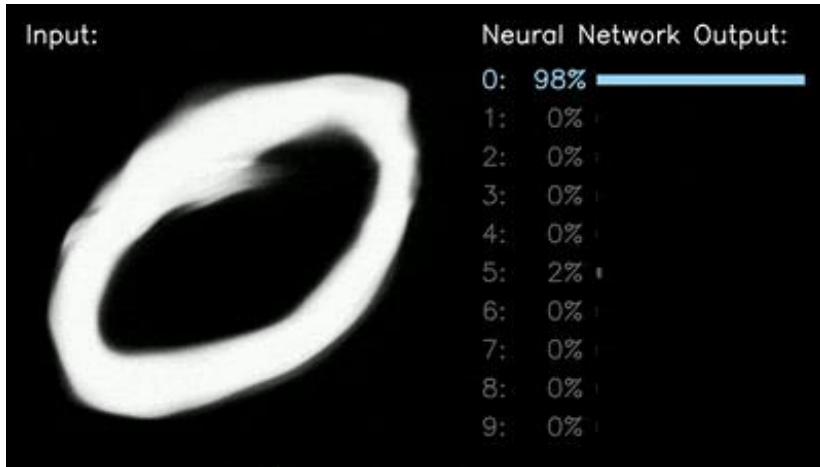
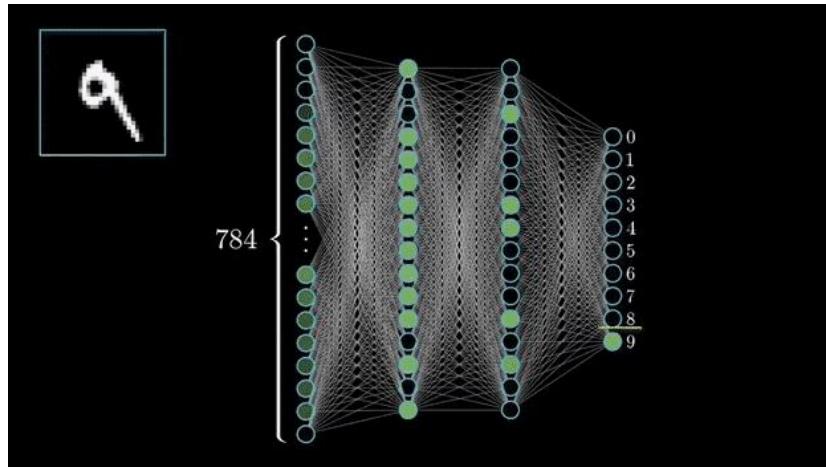
## Session IV

### Deep Learning; Example

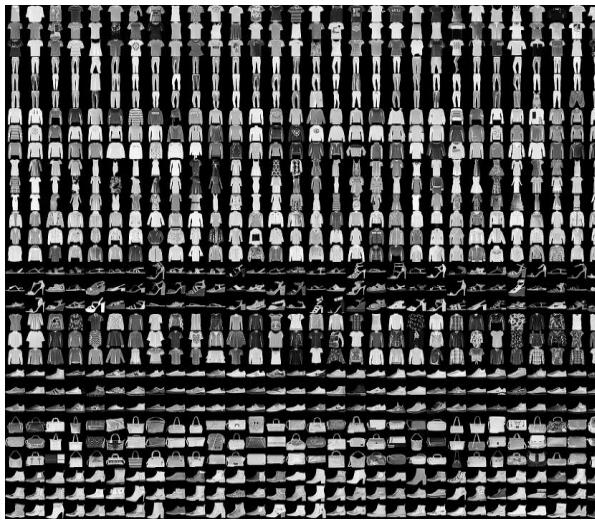


# Digit Classification

## MNIST Dataset

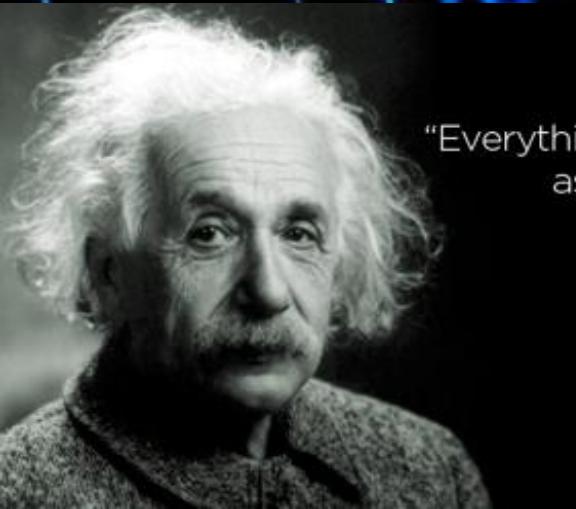
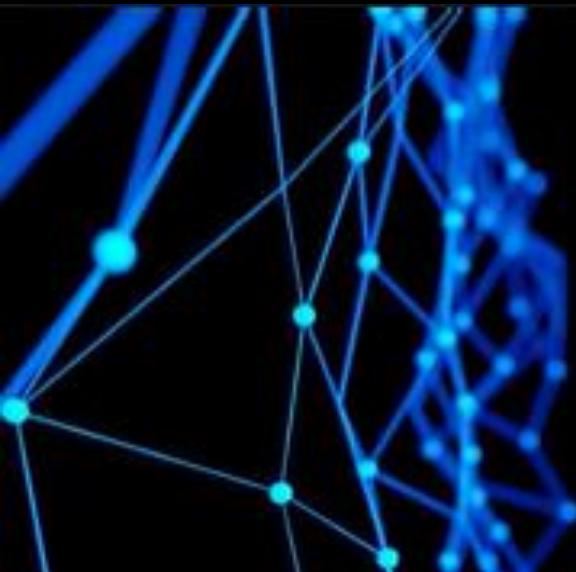


# The Fashion MNIST Dataset



Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For Future Jobs



“Everything should be made  
as simple as possible.  
But not simpler.”

*Albert Einstein*





# TERIMA KASIH

## THANK YOU

### Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

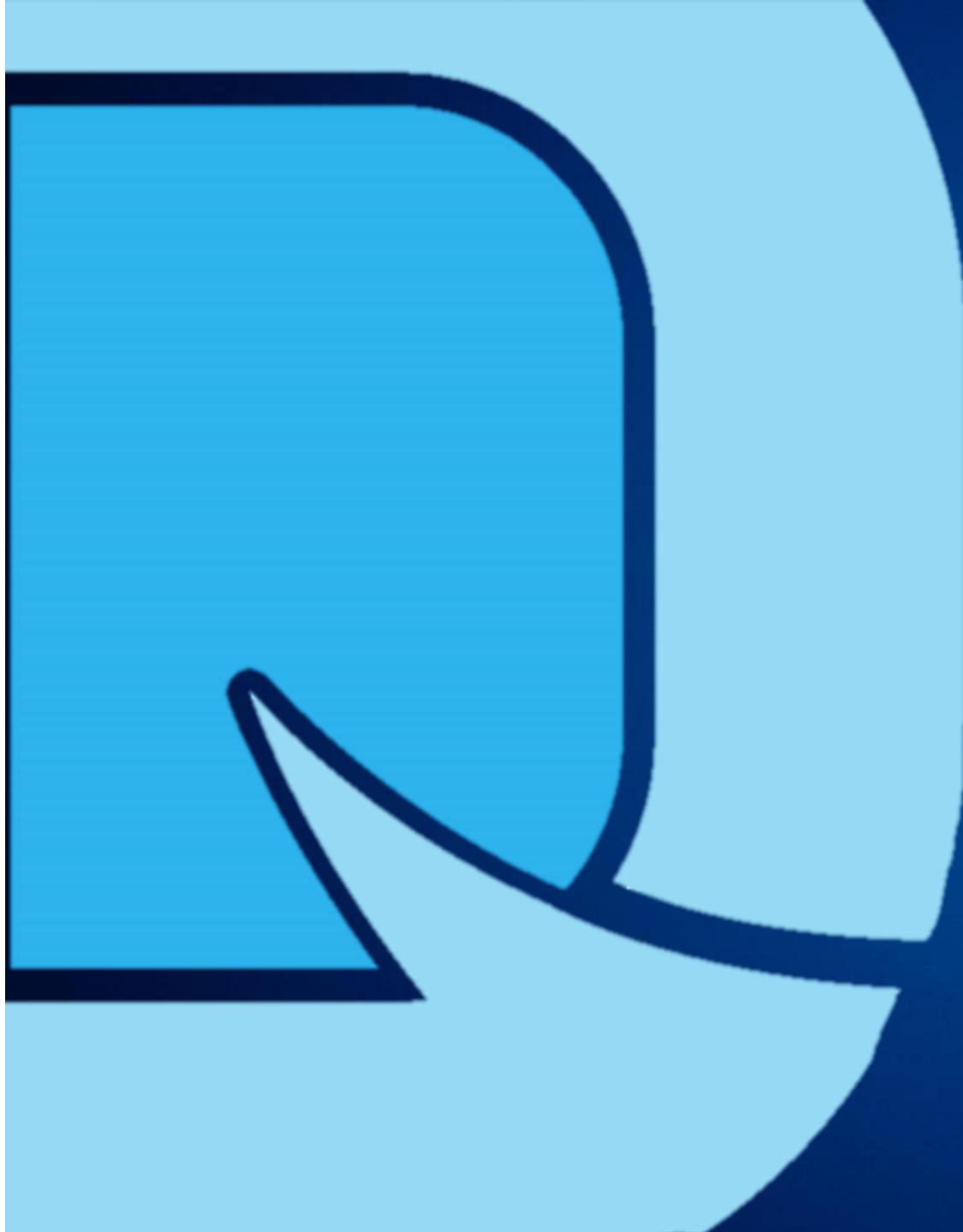
- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

### Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

### Social Media

-  Orbit Future Academy
-  @OrbitFutureAcademyIn1
-  OrbitFutureAcademy
-  Orbit Future Academy



# Module

## Artificial Intelligence Fundamental

### Section

#### Recap AI, ML, DL

# Traditional Programming vs Machine Learning



```
if(speed<4){  
    status=WALKING;  
}
```



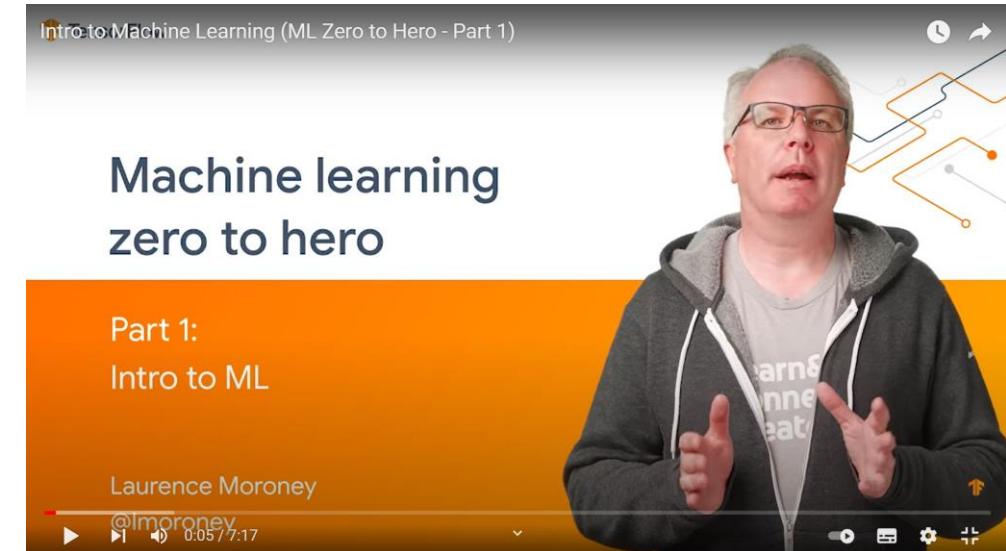
```
if(speed<4){  
    status=WALKING;  
} else {  
    status=RUNNING;  
}
```



```
if(speed<4){  
    status=WALKING;  
} else if(speed<12){  
    status=RUNNING;  
} else {  
    status=BIKING;  
}
```



?



Intro to Machine Learning (ML Zero to Hero - Part 1)

## Machine learning zero to hero

Part 1:  
Intro to ML

Laurence Moroney  
@l moroney

<https://youtu.be/KNAWp2S3w94>

Images Source: <https://developers.google.com/codelabs/tensorflow-1-helloworld#0>

# Machine Learn from Data



# Contoh

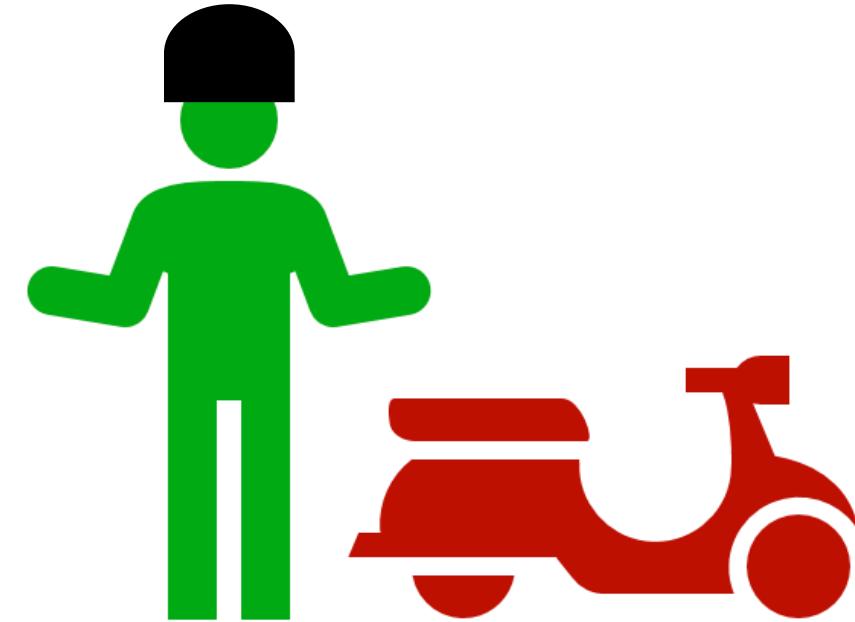


Ojek Online

We have problem...



**Penumpang**

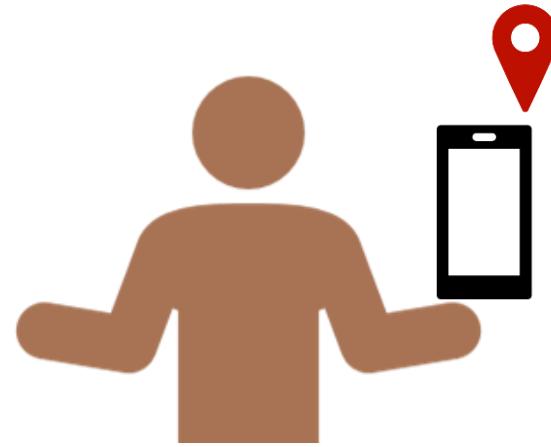


**Pengemudi**



**Penumpang**

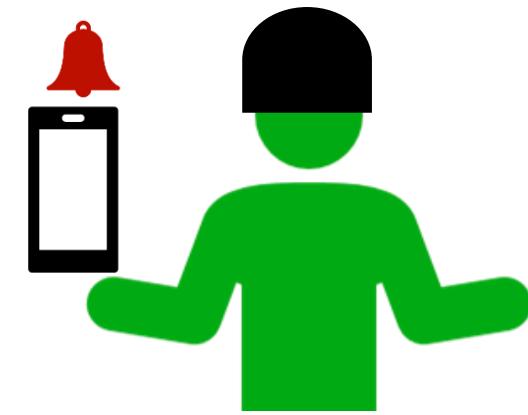
**Pengemudi**



- Lokasi
- Tujuan



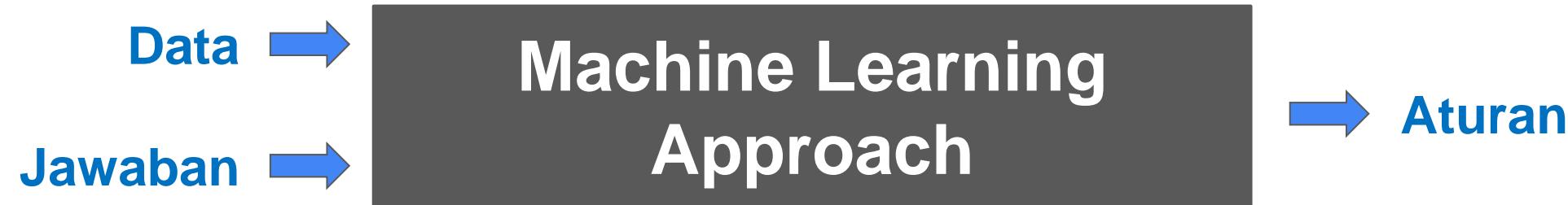
- Tarif
- Rute



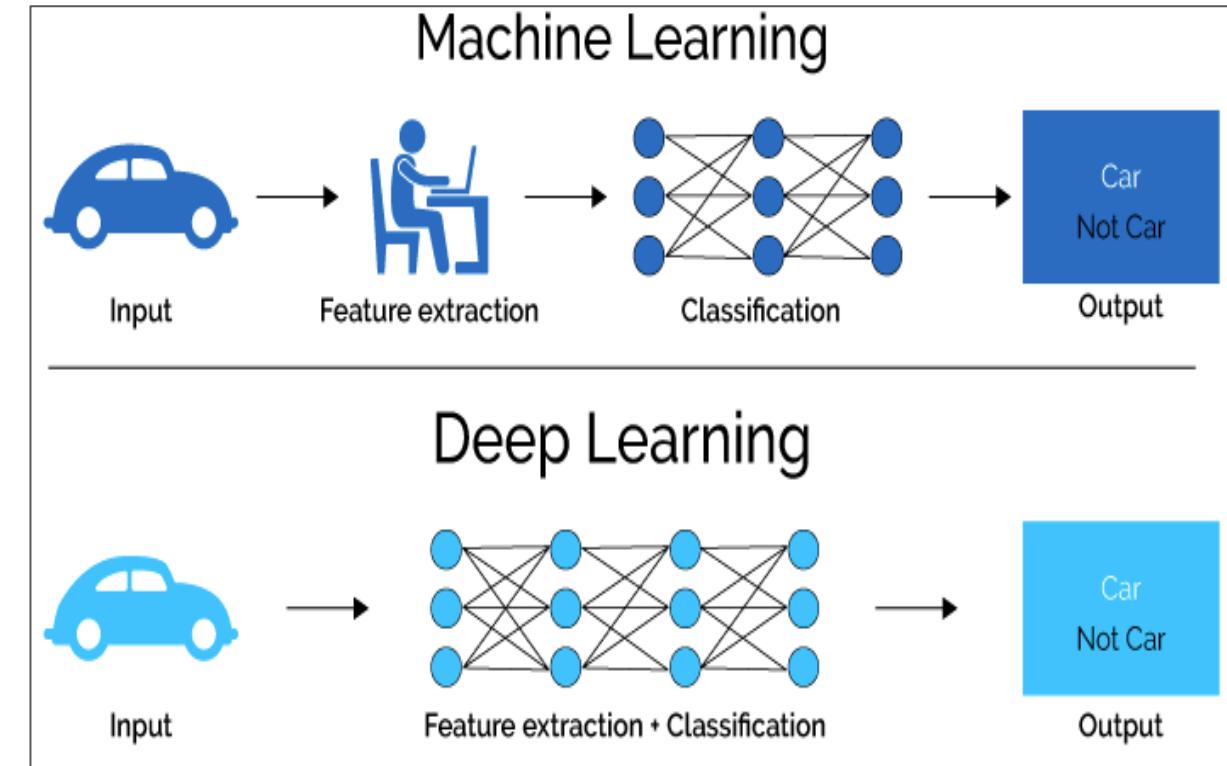
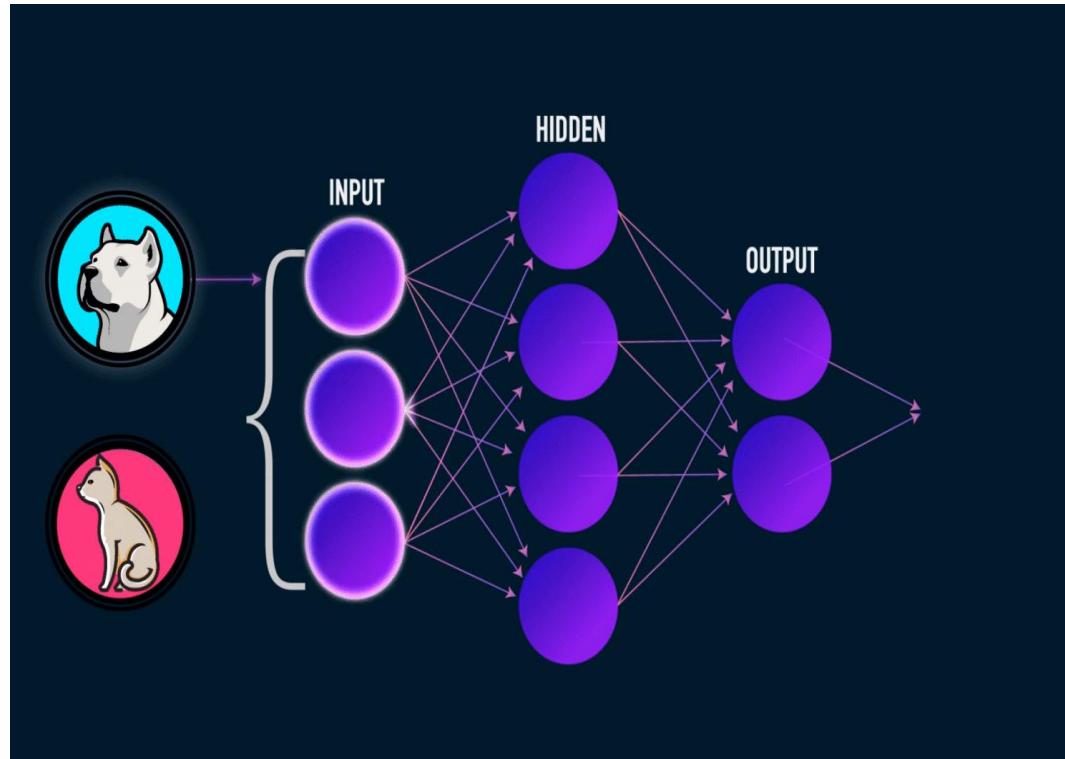
- Lokasi
- Tujuan
- Tarif
- Rute



# Traditional Programming vs Machine Learning Approach



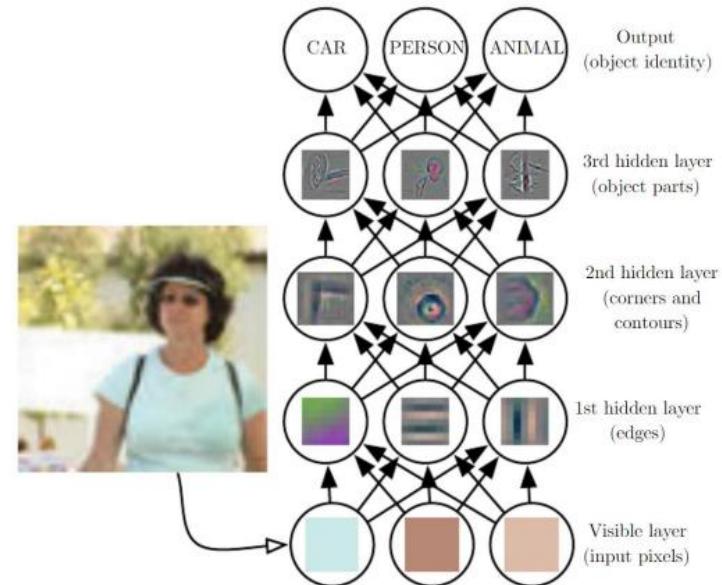
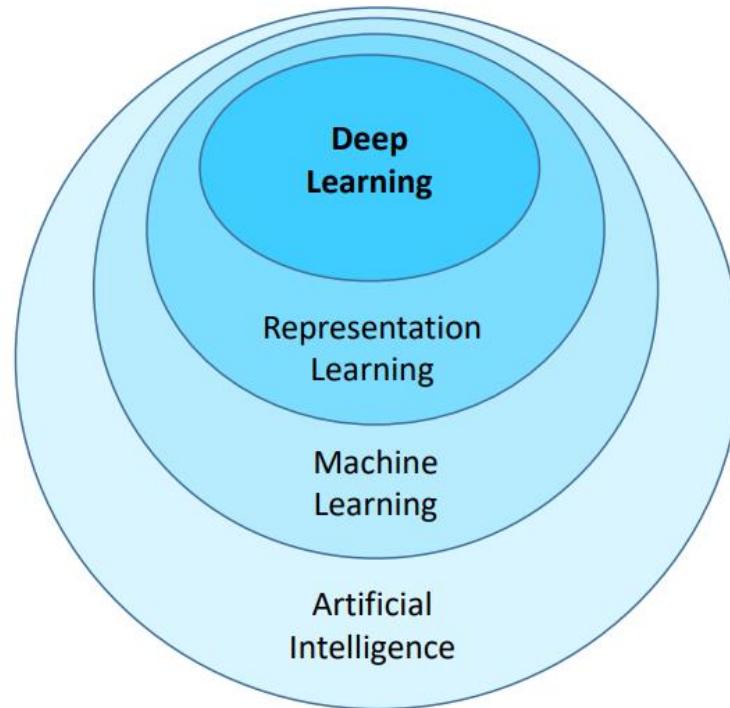
# Machine Learning v Deep Learning



Images Source: Google Image

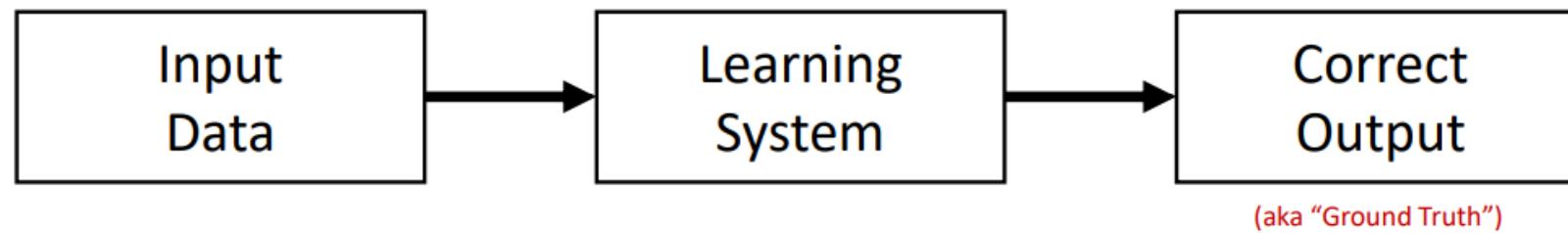
# Deep Learning is Representation Learning

(aka Feature Learning)

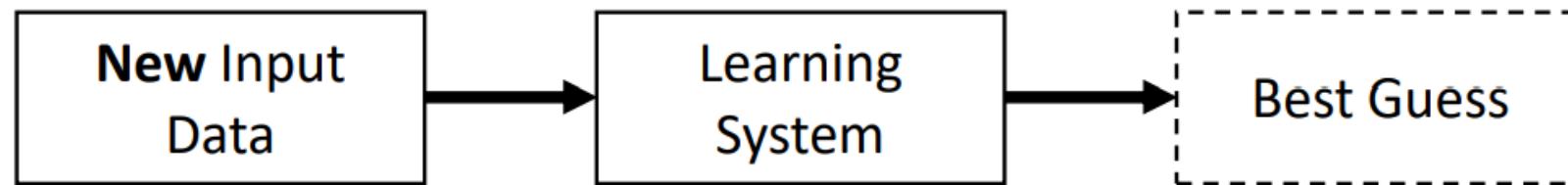


# Deep Learning: Training and Testing

## Training Stage:

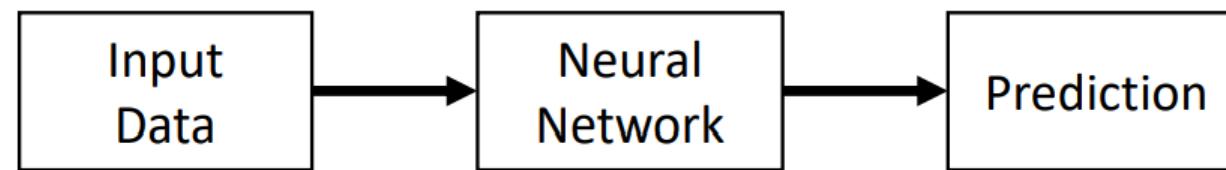


## Testing Stage:

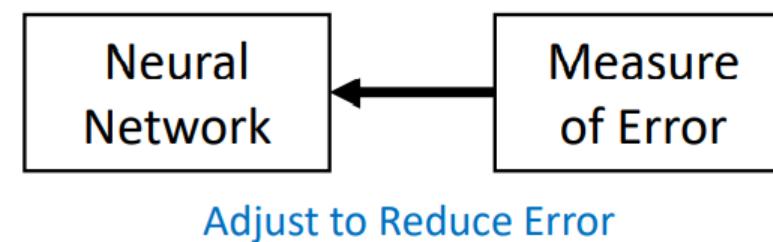


## How Neural Networks Learn: Backpropagation

Forward Pass:



Backward Pass (aka Backpropagation):



## Regression vs Classification



### Regression

What is the temperature going to be tomorrow?

PREDICTION

84°

Fahrenheit  
°F



### Classification

Will it be Cold or Hot tomorrow?

PREDICTION

COLD

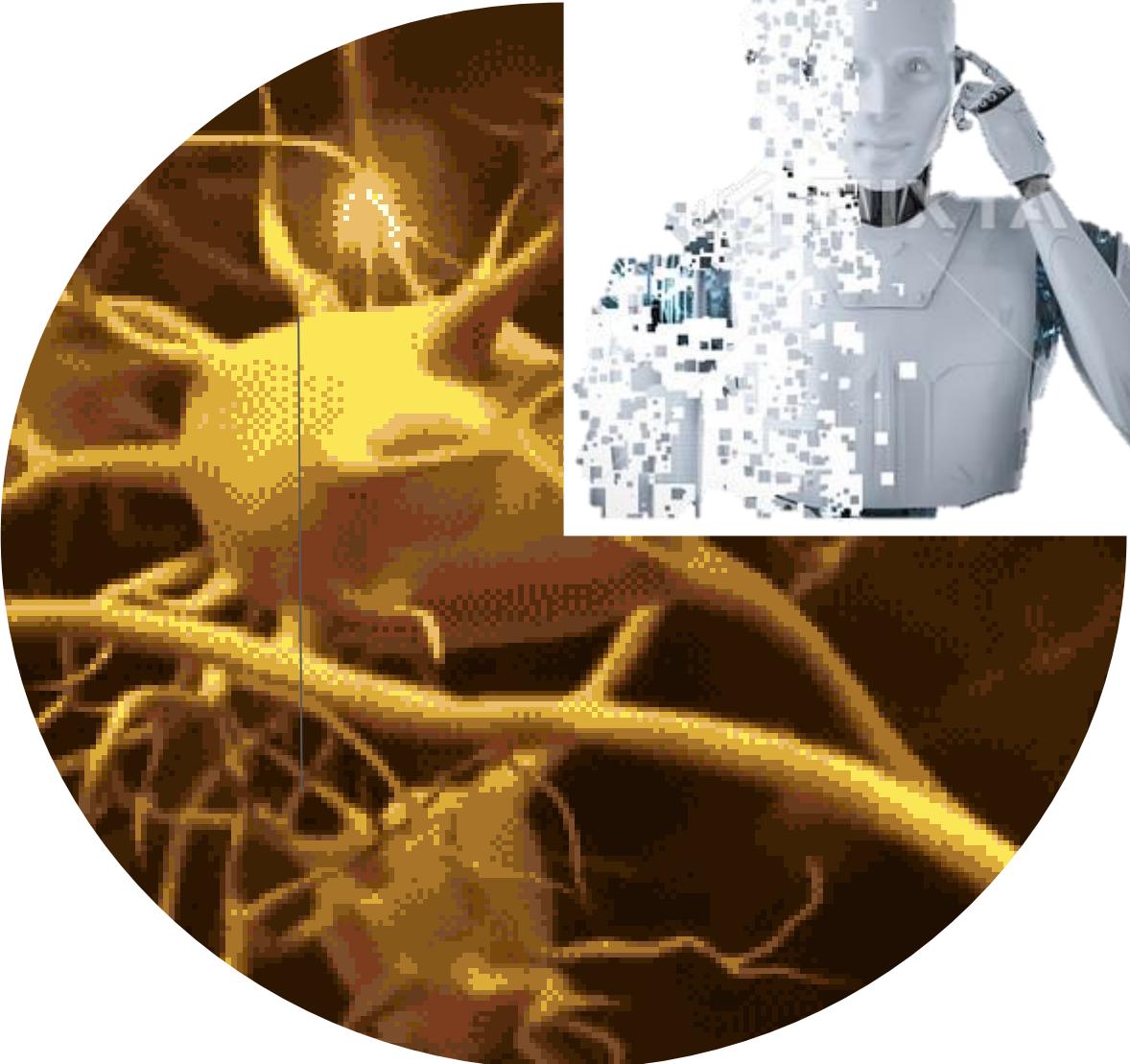
HOT

Fahrenheit  
°F

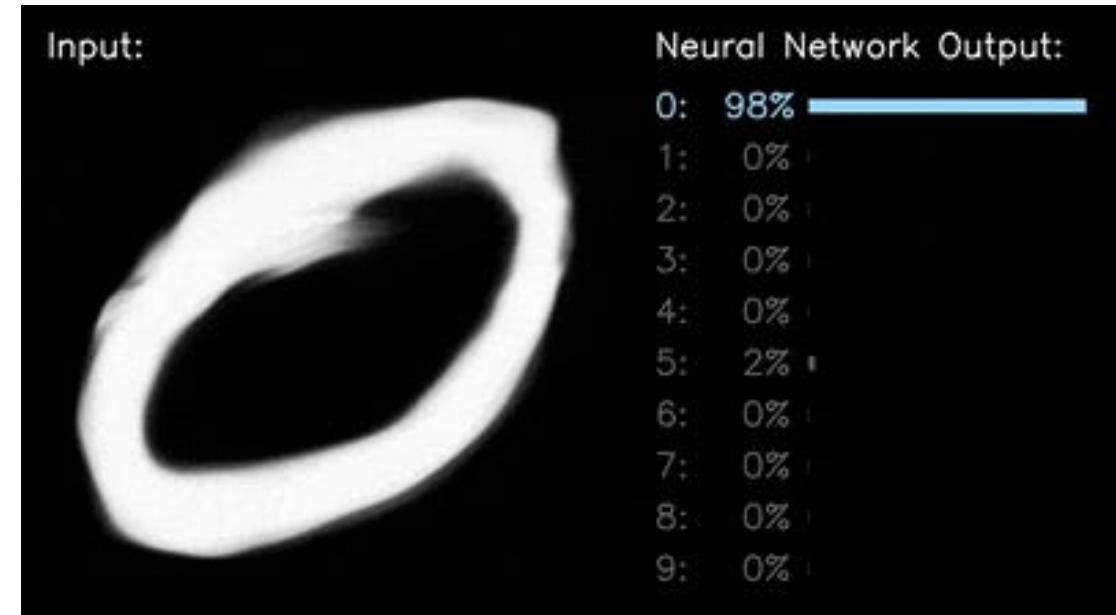
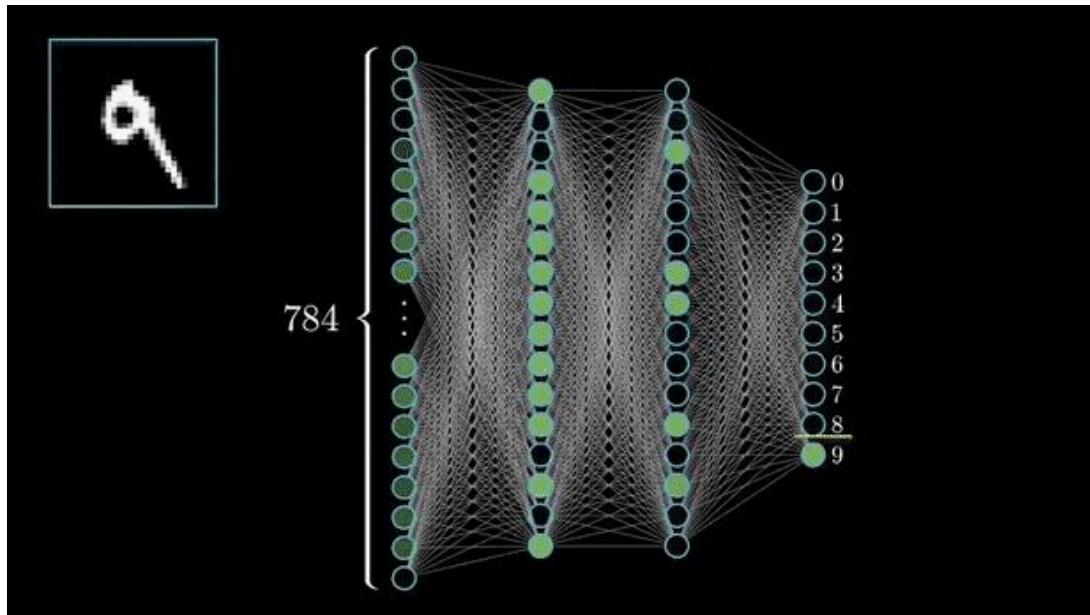


## Deep Learning; Example

---

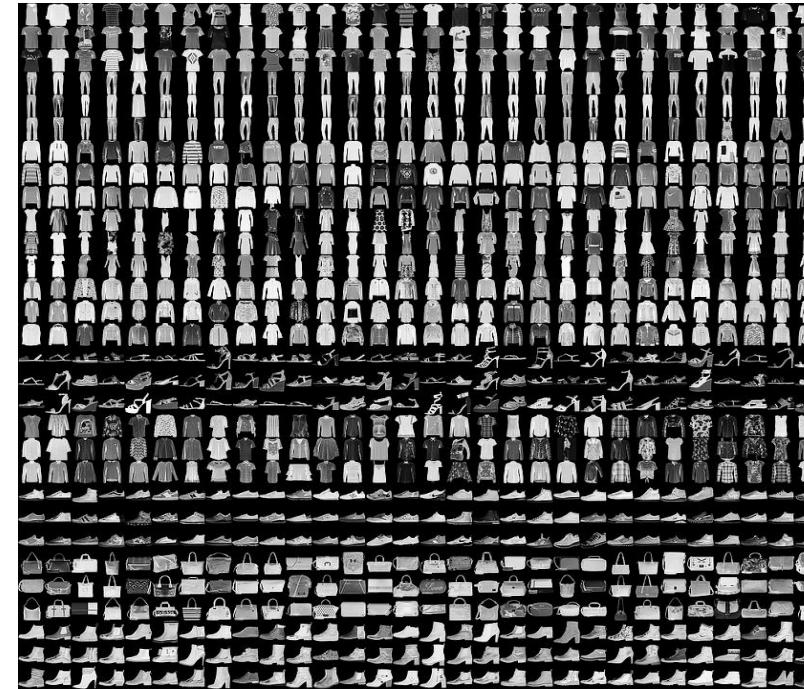


# Digit Classification MNIST Dataset



<https://colab.research.google.com/drive/12Go8c7bA2vFlZzAdLdpauwpBP04QOKGA?usp=sharing>

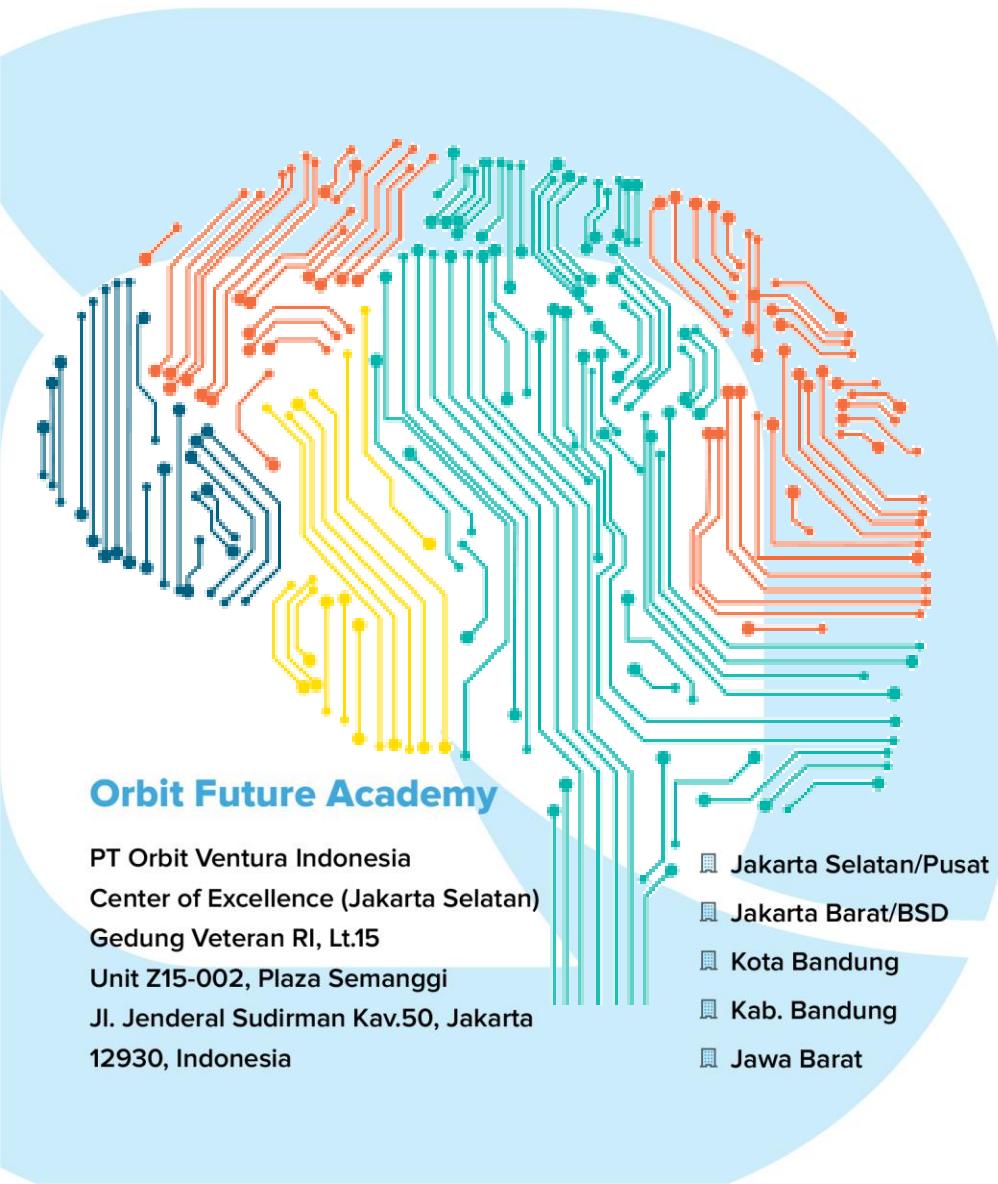
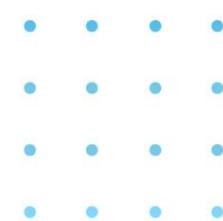
# The Fashion MNIST Dataset



Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY | Skills  
For Future Jobs

<https://colab.research.google.com/github/lmoroney/mlday-tokyo/blob/master/Lab2-Computer-Vision.ipynb>



## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

# TERIMA KASIH

## THANK YOU

## Social Media

- [Orbit Future Academy](#)
- [@OrbitFutureAcademyIn1](#)
- [OrbitFutureAcademy](#)
- [Orbit Future Academy](#)

# AI Mastery Course

- • •
- • •
- • •
- • •



*Artificial Intelligence  
Mastery Program*

## Module 3

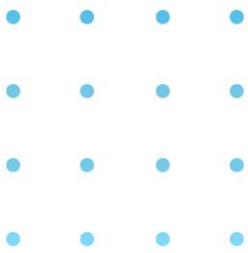
Data analytics with python-  
applied analytics

### Section

Data manipulation  
with pandas



- • •
- • •
- • •
- • •



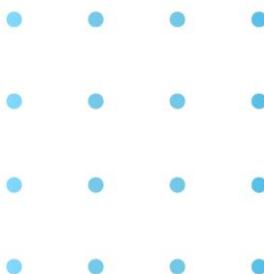
# Learning Objectives

Di akhir modul ini, Anda akan dapat:

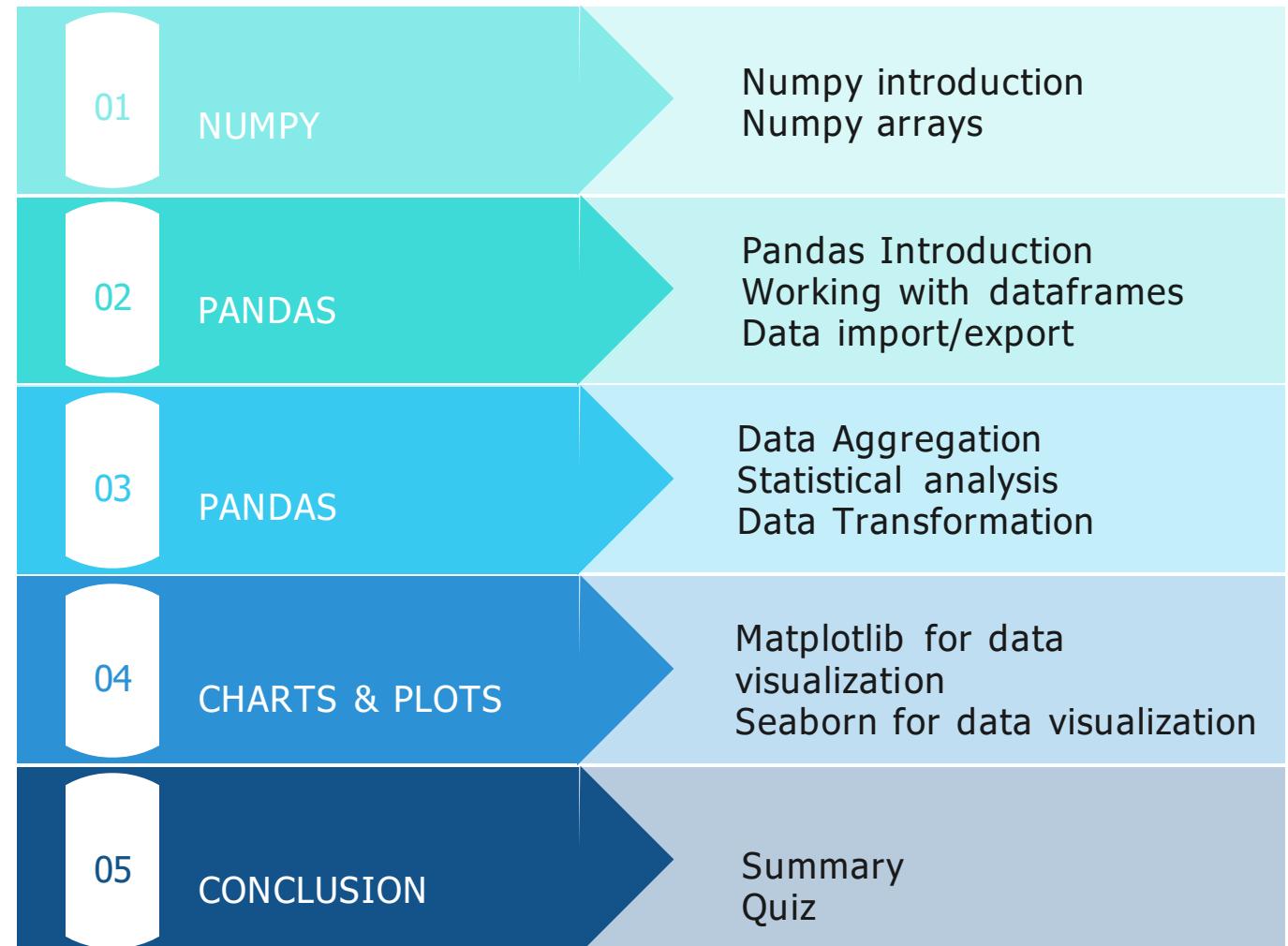
- Pengertian numpy dan penggunaannya.
- Mempelajari dasar-dasar preprocessing data.
- Mempelajari prosedur instalasi dan konsep dasar pada Pandas.
- Memahami data import export, selection dan filtering dengan dataframes.
- Data transformation menggunakan pandas.



*Artificial Intelligence  
Mastery Program*



# Agenda





01

## NUMPY

- Numpy introduction
- Numpy arrays

# Numpy Introduction

A powerful N-dimensional array object.

Sophisticated  
(broadcasting/universal)  
functions.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data.



Useful linear algebra,  
Fourier transform, and  
random number  
capabilities.

# Numpy data types

NumPy mendukung lebih luas berbagai type data jika dibandingkan dengan Python. Numpy didefinisikan oleh `numpy.dtype` class dan meliputi:

- `intc` (sama dengan data integer di bahasa C) and `intp` (used for indexing)
- `int8`, `int16`, `int32`, `int64`
- `uint8`, `uint16`, `uint32`, `uint64`
- `float16`, `float32`, `float64`
- `complex64`, `complex128`
- `bool_`, `int_`, `float_`, `complex_` are shorthand for defaults.

Ini dapat digunakan sebagai fungsi untuk cast literals atau sequence types, Serta argumen untuk numpy functions, yang menerima Argument kata kunci `dtype`.



Data type	Description
<code>bool_</code>	Boolean (True or False) stored as a byte
<code>int_</code>	Default integer type (same as C <code>long</code> ; normally either <code>int64</code> or <code>int32</code> )
<code>intc</code>	Identical to C <code>int</code> (normally <code>int32</code> or <code>int64</code> )
<code>intp</code>	Integer used for indexing (same as C <code>ssize_t</code> ; normally either <code>int32</code> or <code>int64</code> )
<code>int8</code>	Byte (-128 to 127)
<code>int16</code>	Integer (-32768 to 32767)
<code>int32</code>	Integer (-2147483648 to 2147483647)
<code>int64</code>	Integer (-9223372036854775808 to 9223372036854775807)
<code>uint8</code>	Unsigned integer (0 to 255)
<code>uint16</code>	Unsigned integer (0 to 65535)
<code>uint32</code>	Unsigned integer (0 to 4294967295)
<code>uint64</code>	Unsigned integer (0 to 18446744073709551615)
<code>float_</code>	Shorthand for <code>float64</code>
<code>float16</code>	Half-precision float: sign bit, 5 bits exponent, 10 bits mantissa
<code>float32</code>	Single-precision float: sign bit, 8 bits exponent, 23 bits mantissa
<code>float64</code>	Double-precision float: sign bit, 11 bits exponent, 52 bits mantissa
<code>complex_</code>	Shorthand for <code>complex128</code>
<code>complex64</code>	Complex number, represented by two 32-bit floats
<code>complex128</code>	Complex number, represented by two 64-bit floats

# Numpy Arrays



Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

```
1 import numpy as np
2 # Create an array
3 civilian_birth = np.array([4352, 233, 3245, 256, 2394])
4 civilian_birth

array([4352, 233, 3245, 256, 2394])

1 np.zeros((3, 6)) # special arrays with numpy

array([[0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0.]])]

1 np.linspace(0, 1, 5) # 0 to 1 (inclusive) with 5 points

array([0. , 0.25, 0.5 , 0.75, 1. ])

1 a = np.array([0, 1])
2 b = np.array([2, 3])
3 ab = np.stack((a, b)).T # stacking arrays
4 print(ab)

[[0 2]
 [1 3]]
```

# Linear Algebra with numpy

```
1 #(4x+5y=23)
2 #(6x-3y=3)
3
4 import numpy as np
5 A = np.array([[4,5],[6,-3]])
6 b = np.array([23, 3])
7 x,y = np.linalg.solve(A,b)
8 print("value of x = ",x," value of y = ",y)
```

```
value of x =  2.0
value of y =  3.0
```

```
1 matrix = np.matrix([[1, 5, 3], [9, 5, 6], [2, 8, 9]])
2 matrix.flatten() # Flatten matrix

matrix([[1, 5, 3, 9, 5, 6, 2, 8, 9]])
```

```
1 np.linalg.det(matrix) # Return determinant of matrix
-162.00000000000009
```

```
1 matrix.diagonal().sum() # Calculate the trace of the matrix
15
```

```
1 np.linalg.inv(matrix) # Calculate inverse of matrix

matrix([[ 0.01851852,  0.12962963, -0.09259259],
       [ 0.42592593, -0.01851852, -0.12962963],
       [-0.38271605, -0.01234568,  0.24691358]])
```



02

## PANDAS

- Pandas Introduction
- Working with dataframes
- Data import/export

# Pengenalan Pandas



Pandas adalah library Python bersifat open source yang banyak digunakan sebagai tools analisis data pada Python.

Pandas adalah tools yang sangat cocok untuk analisis dan pemodelan data.

Pandas adalah Library Python yang lebih baik dari NumPy dari sisi fungsionalitas, dan menyediakan implementasi DataFrame yang efisien.

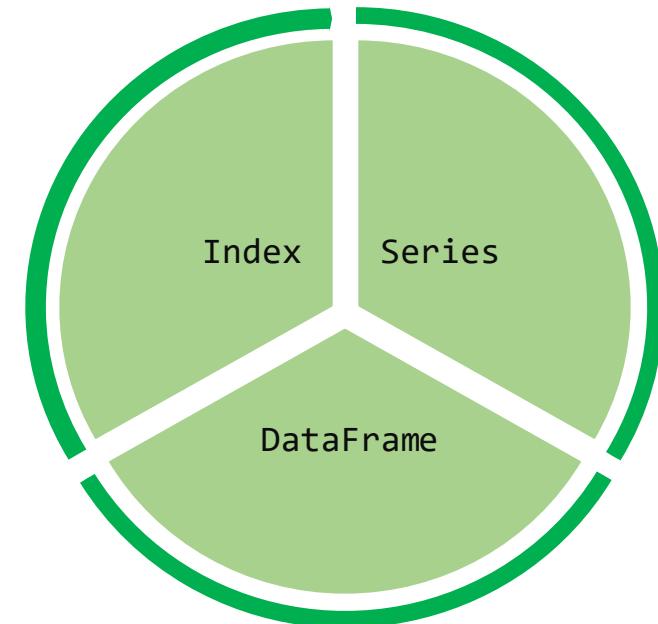
DataFrames pada dasarnya adalah array multidimensi dengan label baris dan kolom.

# Pandas objects

Objek Pandas adalah versi array terstruktur NumPy yang disempurnakan, di mana baris dan kolom diidentifikasi menggunakan label daripada indeks integer sederhana.

Ada banyak alat, metode, dan fungsi berguna yang ditawarkan Panda, selain dari struktur data dasar.

Tiga struktur data dasar pada Pandas, seperti pada gambar disamping kanan:



# Pandas series and dataframes

Dimensions	Name	Description
1	Series	1D labeled homogeneously-typed array
2	DataFrame	General 2D labeled, size-mutable tabular structure with potentially heterogeneously-typed column

The image shows a laptop screen with two code snippets and a resulting DataFrame visualization.

**Code Snippet 1:**

```
1 import pandas as pd
2 data = ['a', 'b', 'c', 'd']
3 s = pd.Series(data) # creating a pandas series
4 print(s)
```

**Output 1:**

```
0    a
1    b
2    c
3    d
dtype: object
```

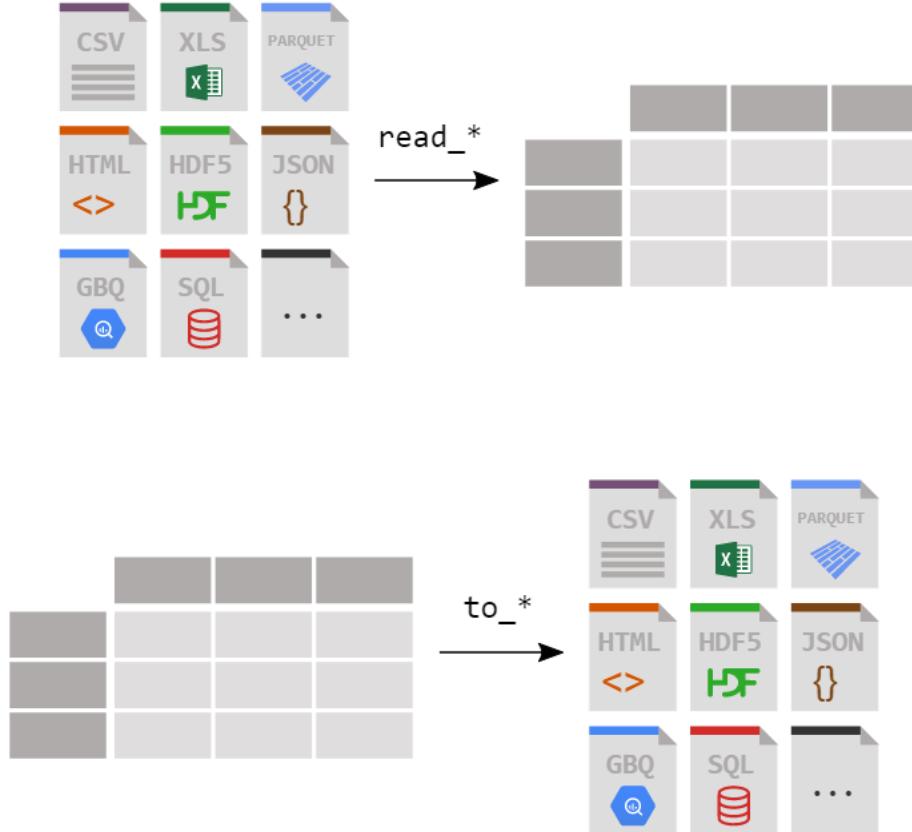
**Code Snippet 2:**

```
1 data = {'name': ['Jason', 'Molly', 'Tina', 'Jake'],
2         'year': [2012, 2012, 2013, 2014], 'reports': [4, 24, 31, 2]}
3 index = ['Cochice', 'Pima', 'Santa Cruz', 'Maricopa']
4 df = pd.DataFrame(data, index = index) # Create an dataframe
5 df
```

**Output 2:**

	name	year	reports
Cochice	Jason	2012	4
Pima	Molly	2012	24
Santa Cruz	Tina	2013	31
Maricopa	Jake	2014	2

# Pandas: Data import, export



Format Type	Data Description	Reader	Writer
text	CSV	read_csv	to_csv
text	Fixed-Width Text File	read_fwf	
text	JSON	read_json	to_json
text	HTML	read_html	to_html
text	LaTeX		Styler.to_latex
text	XML	read_xml	to_xml
text	Local clipboard	read_clipboard	to_clipboard
binary	MS Excel	read_excel	to_excel
binary	OpenDocument	read_excel	
binary	HDF5 Format	read_hdf	to_hdf
binary	Feather Format	read_feather	to_feather
binary	Parquet Format	read_parquet	to_parquet
binary	ORC Format	read_orc	
binary	Stata	read_stata	to_stata
binary	SAS	read_sas	
binary	SPSS	read_spss	
binary	Python Pickle Format	read_pickle	to_pickle
SQL	SQL	read_sql	to_sql
SQL	Google BigQuery	read_gbq	to_gbq

# Pandas: Data import, export

```
1 # Load a csv
2 url = 'https://raw.githubusercontent.com/anshupandey/WileyNXT/main/DataEngin
3 df = pd.read_csv(url)
4 df.head(2)
```

	Dates	Temperature	Humidity	Pressure	Air Quality
0	30-04-2018	218	182	4	2
1	01-05-2018	2592	182	3	2

```
1 # Load a JSON file
2 url = 'https://raw.githubusercontent.com/anshupandey/WileyNXT/main/DataEngin
3 df = pd.read_json(url)
4 df.head(2)
```

	id	dates	Accepted	Rejected	App	Server
0	0	1525046400000	218	182	App_4	Server_02
1	1	1525132800000	2592	182	App_3	Server_02

```
1 # Save dataframe as csv in the working director
2 df.to_csv('server_data.csv')
```

# Pandas: Selection & Filtering



Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

```
1 import pandas as pd
2 data = {'name': ['Jason', 'Molly', 'Tina'],
3         'year': [2012, 2012, 2013],
4         'reports': [4, 5, 9],
5         'coverage': [25, 62, 70]}
6 df = pd.DataFrame(data)
7 df['name'] # View Column
```

```
0 Jason
1 Molly
2 Tina
Name: name, dtype: object
```

```
1 df[df['coverage'] > 50] # Rows Where Coverage>50
```

	name	year	reports	coverage
1	Molly	2012	24	62
2	Tina	2013	31	70

```
1 df[['name', 'reports']] # View Two Columns
```

```
name reports
0 Jason      4
1 Molly      24
2 Tina       31
```

```
1 # View Rows Where Coverage > 20 & Reports < 6
2 df[(df['coverage'] > 20) & (df['reports'] < 6)]
```

	name	year	reports	coverage
0	Jason	2012	4	25

```
1 df[:2] # View First Two Rows
```

```
name year reports coverage
0 Jason 2012      4      25
1 Molly 2012     24      62
```



03

## PANDAS

- Data Aggregation
- Statistical analysis
- Data Transformation

# Data Aggregation: groupby

```
1 import pandas as pd
2 data = {'name': ['Jen', 'Max', 'Tina', 'John', 'Kay', 'Ish', 'Ami'],
3         'year': [2012, 2012, 2013, 2012, 2013, 2014, 2014],
4         'score': [4, 5, 9, 8, 9, 5, 4]}
5 df = pd.DataFrame(data)
6
7 # calculating year wise average score
8 df.groupby(['year'])['score'].mean()
```

```
year
2012    5.666667
2013    9.000000
2014    4.500000
Name: score, dtype: float64
```

# Pandas: Statistical analysis

```
1 import pandas as pd
2 data = {'name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
3         'age': [42, 52, 36, 42, 73],
4         'score': [4, 24, 31, 2, 3]}
5 df = pd.DataFrame(data)
6 df.describe().T #Summary statistics
```

	count	mean	std	min	25%	50%	75%	max
age	5.0	49.0	14.594520	36.0	42.0	42.0	52.0	73.0
score	5.0	12.8	13.663821	2.0	3.0	4.0	24.0	31.0

```
1 print("mean of score ");print(df['score'].mean())
2 print("Median of score ");print(df['score'].median())
3 print("Mode of age ");print(df['age'].mode())
4 print("Variance of age ");print(df['age'].var())
5 print("Skewness of age ");print(df['age'].skew())
```

```
mean of score
12.8
Median of score
4.0
Mode of age
0    42
dtype: int64
Variance of age
213.0
Skewness of age
1.4700997201463943
```

# Pandas : Transformasi Data menggunakan map



Teknik ini diterapkan jika kita perlu melakukan transformasi apa pun berdasarkan nilai dalam array, Seri, atau kolom dalam DataFrame.



Map metodologi pada Seri mengambil fungsi atau objek seperti dikte yang berisi pemetaan.



Menggunakan map method adalah cara yang nyaman untuk melakukan transformasi elemen-bijaksana dan operasi pembersihan data terkait lainnya.



Hasil yang sama juga dapat dicapai dengan melewati fungsi.

# Pandas : Data Transformation using map

```
1 import pandas as pd
2 data = {'name': ['Jen K Forst', 'Max M Brown', 'Tina W Musk',
3                  'John M Gates', 'Kay W Carter', 'Ish T Sodhi'],
4          'score': [4, 5, 9, 8, 9, 5]}
5 df = pd.DataFrame(data)
6 df.T
```

	0	1	2	3	4	5
name	Jen K Forst	Max M Brown	Tina W Musk	John M Gates	Kay W Carter	Ish T Sodhi
score	4	5	9	8	9	5

```
1 def get_code(name):
2     return "".join([k[0].upper() for k in name.split(' ')])
3 # Creating a column with initials
4 df['code'] = df['name'].map(get_code)
5 df.T
```

	0	1	2	3	4	5
name	Jen K Forst	Max M Brown	Tina W Musk	John M Gates	Kay W Carter	Ish T Sodhi
score	4	5	9	8	9	5
code	JKF	MMB	TWM	JMG	KWC	ITS

# Pandas : Transformasi Data – mengganti nilai



Metode penggantian dapat dengan mudah digunakan untuk mengganti nilai.



Kita dapat mengganti nilai sentinel dengan NA menggunakan metode ganti, yang akan menghasilkan seri baru.



Untuk mengganti beberapa nilai sekaligus, kita dapat melewati daftar dan kemudian nilai pengganti.



Untuk menggunakan pengganti yang berbeda untuk setiap nilai, berikan daftar pengganti. Sebuah dikte juga dapat dilewatkan sebagai argumen.



Metode data.replace berbeda dari data.str.replace, yang melakukan elemen substitusi string

# Pandas : Data Transformation – replacing values

```
1 import pandas as pd
2 data = {'name': ['Jen K Forst', 'Max M Brown', 'Tina W Musk',
3                  'John M Gates', 'Kay W Carter', 'Ish T Sodhi'],
4         'score': [4, 5, 9, 8, 9, 5]}
5 df = pd.DataFrame(data)
6 df.T
```

	0	1	2	3	4	5
name	Jen K Forst	Max M Brown	Tina W Musk	John M Gates	Kay W Carter	Ish T Sodhi
score	4	5	9	8	9	5

```
1 df.replace("Jen K Forst",'Jennifer K Forest',inplace=True)
2 df.T
```

	0	1	2	3	4	5
name	Jennifer K Forest	Max M Brown	Tina W Musk	John M Gates	Kay W Carter	Ish T Sodhi
score	4	5	9	8	9	5

# Pandas : Data Transformation – discretization & binning



Data kontinu umumnya disretisasi atau dipisahkan menjadi “bins” Untuk analisis lebih lanjut.

Untuk membagi data menjadi bins, Kita bisa menggunakan cut function di pandas, yang akan mengembalikan objek Kategoris khusus.

Objek Kategoris mirip dengan array string, yang menunjukkan nama bin, yang secara internal berisi array, categories, menentukan nama kategori yang berbeda.

Fungsi lain yang disebut qcut, bins data berdasarkan kuanttil sampel. qcut menggunakan kuanttil sampel yang akan menghasilkan bins berukuran kira-kira sama.

# Pandas : Data Transformation – discretization & binning

```
1 import pandas as pd
2 data = {'name': ['Jen', 'Max', 'Tina', 'John', 'Kay', 'Ish', 'Ami'],
3         'year': [2012, 2012, 2013, 2012, 2013, 2014, 2014],
4         'score': [4, 5, 9, 8, 9, 5, 4]}
5 df = pd.DataFrame(data)
6
7 # creating a categorical attribute for score
8 df['score_cat'] = pd.cut(df['score'], bins=[0,3,6,10],
9                          labels=['Low','Medium','High'])
10 df.T
```

	0	1	2	3	4	5	6
name	Jen	Max	Tina	John	Kay	Ish	Ami
year	2012	2012	2013	2012	2013	2014	2014
score	4	5	9	8	9	5	4
score_cat	Medium	Medium	High	High	High	Medium	Medium



04

## CHARTS & PLOTS

- Matplotlib for data visualization
- Seaborn for data visualization

# Data Visualization

Visualisasi Data digunakan untuk mengkomunikasikan informasi dengan jelas dan efisien kepada pengguna dengan menggunakan grafik informasi seperti tabel dan bagan. Ini membantu pengguna dalam menganalisis sejumlah besar data dengan cara yang lebih sederhana. Itu membuat data kompleks lebih mudah diakses, dimengerti, dan dapat digunakan.



# Matplotlib



Matplotlib adalah salah satu paket Python paling populer yang digunakan untuk visualisasi data.

Matplotlib adalah cross-platform library Untuk membuat 2D plots dari data dalam array.

Matplotlib ditulis dengan Python dan menggunakan NumPy, ekstensi matematika numerik Python.

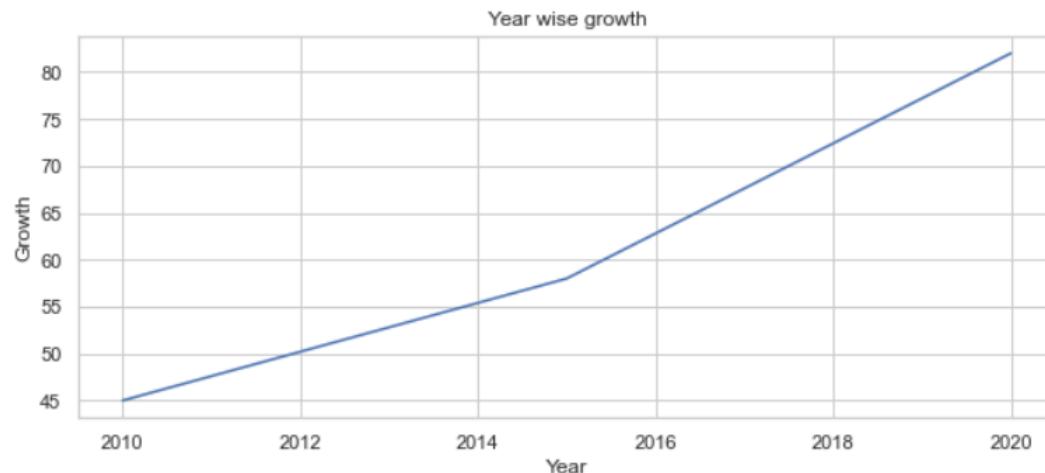
Matplotlib memberikan sebuah object-oriented API yang membantu dalam menanamkan plot dalam aplikasi menggunakan toolkit Python GUI seperti PyQt, WxPython or Tkinter.

# Matplotlib Functions

Sr. No	Function	Description
1	<b>Axes</b>	Add axes to the figure.
2	plt.text	Add text to the axes.
3	plt.title	Set a title of the current axes.
4	plt.xlabel	Set the x axis label of the current axis.
5	plt.xlim	Get or set the x limits of the current axes.
6	plt.xscale	Set the scaling of x axis
7	plt.xticks	Get or set the x-limits of the current tick locations and labels.
8	plt.yscale	Set the y axis label of the current axis.
9	plt.ylim	Get or set the y-limits of the current axes.
10	plt.yticks	Set the scaling of the y-axis.
11	plt.yticks	Get or set the y-limits of the current tick locations and labels.

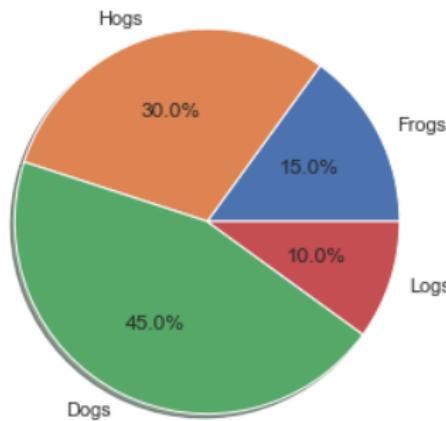
# Line plot with matplotlib

```
1 import matplotlib.pyplot as plt
2 year = [2010,2015,2020]
3 growth = [45,58,82]
4
5 plt.figure(figsize=(10,4))
6 plt.plot(year,growth)
7 plt.xlabel("Year")
8 plt.ylabel("Growth")
9 plt.title("Year wise growth")
10 plt.show()
```



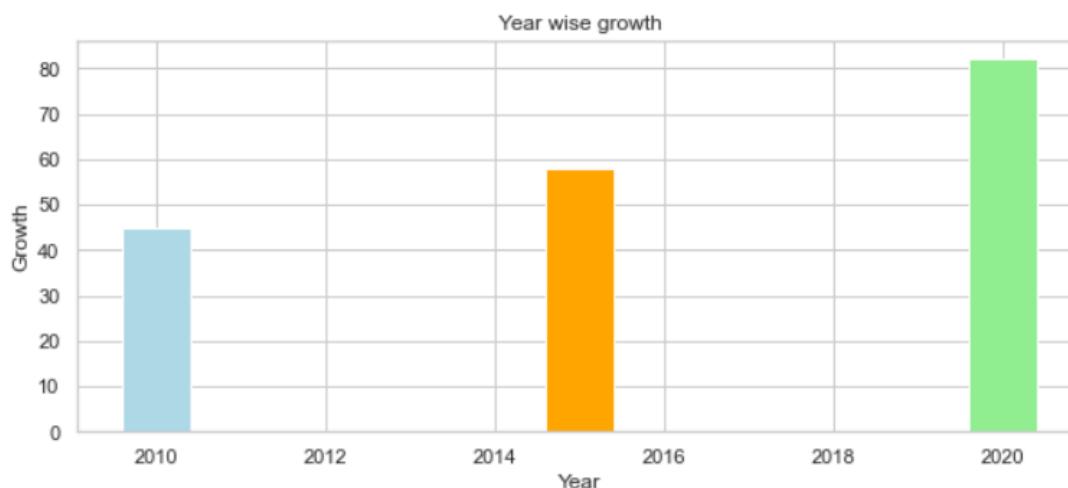
# Pie chart with matplotlib

```
1 import matplotlib.pyplot as plt
2 from matplotlib.gridspec import GridSpec
3
4 labels = 'Frogs', 'Hogs', 'Dogs', 'Logs'
5 fracs = [15, 30, 45, 10]
6 explode = (0, 0.05, 0, 0)
7
8 plt.figure(figsize=(5,5))
9 plt.pie(fracs, labels=labels, autopct='%1.1f%%', shadow=True)
10 plt.show()
```



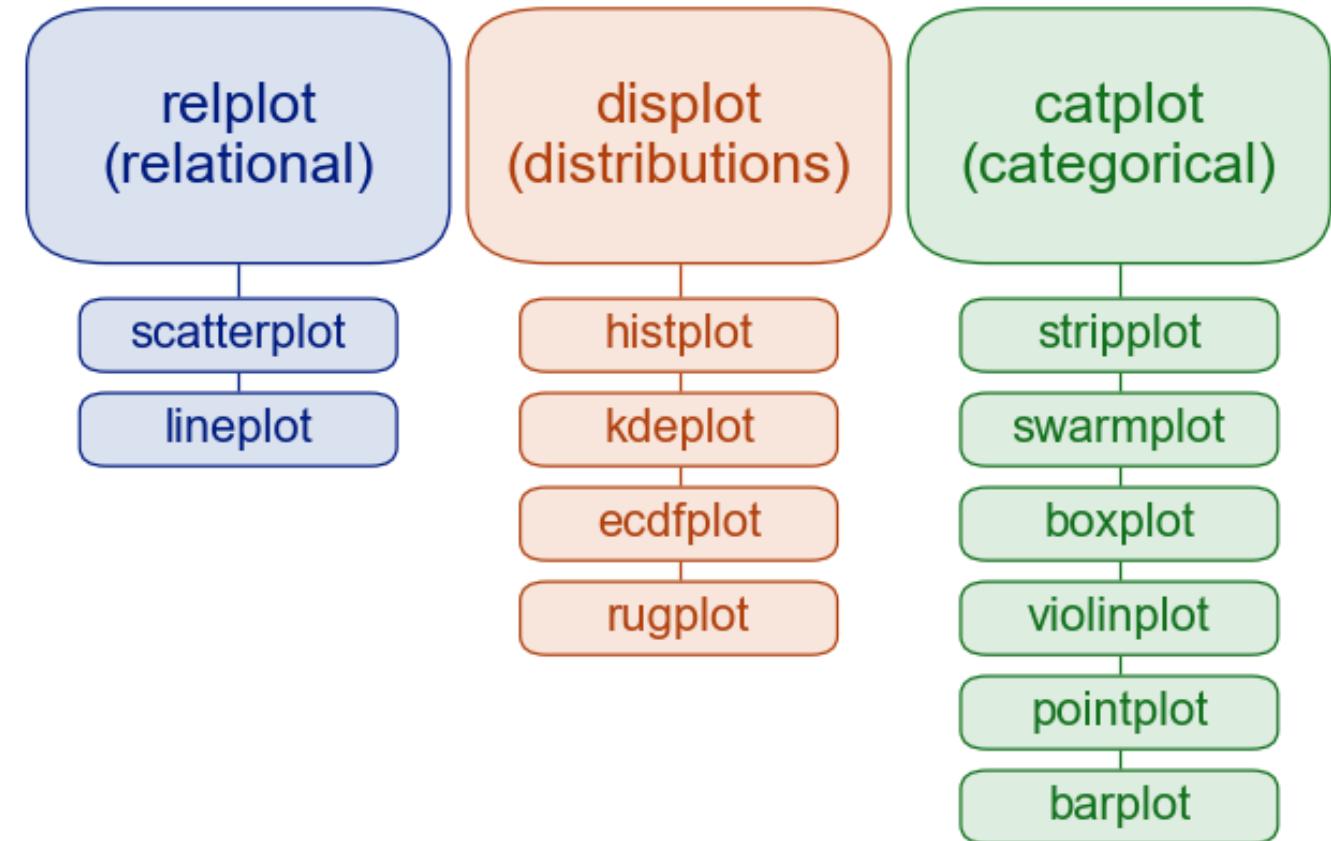
# Bar plot with matplotlib

```
1 import matplotlib.pyplot as plt
2 year = [2010,2015,2020]
3 growth = [45,58,82]
4
5 plt.figure(figsize=(10,4))
6 plt.bar(year,growth,color=['lightblue','orange','lightgreen'])
7 plt.xlabel("Year")
8 plt.ylabel("Growth")
9 plt.title("Year wise growth")
10 plt.show()
```



# Seaborn

Seaborn adalah library untuk membuat grafik statistik di Python. Seaborn dibangun di atas matplotlib dan terintegrasi dengan struktur data di Pandas.

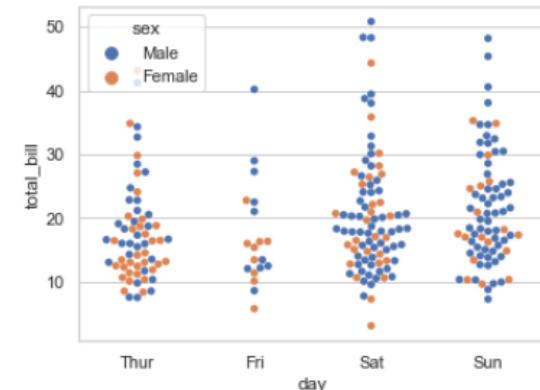
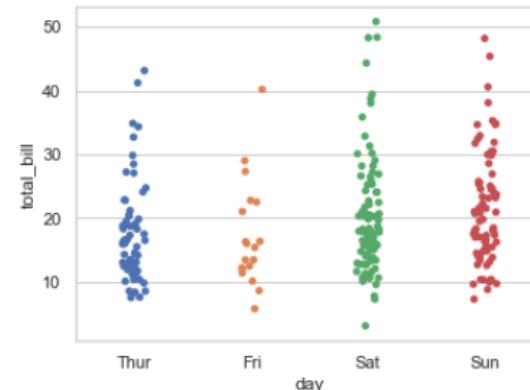


# Stripplot and swarmplot with seaborn

Strip plot digunakan untuk menganalisis numerik v/s categorical data

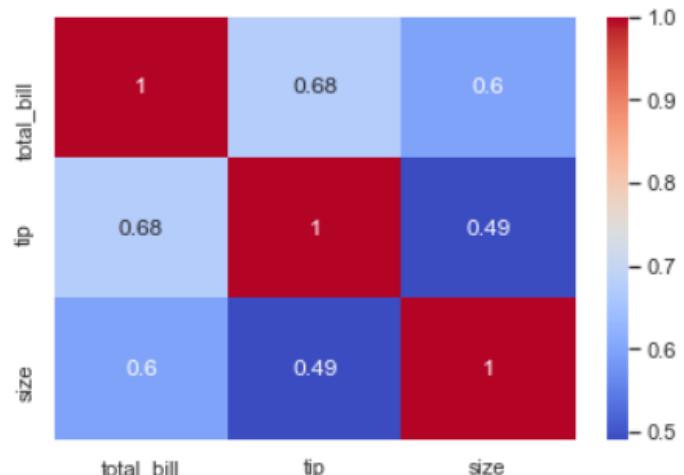
Swarm plot dapat digunakan untuk melakukan analisis multivariat antara numerik v/s categorical v/s categorical

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 sns.set(style="whitegrid", color_codes=True)
5 tips = sns.load_dataset("tips")
6
7 plt.figure(figsize=(12,4))
8 plt.subplot(121)
9 sns.stripplot(x="day", y="total_bill", data=tips, jitter=True)
10 plt.subplot(122)
11 sns.swarmplot(x="day", y="total_bill", hue="sex", data=tips)
12 plt.show()
```



# Heatmap with Seaborn

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3
4 sns.set(style="whitegrid", color_codes=True)
5 tips = sns.load_dataset("tips")
6
7 sns.heatmap(tips.corr(), annot=True, cmap='coolwarm')
8 plt.show()
9
```





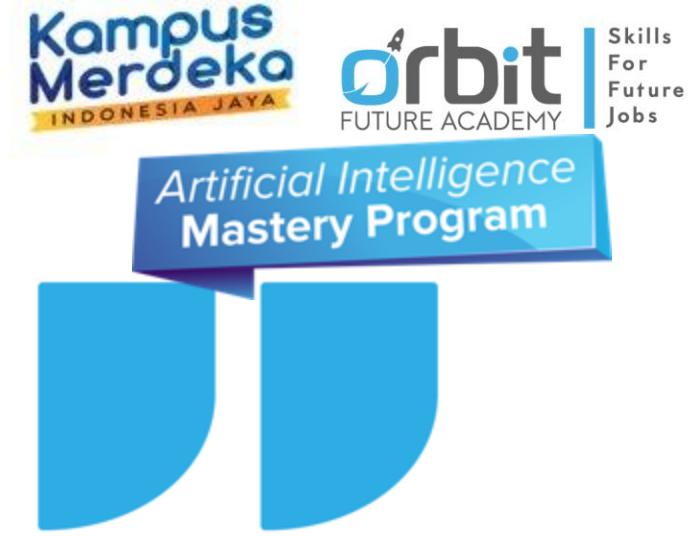
05

## CONCLUSION

- Summary

# Summary

1. Numpy adalah paket python yang banyak digunakan untuk perhitungan matematis dengan python.
2. Pandas dapat digunakan untuk melakukan data import, eksport, data cleaning, data wrangling, data aggregation.
3. Series dan dataframes adalah primary data types dalam python.
4. Matplotlib dan seaborn sebagai paket yang digunakan untuk visualisasi data dengan python.



# Quiz



## Question

manakah dari paket berikut yang tidak digunakan untuk visualisasi data?

- A.Matplotlib
- B.Seaborn
- C.Numpy
- D.plotly

# Quiz



## Question

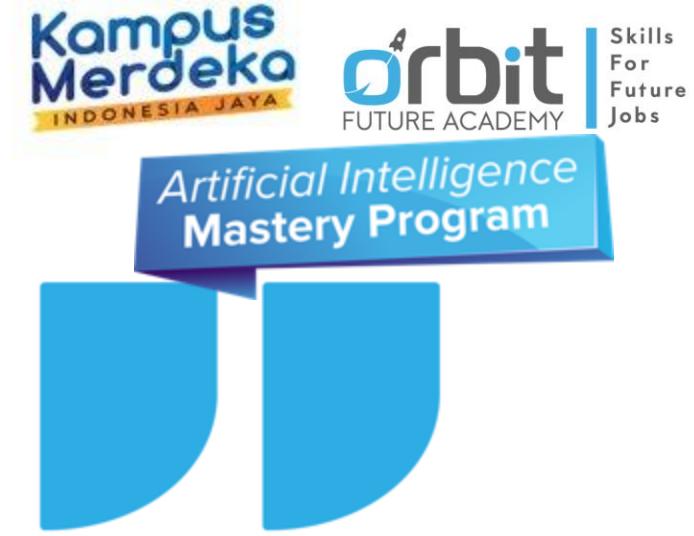
manakah dari paket berikut yang tidak digunakan untuk visualisasi data?

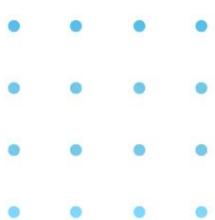
- A.Matplotlib
- B.Seaborn
- C.Numpy
- D.plotly

Answer: C

# Refference

- Data Analysis with Pandas, Stefani Molin
- Effective Pandas, Matt Harrison





Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY  
Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

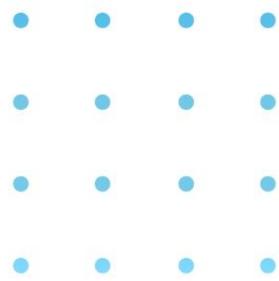
Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

- Orbit Future Academy
- @OrbitFutureAcademyIn1
- OrbitFutureAcademy
- Orbit Future Academy



# AI Mastery Course



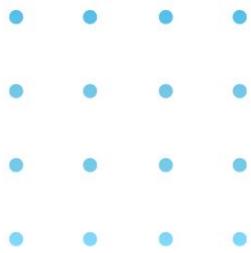
## Module 3

Data analytics with python- applied  
analytics

### Section

Exploratory Data Analytics (EDA)



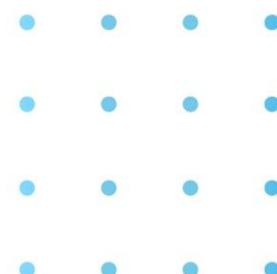


# Learning Objectives

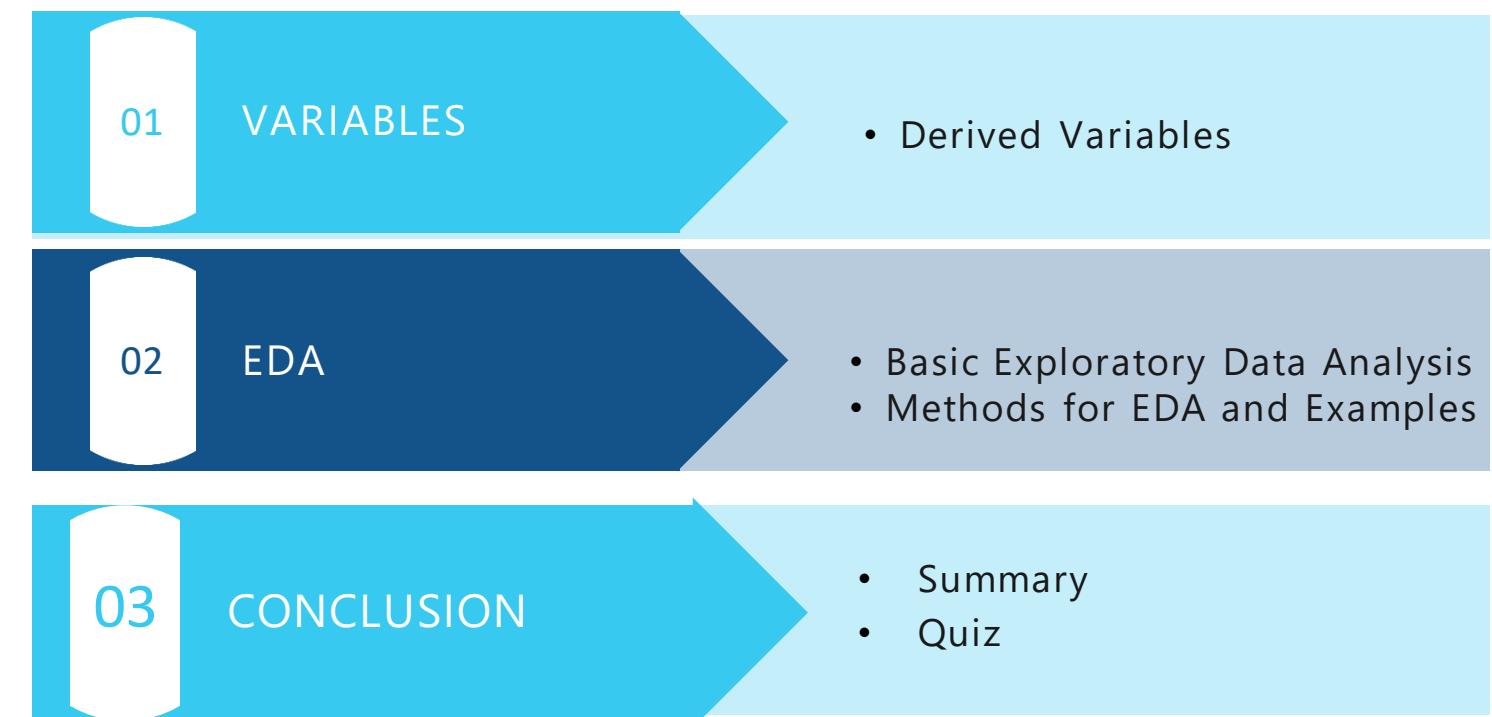
Di akhir modul ini, Anda akan dapat:

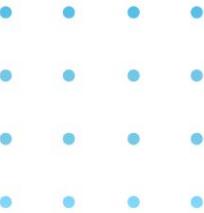


- Menjelaskan derived variables/ variabel turunan dan metode yang digunakan untuk menganalisisnya
- Memahami overview dari exploratory data analysis (EDA) dan manfaatnya
- Mengidentifikasi perbedaan metode yang tersedia di EDA dan 1 contohnya



# Agenda

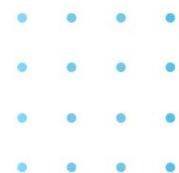




01

## VARIABLES

- Derived Variables



# Derived Variables/Variabel Turunan

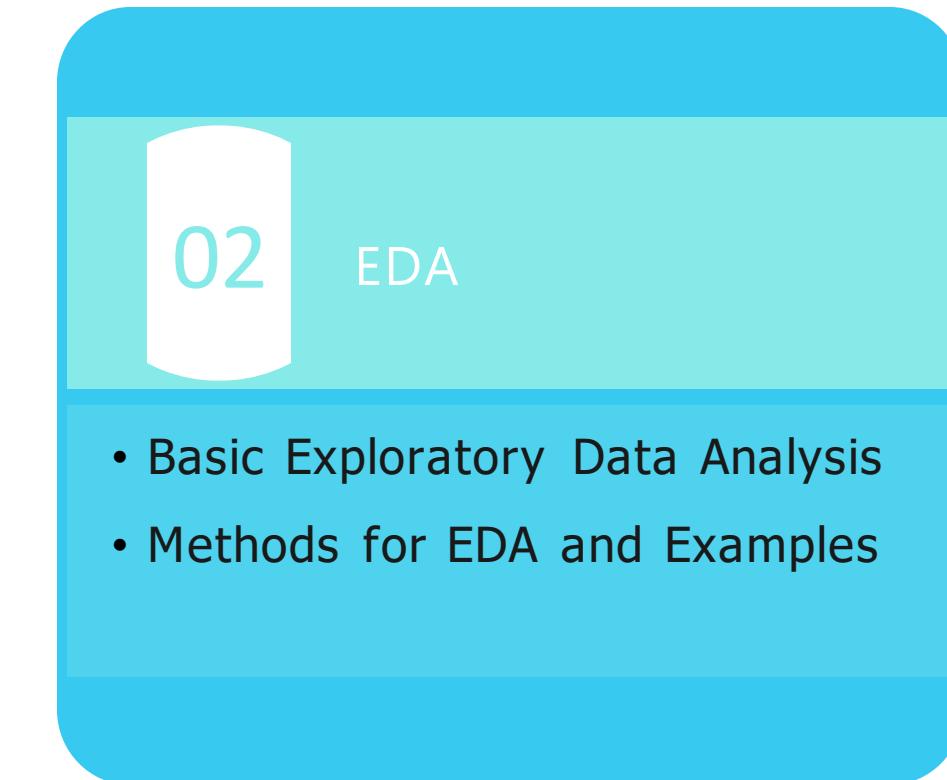
**Di bawah ini adalah beberapa poin penting dari variable turunan:**

- Dalam konteks statistical analysis, variable turunan merupakan variabel yang diturunkan dari dua atau lebih variabel utama
- Variabel turunan dibentuk dari melakukan kalkulasi atau kategorisasi dari variabel yang ada di dataset
- Beberapa contoh dari variabel turunan adalah rasio, persentase, index dan rate
- Variabel turunan berdasarkan pada variabel utama, maka perubahan yang terjadi pada variabel utama akan memiliki dampak pada variabel turunan
- Variable turunan bisa memiliki beberapa properti yang tidak diharapkan dan memerlukan metode penanganan yang berbeda

# Analysis of Derived Variables

**Beberapa hal terkait analisa dari variabel turunan.**

- Analisa dari variabel turunan memuat beberapa metode untuk mengambil sekumpulan pengukuran dan mengombinasikannya menjadi fitur ringkas yang penting
- Metode statistik digunakan untuk membentuk model atau prediksi, menghasilkan nilai yang diprediksi dan residual
- Nilai yang diprediksi adalah nilai yang diprediksi oleh model, dimana untuk kebanyakan kasus merupakan single variable. Kita bisa menganalisa variabel berikut juga relasinya terhadap variabel lain di model tersebut atau di luar model tersebut
- Residual merupakan hal dimana model tidak dapat prediksi. Residual digunakan untuk mediagnosa model dan mengidentifikasi masalah yang terkait dengan model tersebut
- Pada praktiknya, variabel turunan dibuat selama melakukan analisa statistik, dengan banyak variasi, yang tergantung pada teknik analisis yang digunakan
- Ada banyak tools yang tersedia untuk menganalisa variabel turunan



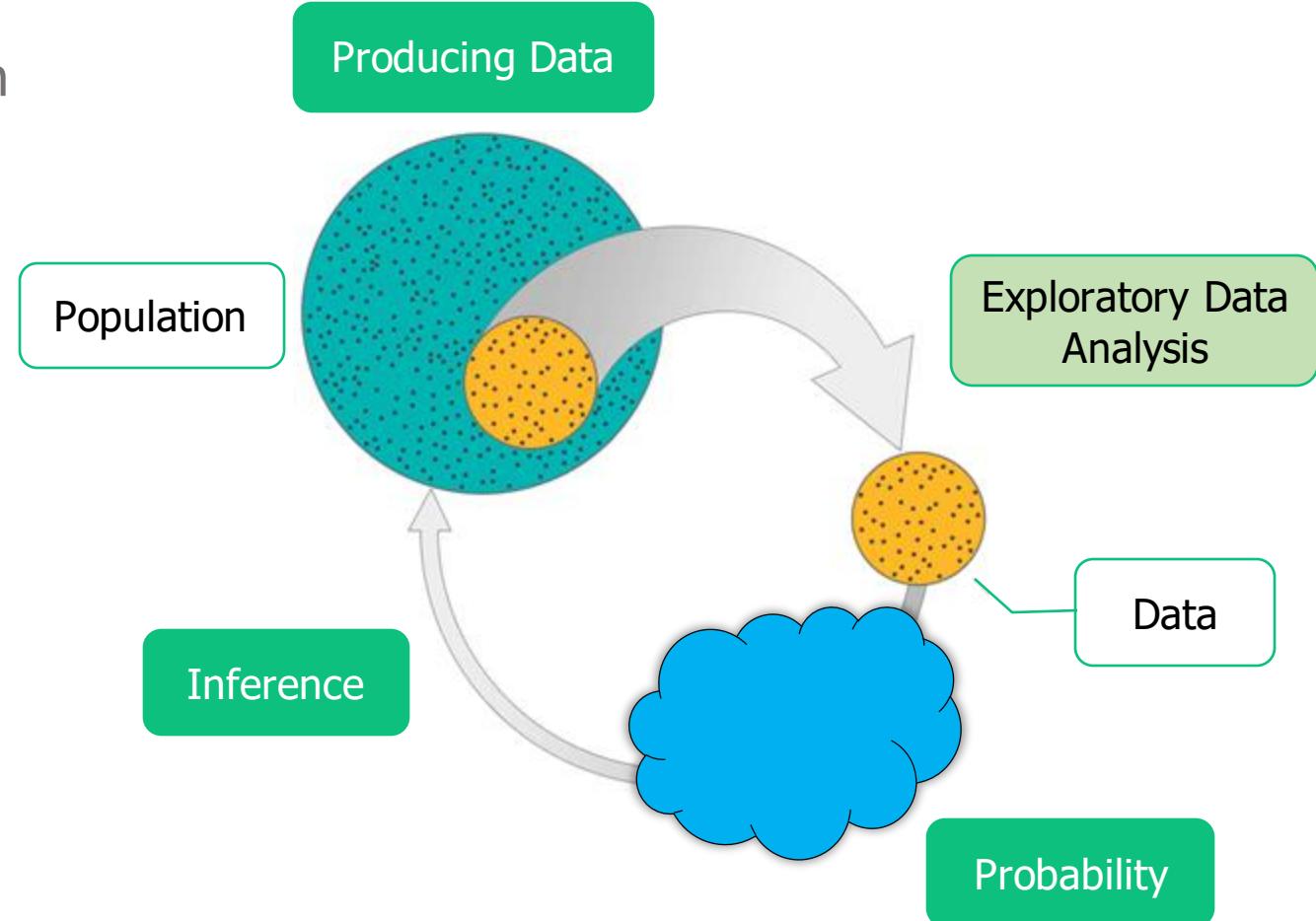
02 EDA

- Basic Exploratory Data Analysis
- Methods for EDA and Examples



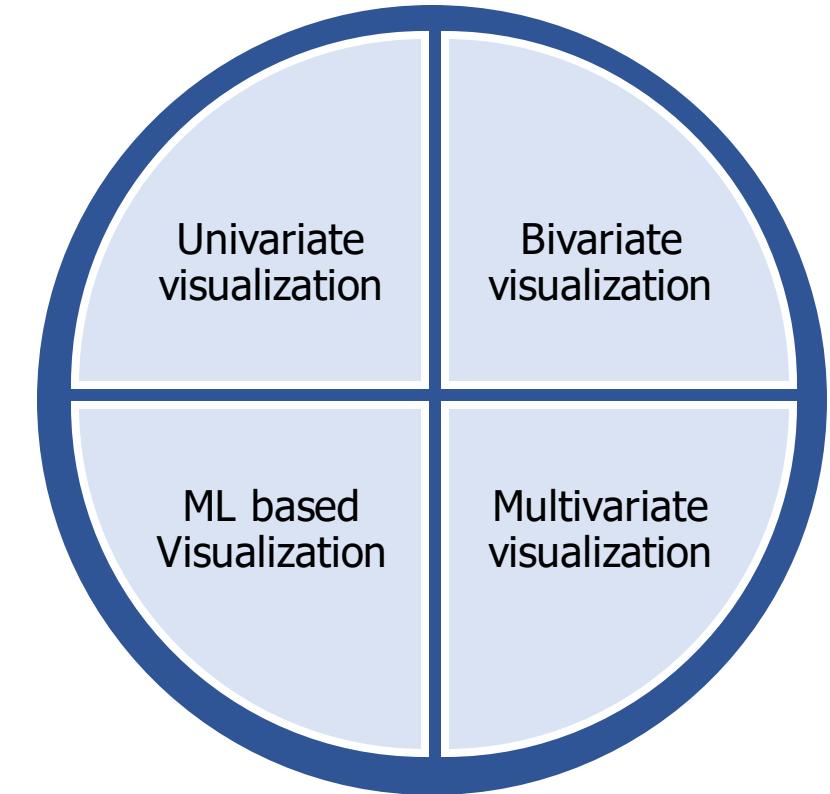
# Exploratory Data Analysis

- Exploratory data analysis (EDA) dilakukan pada sebuah data sebagai serangkaian upaya investigasi awal, untuk menemukan suatu pola, mendeteksi anomali, hypothesis testing dan untuk memvalidasi asumsi menggunakan ringkasan statistik dan representasi visual
- EDA digunakan untuk memastikan bahwa data yang dimiliki siap untuk digunakan oleh algoritma machine learning, dan membantu untuk menentukan algoritma yang tepat dari dataset yang ada
- EDA juga digunakan untuk menentukan feature variable yang dapat digunakan oleh machine learning



# Methods of Exploratory Data Analysis

- Ada beberapa teknik eksplorasi yang tersedia untuk melakukan EDA dan hasilnya bisa dikomparasikan
- Setelah melakukan analisa data dan memahaminya, kita bisa melakukan data collection dan data cleaning untuk mentransformasi data agar sesuai dengan business requirements



# EDA with Data Visualization

- Univariate plot biasa digunakan untuk menampilkan distribusi data dengan satu variabel. Variabel tersebut bisa berbentuk kategori (mis: jenis kelamin, ras, negara, kota, dll) atau berbentuk kuantitatif (mis: umur, berat badan, inflasi, penilaian, dll).
- Data Bivariate menampilkan hubungan antara dua variabel. Tipe grafik ini tergantung level pengukuran dari variabel (kategori atau kuantitatif)
- Data Multivariate menampilkan hubungan antara tiga atau lebih variabel.
- Dimensionality reduction digunakan untuk memahami fields dari data yang menjelaskan paling banyak varians antara data observasi dan kemungkinan volume data yang dikurangi
- Pola perilaku dapat lebih mudah diidentifikasi dengan melakukan pengelompokan (clustering) data observasi yang serupa pada dataset ke dalam pengelompokan yang berbeda (dengan cara mengelompokkan data menjadi beberapa kumpulan data kecil)

# Approaches and Techniques

**Categorical V/s  
Continuous**

1. Probability Distribution analysis - using distplot
2. Swarm plot and boxplot
3. ANOVA - f\_classif

**Continuous v/s  
continuous**

1. Scatterplot
2. Correlation Analysis
3. ANOVA - f\_regress

**Categorical v/s  
Categorical**

1. Barplot - of ratio of frequency/count in multiple groups
2. Chi Square Test
3. Histogram/countplot



03 CONCLUSION

- Summary
- Quiz



# Summary

1. Variabel turunan dan bagaimana cara menganalisanya
2. EDA untuk menemukan suatu pola, mendeteksi anomali, hypothesis testing dan untuk memvalidasi asumsi menggunakan ringkasan statistik dan representasi visual juga digunakan untuk menentukan feature variable yang dapat digunakan oleh machine learning
3. Ragam metode yang berbeda dan tersedia untuk EDA

# Quiz

- Mana dibawah ini yang bukan merupakan teknik dari EDA?

- A. Univariate visualization
- B. Bivariate visualization
- C. Multivariate visualization
- D. Logistic regression



# Quiz

- Mana dibawah ini yang bukan merupakan teknik dari EDA?

- A. Univariate visualization
- B. Bivariate visualization
- C. Multivariate visualization
- D. Logistic regression

**Jawaban: D**



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

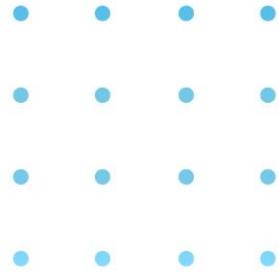
Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)



# AI Mastery Course



## Module 4

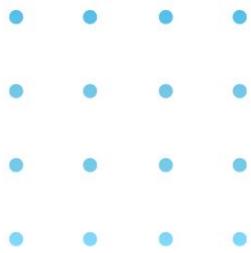
### Supervised Machine Learning and predictive modelling



#### Section

##### Logistic regression





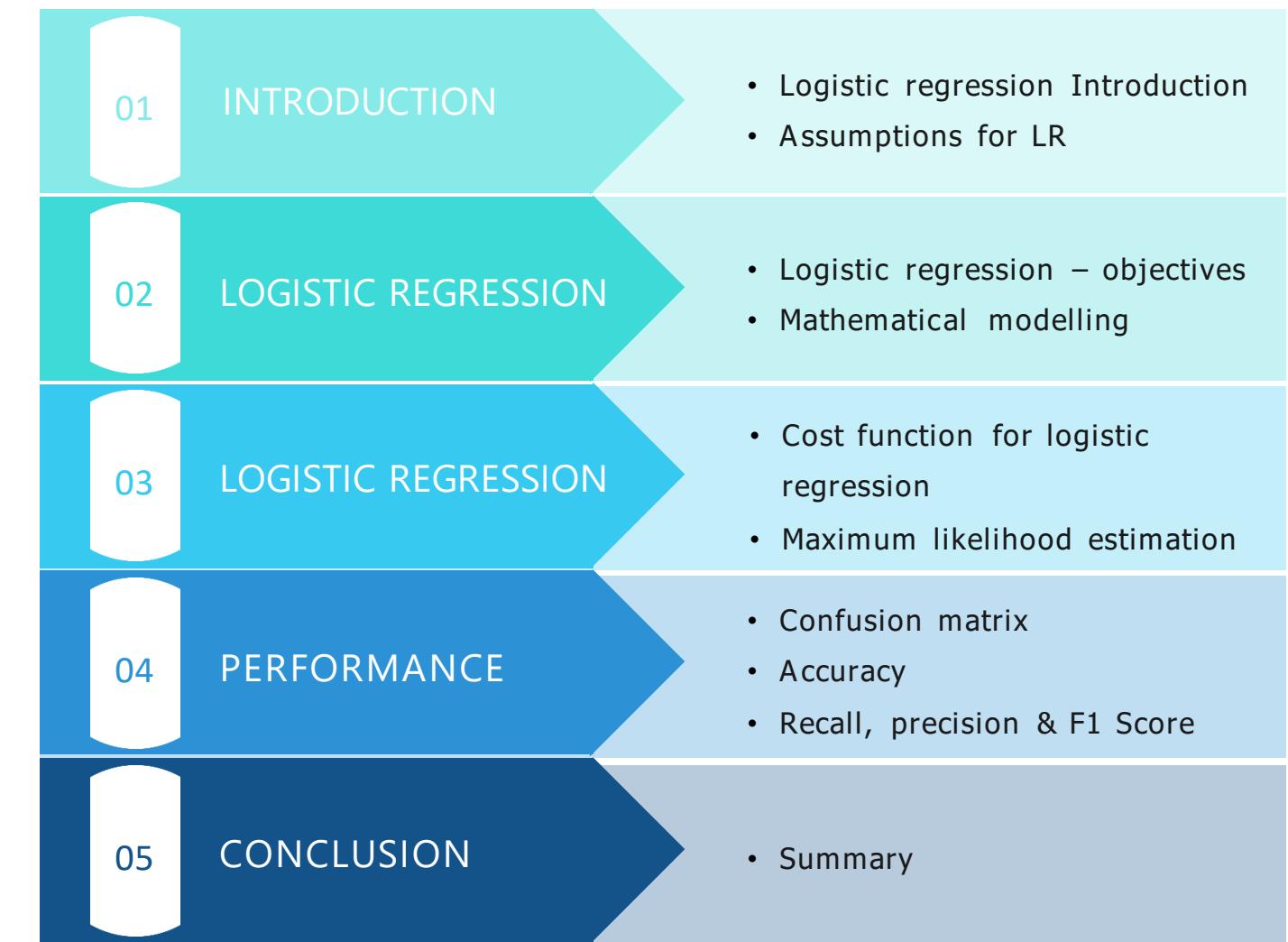
# Learning Objectives

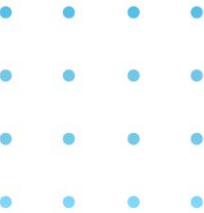
Di akhir modul ini, Anda akan dapat:

- Memahami algoritma Logistic Regression
- Menjelaskan intuisi matematika di balik Logistic Regression
- Memahami Cost function untuk Logistic Regression
- Memahami Maximum likelihood estimation
- Melakukan analisa performa dari persoalan klasifikasi



# Agenda





## 01

### INTRODUCTION

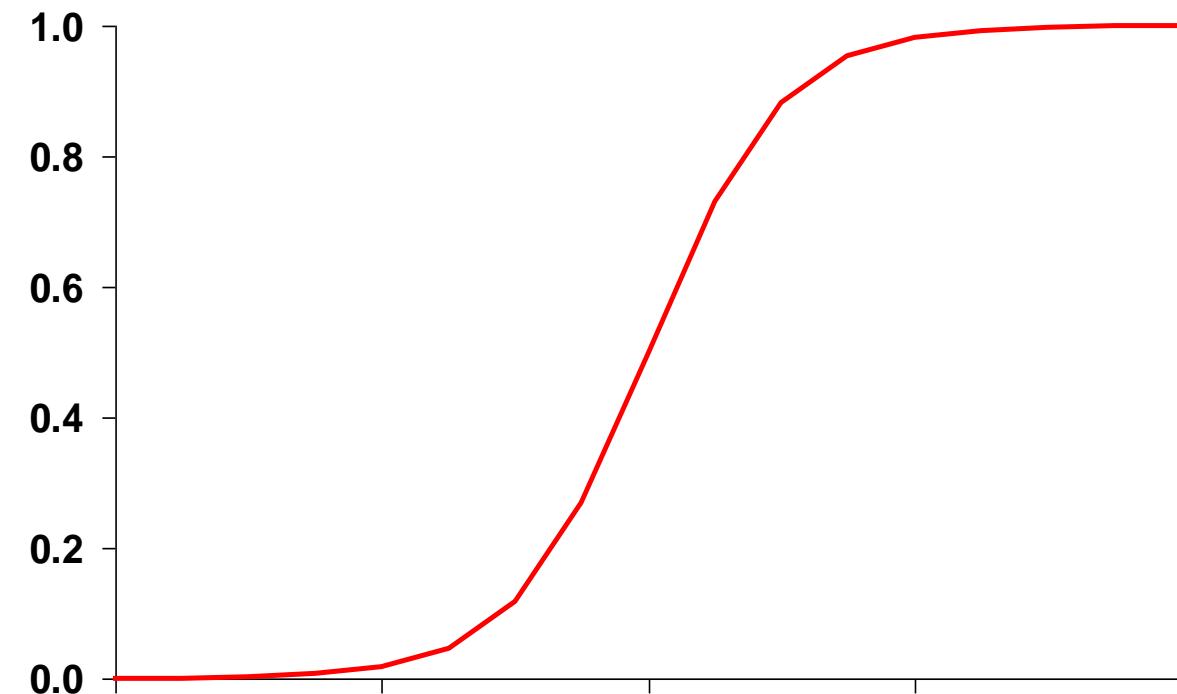
- Logistic regression Introduction
- Assumptions for LR



# What is Logistic Regression?

- Logistic Regression adalah salah satu kelas klasifikasi yang melakukan pemodelan terhadap probabilitas dari output kelas
- Algoritma ini mengestimasi relasi antara dependent variable (target/label) dan 1 atau lebih independent variable (predictors) dimana dependent variable nya bersifat categorical.

$$P(y|x) = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}}$$

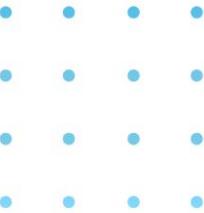


# Logistic Regression

1. Mengharapkan relasi linear yang "smooth" dari predictors.
2. Logistic Regression akan menghasilkan probabilitas dari outcome yang bersifat diskrit (yes or not, 1/0).
3. Memiliki sedikit resiko terhadap over-fitting.
4. Logistic Regression digunakan untuk menyelesaikan masalah klasifikasi.

# Assumptions

- \* Binary Logistic Regression mengharuskan dependent variable berbentuk binary.
- \* Untuk binary regression, dependent variable semestinya merepresentasikan output yang diharapkan (yes or not, 1 or 0).
- \* Hanya variables yang bernilai/berdampak yang semestinya dilibatkan.
- \* Independent variable bersifat independen satu sama lainnya sehingga model hanya memiliki sedikit atau bahkan tidak memiliki masalah multicollinearity.
- \* Logistic regression memerlukan ukuran sample yang cukup besar.



02

## LOGISTIC REGRESSION

- Logistic regression – objectives
- Mathematical modelling



# Objective of Logistic Regression

## Binary Classification:

1. Prediksi apakah spam atau tidak dengan memanfaatkan informasi subjek dan isi email
2. Prediksi cerah atau hujan, menggunakan informasi cuaca
3. Berdasarkan data historis dari kustomer bank, memprediksi apakah pihak bank sebaiknya memberikan loan atau tidak

# Logistic Regression – Diabetes Dataset

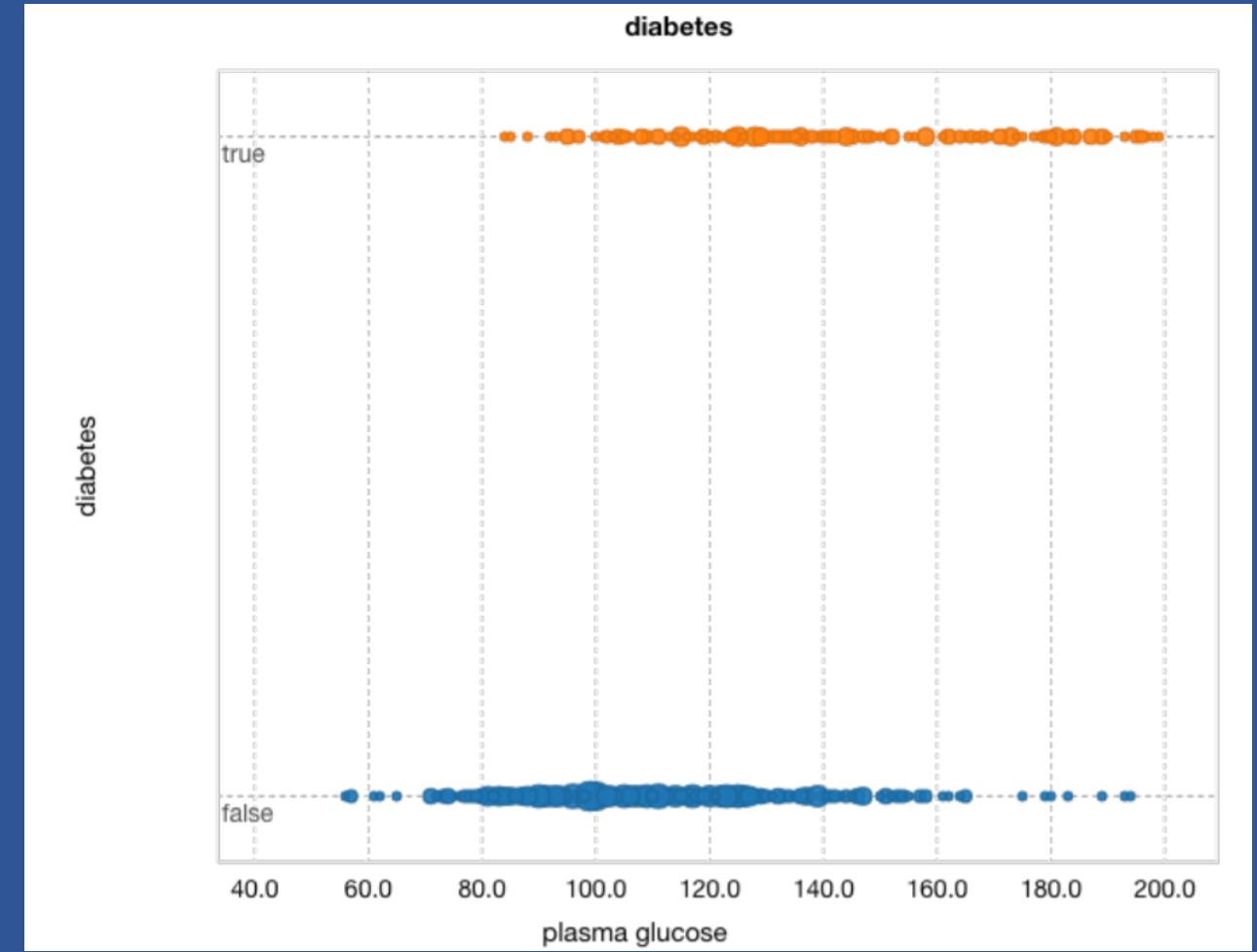
Y axis

True: Person has diabetes

False: No diabetes

X Axis

Feature – plasma glucose

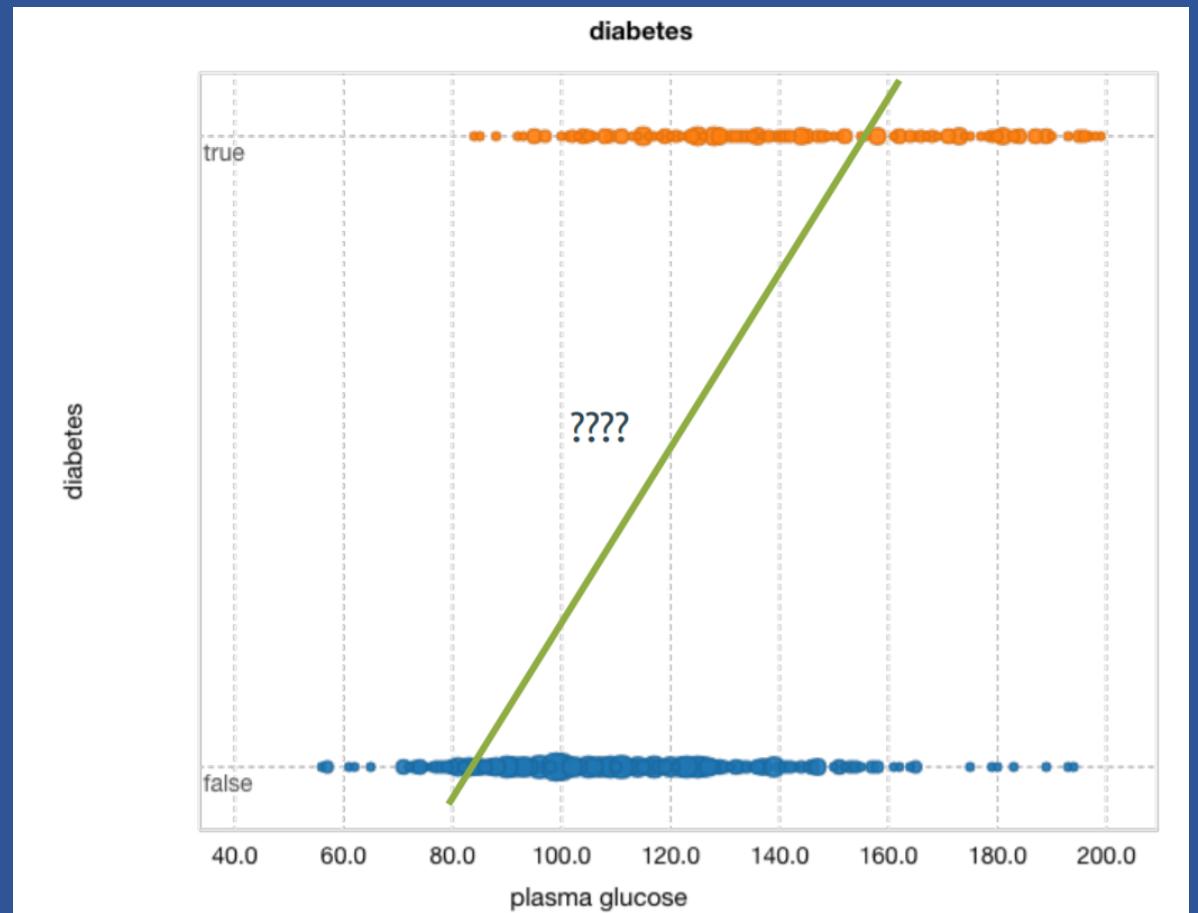


# Logistic Regression – Diabetes Dataset

$$\hat{y} = mx + c$$

$\hat{y}$  = Value predicted by current Algorithm

Linear Regression in one Variable

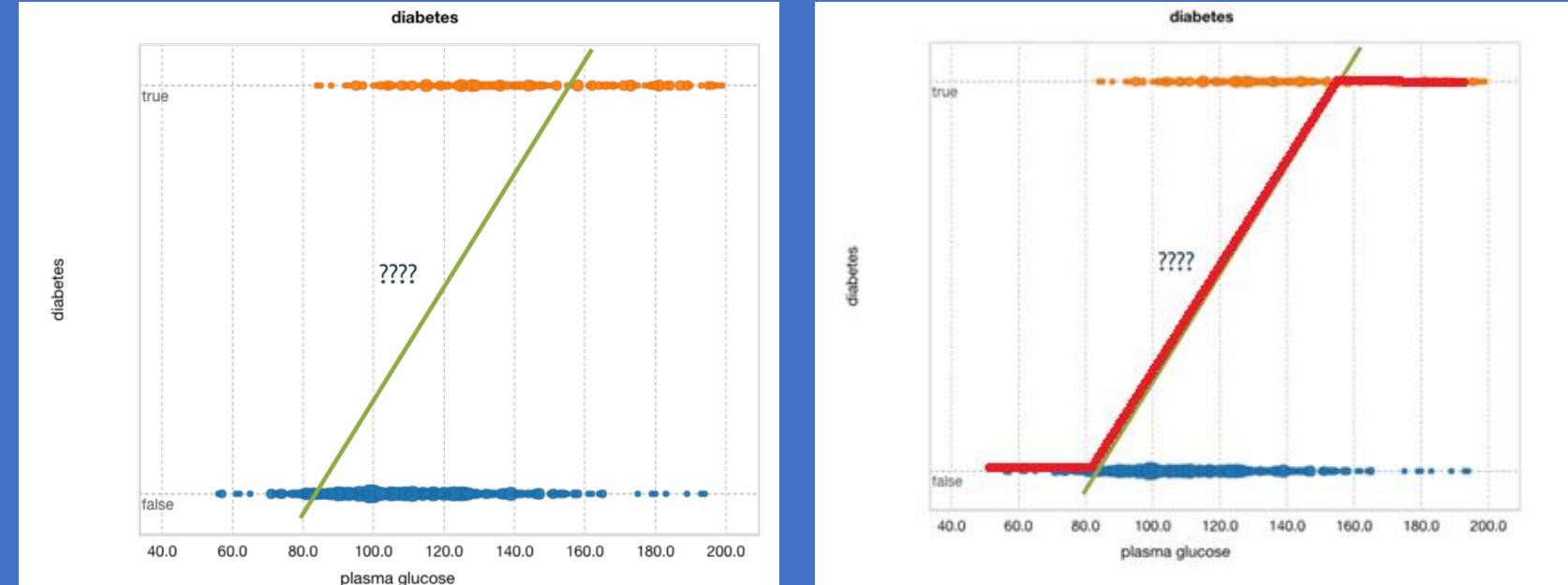


# Logistic Regression – Diabetes Dataset

$$\hat{y} = mx + c$$



$$p = \frac{1}{1 + e^{-\hat{y}}}$$



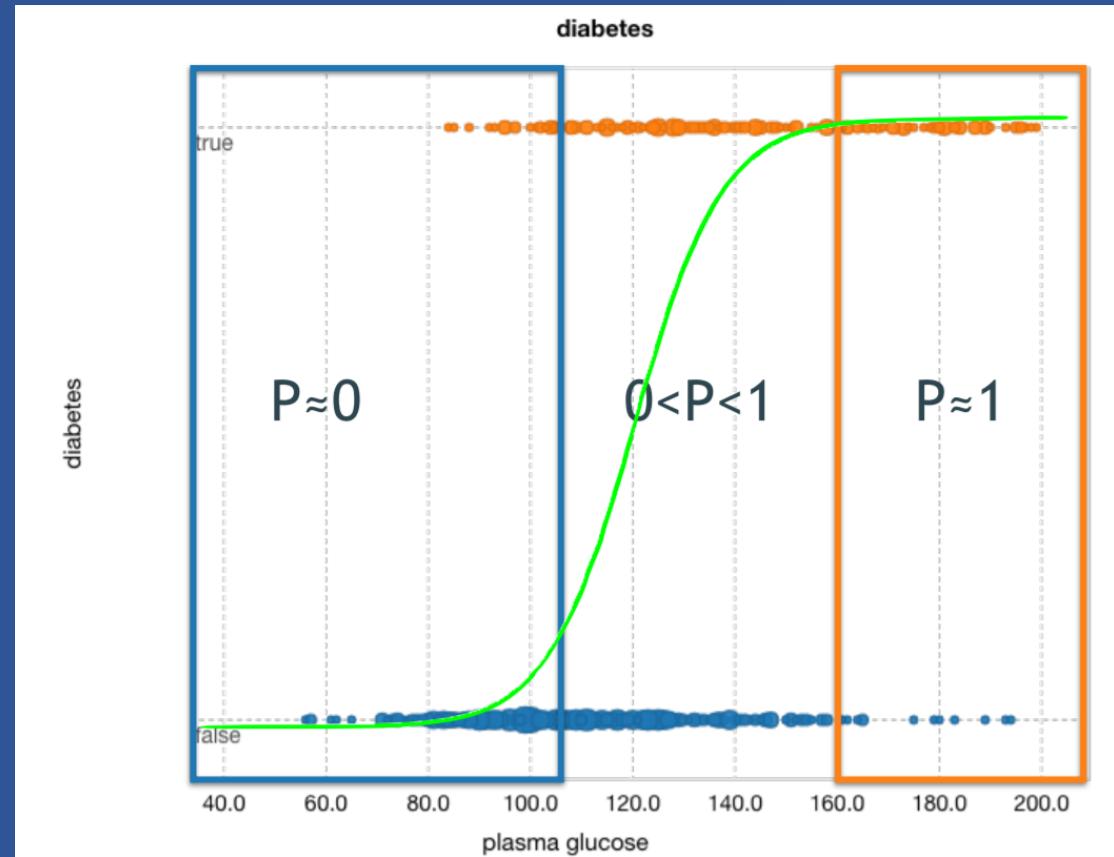
# Logistic Regression – Diabetes Dataset

•

$$p = \frac{1}{1 + e^{-\hat{y}}}$$

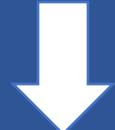


$$p = \frac{1}{1 + e^{-(mx+c)}}$$

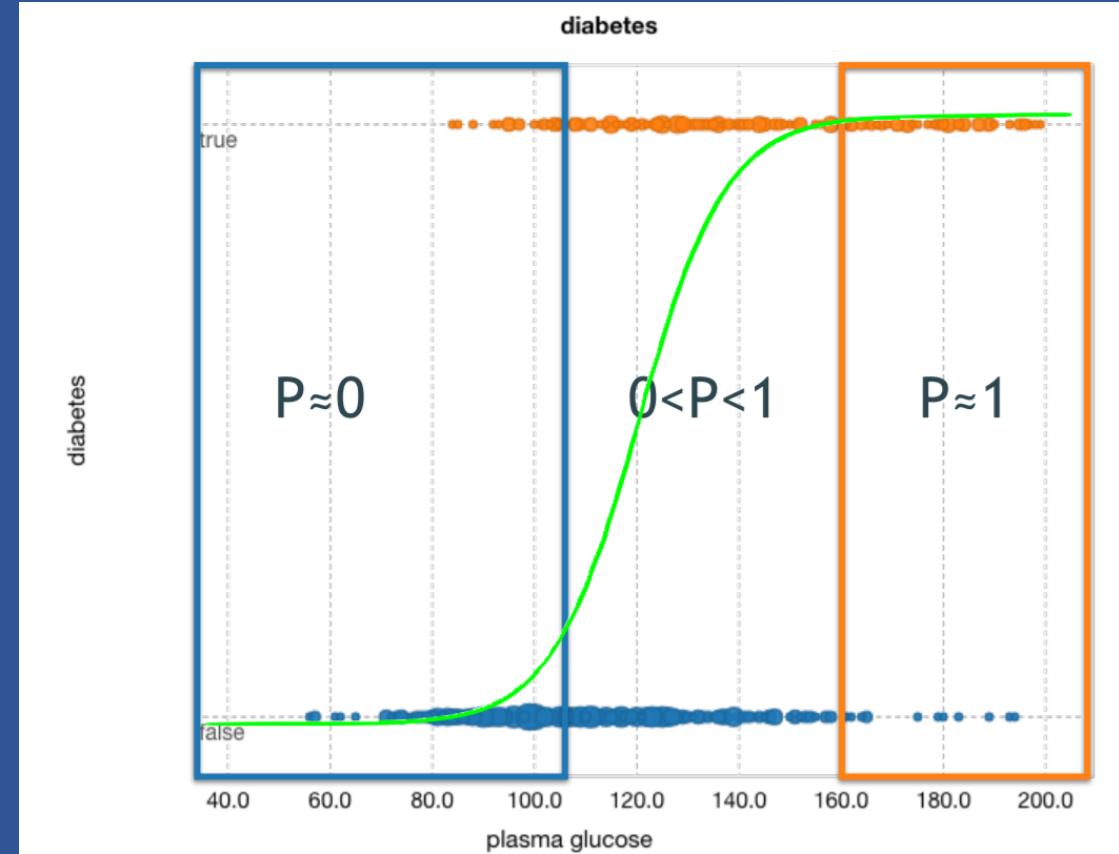


# Logistic Regression – Diabetes Dataset

$$p = \frac{1}{1 + e^{-(mx+c)}}$$



$$\ln\left(\frac{p}{1-p}\right) = mx + c$$





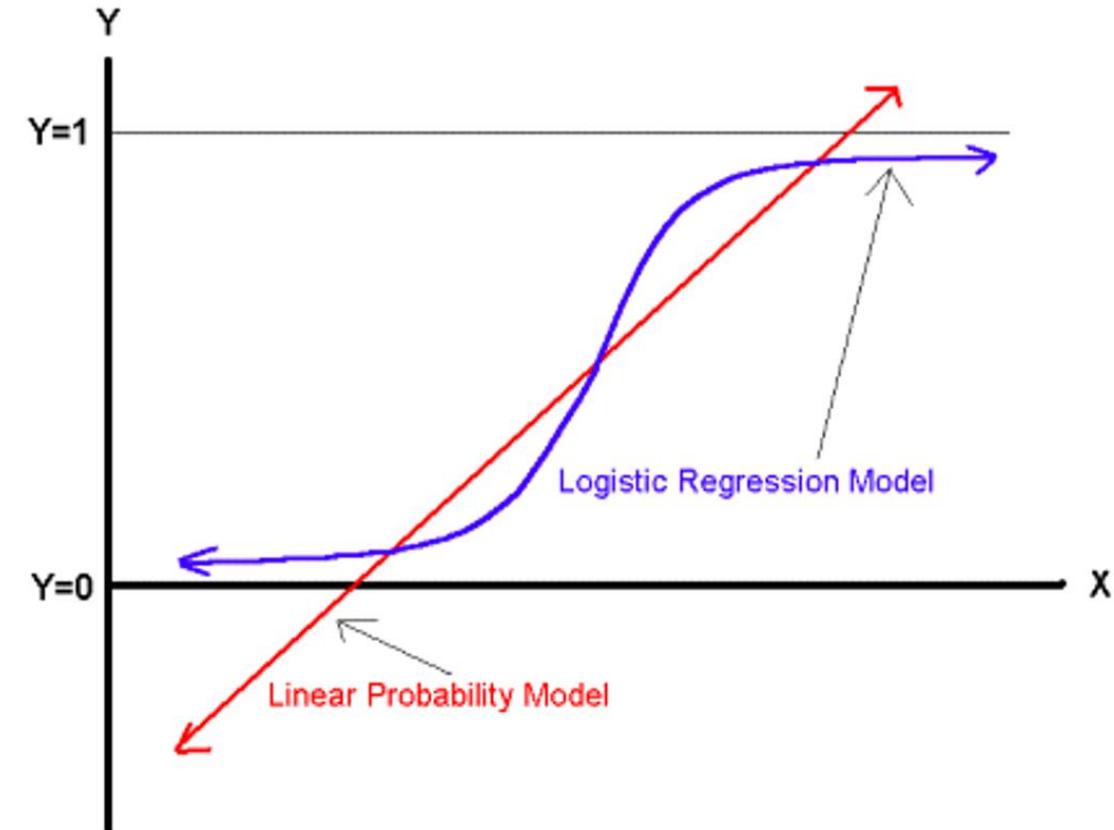
03

## LOGISTIC REGRESSION

- Cost function for logistic regression
- Maximum likelihood estimation

# Logistic Regression

Perbandingan  
**Linear Probability Model**  
dengan  
**Logistic Regression Model**



# Logistic Regression

Binary Classification using  
Logistic Regression

*Sigmoidal Function*

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Multinomial Classification  
using Logistic Regression

*Softmax function*

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, 2, \dots, K$$

# Cost function for logistic regression

- Log loss digunakan sebagai cost function untuk logistic regression

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^N -(y_i * \log(p_i) + (1-y_i) * \log(1-p_i))$$

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^N -(y_i * \log(\hat{Y}_i) + (1-y_i) * \log(1-\hat{Y}_i))$$

# Cost function for logistic regression

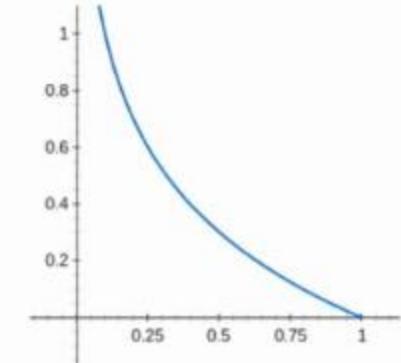
- Case 1: Actual Class nya bernilai 1 dan probability mendekati 1, sisi kiri dari persamaan berikut menjadi aktif dan sisi kanan akan hilang. Plot dari probability yang diprediksi akan bergeser ke 0 dan cost meningkat.

Logistic Regression Cost Function

$$J = \frac{1}{N} \sum_{i=1}^N -(y_i * \log(\hat{Y}_i) + (1-y_i) * \log(1-\hat{Y}_i))$$

$$\hat{Y} = \frac{1}{1+e^{-z}}$$

When  $y=1$



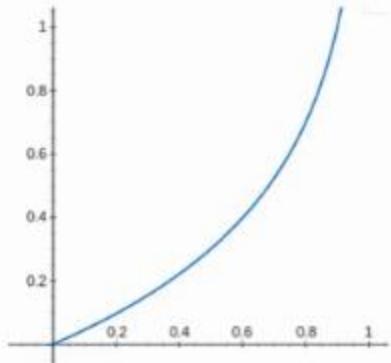
- Case 2: Actual class bernilai 0 dan probability bernilai 0, sisi kanan akan menjadi aktif sedangkan sisi kiri akan hilang.

Logistic Regression Cost Function

$$J = \frac{1}{N} \sum_{i=1}^N -(y_i * \log(\hat{Y}_i) + (1-y_i) * \log(1-\hat{Y}_i))$$

$$\hat{Y} = \frac{1}{1+e^{-z}}$$

When  $y=0$



# Maximum Likelihood Estimation (MLE)

- MLE adalah metode statistik untuk mengestimasi coefficients dari sebuah model.
- The likelihood function (L) mengukur probabilitas dari proses observasi. terhadap sekumpulan dependent variable tertentu ( $p_1, p_2, \dots, p_n$ ) yang ada di suatu sampel:

$$L = \text{Prob} (p_1 * p_2 * \dots * p_n)$$

- Semakin besar nilai L, semakin besar probabilitas dari observasi variable p di sampel tersebut



## 04

## PERFORMANCE

- Confusion matrix
- Accuracy
- Recall, precision & F1 Score

# Confusion Matrix

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

- True positive = correctly identified
- False positive = incorrectly identified
- True negative = correctly rejected
- False negative = incorrectly rejected

## Example

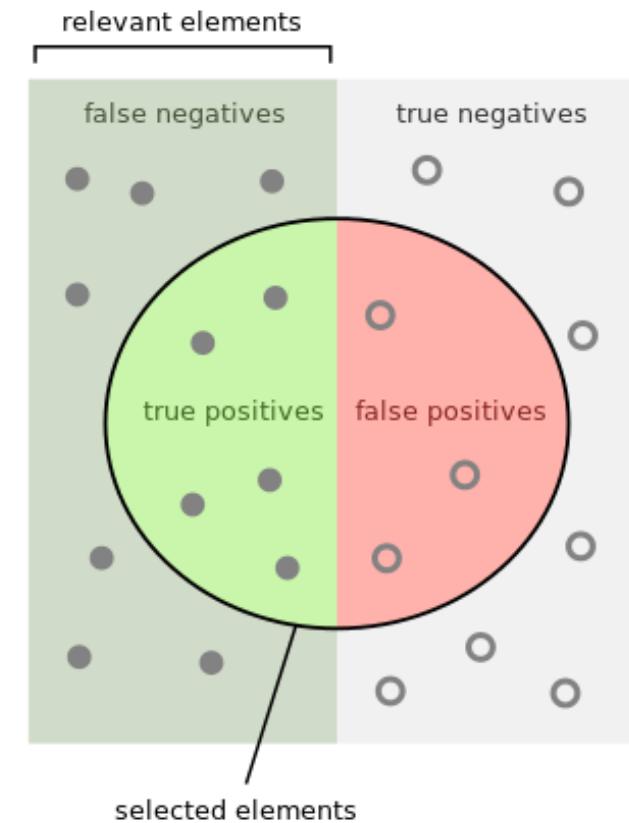
- True positive: Orang sakit diprediksi sakit
- False positive: Orang sehat diprediksi sakit
- True negative: Orang sehat diprediksi sehat
- False negative: Orang sakit diprediksi sehat

# Accuracy

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

# Relevancy Score

1. Precision
2. Recall



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

# Recall

- Recal secara intuitif berarti kemampuan dari classifier untuk menemukan semua sample yang positif

$$\text{Recall} = \frac{tp}{tp + fn}$$

# Precision

- Precision secara intuitif berarti kemampuan dari classifier untuk tidak mengklasifikasi sample negatif sebagai positif

$$\text{Precision} = \frac{tp}{tp + fp}$$

# F1 Score

- F1 Score adalah pengukuran yang mengkombinasikan precision dan recall sebagai bentuk harmonik dari keduanya

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



**05** CONCLUSION

- Summary
- Quiz



• • •  
• • •  
• • •  
• • •

Confusion matrix :

```
[[2 2]  
 [1 5]]
```

Outcome values :

```
2 2 1 5
```

Classification report :

	precision	recall	f1-score	support	
1	0.67	0.50	0.57	4	
0	0.71	0.83	0.77	6	
micro avg	0.70	0.70	0.70	10	.....
macro avg	0.69	0.67	0.67	10	....
weighted avg	0.70	0.70	0.69	10	....

# Summary

1. Logistic regression merupakan linear model yang digunakan untuk persoalan klasifikasi
2. Logistic regression bukan merupakan pilihan yang baik ketika dataset yang ada memuat feature yang memiliki relasi tidak linear dengan target
3. Logloss digunakan sebagai cost function untuk logistic regression
4. Accuracy, recall, precision dan f1 score digunakan untuk menilai performa dari model di persoalan klasifikasi



# Quiz

## Question

Manakah di bawah ini yang bukan merupakan metrik untuk menilai performa dari model klasifikasi?

- A. R2 score
- B. Accuracy
- C. Recall
- D. Precision



# Quiz

## Question

Manakah di bawah ini yang bukan merupakan metrik untuk menilai performa dari model klasifikasi?

- A. R2 score
- B. Accuracy
- C. Recall
- D. Precision



**Answer:** A



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

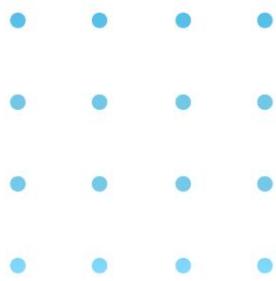
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# AI Mastery Course

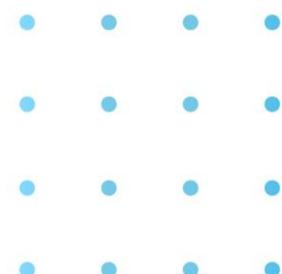


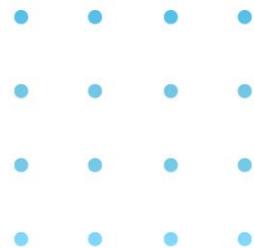
## Module 5

Supervised machine learning and  
predictive modelling

### Section

K nearest neighbor (KNN)

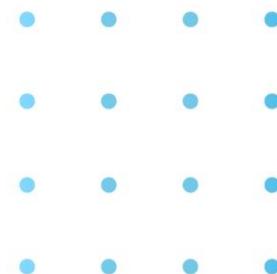




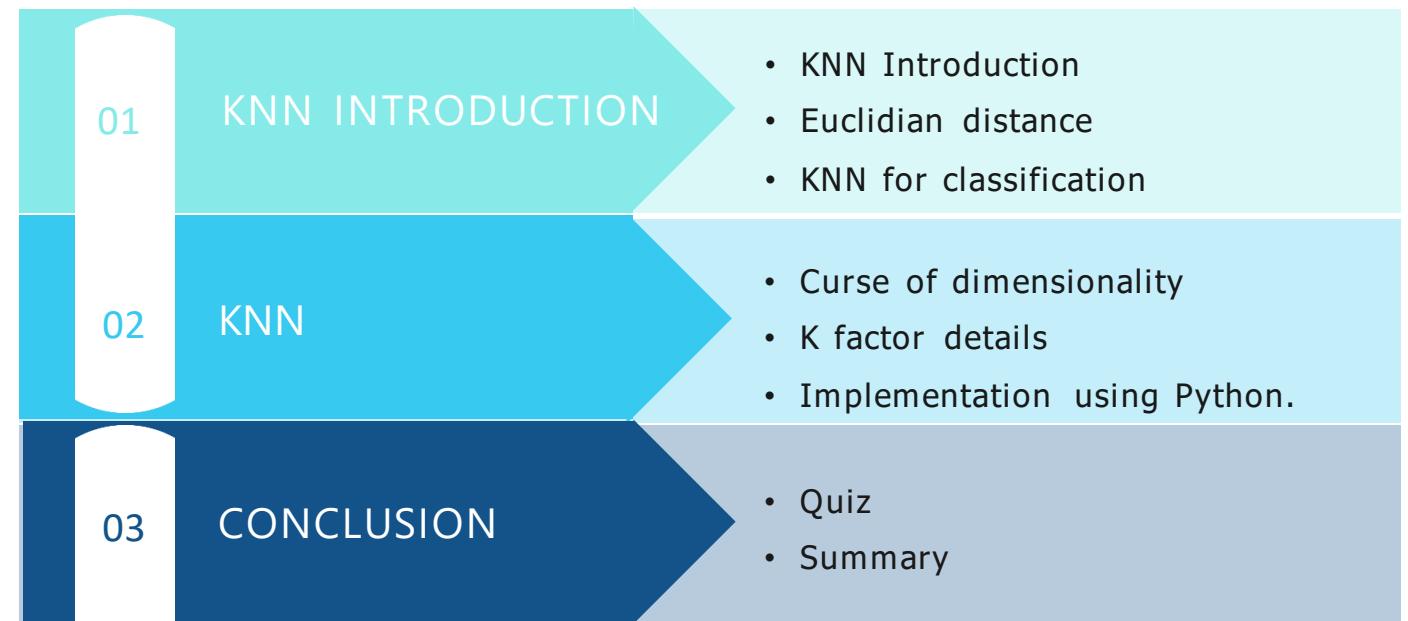
# Learning Objectives

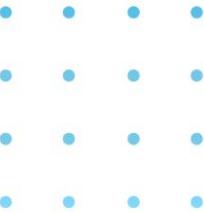
Di akhir modul ini, Anda akan dapat:

- Menjelaskan konsep K Nearest Neighbor (KNN) untuk klasifikasi dan regresi
- Memahami kapan kita menggunakan algoritma KNN
- Implementasi KNN menggunakan Python



# Agenda





01

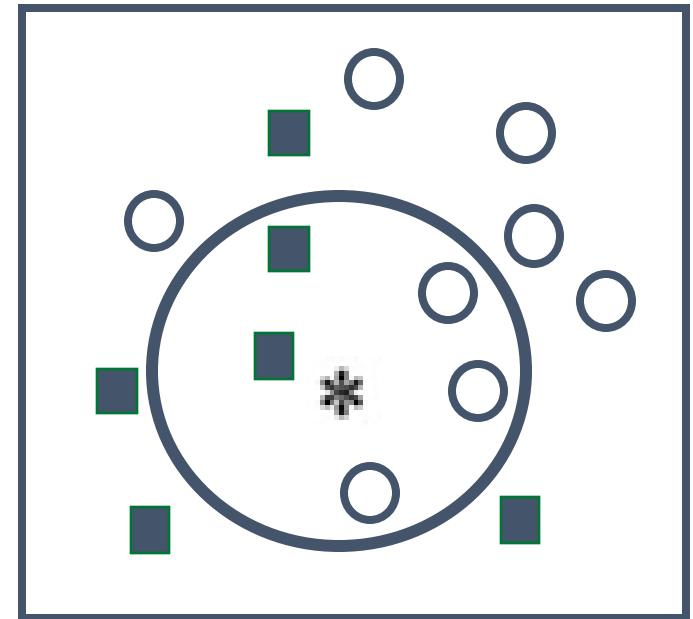
## KNN INTRODUCTION

- KNN Introduction
- Euclidian distance
- KNN for classification



# K Nearest Neighbor classification

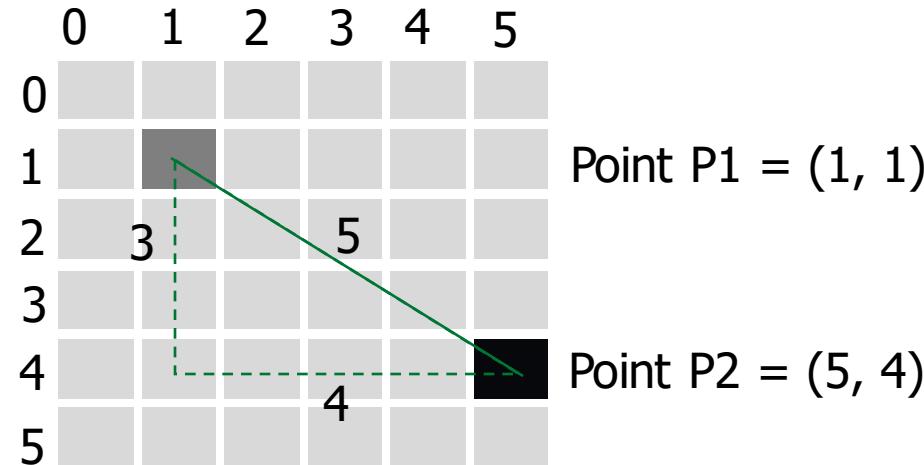
- Nearest-neighbor classifiers belajar berdasarkan analogi dengan cara mengkomparasi tuple test yang diberikan dengan tuple training yang memiliki kemiripan.
- Tuple training dideskripsikan dengan atribut n.
- Saat  $k=1$ , tuple yang tidak diketahui akan ditaruh di kelas dari tuple training yang paling dekat dan memiliki kesamaan pola



# K Nearest Neighbor Classification (Contd.)

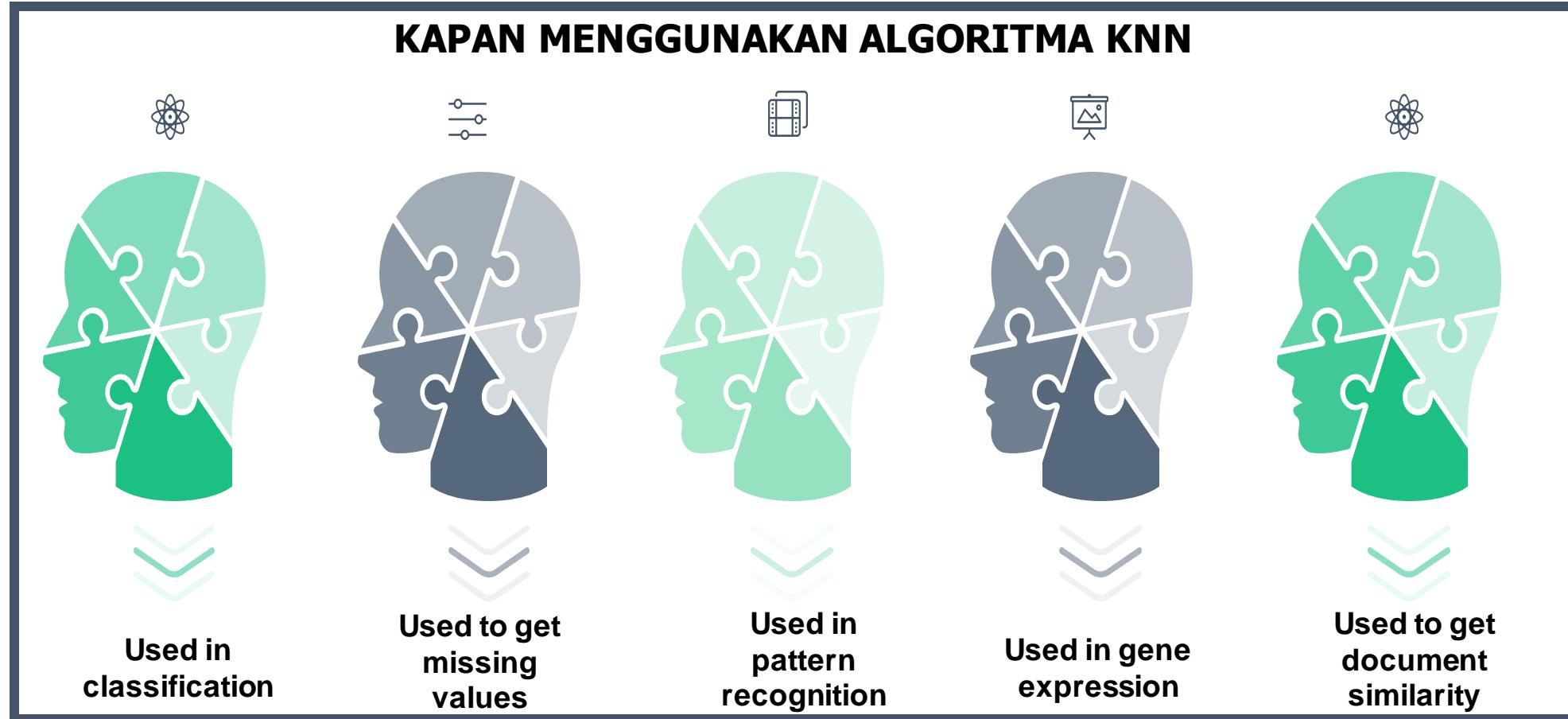
Euclidean distance dihitung dari akar kuadrat dari jumlah selisih kuadrat di antara dua titik.

$$EuclideanDistance(x_i, y_i) = \sqrt{\sum((x_i - y_i)^2)}$$

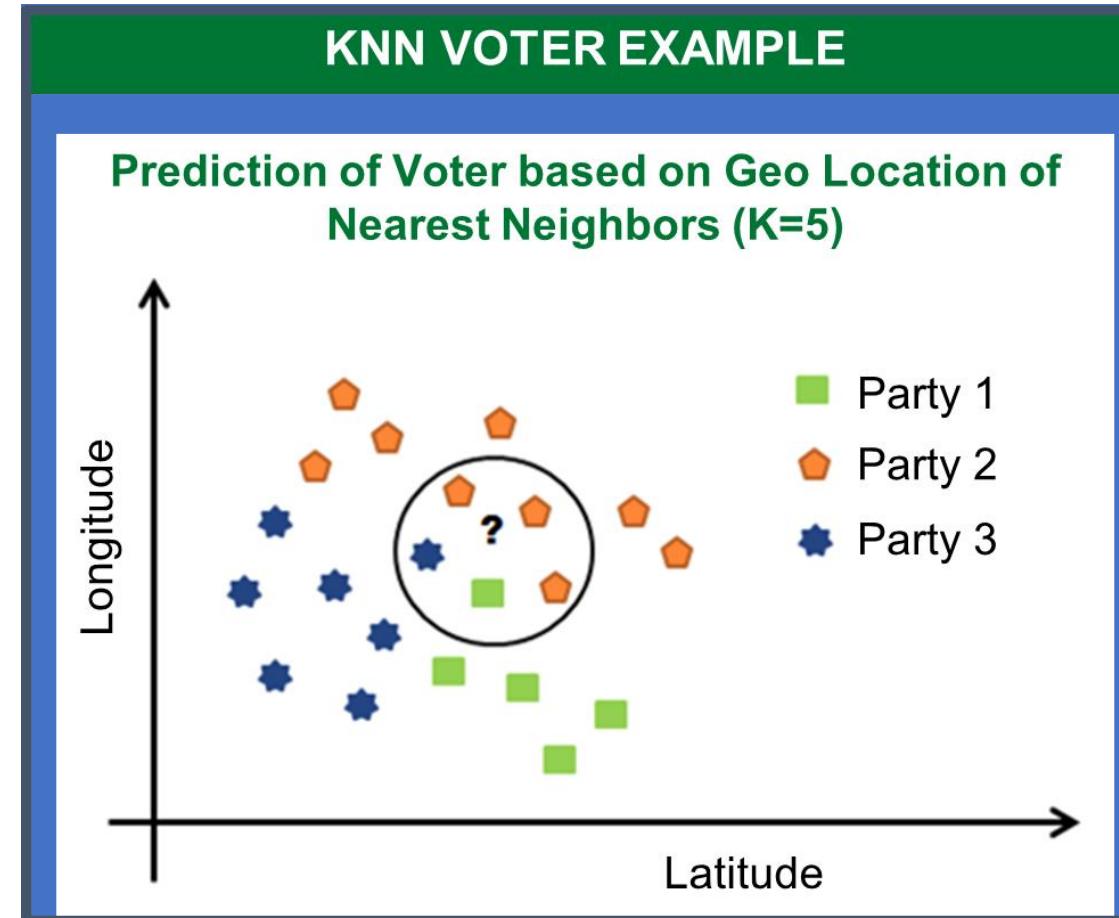


$$\text{Euclidean distance} = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

# K Nearest Neighbor Classification (Contd.)



# K Nearest Neighbor Classification (Contd.)



# K Nearest Neighbor Classification (Contd.)

Bagaimana algoritma KNN bekerja?



**Instance-based**

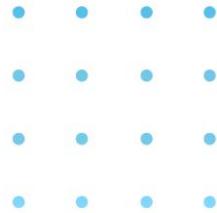


**Competitive learning**



**Lazy learning**





02

## KNN

- Curse of dimensionality
- K factor details
- Implementation using Python.



# Curse of Dimensionality

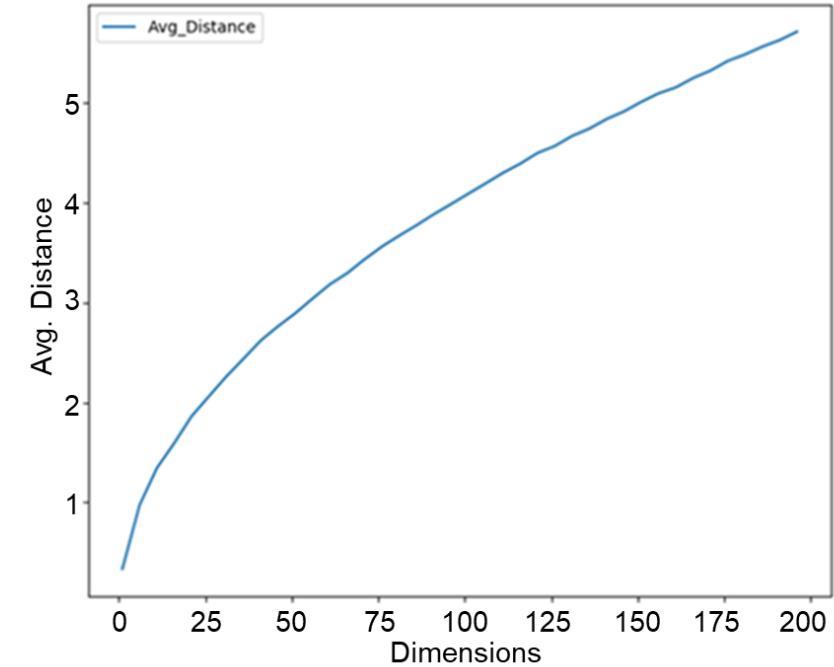
**Data dengan dimensi tinggi akan sulit karena**

- Menambahkan lebih banyak fitur akan meningkatkan kemungkinan noise.
- Biasanya sulit mendapatkan jumlah observasi yang cukup untuk melakukan estimasi yang lebih baik.

**Ini bisa menyebabkan,**

- Meningkatnya running time
- Over fitting
- Jumlah samples yang diperlukan

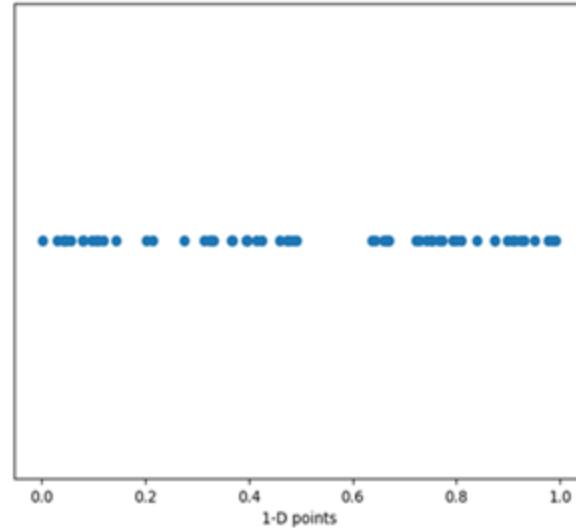
Average Distance change with Number of Dimensions for 1k Observations



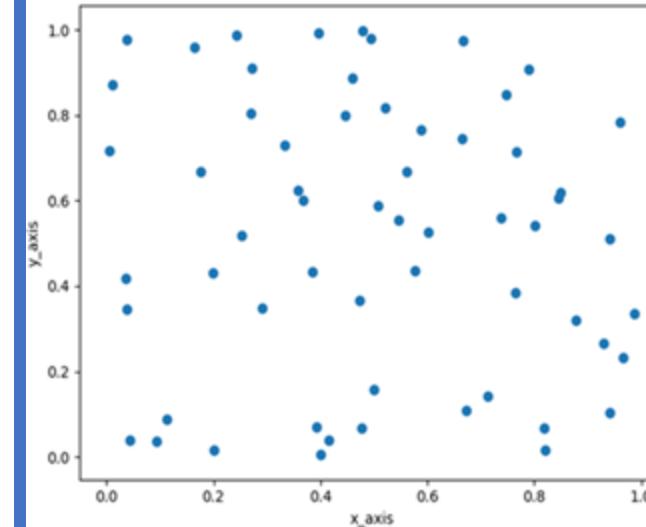
# Curse of Dimensionality (Contd.)

## Curse of dimensionality with 1D, 2D, and 3D example

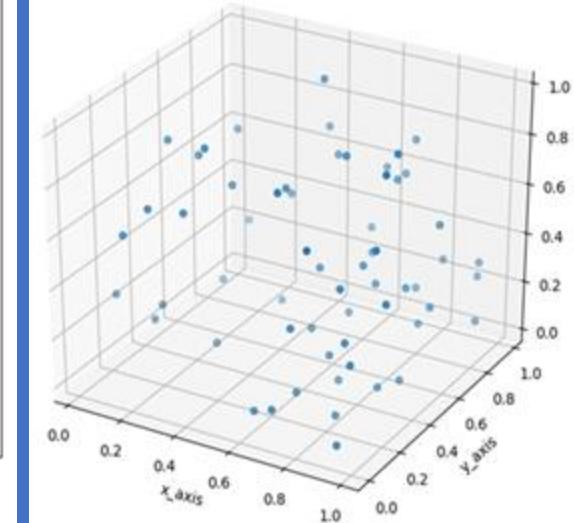
60 Random Points Generated on 1-D



60 Random Points Generated on 2-D



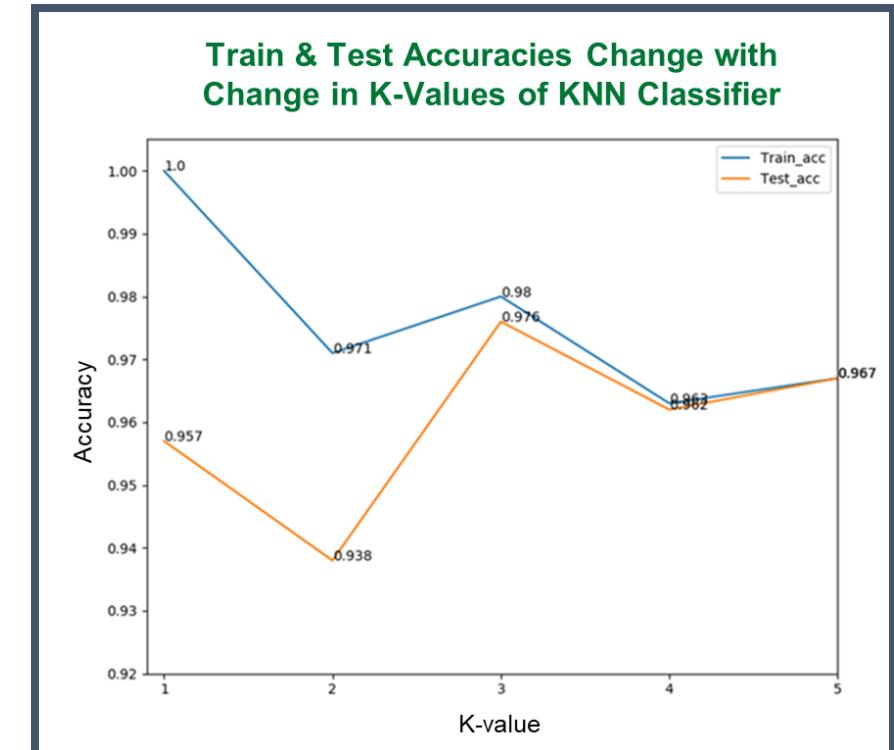
60 Random Points Generated on 3-D



# K-Factor

## Berikut beberapa hal penting terkait K-Factor

- Jika memiliki jumlah sample yang tidak terbatas, semakin besar K akan semakin baik kualitas klasifikasi
- K=1 biasa digunakan untuk efficiency, tapi akan sensitif terhadap "noise"
- Semakin besar K memberikan boundary yang lebih halus
- Relasi untuk menemukan K untuk sampel data yang besar:  $k = \sqrt{n}/2$  dimana n adalah jumlah example
- Kita bisa memilih nilai K dengan cross-validation



# Use Case Of KNN

- **Banking System**

- KNN bisa digunakan di sistem perbankan untuk memprediksi apakah seorang nasabah bisa mendapatkan loan approval? Apakah nasabah tersebut memiliki karakteristik yang mirip dengan nasabah biasanya ?

- **Calculating Credit Ratings**

- Algoritma KNN bisa digunakan untuk menentukan credit rating dari seorang nasabah dengan membandingkan dengan orang-orang yang memiliki kemiripan sifat.

- **Politics**

- Dengan bantuan algoritma KNN, kita bisa mengklasifikasi potential voter ke beberapa kelompok kelas seperti "Will Vote", "Will not Vote", "Will Vote to Party 'A', "Will Vote to Party 'B'".
- Other areas in which KNN algorithm can be used are Speech Recognition, Handwriting Detection, Image Recognition and Video Recognition.

# Implementation of KNN using Python

## 1. Import libraries

```
1 # KNN Classifier - Breast Cancer
2 import numpy as np
3 import pandas as pd
4 from sklearn.metrics import accuracy_score,classification_report
5 breast_cancer = pd.read_csv("breast-cancer-wisconsin.data")
```

## 2. Visualize sample data

```
1 breast_cancer.head(5)
2
```

	Sample code number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class
0	1000025	5	1	1	1	2	1	3	1	1	2
1	1002945	5	4	4	5	7	10	3	2	1	2
2	1015425	3	1	1	1	2	2	3	1	1	2
3	1016277	6	8	8	1	3	4	3	7	1	2
4	1017023	4	1	1	3	2	1	3	1	1	2

# Implementation of KNN using Python (Contd.)

## 3. New classifier for class variable

```
1 breast_cancer['Cancer_Ind'] = 0  
2 breast_cancer.loc[breast_cancer['Class']==4, 'Cancer_Ind'] = 1
```

```
1 breast_cancer.head(5)
```

	Sample code number	Clump Thickness	Uniformity of Cell Size	Uniformity of Cell Shape	Marginal Adhesion	Single Epithelial Cell Size	Bare Nuclei	Bland Chromatin	Normal Nucleoli	Mitoses	Class	Cancer_Ind
0	1000025	5	1	1	1	2	1	3	1	1	2	0
1	1002945	5	4	4	5	7	10	3	2	1	2	0
2	1015425	3	1	1	1	2	2	3	1	1	2	0
3	1016277	6	8	8	1	3	4	3	7	1	2	0
4	1017023	4	1	1	3	2	1	3	1	1	2	0

# Implementation of KNN using Python (Contd.)

## 4. Train and Test the model using required variables

```
1 x_vars = breast_cancer.drop(['Sample code number','Class','Cancer_Ind'],axis=1)
2 y_var = breast_cancer['Cancer_Ind']
3 from sklearn.preprocessing import StandardScaler
4 x_vars_stdsclle = StandardScaler().fit_transform(x_vars.values)
5 from sklearn.model_selection import train_test_split

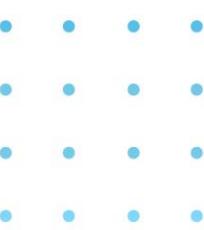
1 x_vars_stdsclle_df = pd.DataFrame(x_vars_stdsclle, index=x_vars.index, columns=x_vars.columns)
2 x_train,x_test,y_train,y_test = train_test_split(x_vars_stdsclle_df,y_var, train_size = 0.7,random_state=42)
```

# Implementation of KNN using Python (Contd.)

## 5. Checking the accuracy using KNN

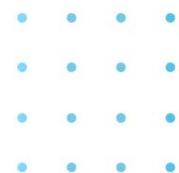
```
1 from sklearn.neighbors import KNeighborsClassifier  
2 knn_fit = KNeighborsClassifier(n_neighbors=3,p=2,metric='minkowski')  
3 knn_fit.fit(x_train,y_train)  
4  
5 print ("K-Nearest Neighbors - Train accuracy:",round(accuracy_score(y_train, knn_fit.predict(x_train)),3))  
6 print ("K-Nearest Neighbors - Test accuracy:",round(accuracy_score( y_test,knn_fit.predict(x_test)),3))  
7
```

```
('K-Nearest Neighbors - Train accuracy:', 0.98)  
('K-Nearest Neighbors - Test accuracy:', 0.976)
```



**03** Conclusion

- Quiz
- Summary



# Quiz

## 1. State True or False

Algoritma KNN melakukan lebih banyak perhitungan pada waktu pengujian daripada waktu latihan.



- A. TRUE
- B. FALSE

# Quiz

## 1. State True or False

Algoritma KNN melakukan lebih banyak perhitungan pada waktu pengujian daripada waktu latihan.



- A. TRUE
- B. FALSE

**Answer: A**

# Quiz

2. Ketika Anda menemukan noise dalam data yang mana dari opsi berikut yang akan Anda pertimbangkan di KNN?

- A. Meningkatkan nilai K
- B. Mengurangi nilai K
- C. Noise tidak dipengaruhi oleh nilai K
- D. Tidak ada yang benar



# Quiz

2. Ketika Anda menemukan noise dalam data yang mana dari opsi berikut yang akan Anda pertimbangkan di KNN?

- A. Meningkatkan nilai K
- B. Mengurangi nilai K
- C. Noise tidak dipengaruhi oleh nilai K
- D. Tidak ada yang benar



**Jawaban: A**

# Summary

- KNN yang merupakan singkatan dari K Nearest Neighbor adalah algoritma Supervised Machine Learning yang mengklasifikasikan titik data baru ke dalam kelas target, mengandalkan fitur titik data tetangganya.
- 
- Euclidean Jarak antara dua vektor, Kurangi satu dari yang lain, kuadrat perbedaan, jumlahkan mereka, dan mengambil akar kuadrat
- 
- KNN regresi mencoba memprediksi nilai variabel output dengan menggunakan rata-rata lokal.
- 
- KNN classification mencoba untuk memprediksi class, dimana output variable dengan menghitung local probability.





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

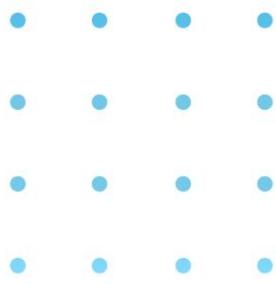
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# AI Mastery Course



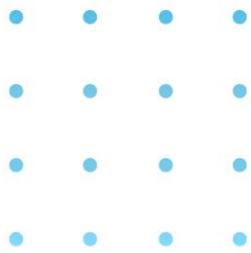
## Module 8

### Deep learning basics

#### Section

Artificial neural network basics

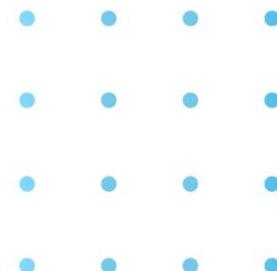




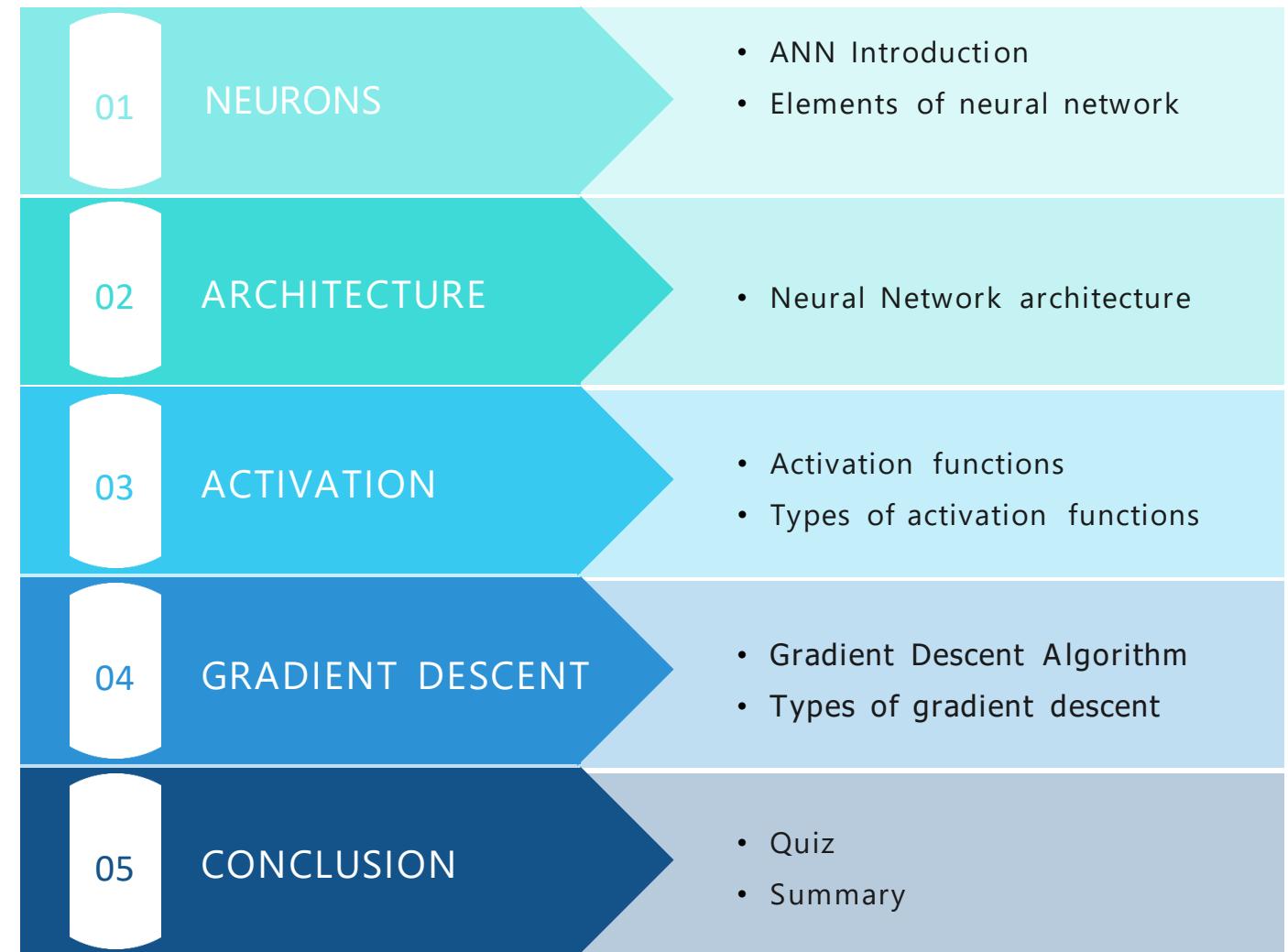
# Learning Objectives

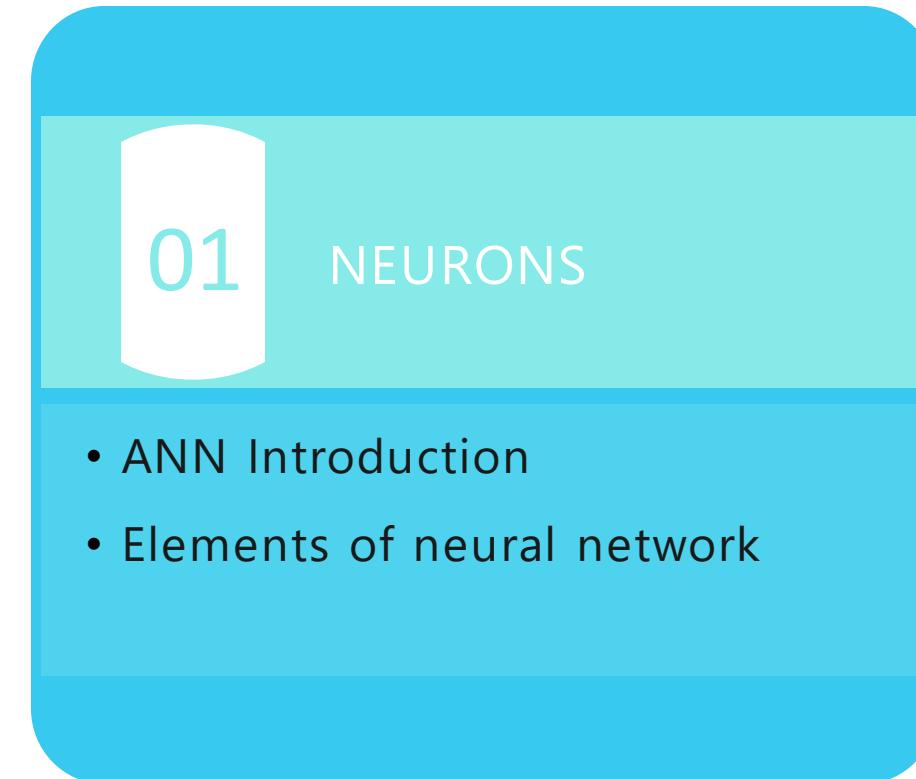
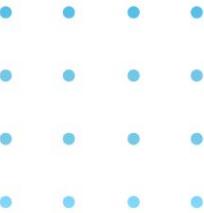
At the end of this module, you will be able to:

- Understand neural networks and intuition behind neural networks
- Understand architecture of the Neural networks
- Define the activation functions (need and types)
- Understand gradient descent and its types



# Agenda





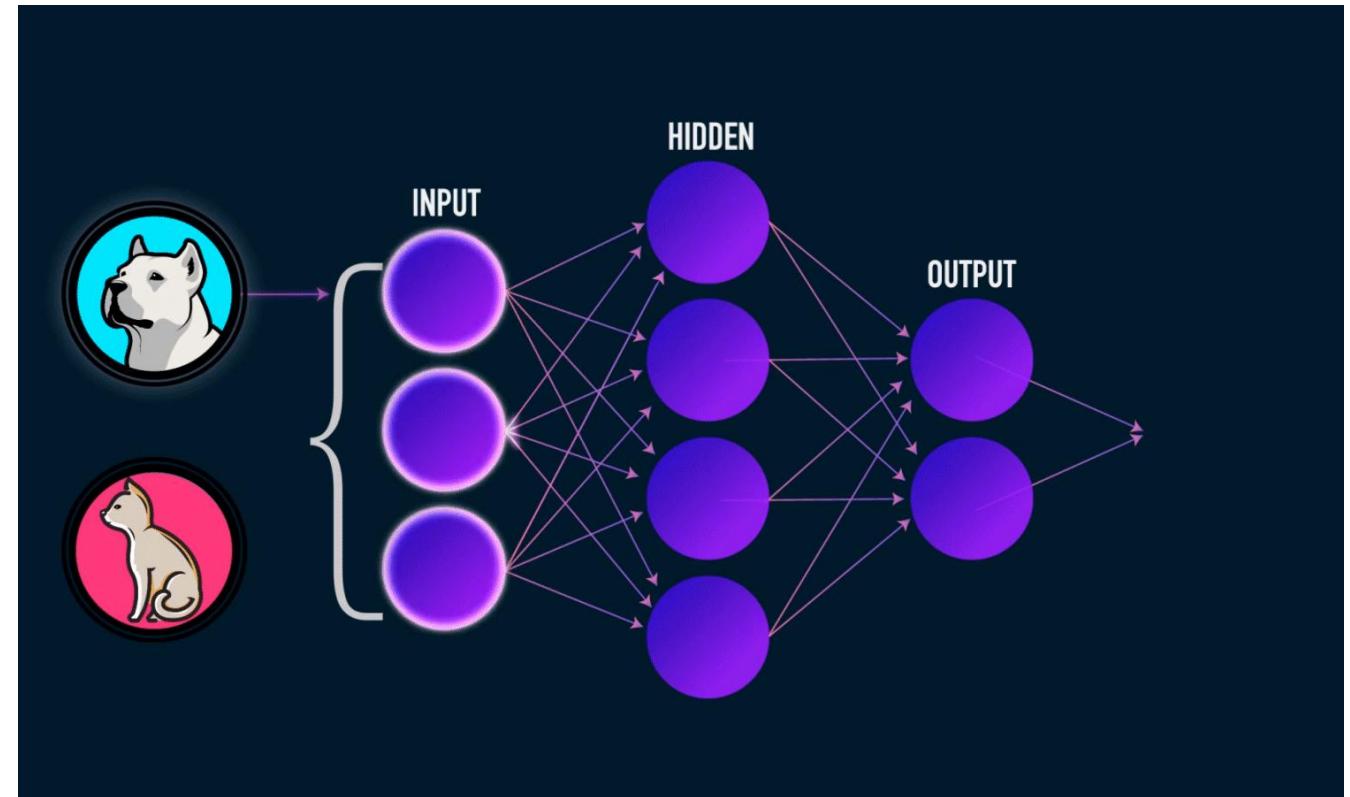
01 NEURONS

- ANN Introduction
- Elements of neural network

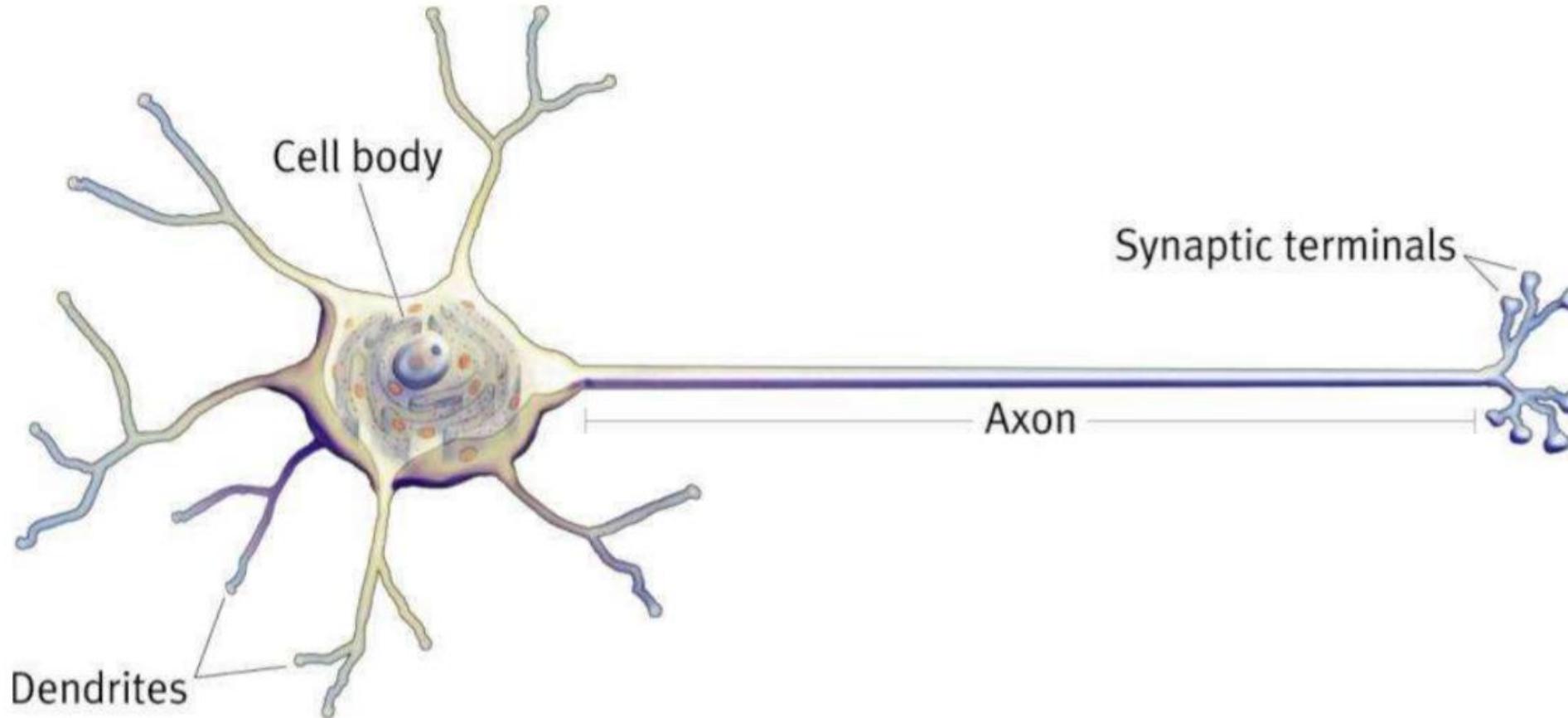


# Introduction to neural networks

- Artificial Neural Networks (ANN) are Information processing paradigm inspired by biological nervous systems
- ANN is composed of a system of neurons connected by synapses
- ANN learn by an example
- Adjust synaptic connections between neurons

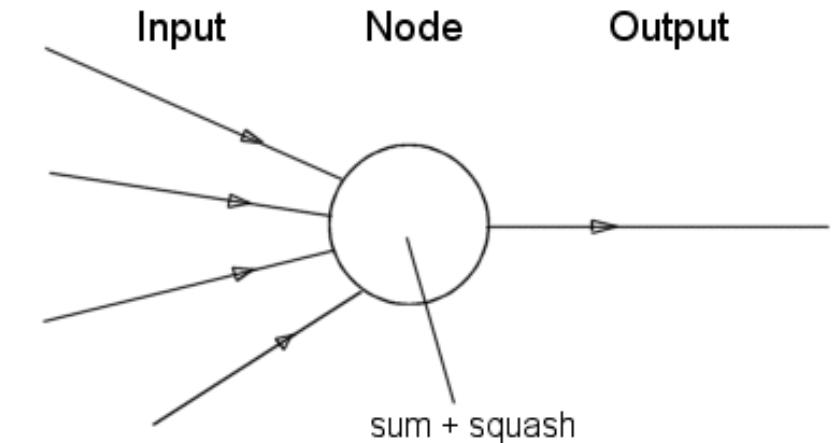
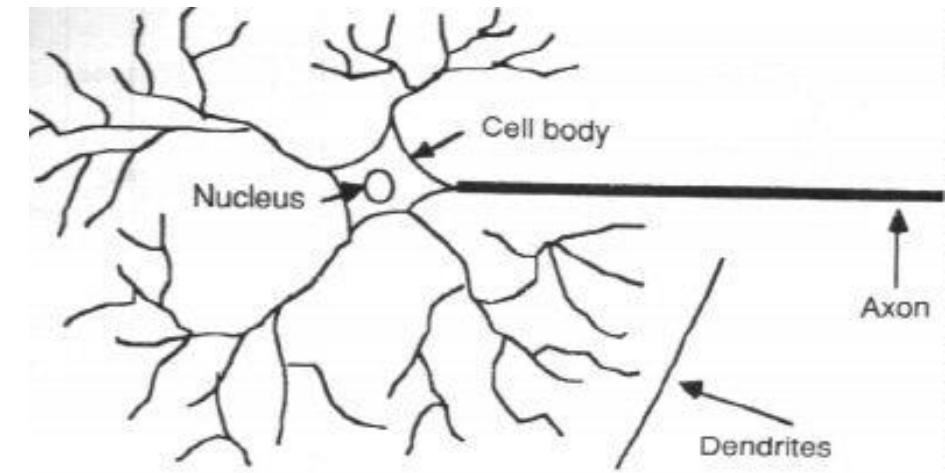


# A natural neuron

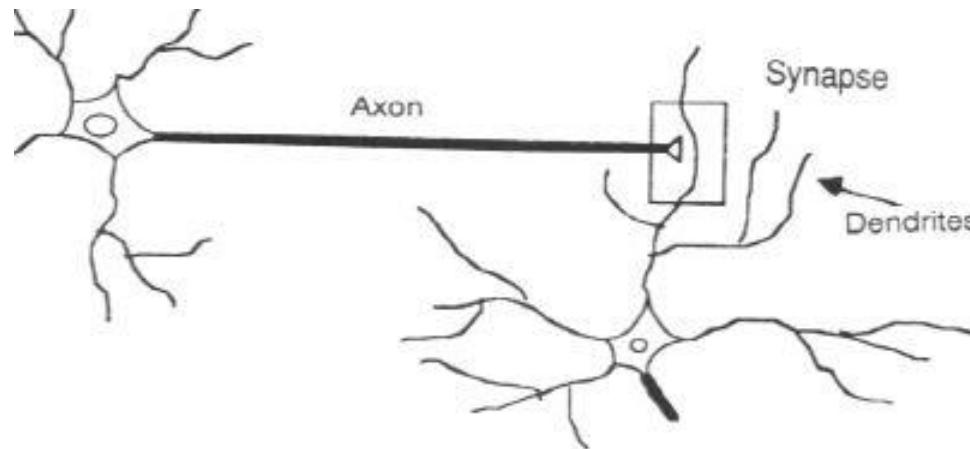


# Neuron v/s nodes

- Neuron in Brain
- A Neuron/Node in Artificial Neural Network

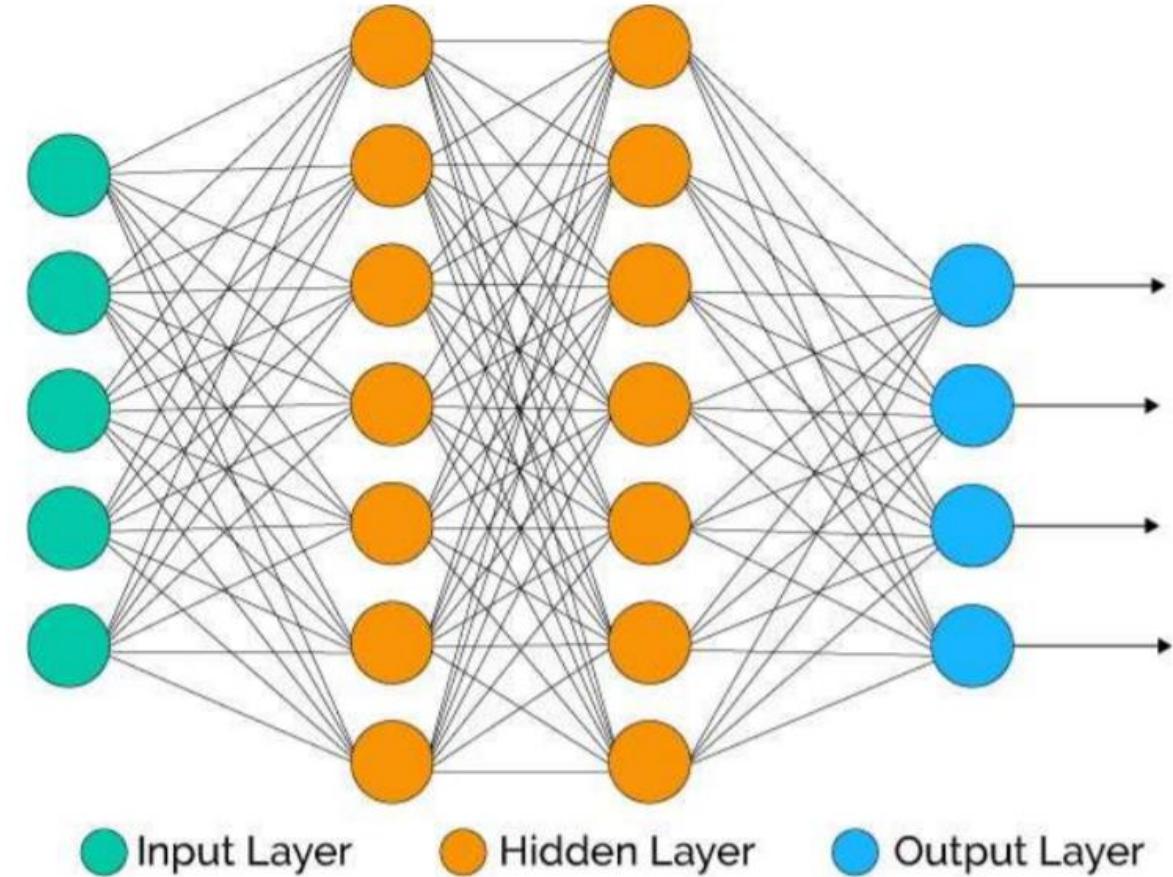


# Neurons v/s nodes



# A neural network

- A typical neural network contains a large number of artificial neurons called units arranged in a series of layers.
- Each neuron of one layer is connected to all the neurons of the next layer through connections called as weighted connections. Such layers are called as Dense Layers.

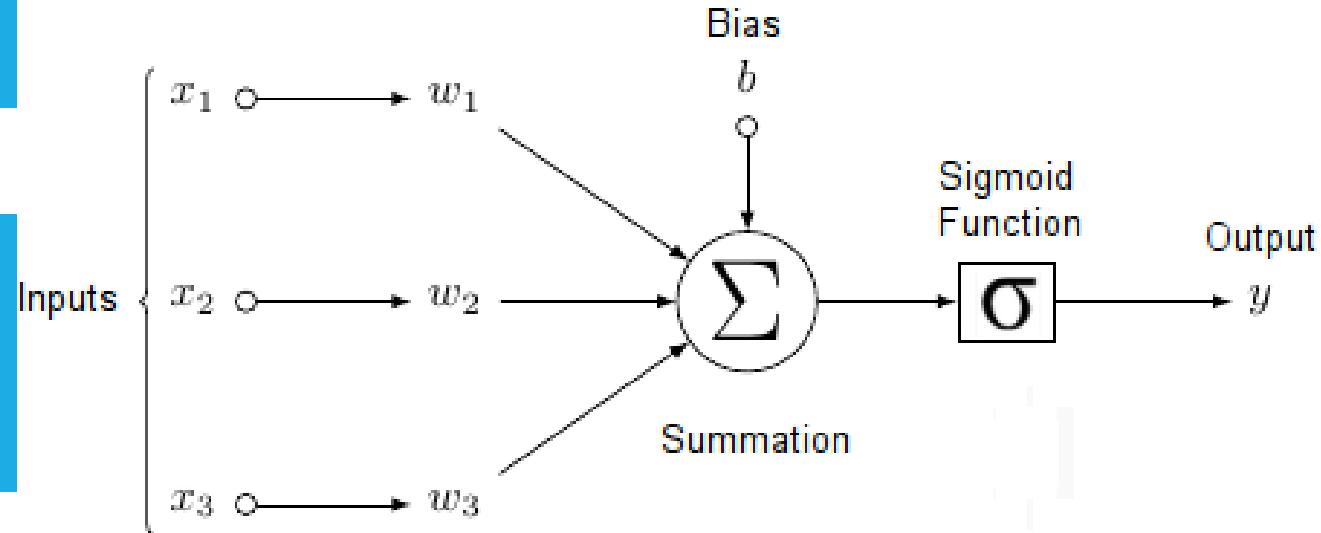


# Elements of Neural Networks

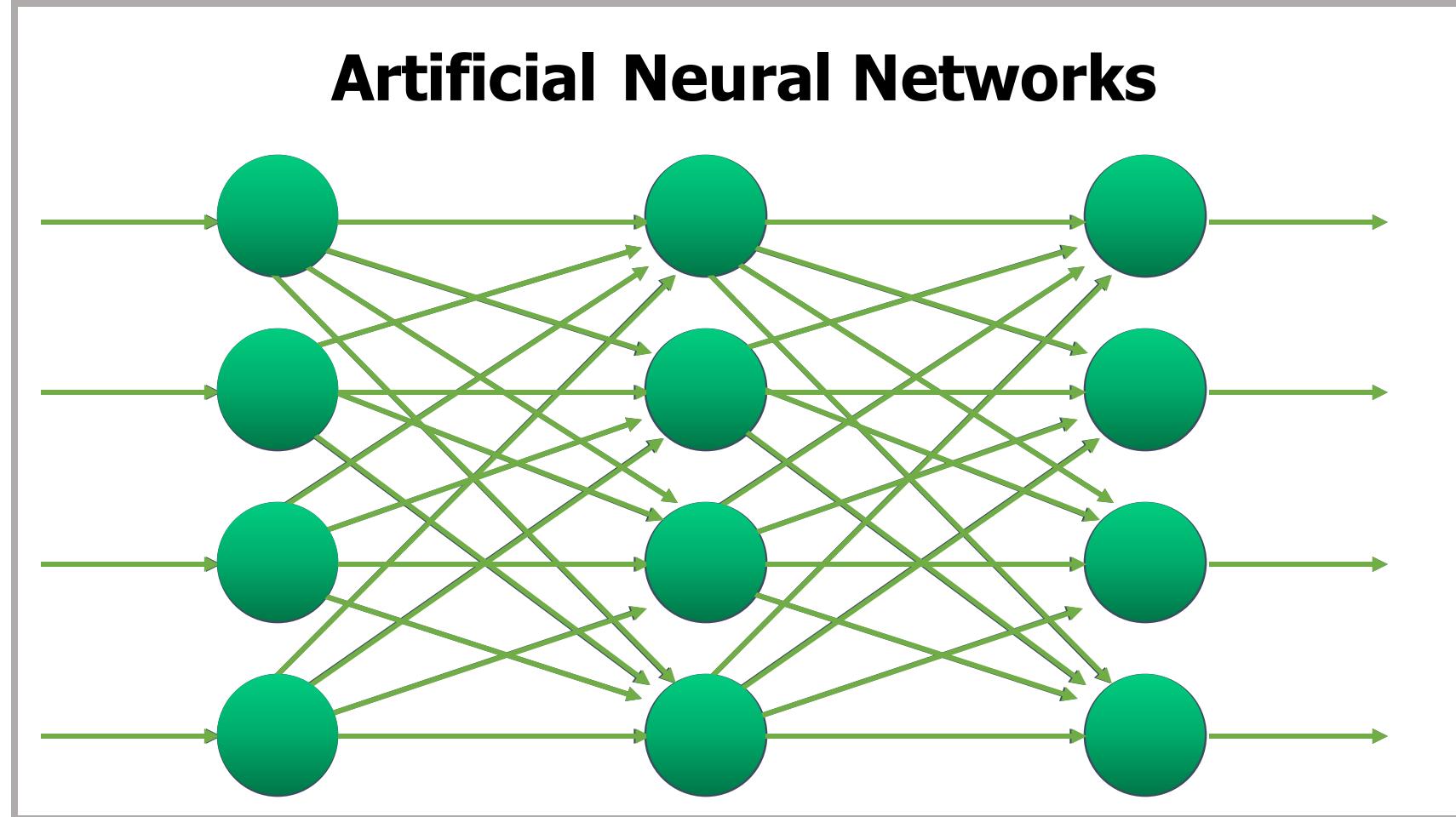
1. No Computation on Input layer Neurons

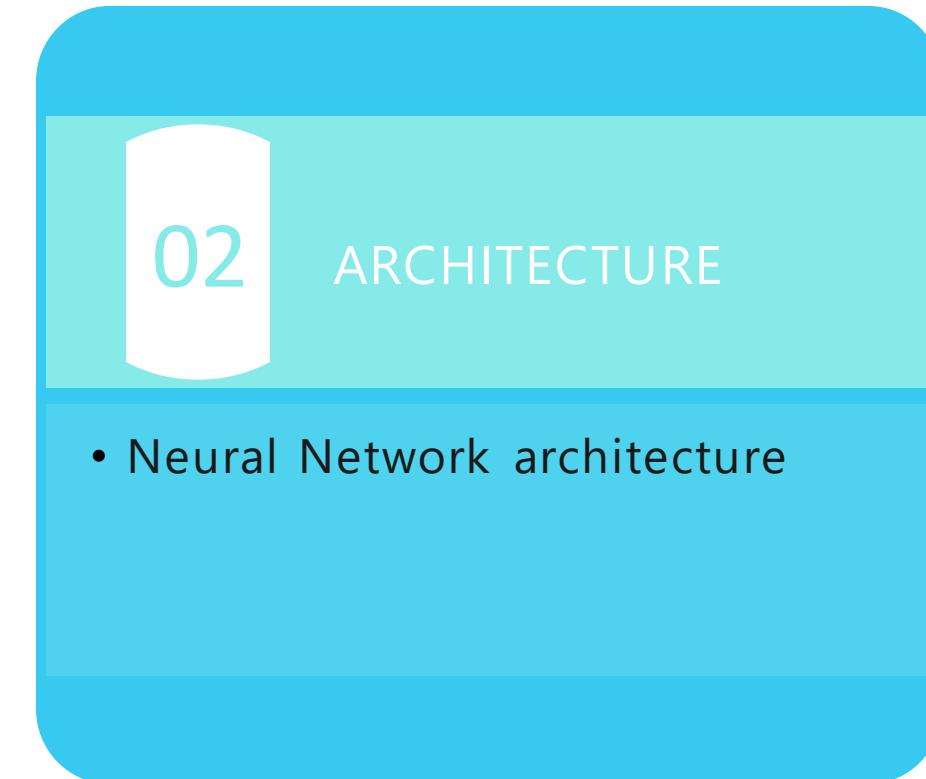
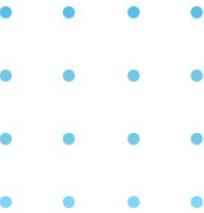
2. Two process on every computational Neuron.

- Weighted Sum
- Activation Function



# Flow of Data in ANN





02 ARCHITECTURE

- Neural Network architecture



# Network Architecture

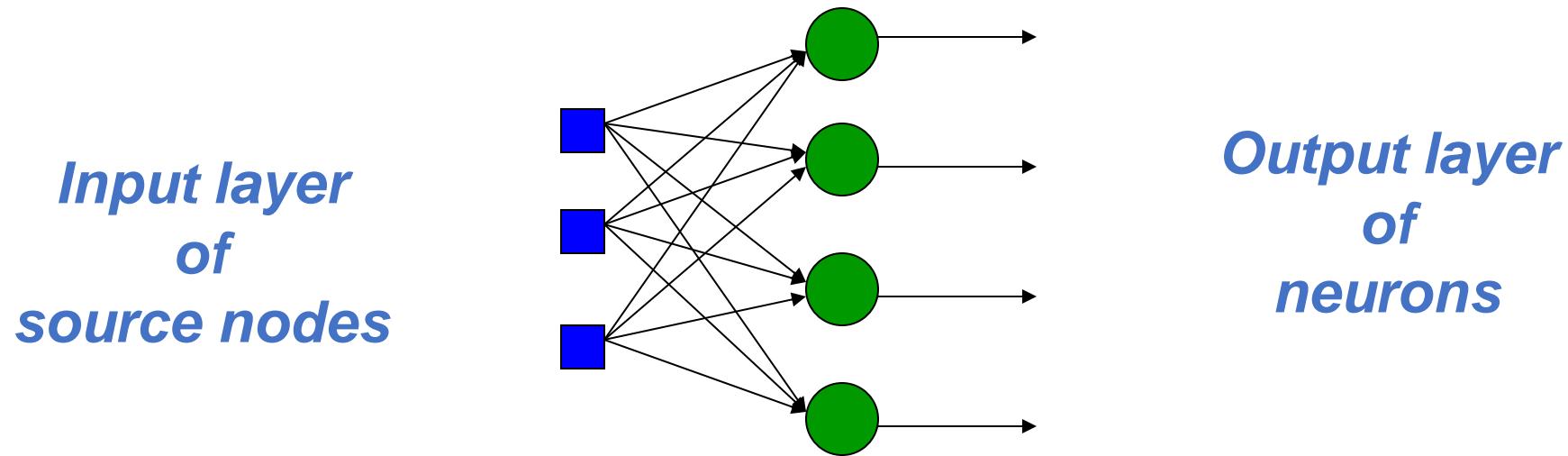
Three basic different classes of network architectures

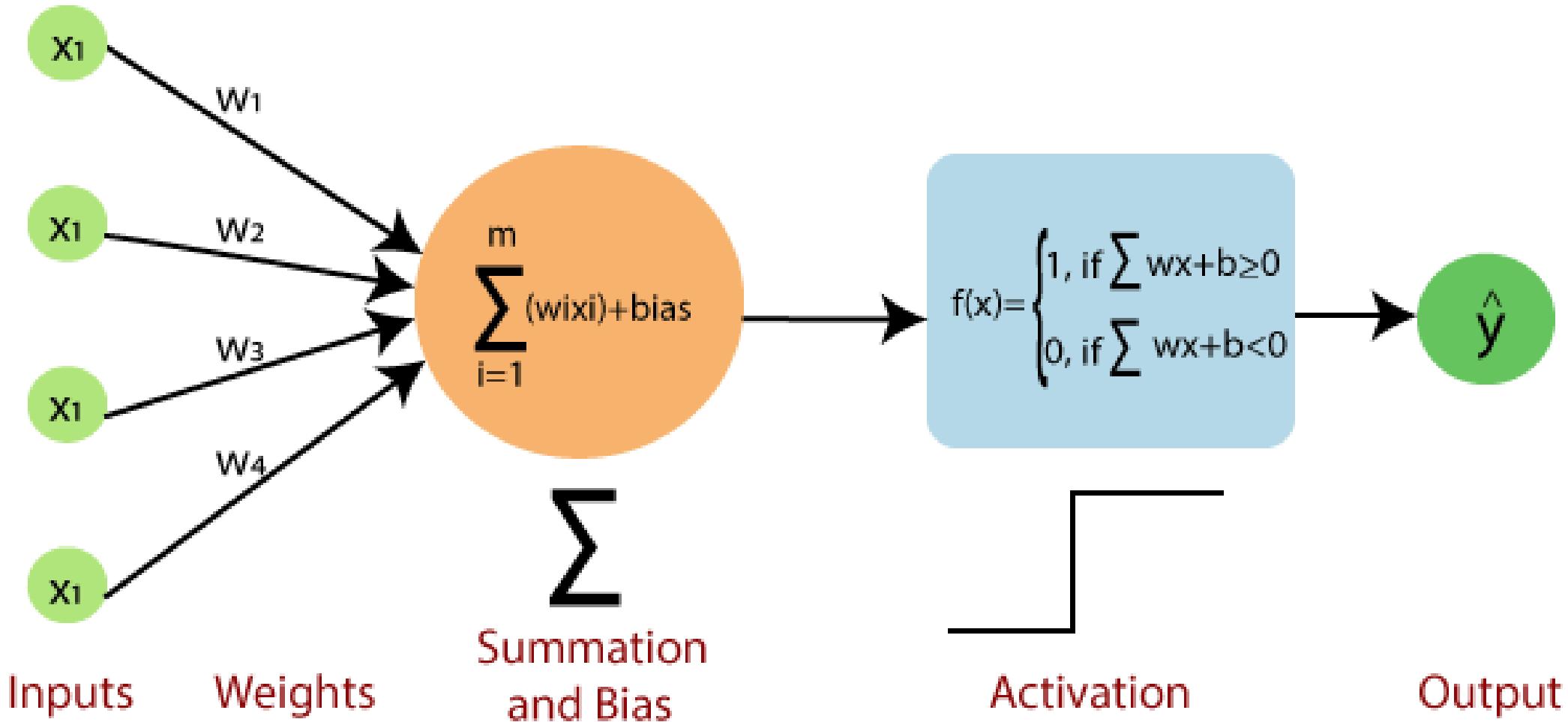
- Single-layer feed-forward Neural Networks
- Multi-layer feed-forward Neural Networks
- Recurrent Neural Networks

The architecture of a neural network is linked with the learning algorithm used to train

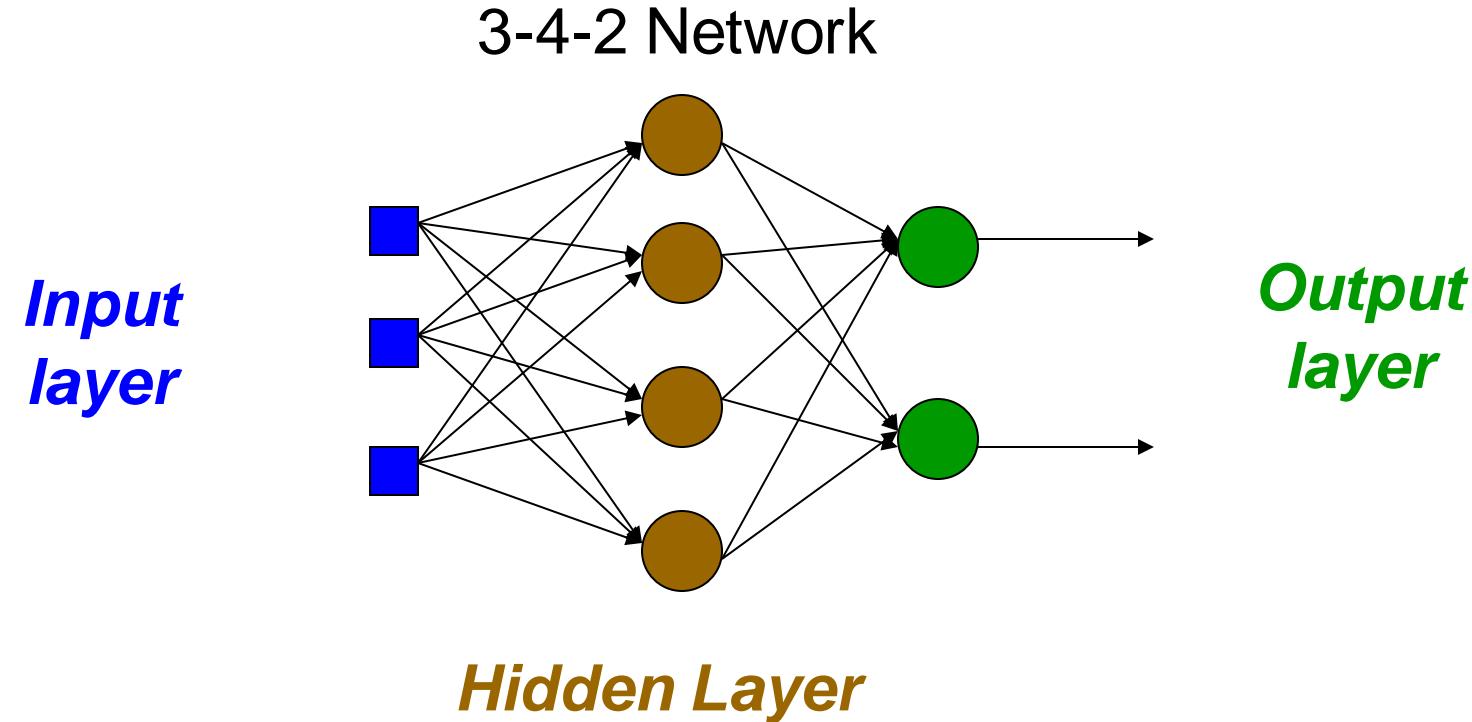
# Single Layer Feed-forward

Neural Network having two input units and one output units with no hidden layers. These are also known as 'single layer perceptron'.



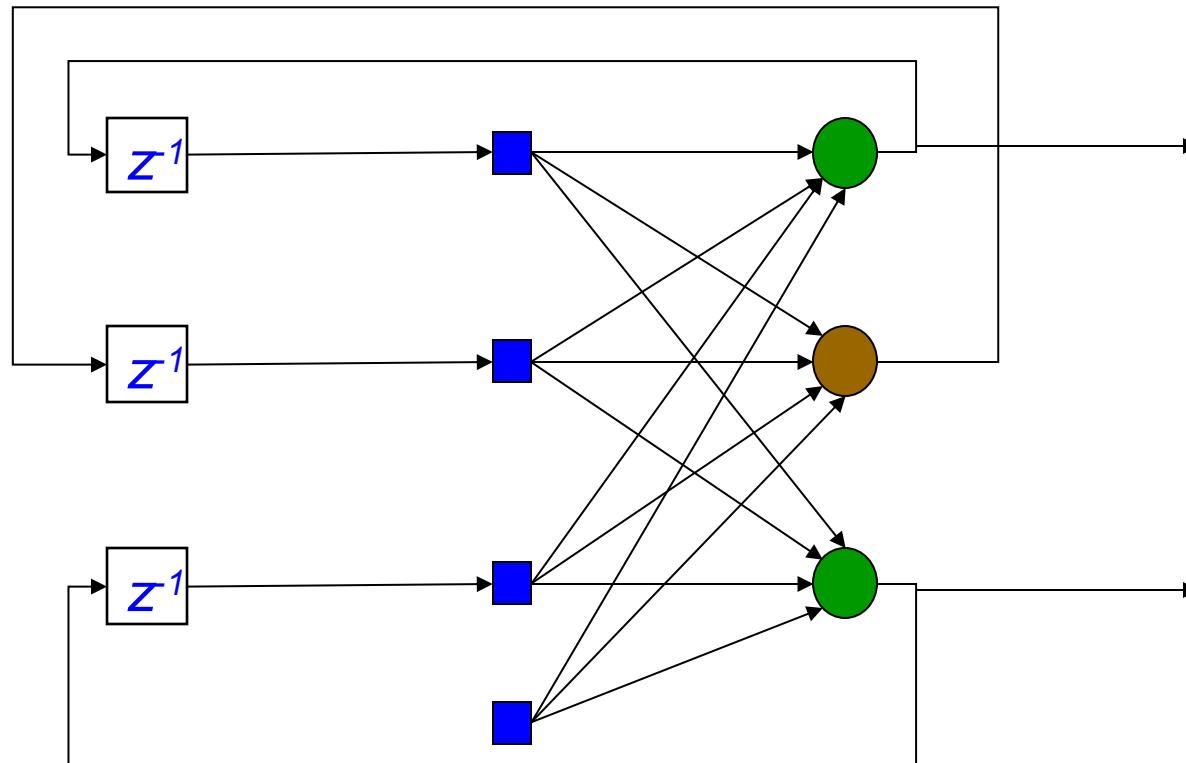


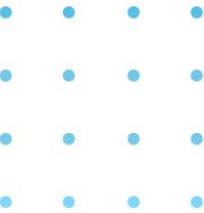
# Multi Layer Feed-forward



# Recurrent Network

Recurrent Network with hidden neuron(s): unit delay operator  $z^{-1}$  implies dynamic system





03

## ACTIVATION

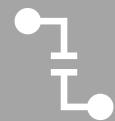
- Activation functions
- Types of activation functions



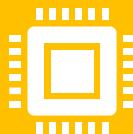
# Activation Functions



Activation functions are simple mathematical functions which help in making sense of something really complicated and non-linear complex functional mapping between the input provided and the output.



They are used to capture non-linear properties of our network. The main purpose of this function is to convert the input signal to the output signal.



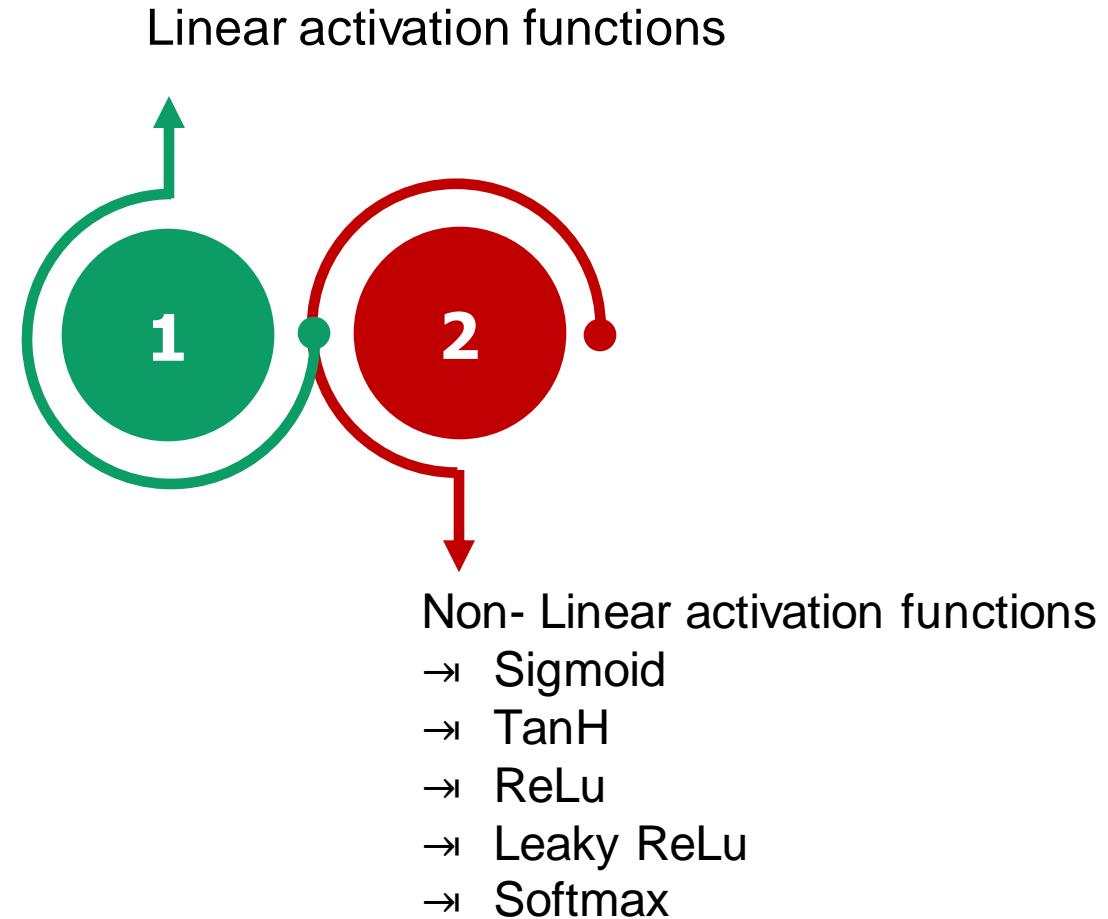
We do sum of products of the weights and the inputs which goes into the activation function. The output of the activation function is the input of the next layer.

# Need of Activation Function

- The need of activation function arises as it helps in capturing complex features from the inputs. If there is no activation function, then it would be just a simple linear mapping between the input and the output (simple linear function).
- The deep neural network is used to make sense out of high dimensional and complicated data, where an increase in the number of layers increases the model to capture the complex insights of the data.
- Also, we want the activation function to be differentiable as it would be needed to train the model using back propagation which involves gradient descent.

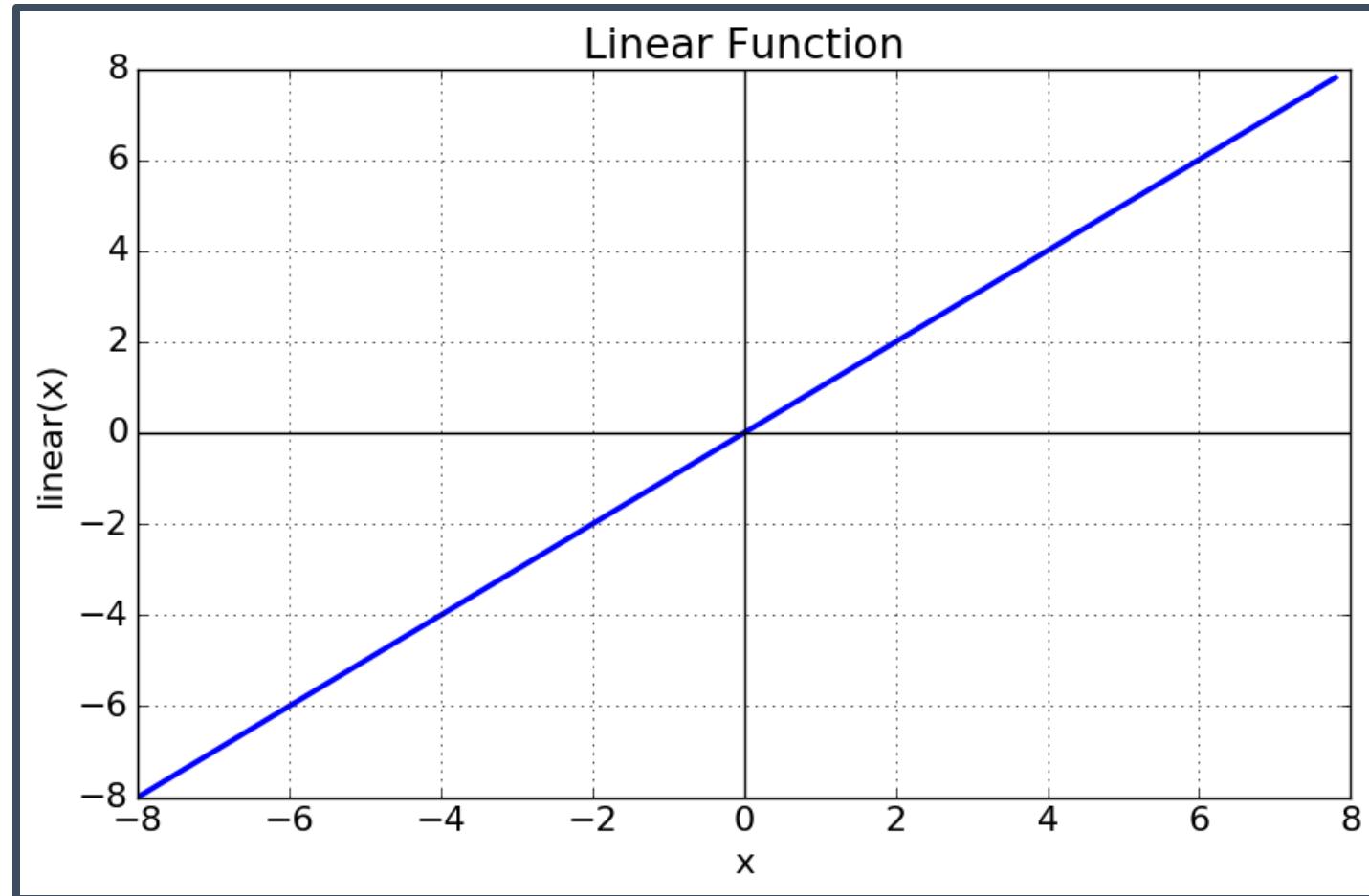
# Types of Activation Functions

The activation functions can be divided into 2 types. They are as follows:

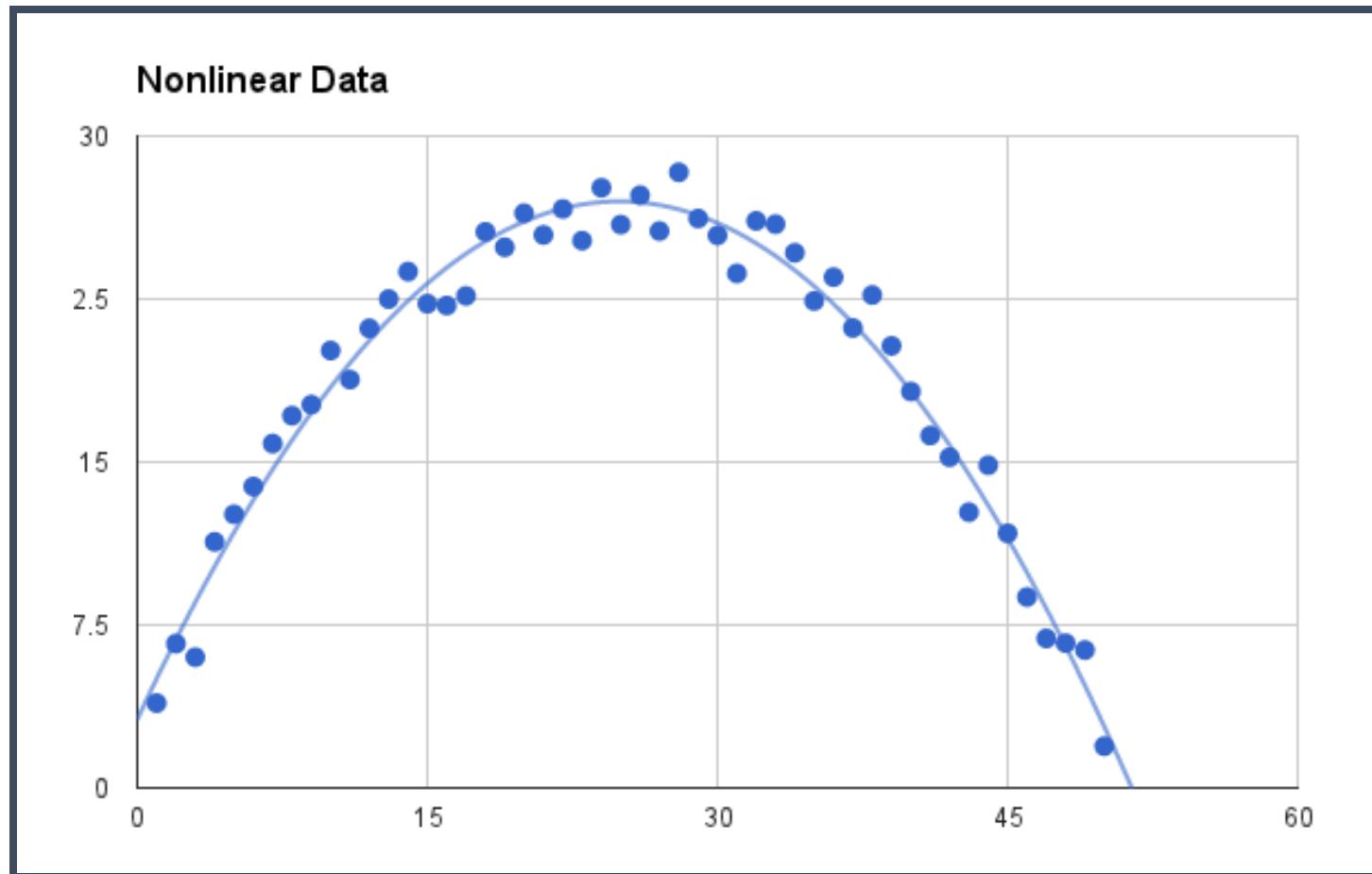


# Linear Activation Function

The function is a simple straight line.

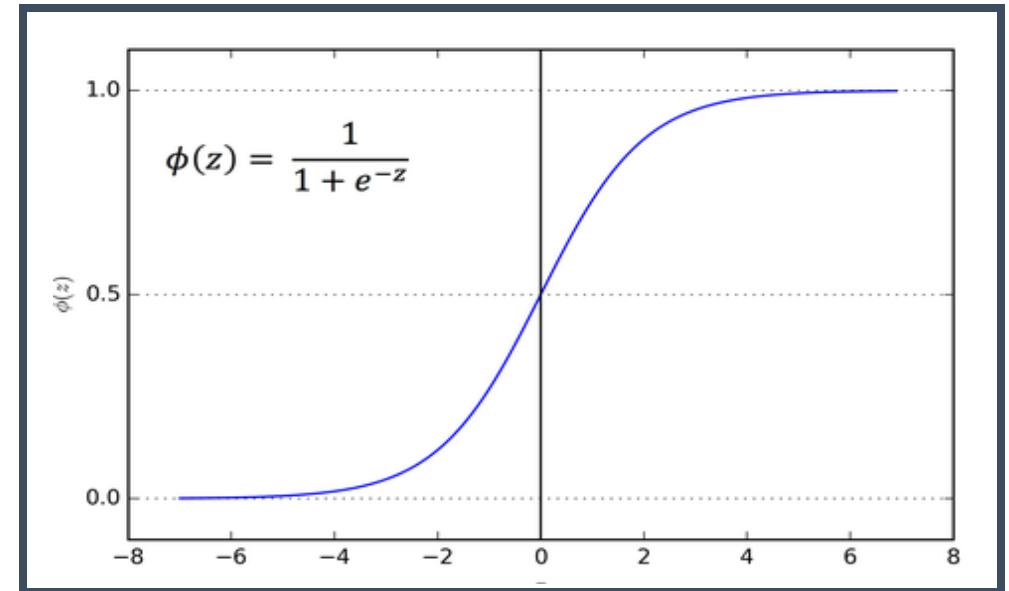


# Non- Linear Data



# Non-Linear Activation Function (Sigmoid)

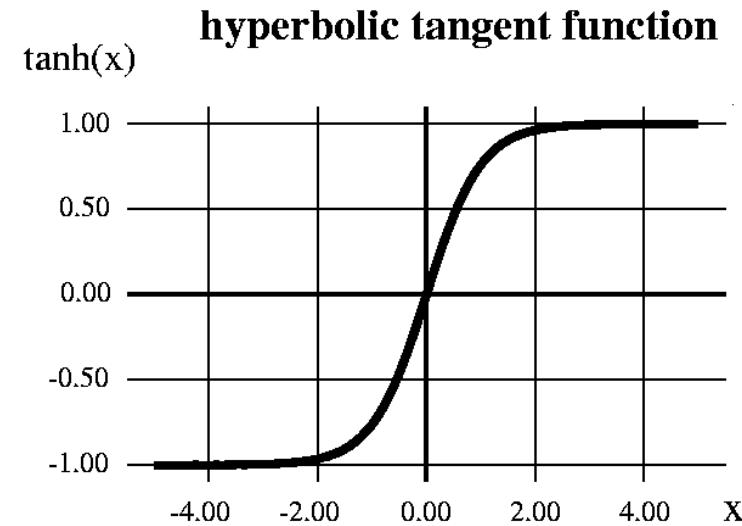
- The function is differentiable. That means, we can find the slope of the sigmoid curve at any two points.
- The function is monotonic, but function's derivative is not.
- The logistic sigmoid function can cause a neural network to get stuck at the training time.
- The SoftMax function is a more generalized logistic activation function which is used for multiclass classification.



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

# Non- Linear Activation Function (Tanh)

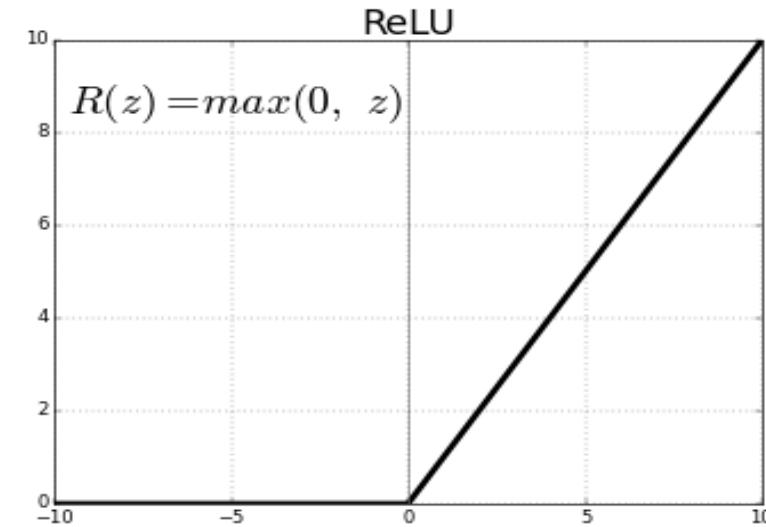
- The function is differentiable.
- The function is monotonic while its derivative is not.
- The tanh function is mainly used classification between two classes.
- output is zero centered because its range is between -1 to 1 i.e.,  $-1 < \text{output} < 1$ . Hence optimization is easier in this method hence in practice it is always preferred over Sigmoid function .



$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

# Non-Linear Activation Function (ReLU)

- Rectified Linear Unit (ReLU) -  $\max(0, Z)$
- The function and its derivative both are monotonic.
- It should only be used within Hidden layers of a Neural Network Model.
- Hence for output layers we should use a SoftMax function for a Classification problem to compute the probabilities for the classes , and for a regression problem it should simply use a linear function.

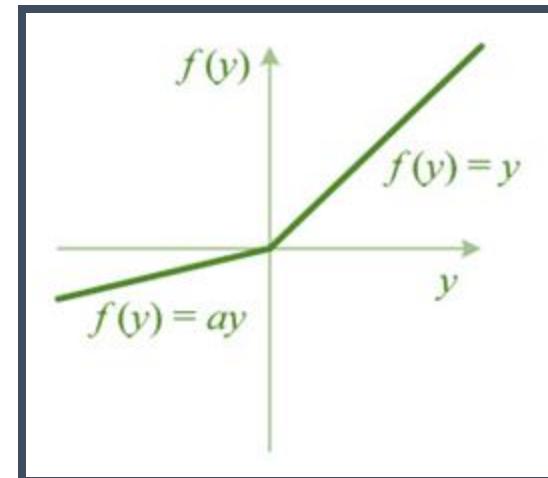
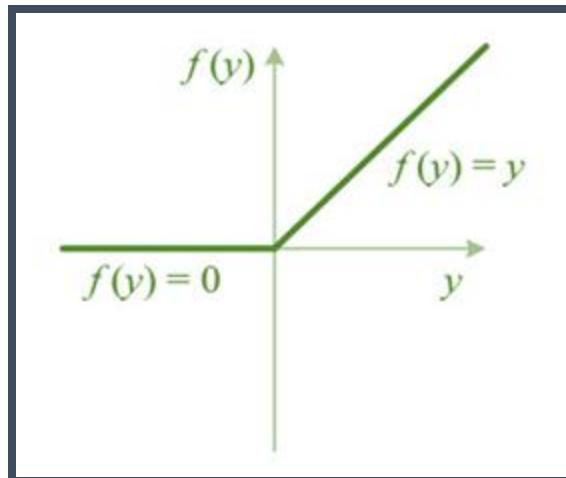


$$f(x) = \max(x, 0)$$

# Non-Linear Activation Function (Leaky ReLu)

The problem with ReLu is that some gradients can be fragile during training and can die. It can cause a weight update which will makes it never activate on any data point again. Simply saying that ReLu could result in Dead Neurons.

- To fix the problem of dying neurons another modification was introduced called Leaky ReLu. It introduces a small slope to keep the updates alive.



$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases}$$



04

## GRADIENT DESCENT

- Gradient Descent Algorithm
- Types of gradient descent

# Gradient Descent

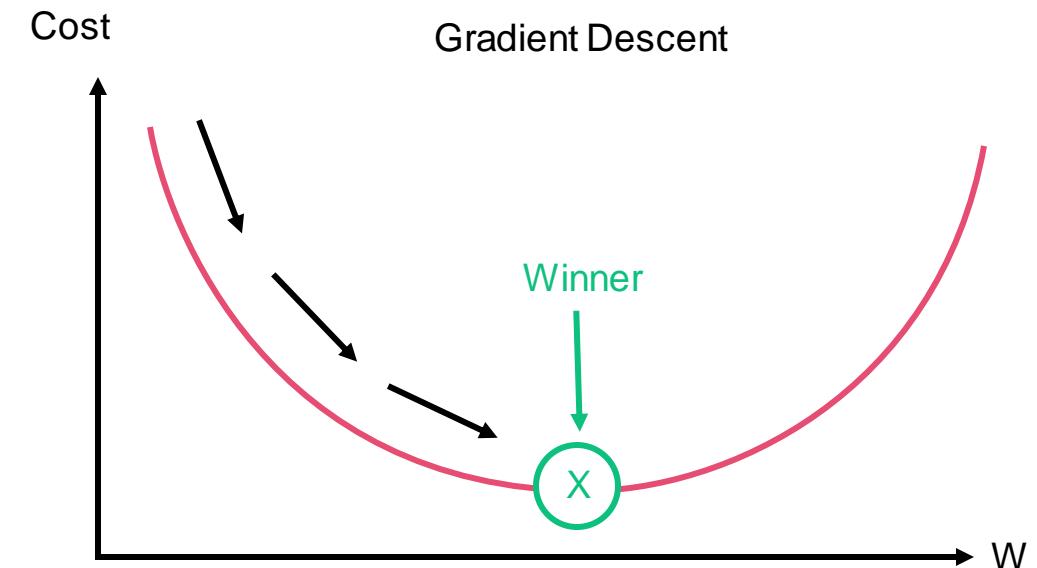
Following are the key details about gradient descent:

- Optimization technique used to reduce the error or cost function.
- It moves iteratively towards the steepest direction in order to reduce the value of the cost function.
- Gradient descent is used to update the values of parameters in any machine learning algorithm.
- It is used to update the values of Bias and Weights in any neural network.

# Gradient Descent Terminologies

Some important terminologies to be understood for Gradient Descent are as follows:

- Move towards to lowest point of the slope also called as local minima
- The size of the steps taken to reach the lowest point is called Learning Rate
- Cost Function tells how good the models are performing. Different cost functions are used in different algorithms



# Types of Gradient Descents

There are three types of gradient descent. They are as follows:

## Batch gradient descent

All training observations are taken at once for computation of gradients.

## Stochastic Gradient Descent

One training observation is taken at a time to calculate gradients.

## Mini-Batch Gradient Descent

Small batches with defined number of training observations are taken at a time for calculation of gradients.



05

## CONCLUSION

- Quiz
- Summary

# Quiz

## Question

Which activation function in the output layer to predict probabilities for n classes ( $n > 2$ ) ?

- A. Sigmoid
- B. TanH
- C. Softmax
- D. ReLu



# Quiz

## Question

Which activation function in the output layer to predict probabilities for n classes ( $n > 2$ ) ?

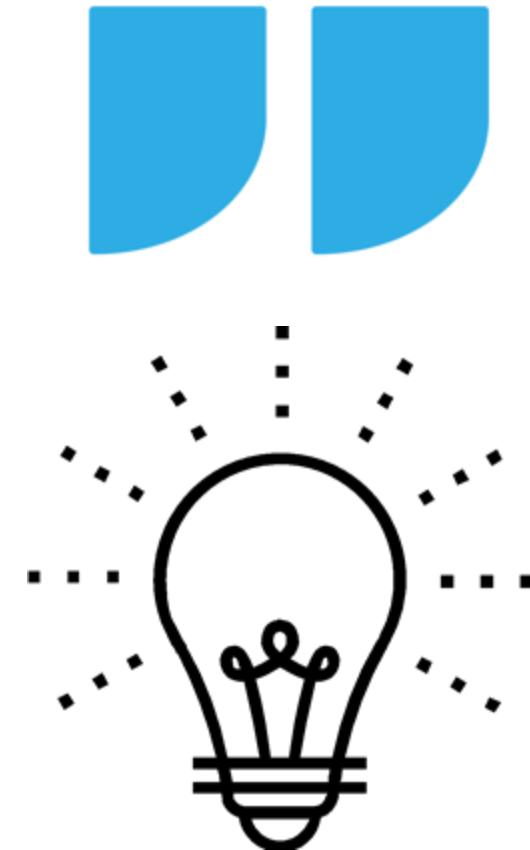


- A. Sigmoid
- B. TanH
- C. Softmax
- D. ReLu

Answer: C

# Summary

- Neural networks are composed of neurons (linear functions) arranged in layers, three types of layers – input, hidden and output layers.
- Each computational neuron performs weighted sum and activation function. Input layer neurons do not perform any computation on data
- In a dense neural network, each neuron of one layer is connected to each neuron of next layer
- There are 3 types of network architecture- single feed, multi feed and RNN.
- There are 2 types of activation functions- linear and non-linear.





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

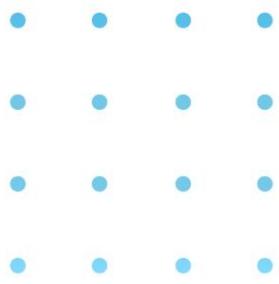
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# AI Mastery Course



Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY  
Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

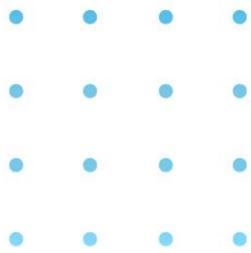
## Module 3

Data analytics with python- applied statistics

### Section

Data Visualization in Python using  
Matplotlib & Seaborn

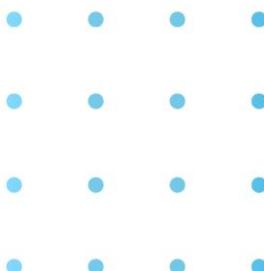




# Learning Objectives

Pada akhir modul ini, Anda akan dapat:

- Memahami **teknik visualisasi** data dengan python
- Memahami fungsi visualisasi dasar dengan matplotlib
- Membuat grafik dan visualisasi yang berbeda seperti lineplot, pie charts, box plots, strip plots, swarmplots dengan matplotlib dan seaborn
- Membuat visualisasi untuk time series data



# Agenda

01

## VISUALIZATION

- Data Visualization basics

02

## MATPLOTLIB

- Matplotlib
- Matplotlib functions

03

## GRAPHS

- Visualization using matplotlib
- Visualization with seaborn

04

## ADVANCE PLOTS

- Working with multiple axes,
- Plotting time series data

05

## CONCLUSION

- Summary
- Quiz



01

## VISUALIZATION

- Data Visualization basics

# Data Visualization

- Visualisasi Data digunakan untuk mengkomunikasikan informasi dengan jelas dan efisien kepada pengguna dengan menggunakan grafik informasi seperti tables and charts.
- Ini membantu pengguna dalam menganalisis sejumlah besar data dengan cara yang lebih sederhana. Itu membuat data kompleks lebih mudah diakses, dimengerti, dan dapat digunakan.
- 





02

## MATPLOTLIB

- Matplotlib
- Matplotlib functions

# matplotlib

- Matplotlib Adalah salah satu paket Python paling populer yang digunakan untuk visualisasi data.
- Ini adalah cross-platform library Untuk membuat 2D plots dari data di arrays.
- Matplotlib Ditulis dengan Python dan menggunakan NumPy, Ekstensi Matematika numerik dari Python.
- Ini memberikan sebuah object-oriented API Yang membantu dalam embedding plots dalam aplikasi menggunakan Python GUI toolkits misalnya PyQt, WxPythonnotTkinter.
- Hal ini dapat digunakan dalam Python dan IPython shells, Jupyter notebook dan server aplikasi web juga.

# Matplotlib image functions



Sr.No	Function & Description
1	<b>Imread</b> Baca gambar dari file ke dalam array.
2	<b>Imsave</b> Menyimpan array seperti dalam file gambar.
3	<b>Imshow</b> Menampilkan gambar pada axes.

# Matplotlib figure functions



## Sr.No Function & Description

- 1 **Figtext**  
Tambahkan teks ke gambar.
  
- 2 **Figure**  
Membuat gambar baru.
  
- 3 **Show**  
Menampilkan gambar.
  
- 4 **Savefig**  
Simpan gambar saat ini.
  
- 5 **Close**  
Tutup jendela gambar.

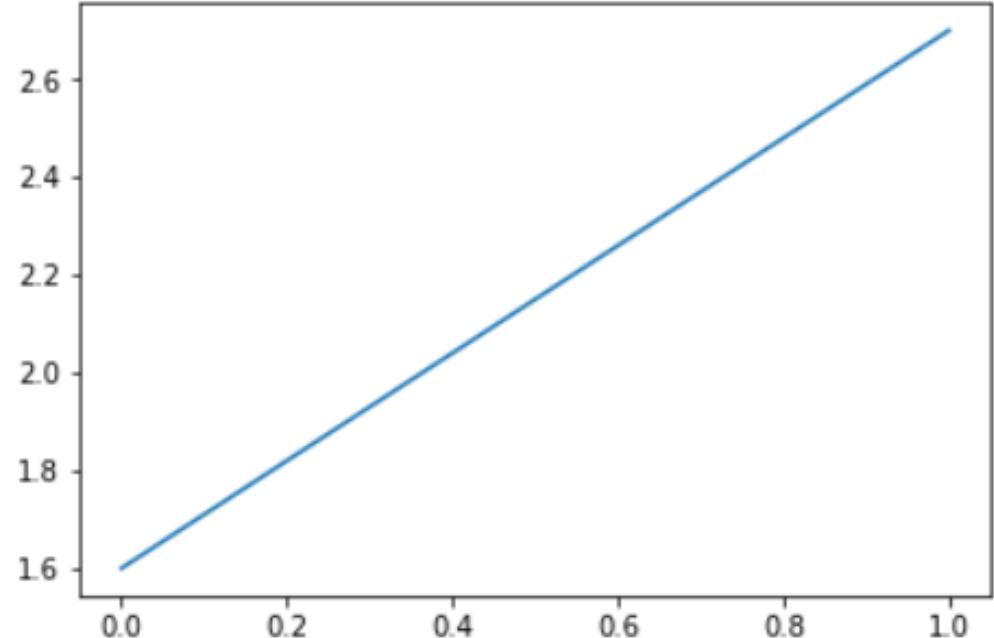


### 03 GRAPHS & PLOTS

- Visualization using matplotlib
- Visualization with seaborn

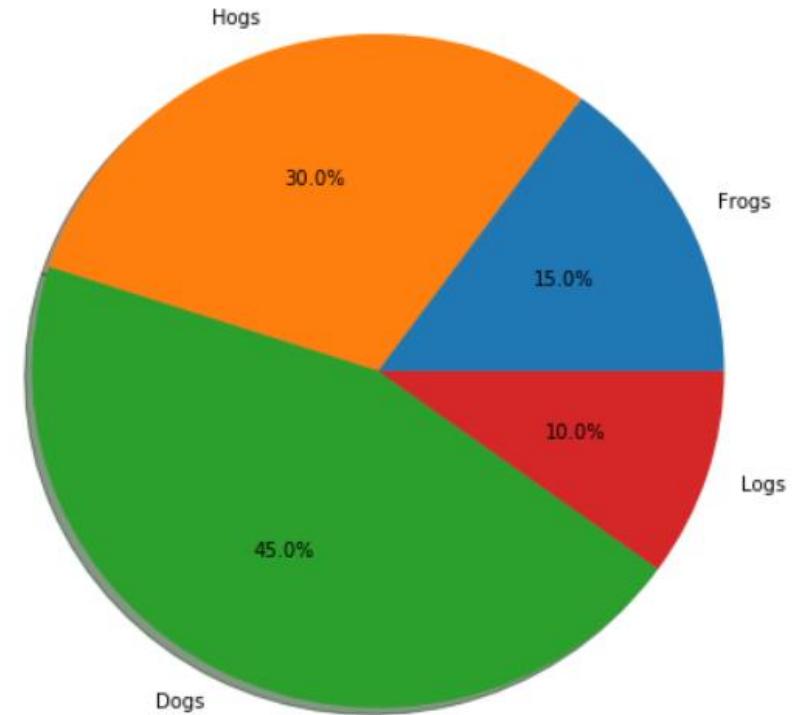
# Line Plot

```
import matplotlib.pyplot as plt  
plt.plot([1.6, 2.7])
```



# Pie chart

```
import matplotlib.pyplot as plt  
  
from matplotlib.gridspec import GridSpec  
  
plt.figure(figsize=(8,8))  
  
labels = 'Frogs', 'Hogs', 'Dogs', 'Logs'  
  
fracs = [15, 30, 45, 10]  
  
explode = (0, 0.05, 0, 0)  
  
plt.pie(fracs, labels=labels,  
autopct='%.1f%%', shadow=True)  
  
plt.show()
```

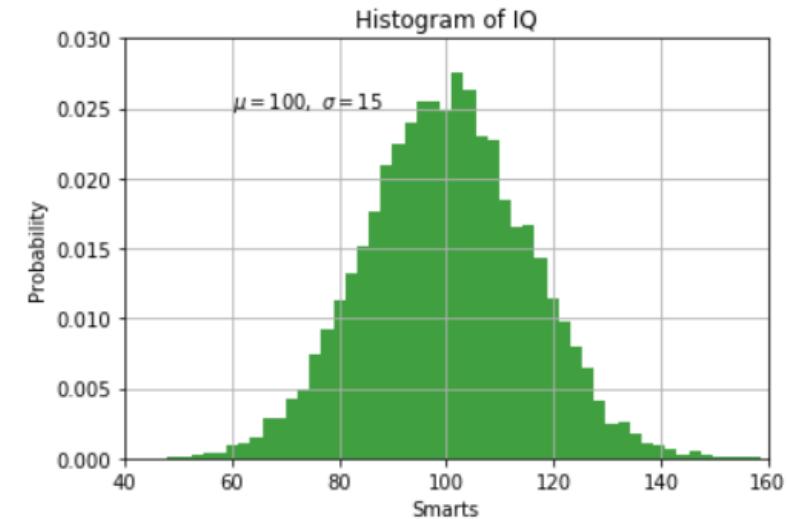


# Working with text

```
mu, sigma = 100, 15
x = mu + sigma * numpy.random.randn(10000)

n, bins, patches = plt.hist(x, 50, normed=1,
facecolor='g', alpha=0.75)

plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60,.025,r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```

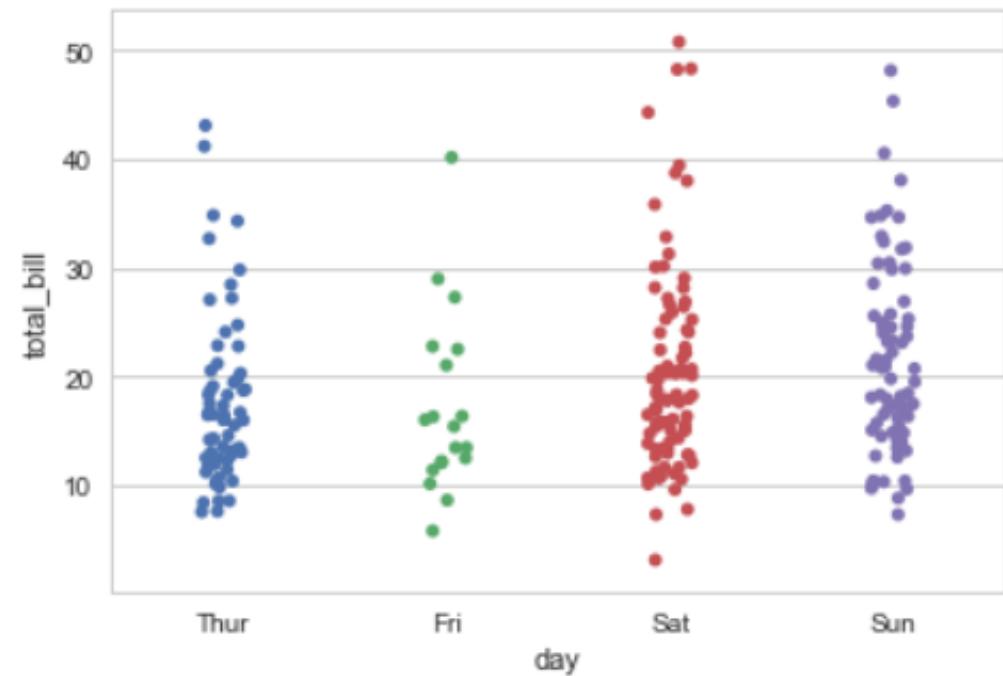


# Plotting with categorical data

```
import numpy as np  
  
import pandas as pd  
  
import matplotlib as mpl  
  
import matplotlib.pyplot as plt  
  
import seaborn as sns  
  
sns.set(style="whitegrid", color_codes=True)  
np.random.seed(sum(map(ord, "categorical")))  
  
titanic = sns.load_dataset("titanic")  
  
tips = sns.load_dataset("tips")  
  
iris = sns.load_dataset("iris")
```

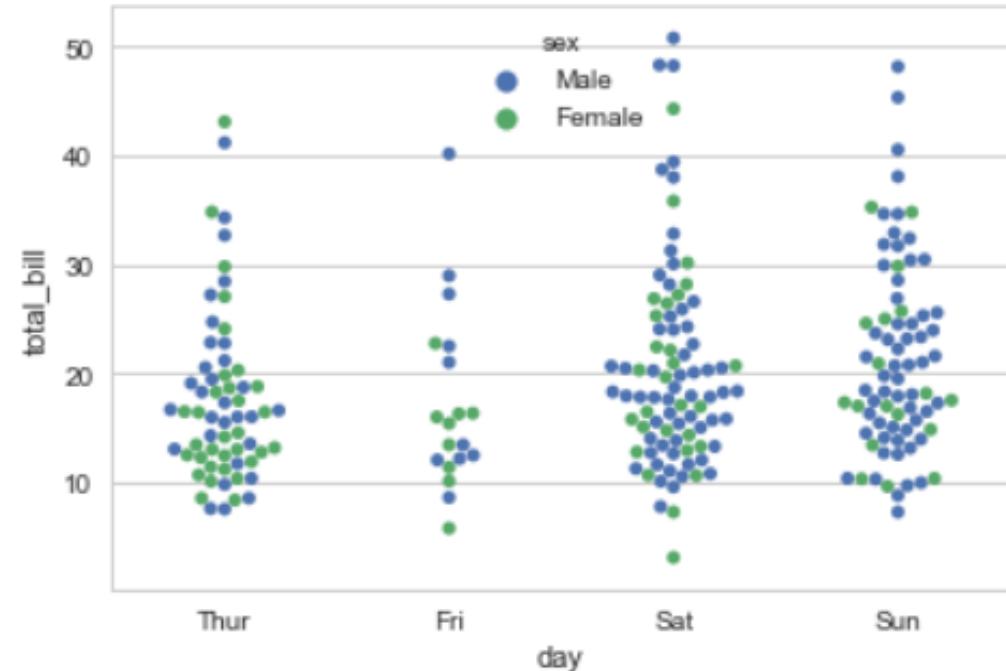
# Categorical scatterplots: Stripplots

```
sns.stripplot(x="day",  
y="total_bill", data=tips,  
jitter=True);
```



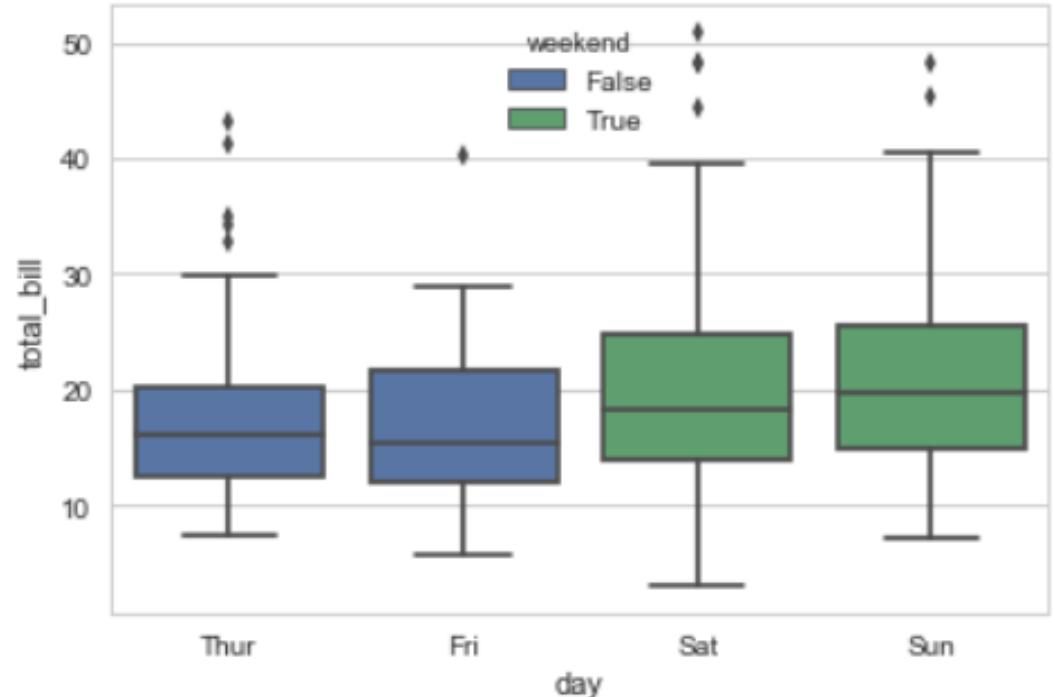
# Categorical scatterplots: swarmplots

```
sns.swarmplot(x="day",  
y="total_bill", hue="sex",  
data=tips);
```



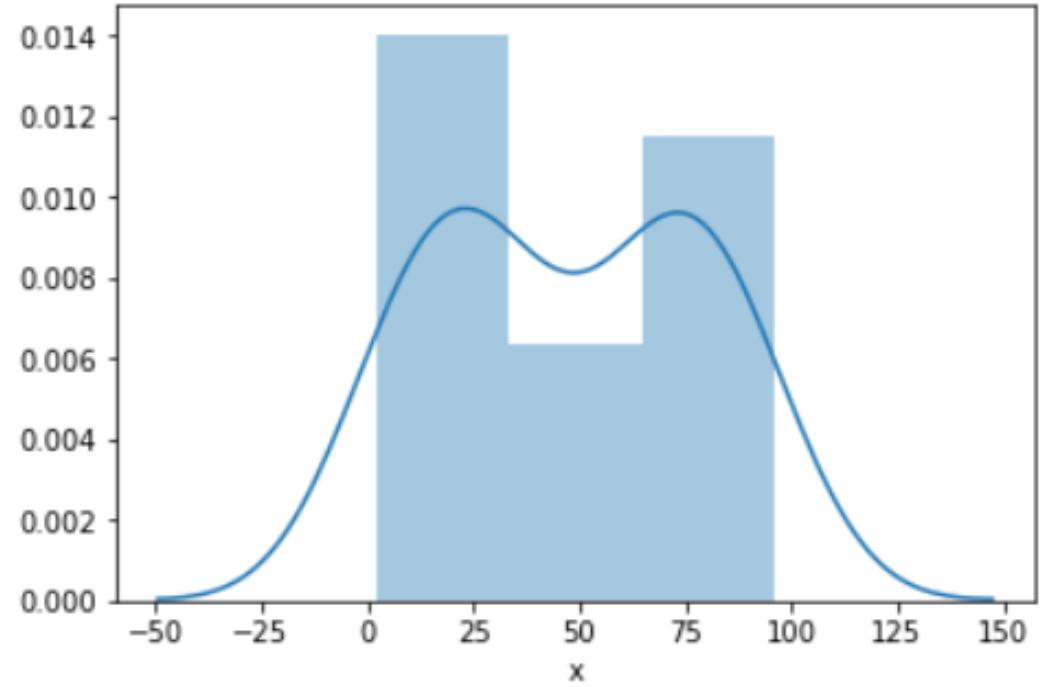
# Boxplot

```
tips["weekend"] =  
tips["day"].isin(["Sat", "Sun"])  
  
sns.boxplot(x="day",  
y="total_bill", hue="weekend",  
data=tips, dodge=False);
```



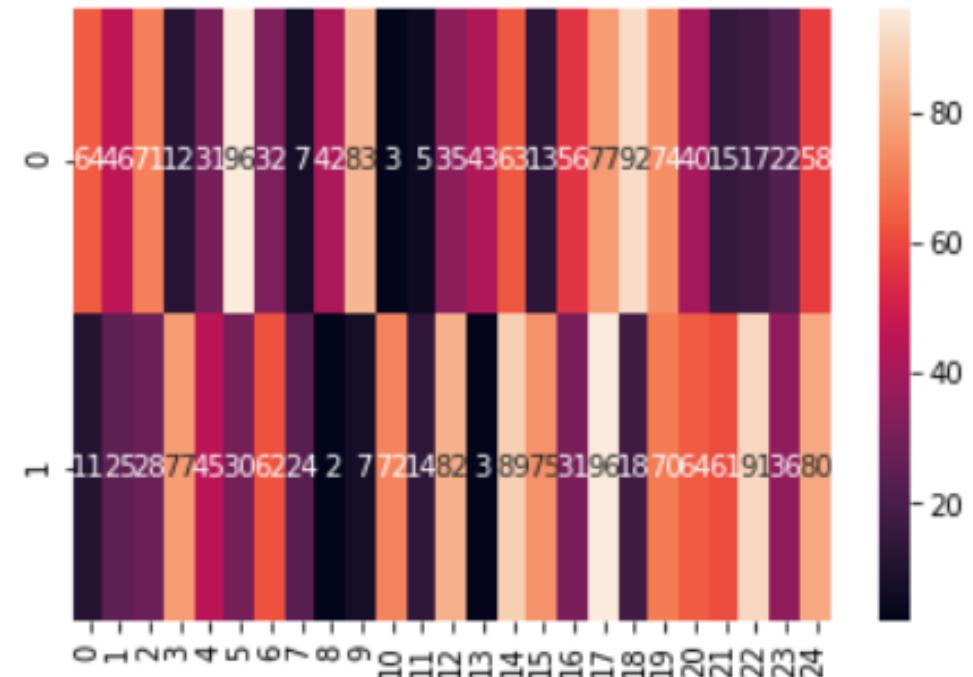
# Density Plot

```
sns.distplot(df.x)
```



# Heatmap

```
sns.heatmap([df.y, df.x],  
            annot=True, fmt="d")
```





04

## ADVANCE PLOTS

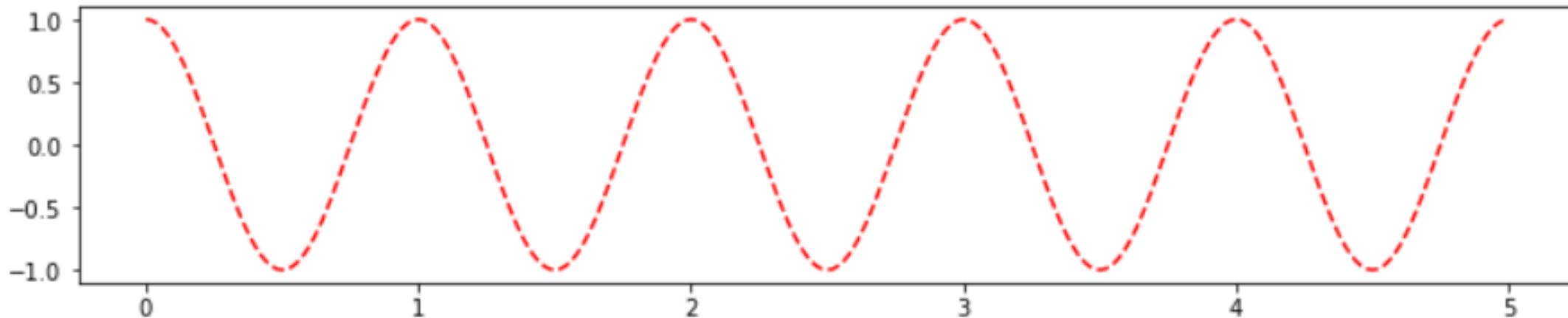
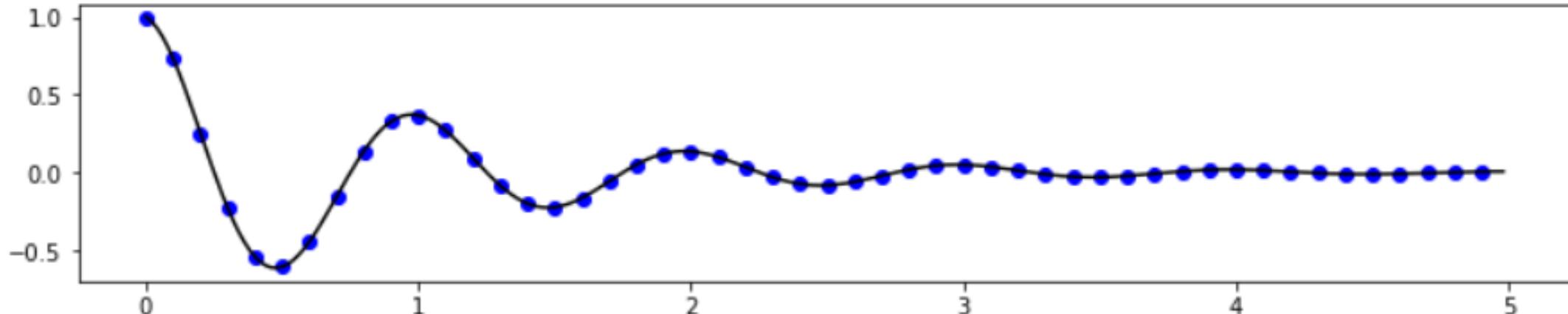
- Working with multiple axes,
- Plotting time series data

# Working with multiple figures and axes



```
import numpy  
  
t1 = numpy.arange(0.0, 5.0, 0.1)  
  
t2 = numpy.arange(0.0, 5.0, 0.02)  
  
f=numpy.exp(-t1)*numpy.cos(2*numpy.pi*t1)  
  
f2=numpy.exp(-t2)*numpy.cos(2*numpy.pi*t2)  
  
plt.figure(figsize=(12,5))  
  
plt.subplot(211)  
  
plt.plot(t1, f, 'bo', t2, f2, 'k')  
  
plt.subplot(212)  
  
plt.plot(t2, numpy.cos(2*numpy.pi*t2), 'r--')  
  
plt.show()
```

# Working with multiple figures and axes

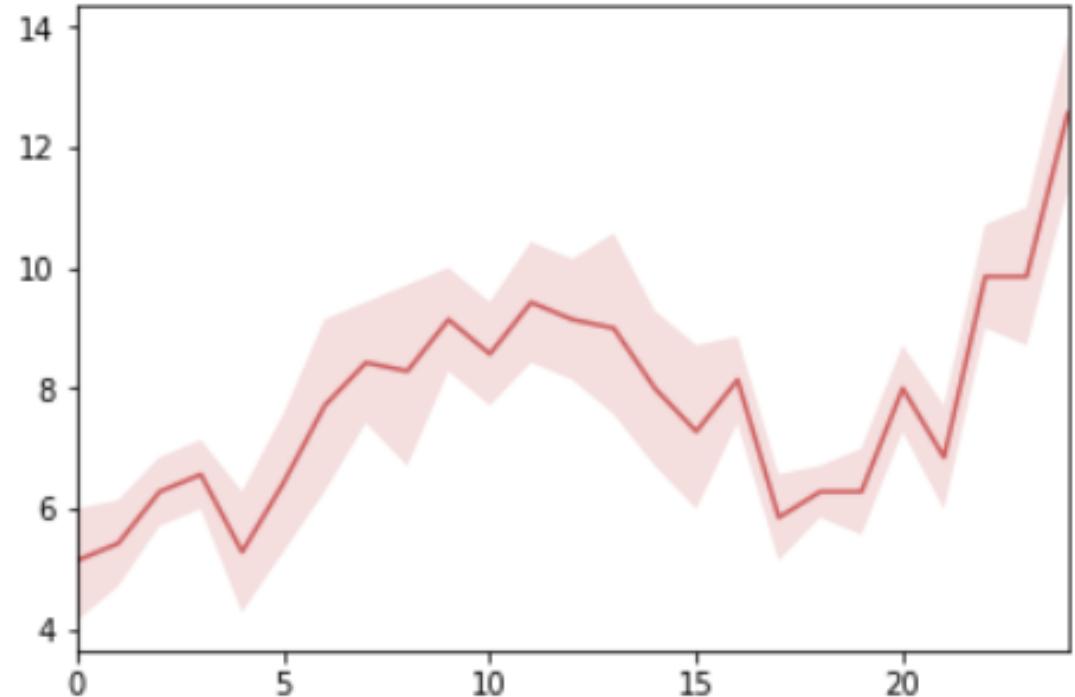


# Creating A Time Series Plot With Seaborn

```
import pandas  
  
data=pandas.read_csv('https://raw.githubusercontent.com/anshupande  
y/datasets/master/iiftfootfall.csv')
```

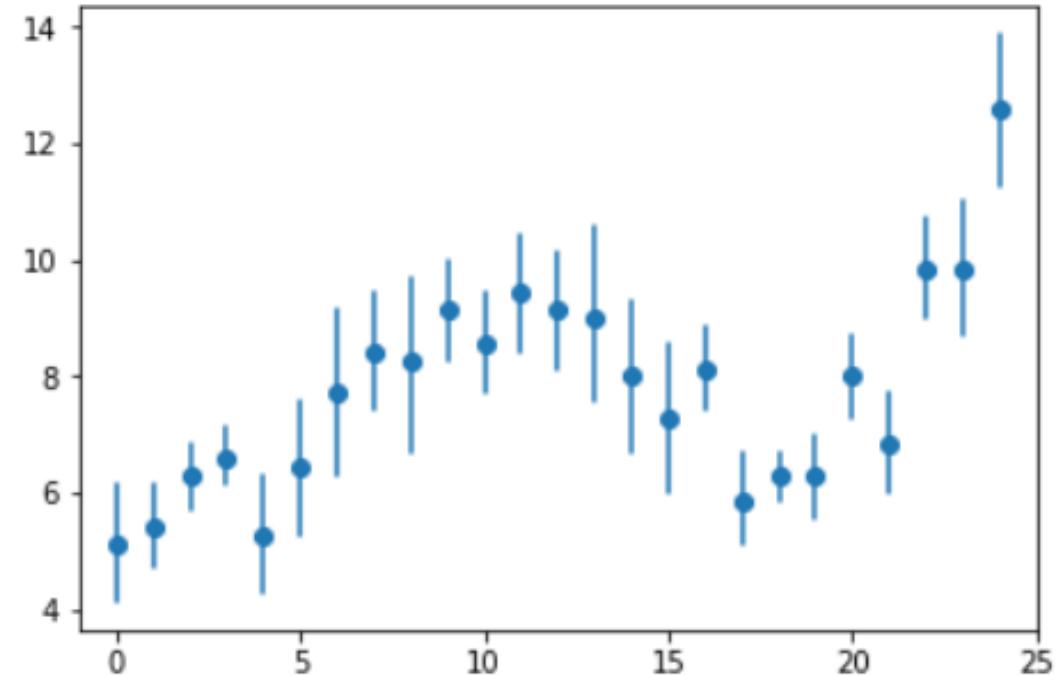
# Time Series Plot

```
sns.tsplot([data.Monday, data.Tuesday, data.Wednesday, data.Thursday, data.Friday, data.Saturday, data.Sunday], color="indianred")
```



# Time Series Splot With Confidence Interval Lines But No Lines

```
sns.tsplot([data.Monday, data.Tues  
day, data.Wednesday, data.Thursday,  
data.Friday, data.Saturday, data.Su  
nday], err_style="ci_bars",  
interpolate=False)
```





05 CONCLUSION

- Summary

# Summary



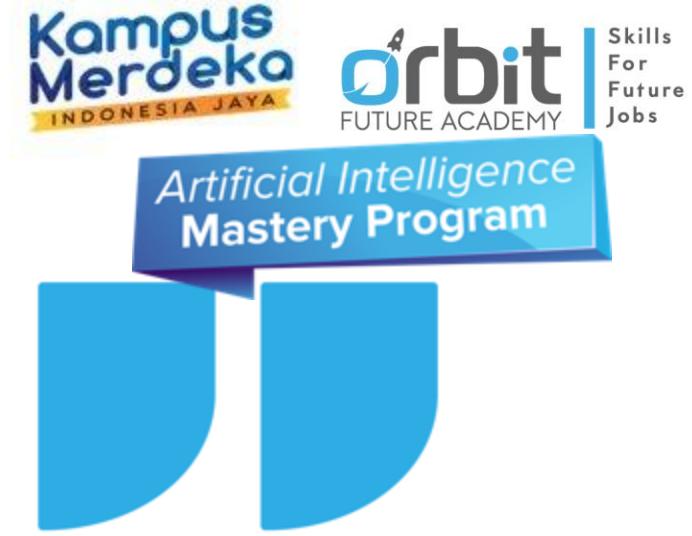
1. Data visualization basics
2. Fungsi visualisasi dan grafik dengan matplotlib
3. Memvisualisasikan data numerik dan kategoris dengan matplotlib dan seaborn
4. Memvisualisasikan data deret waktu dengan matplotlib dan seaborn

# Quiz

## Question

Manakah dari grafik yang dapat digunakan untuk menganalisis distribusi frekuensi sebuah variabel numerik?

- A. Barplot
- B. Histogram
- C. Piechart
- D. Tak satu pun dari di atas



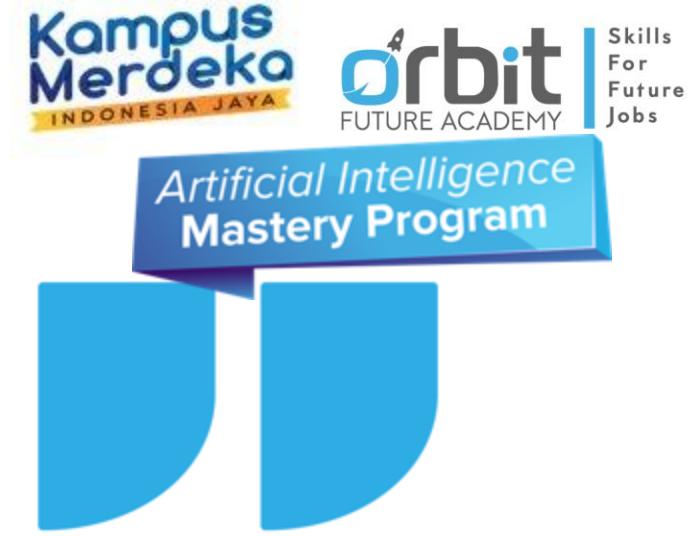
# Quiz

## Question

Manakah dari grafik yang dapat digunakan untuk menganalisis distribusi frekuensi sebuah variabel numerik?

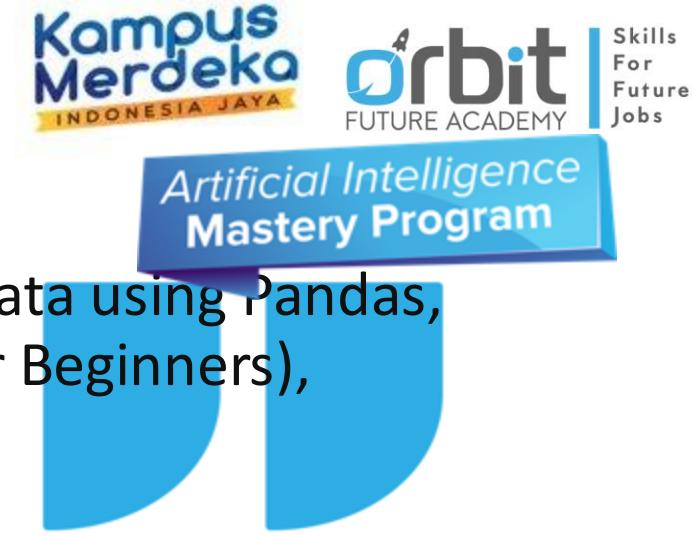
- A. Barplot
- B. Histogram
- C. Piechart
- D. None of above

Answer: B



# Reference

- Data Visualization with Python for Beginners: Visualize Your Data using Pandas, Matplotlib and Seaborn (Machine Learning & Data Science for Beginners), Publishing, AI





Kampus  
Merdeka  
INDONESIA JAYA

orbit  
FUTURE ACADEMY  
Skills  
For  
Future  
Jobs

Artificial Intelligence  
Mastery Program

# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

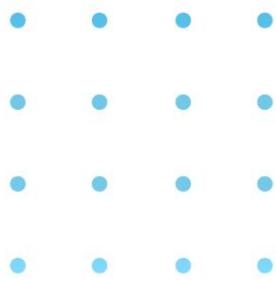
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

- Orbit Future Academy
- @OrbitFutureAcademyIn1
- OrbitFutureAcademy
- Orbit Future Academy

# AI Mastery Course



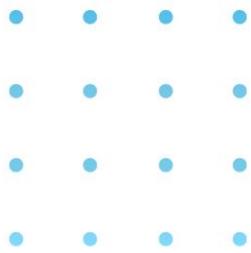
## Module 4

Supervised machine learning and  
predictive modelling

### Section

Linear regression





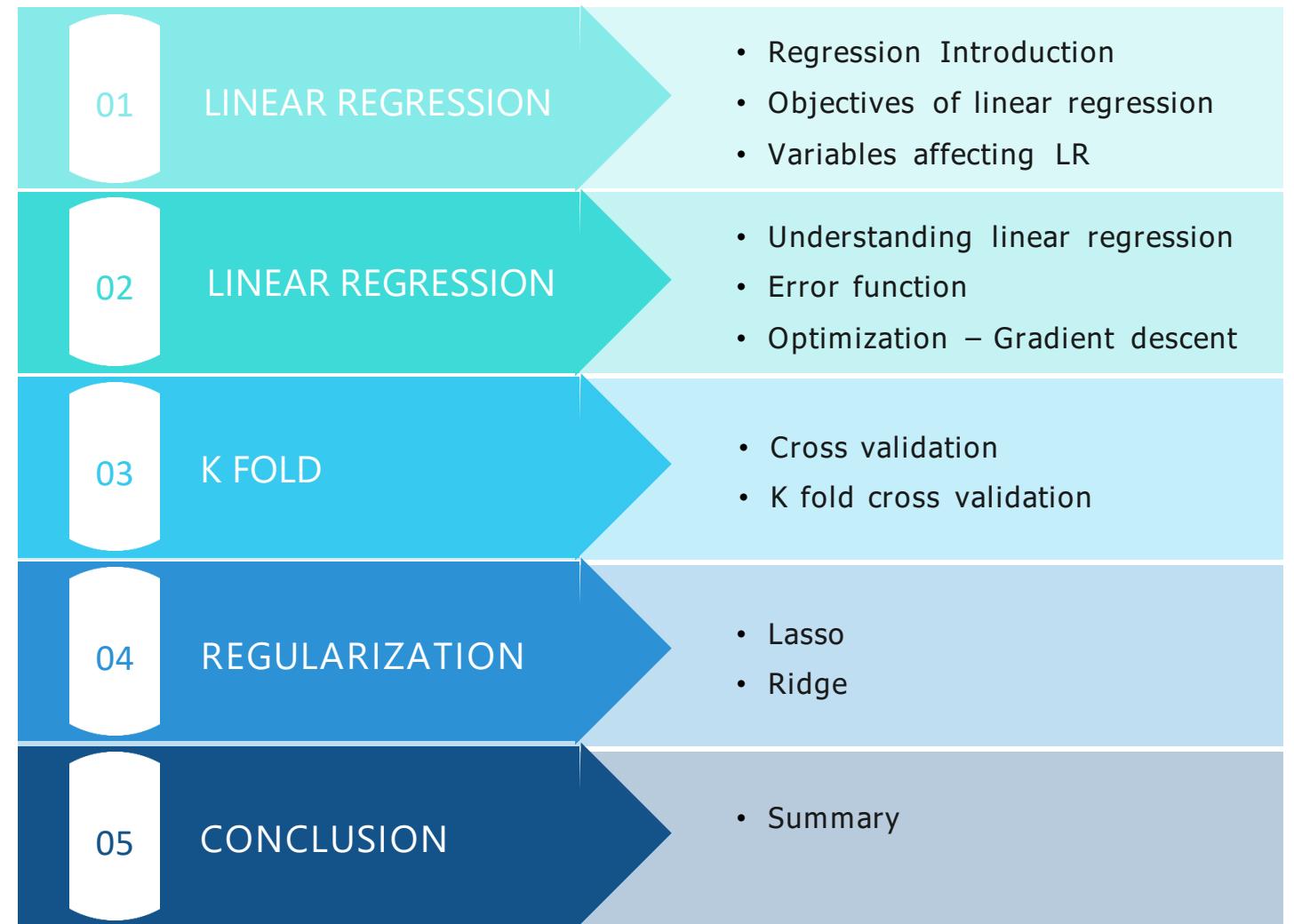
# Learning Objectives

Di akhir modul ini, Anda akan dapat:

- Memahami lebih detil tentang teknik regression dan berbagai use case nya
- Melihat bagaimana konsep gradient descent di linear regression
- Menjelaskan fungsi dari cost function
- Melakukan training model untuk memahami konsep gradient descent
- Mendeskripsikan regularisasi pada regression seperti: Lasso dan Ridge Regression
- Mendefinisikan Generalized Linear Regression (GLM)



# Agenda





## 01

## LINEAR REGRESSION

- Regression Introduction
- Objectives of linear regression
- Variables affecting LR

# Regression Technique

Analisa regression digunakan:

- Untuk memprediksi nilai-nilai dari dependent variable Y, berdasarkan relasinya dengan nilai-nilai dari independent variable X (minimal 1)
- Untuk menjelaskan apa pengaruh dari perubahan pada sebuah independent variable terhadap dependent variable melalui estimasi nilai numerik dari hubungan yang ada

Dependent variable  
(Y)

Variable yang ingin diperjelas

Independent  
variable (X)

Variable yang digunakan untuk menjelaskan dependent variable

Coefficients

Nilai yang memuat penjelasan tentang hubungan terhadap dependent variable yang dihitung dengan regression tool

Residuals

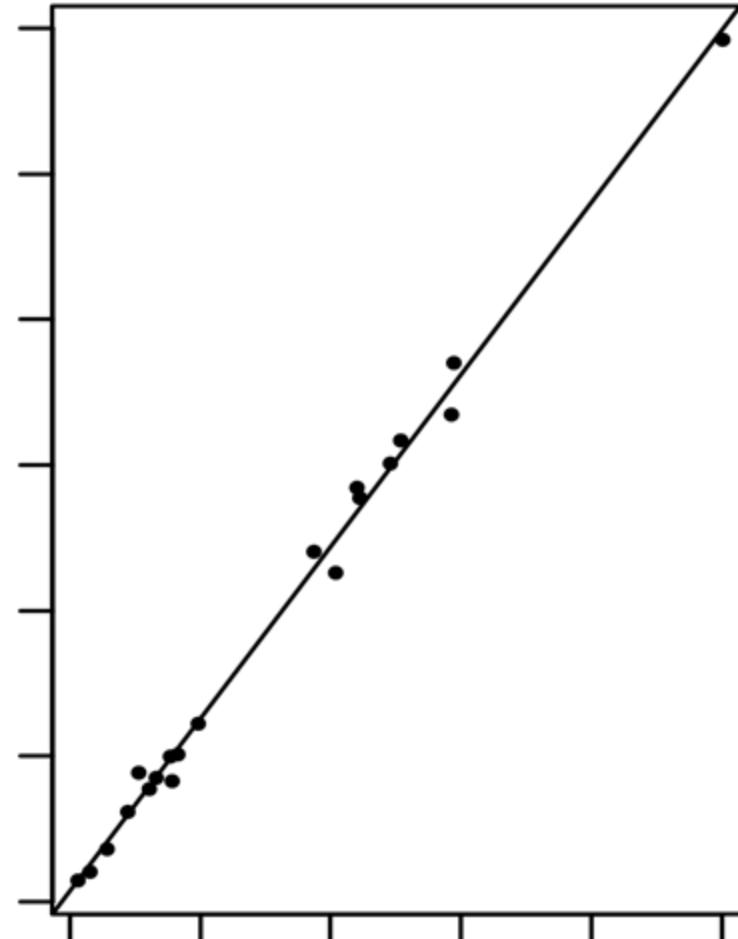
Porsi dari dependent variable yang tidak dijelaskan oleh model (model melakukan under dan over predictions)

# Regression in Real World

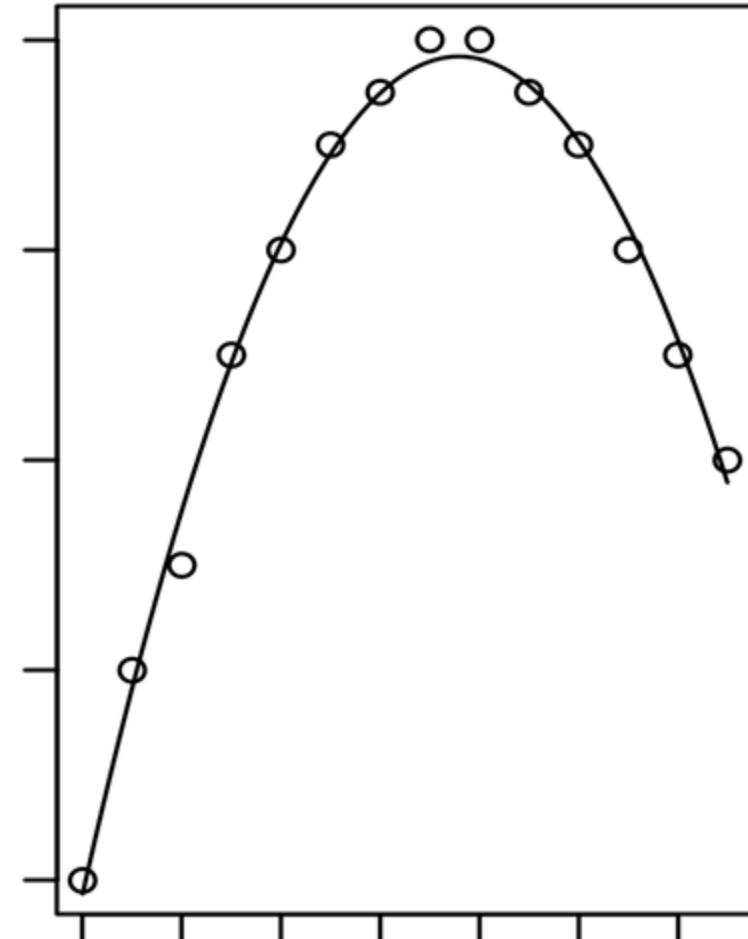
- Dalam penerapannya kita sering mendapati data-data numerik yang disimpan dalam bentuk tabel. Regression analysis sebagai tool yang sangat baik bisa digunakan untuk menganalisa data numerik tersebut. Secara umum, analisa regresi adalah sebuah proses untuk menemukan best fits dari sekumpulan data points.
- Contohnya: Kita memiliki beberapa deskriptor dari sebuah lagu seperti genre, author, duration, lyrics dan lain sebagainya. Tujuan dari masalah ini adalah untuk memprediksi tahun ketika lagu tersebut diproduksi. Pada dasarnya masalah ini bisa dikategorikan sebagai masalah regresi, karena variabel target yang ingin diprediksi adalah angka dalam kisaran antara tahun 1922 dan 2011.

# Regression Types

Linear

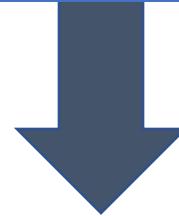


Non - Linear



# Linear Regression Types

**Number of the explanatory variables**



Simple regression



Multiple regression

# Objectives of Linear Regression

- Menganalisa relasi antara dua variable

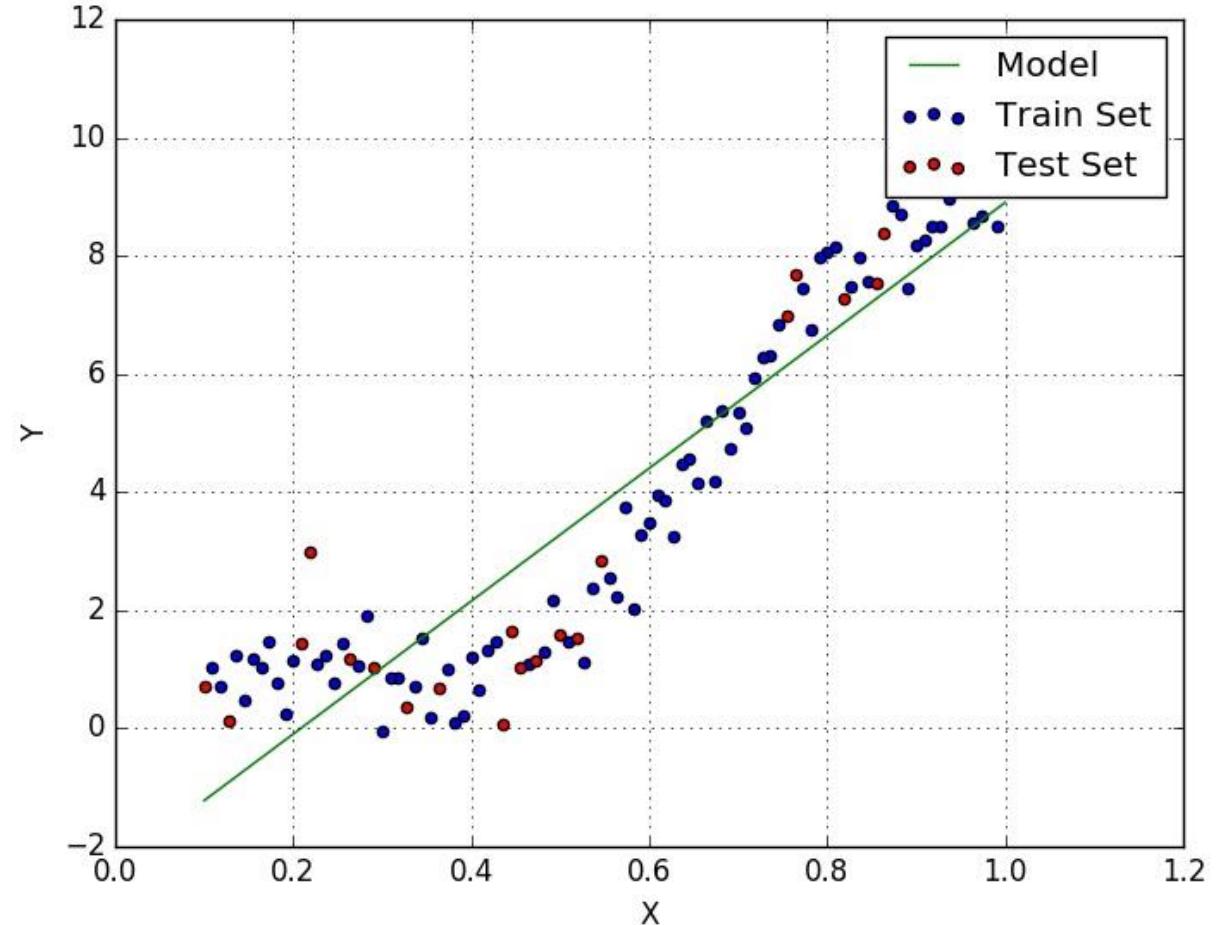
Contoh – relasi antara harga dan area/luas rumah, jumlah jam belajar dengan nilai yang didapat, pendapatan dan pengeluaran dan lain sebagainya

- Memprediksi nilai baru yang memungkin

Berdasarkan area/luas rumah memprediksi harga rumah tersebut di bulan tertentu, berdasarkan jumlah jam belajar memprediksi kemungkinan nilai yang akan didapat, memprediksi total penjualan di 3 bulan berikutnya dan lain sebagainya

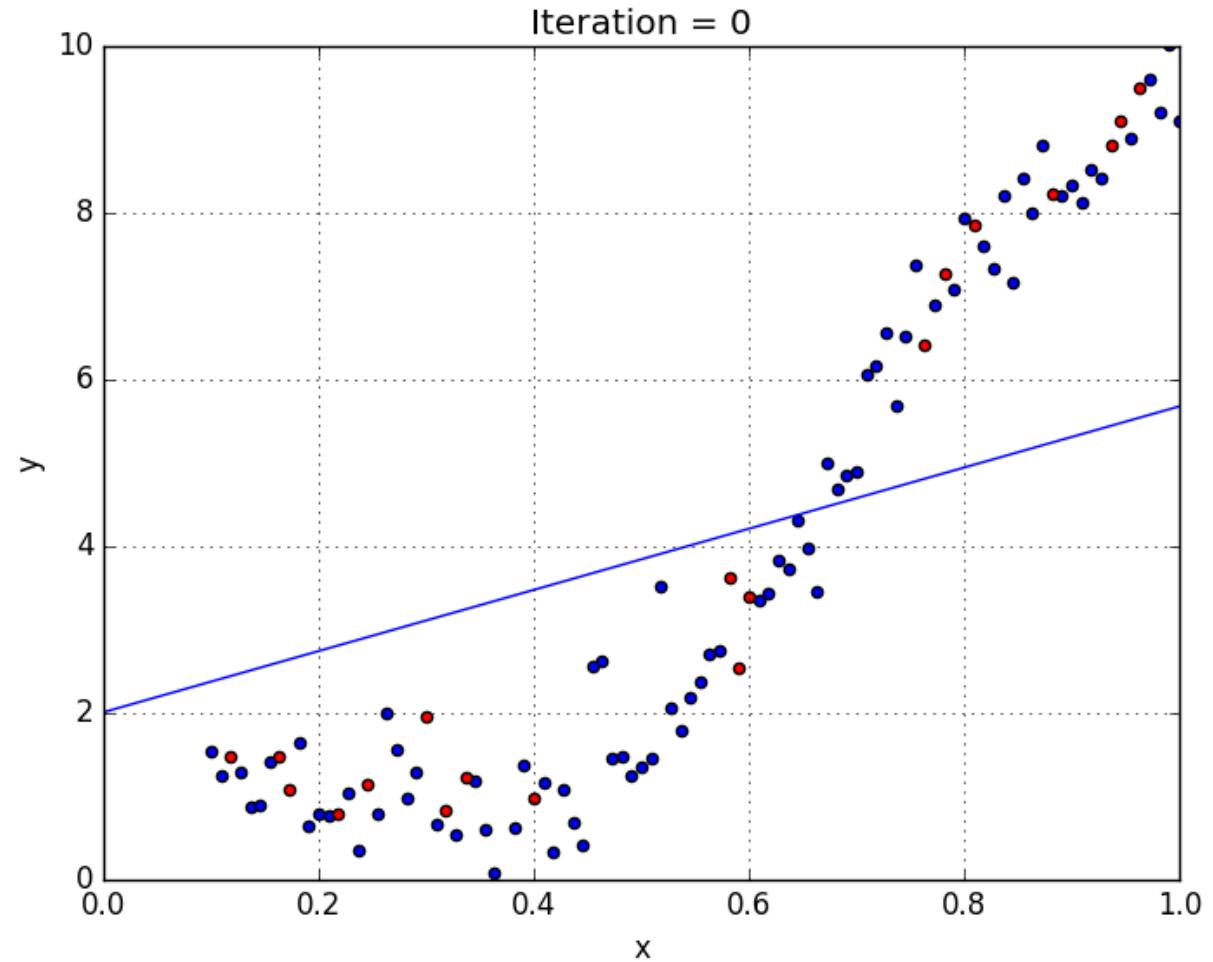
# What is Linear Regression?

- Algoritma supervised learning yang belajar dari sekumpulan sampel data latih
- Linear Regression mengestimasi hubungan antara dependent variable (target/label) dan 1 atau lebih independent variable (predictors).



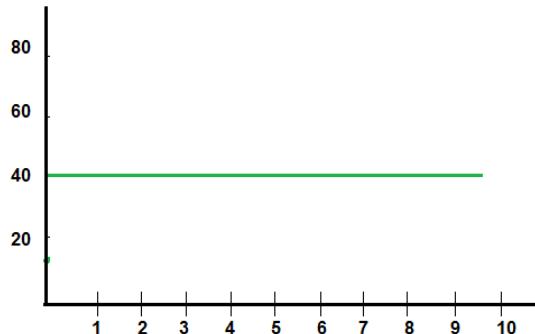
# Univariate Linear Regression

Selama periode latih (training), garis regresi akan sampai pada kondisi "fit".

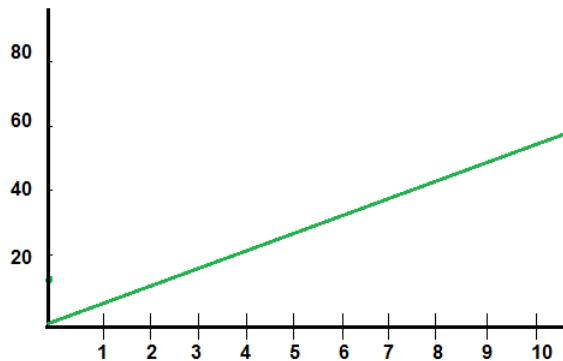


# Linear Regression – Variables affecting Regression Equation

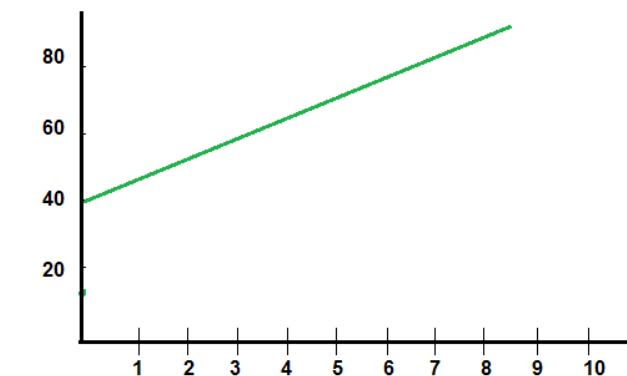
$$m = 0 \\ c = 40$$



$$m = 0.8 \\ c = 0$$



$$m = 0.8 \\ c = 40$$



$$\hat{y} = mx + c$$



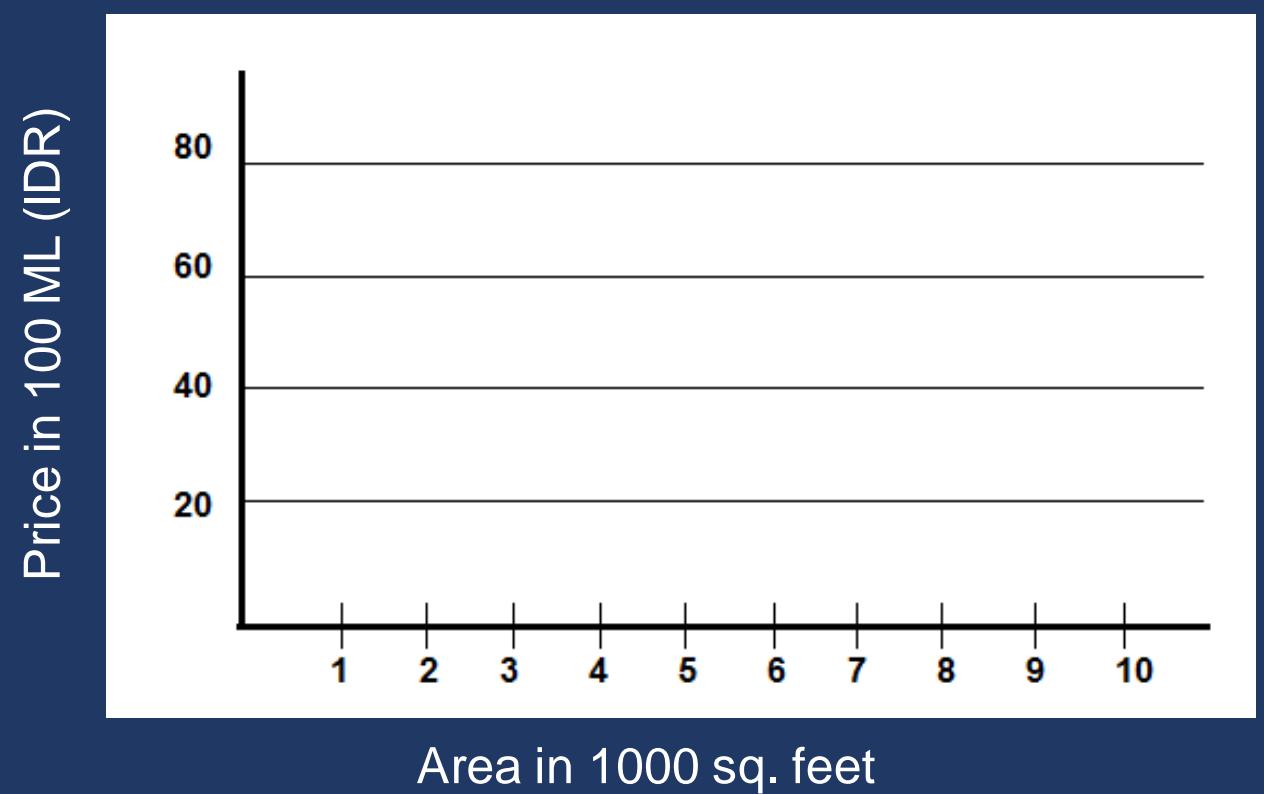
## 02

### LINEAR REGRESSION

- Understanding linear regression
- Error function
- Optimization – Gradient descent

# Linear Regression – Housing prices prediction

Area ( sq ft)	Price In IDR
1200	2,600,000K
1800	4,200,000K
3200	7,400,000K
3800	6,200,000K
4200	7,050,000K

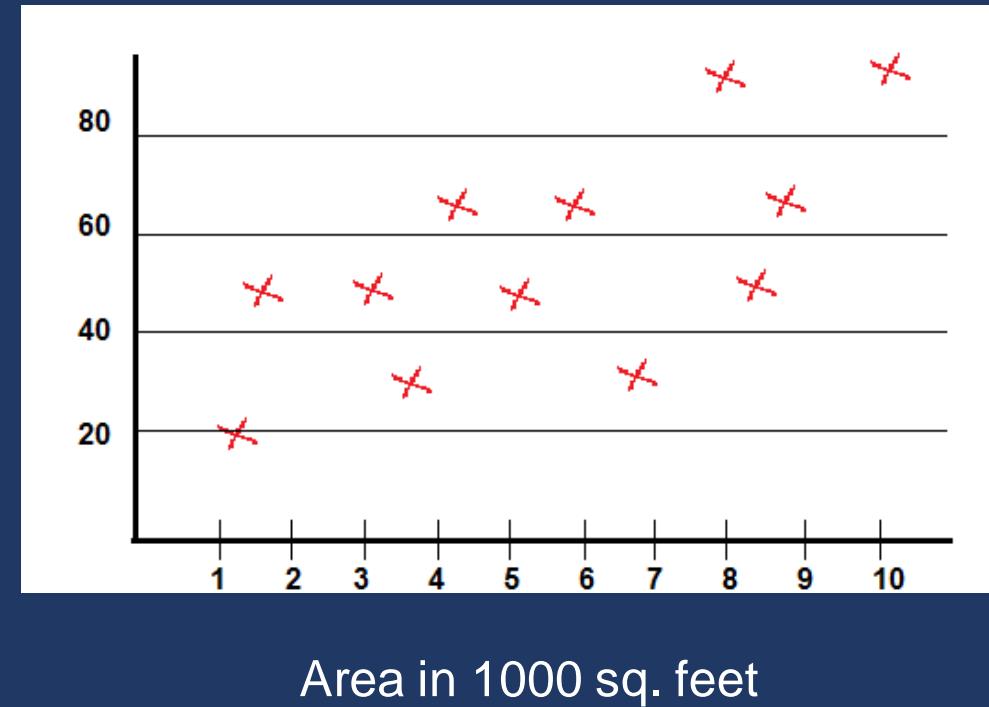


# Linear Regression – Housing prices prediction

Area ( sq ft )	Price In IDR
1200	2,600,000K
1800	4,200,000K
3200	7,400,000K
3800	6,200,000K
4200	7,050,000K

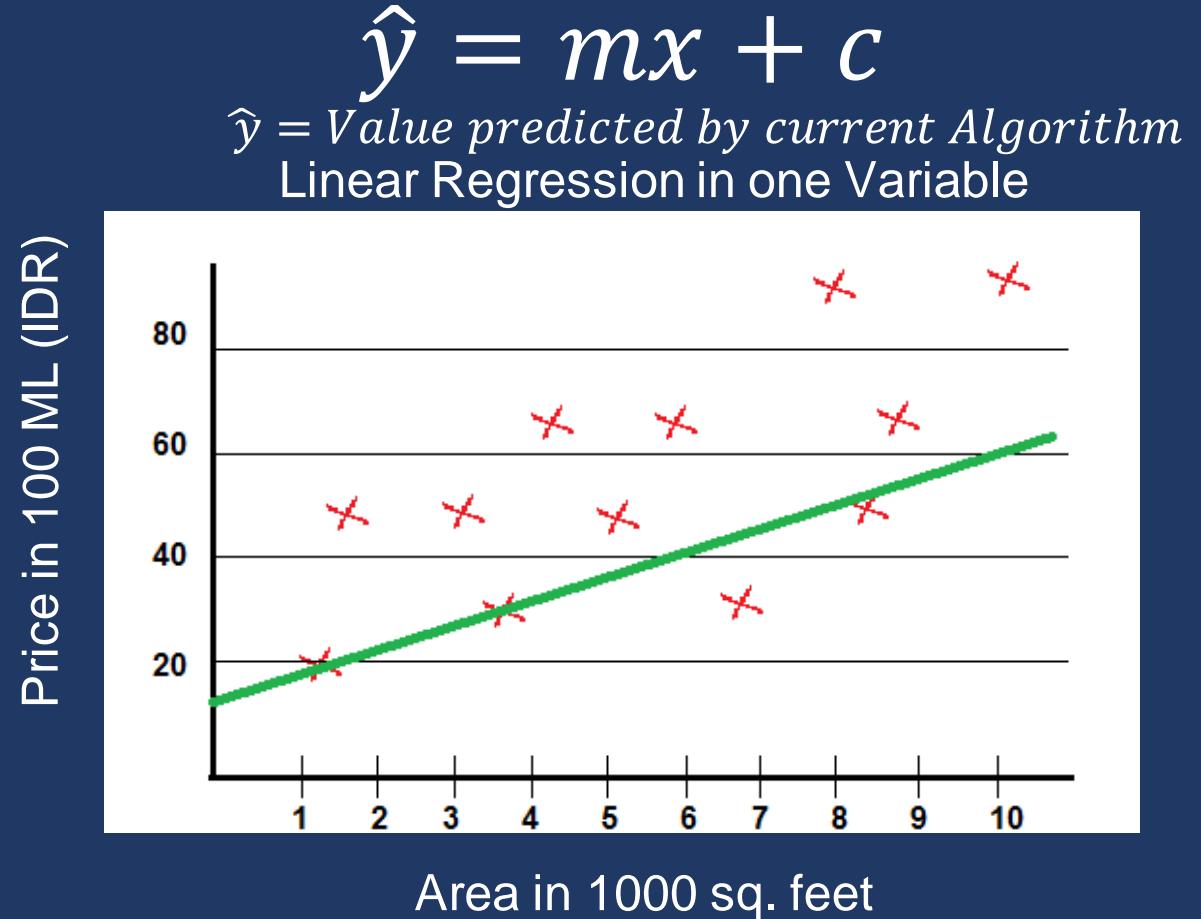
Price in 100 ML (IDR)

y: Dependent Variable, criterion variable.  
x: Independent variable, predictor variables.



# Linear Regression – Housing prices prediction

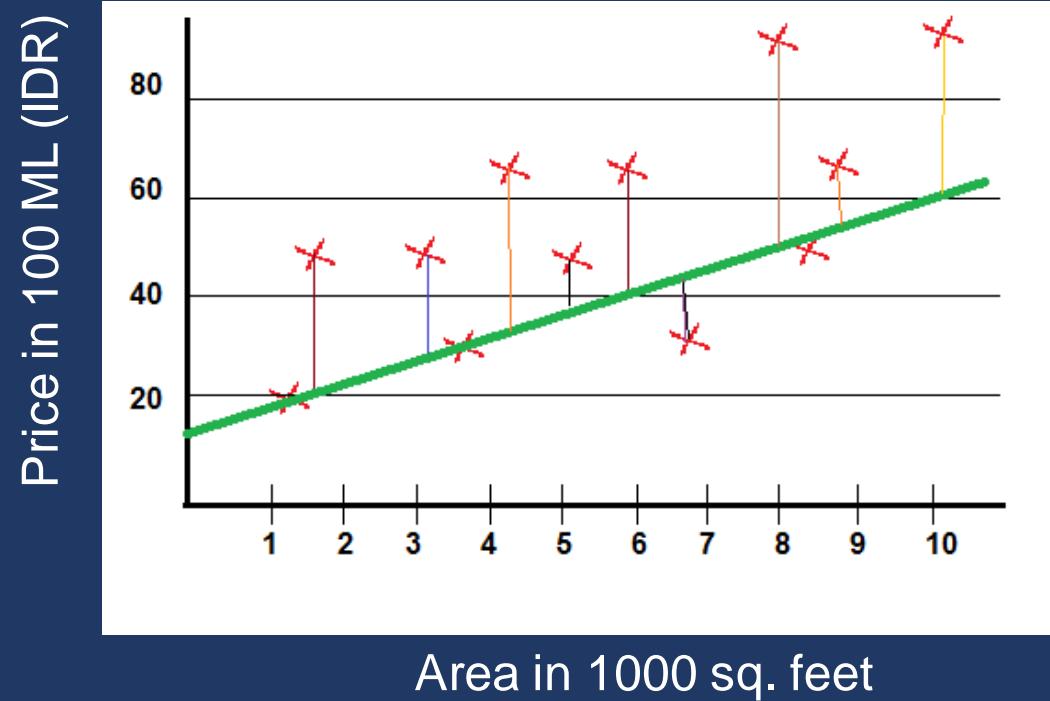
Area ( sq ft)	Price In IDR
1200	2,600,000K
1800	4,200,000K
3200	7,400,000K
3800	6,200,000K
4200	7,050,000K



# Linear Regression – Housing prices prediction

*minimize*  
 $(y - \hat{y})$

*Predictor*  
 $\hat{y} = mx + c$



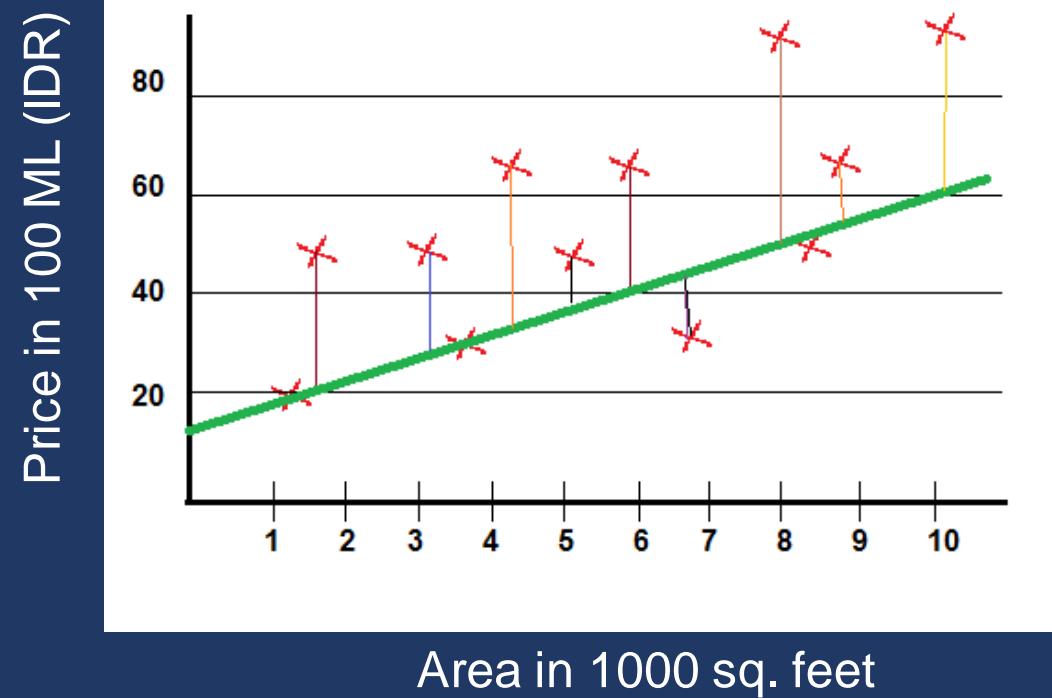
# Linear Regression – Housing prices prediction

*Cost Function*

$$J = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$j(m, c) = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

*Predictor*  
 $\hat{y} = mx + c$



# Linear Regression

Regression Equation:

$$\hat{y} = mx + c$$

Parameters

$$m_i, c$$

Cost Function:

$$j(m_i, c) = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Goal

$$\underset{m_i, c}{\text{minimize}} J(m_i, c)$$

# Linear Regression – Gradient Descent Algorithm

Repeat Until converge

$$w_j := w_j - lr \frac{\partial}{\partial w} J(w_j)$$

simultaneously update,  $j = 0, j = 1$   
where, w=parameter (coefficient & constant)

# Learning Rate - lr

- Learning rate lr akan mengontrol seberapa besar update yang akan terjadi pada parameter weight
- Jika lr terlalu kecil, maka gradient descent menjadi sangat lamban
- Jika lr terlalu besar, maka gradient descent akan mengalami overshoot dan pada akhirnya gagal untuk convergence

# Linear Regression - OLS

Equation Line formula:

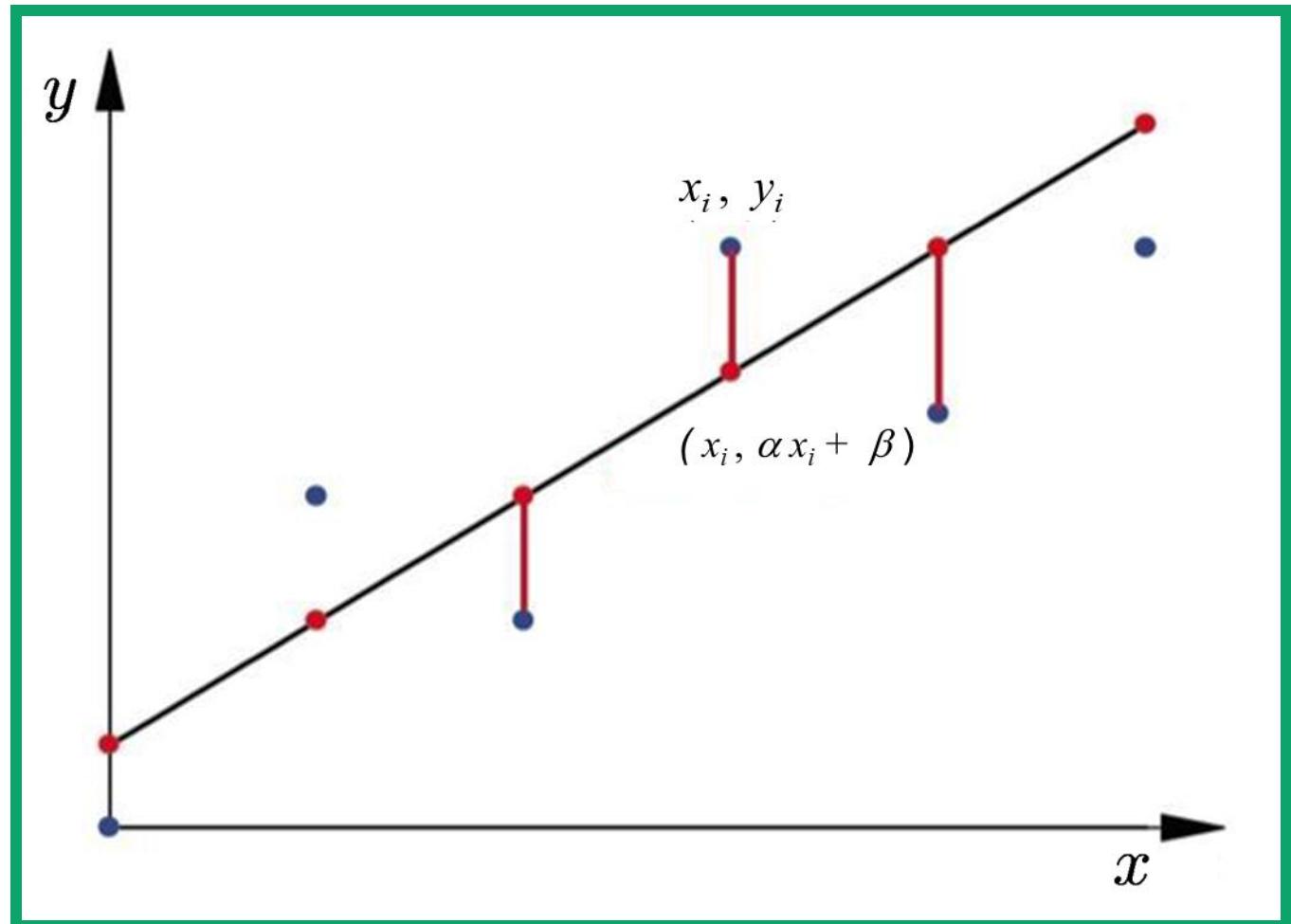
$$y = \alpha * x + \beta$$

Quantity in Minimal

$$E = \sum_{i=1}^n (\alpha x_i + \beta - y_i)^2$$

Sum of the squares of the distances

$$(x_i, \alpha x_i + \beta)$$



# Linear Regression – R squared

**RSS**

$$SS_{res} = \sum_{i=1}^n (y_i - \alpha x_i + \beta)^2$$

**TSS**

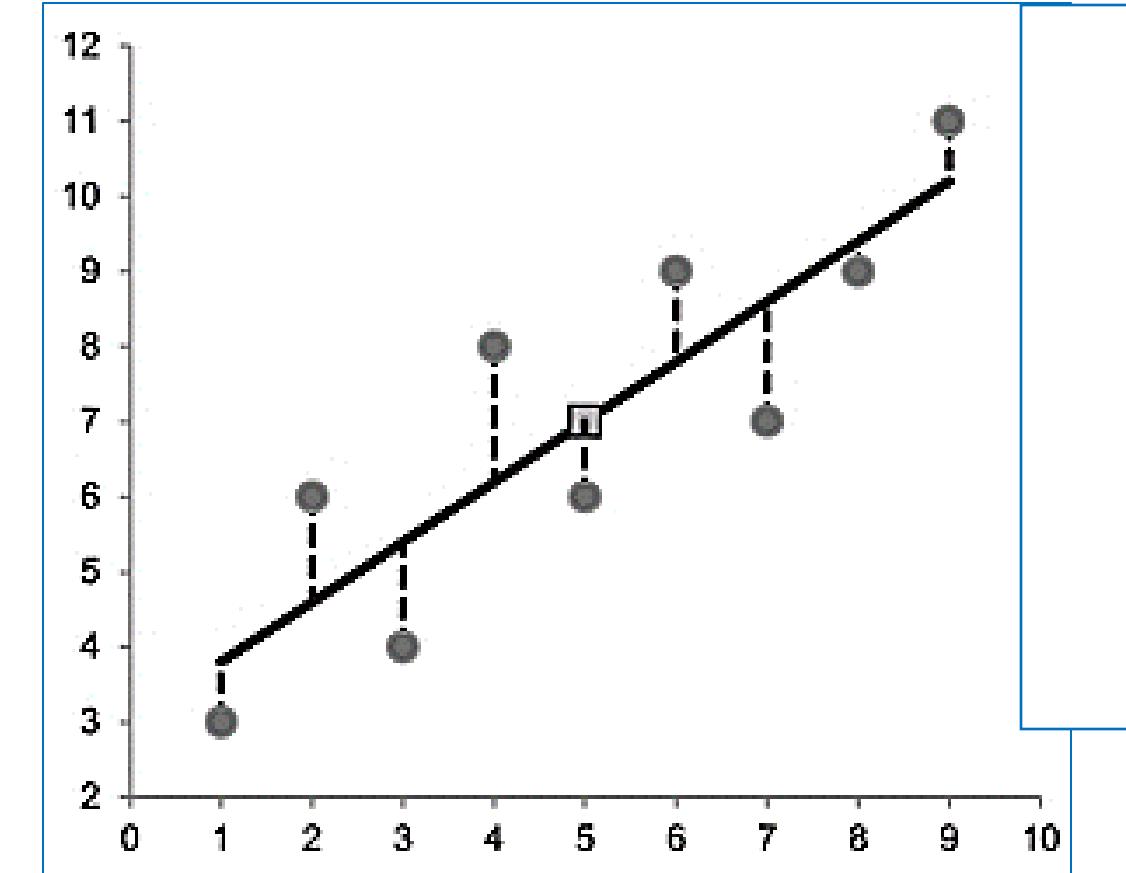
$$SS_{tot} = \sum_{i=1}^n (y_i - y_{mean})^2$$

**R-squared**

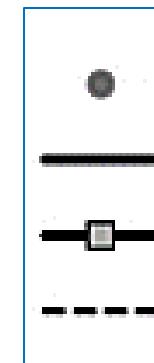
$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^n (y_i - \alpha x_i + \beta)^2}{\sum_{i=1}^n (y_i - y_{mean})^2}$$

# Linear Regression – Goodness of Fit (Contd..)

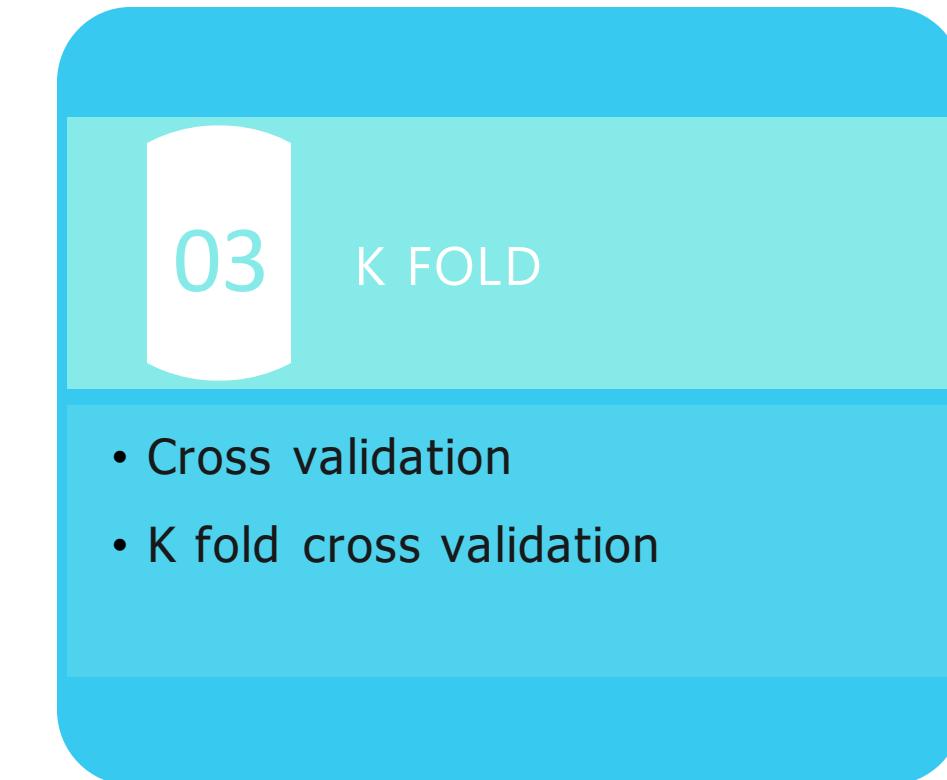
## Least Squares Linear Regression



Meminimalkan jumlah kuadrat dari residu vertikal antara setiap titik dan garis yang berada di antara data

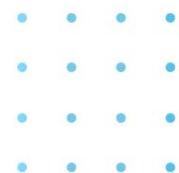


**Actual Data**  
**Line of Best Fit**  
**Data Mean**  
**Residuals or Errors**

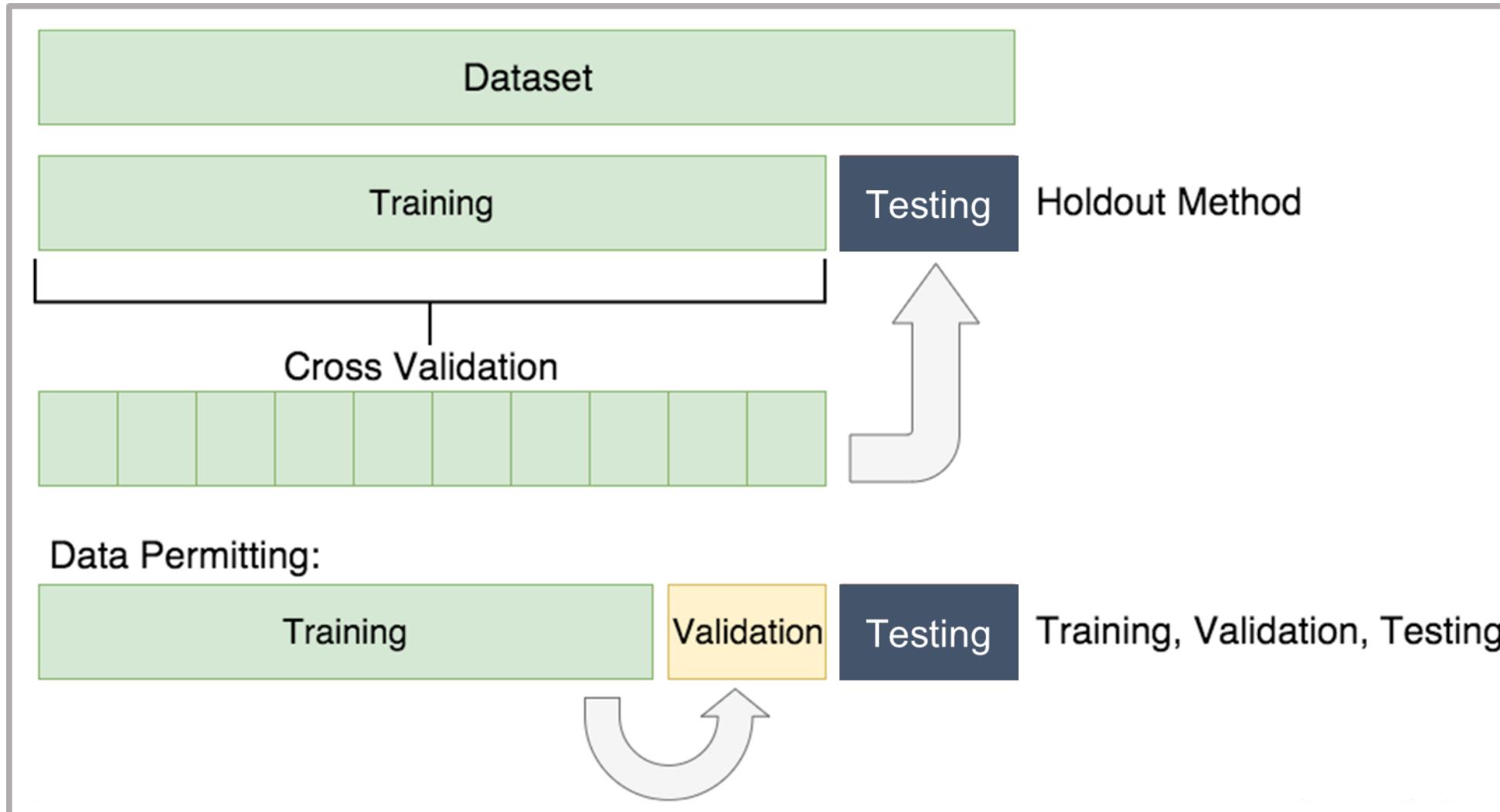


03 K FOLD

- Cross validation
- K fold cross validation



# Testing Model Using Cross Validation



# Cross Validation Types

## K – Fold

Satu subset data digunakan untuk melakukan validasi/pengujian terhadap model sedangkan sisanya digunakan untuk data latih.

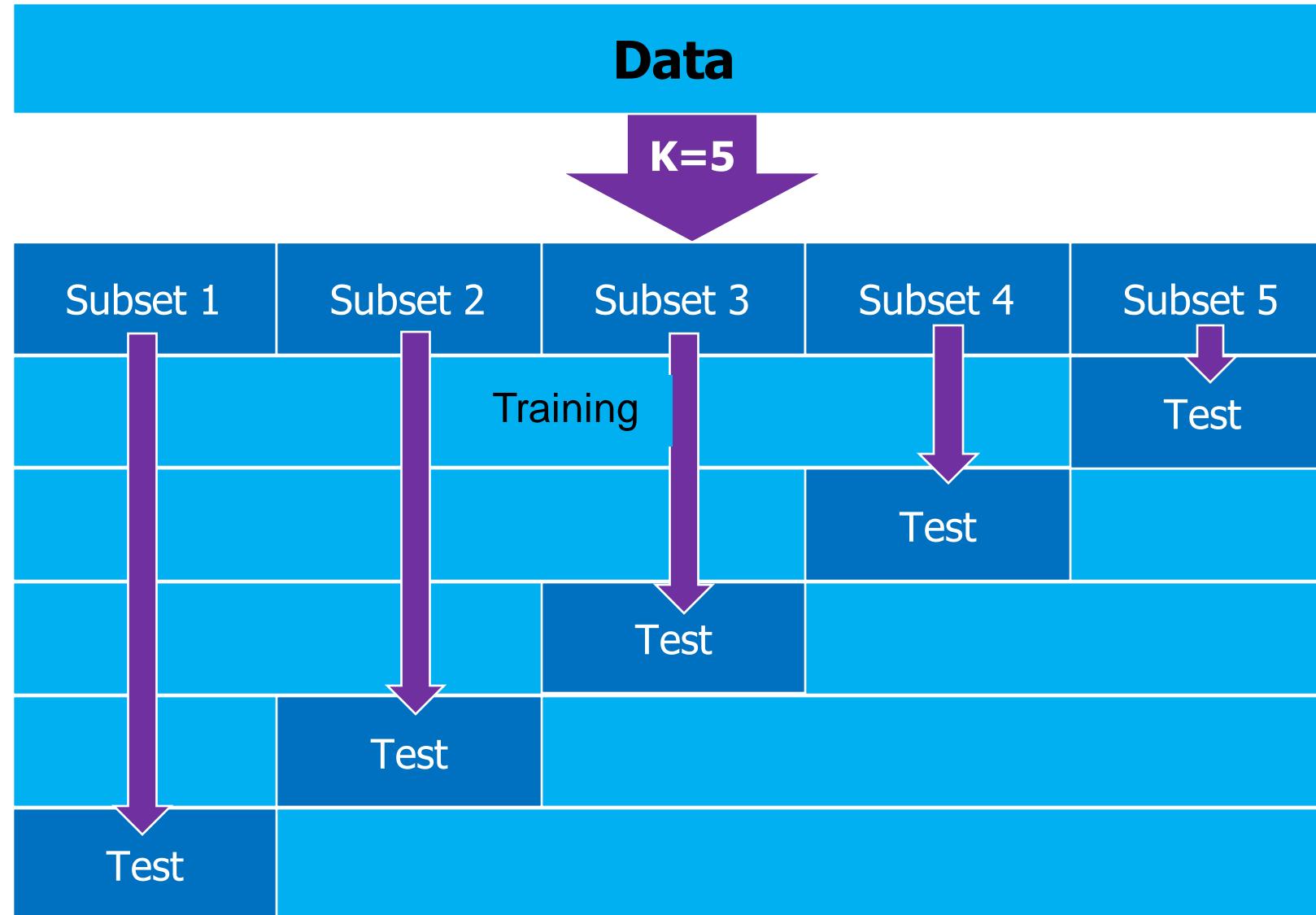
## LeaveOneOut

k sama dengan total observasi yang ada pada data yang mana setiap 1 data observasi digunakan sebagai data validasi atau data uji, sedangkan sisanya digunakan untuk data latih.

## Stratified K– Fold

Lipatan/fold dibuat dengan cara mempertahankan persentase sampel untuk setiap kelas.

# K-Fold Cross Validation



# K-Fold Cross Validation – With an Example

## Import Libraries

```
1 import pandas
2 from sklearn.model_selection import KFold
3 from sklearn.preprocessing import MinMaxScaler
4 from sklearn.model_selection import train_test_split # Import train_test_split function
5 from sklearn.svm import SVR
6 import numpy as np
7 import pandas as pd
```

## Load dataset and set Kfold parameters

```
1 col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
2 # load dataset
3 pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
4 kfold = KFold(3, True, 1)
```



04

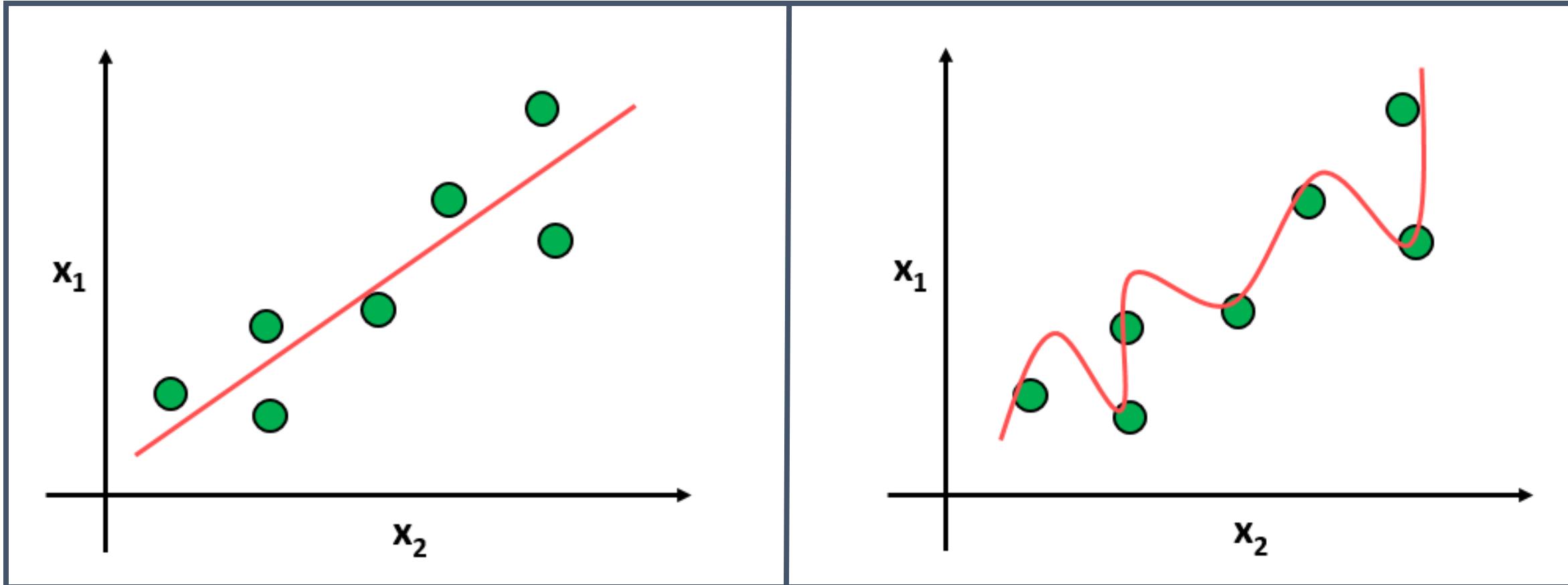
## REGULARIZATION

- LASSO
- RIDGE

# Overfitting & Generalisation

- Ketika kita melatih sebuah model dengan lebih banyak data, model tersebut kemungkinan akan mulai semakin akurat terhadap data latih, namun menjadi lebih buruk dalam menangani data uji yang akan diberikan kemudian
- Kondisi ini dikenal dengan istilah “over-fitting” yang pada akhirnya akan mempengaruhi performa model (generalization error).
- Coefficients yang bernilai besar berpotensi terjadi overfitting
- Teknik untuk menyelesaikan masalah large coefficients: Regularization

# Good fit v/s overfitting



# How to minimize overfitting?

- Untuk meminimalisir generalization error (overfitting) kita harus mengumpulkan data sampel sebanyak mungkin
- Gunakan random subset dari data sampel yang kita miliki untuk proses latih (training)
- Gunakan data sampel yang tersisa untuk melakukan validasi/pengujian seberapa baik performa suatu model ketika diberikan data yang tidak pernah digunakan selama proses latih

# Lasso (L1) Regression

- Least Absolute Shrinkage and Selection Operator (LASSO)
- Memiliki jumlah sample yang sangat besar ( $n$ ) sehubungan dengan jumlah dimensi ( $d$ ) akan meningkatkan kualitas dari suatu model
- Satu cara untuk mengurangi jumlah dimensi adalah dengan memanfaatkan data sampel yang memiliki pengaruh yang paling besar dan mengabaikan data sampel yang tidak berpengaruh (noise)
- L1 regularization bisa bekerja secara demikian dengan menerapkan penalty yang akan menghilangkan bobot dari dimensi yang berperilaku sebagai noise
- L1 regularization memanfaatkan sparse vector (vector dengan banyak nilai 0)

# Lasso (L1) Regression

- Tergantung dari seberapa besar regularization strength, weights tertentu pada akhirnya bisa bernilai zero, yang mana membuat LASSO juga yang mana membuat LASSO juga bermanfaat untuk teknik feature selection:

$$j(w_i) = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \|w_i\|$$

# Ridge (L2) Regression

- Cara lain untuk mengurangi kompleksitas dari suatu model dan mencegah overfitting terhadap outliers adalah dengan menggunakan L2 regression yang juga dikenal ridge regression.
- Di L2 Regularization kita melibatkan term tambahan untuk cost function yang memiliki efek penalty terhadap large weights yang pada akhirnya akan meminimalisir skew.

# Ridge (L2) Regression

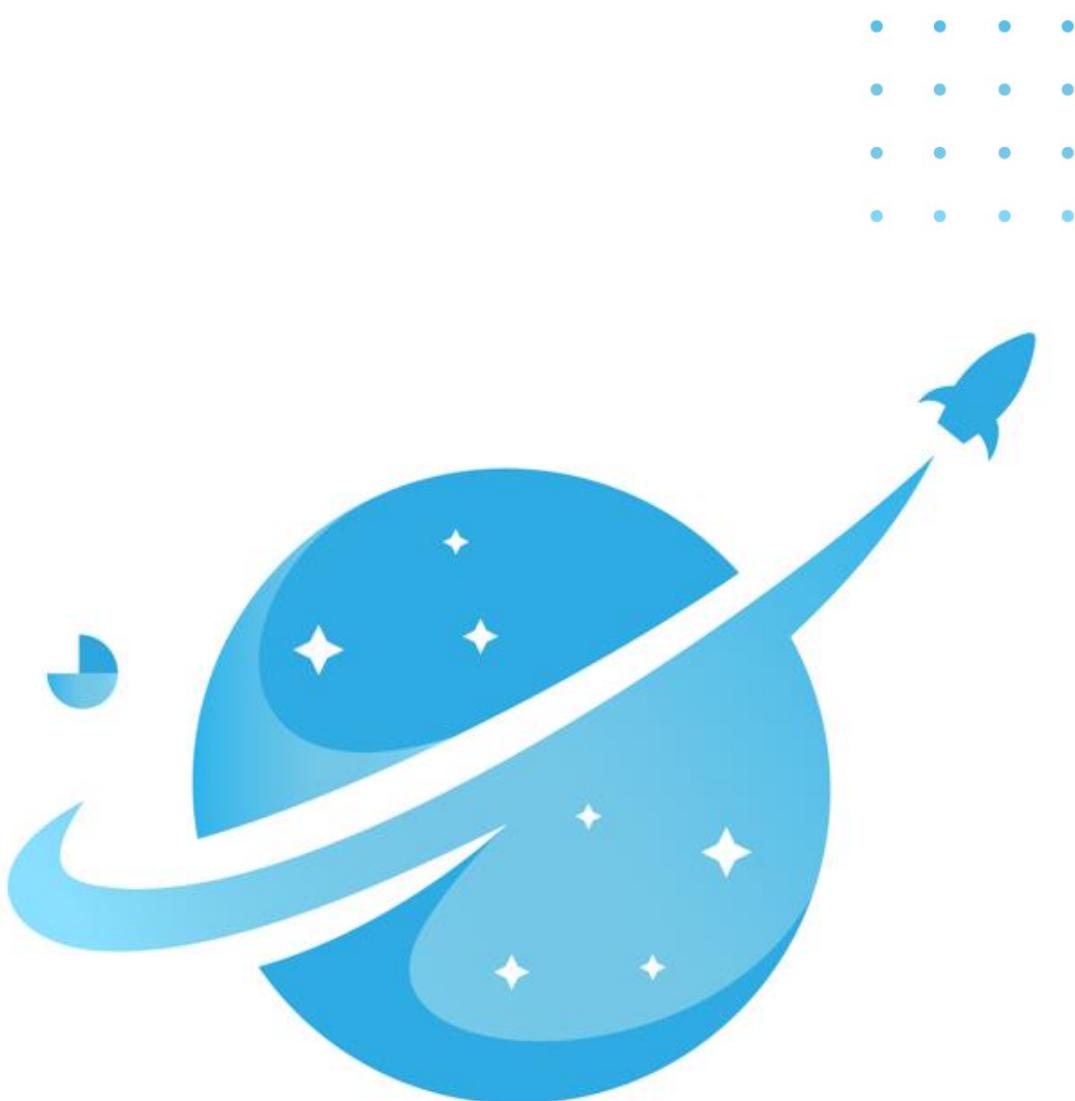
- Ridge regression adalah L2 model yang mendapatkan penalty dimana secara sederhana menambahkan squared sum of the weights pada least-squares cost function:

$$j(w_i) = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \|w_i\|^2$$

- Dengan meningkatkan nilai dari hyperparameter  $\lambda$  , kita bisa meningkatkan kekuatan dari regularization strength dan mengurangi weights dari suatu model

# L1 & L2 Regularisation (Elastic Net)

- L1 Regularisation meminimalisir dampak dari dimensi yang memiliki pengaruh yang lemah atau dianggap sebagai “noise”
- L2 Regularisation meminimalisir dampak dari outliers dari data latih yang ada
- L1 & L2 Regularisation bisa digunakan secara bersama-sama dan kombinasi ini dikenal dengan istilah Elastic Net regularisation
- Karena differential of the error function memuat sigmoid yang tidak memiliki inverse, kita tidak bisa menemukan w dan harus menggunakan gradient descent



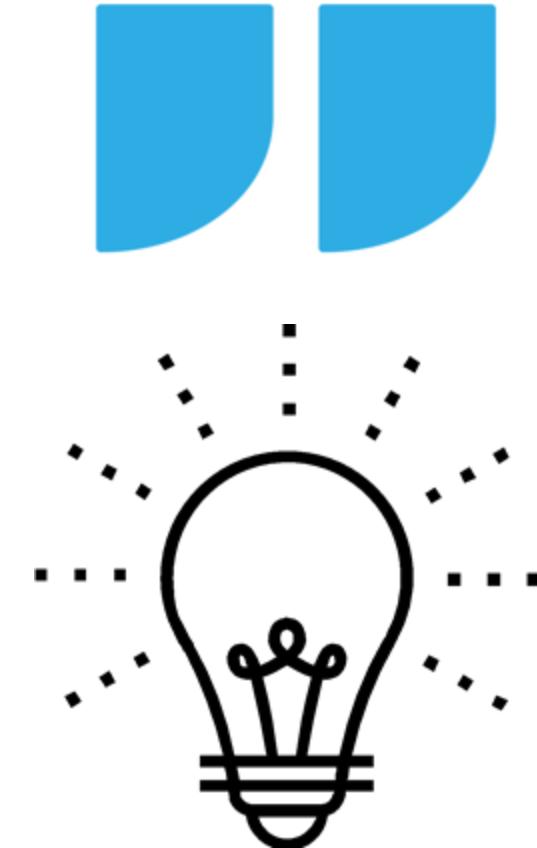
05

## CONCLUSION

- Summary

# Summary

- Regression technique dan use cases
- Gradient descent dengan linear regression
- Fungsi dari Cost function
- Melakukan training model untuk memahami gradient descent di Linear Regression
- Cross validation
- K-fold cross validation
- Regularisation pada regresi yang terdiri dari 2 tipe: Lasso and Ridge Regression



# Quiz

## Question

Di persoalan linear regression, kita menggunakan “R-squared” untuk mengukur goodness-of-fit. Ketika kita menambahkan feature di linear regression model tersebut dan melakukan retraining dengan menggunakan model yang sama, mana dari pernyataan berikut ini yang benar?

- A. Jika R Squared meningkat, maka variable tersebut memiliki pengaruh signifikan
- B. Jika R Squared berkurang, maka variable tersebut tidak memiliki pengaruh signifikan
- C. R squared tidak bisa memberi tahu tentang pengaruh variable
- D. Tidak ada yang benar

# Quiz

## Question

Di persoalan linear regression, kita menggunakan “R-squared” untuk mengukur goodness-of-fit. Ketika kita menambahkan feature di linear regression model tersebut dan melakukan retraining dengan menggunakan model yang sama, mana dari pernyataan berikut ini yang benar?

- A. Jika R Squared meningkat, maka variable tersebut memiliki pengaruh signifikan
- B. Jika R Squared berkurang, maka variable tersebut tidak memiliki pengaruh signifikan
- C. R squared tidak bisa memberi tahu tentang pengaruh variable
- D. Tidak ada yang benar

**Jawaban: C**



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

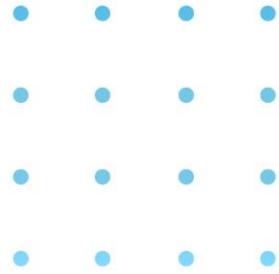
Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)



# AI Mastery Course



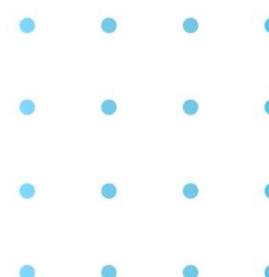
## Module 5

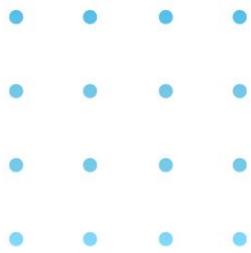
Supervised machine learning  
and predictive modelling



### Section

#### Decision tree

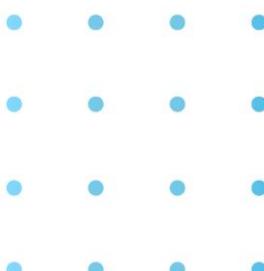




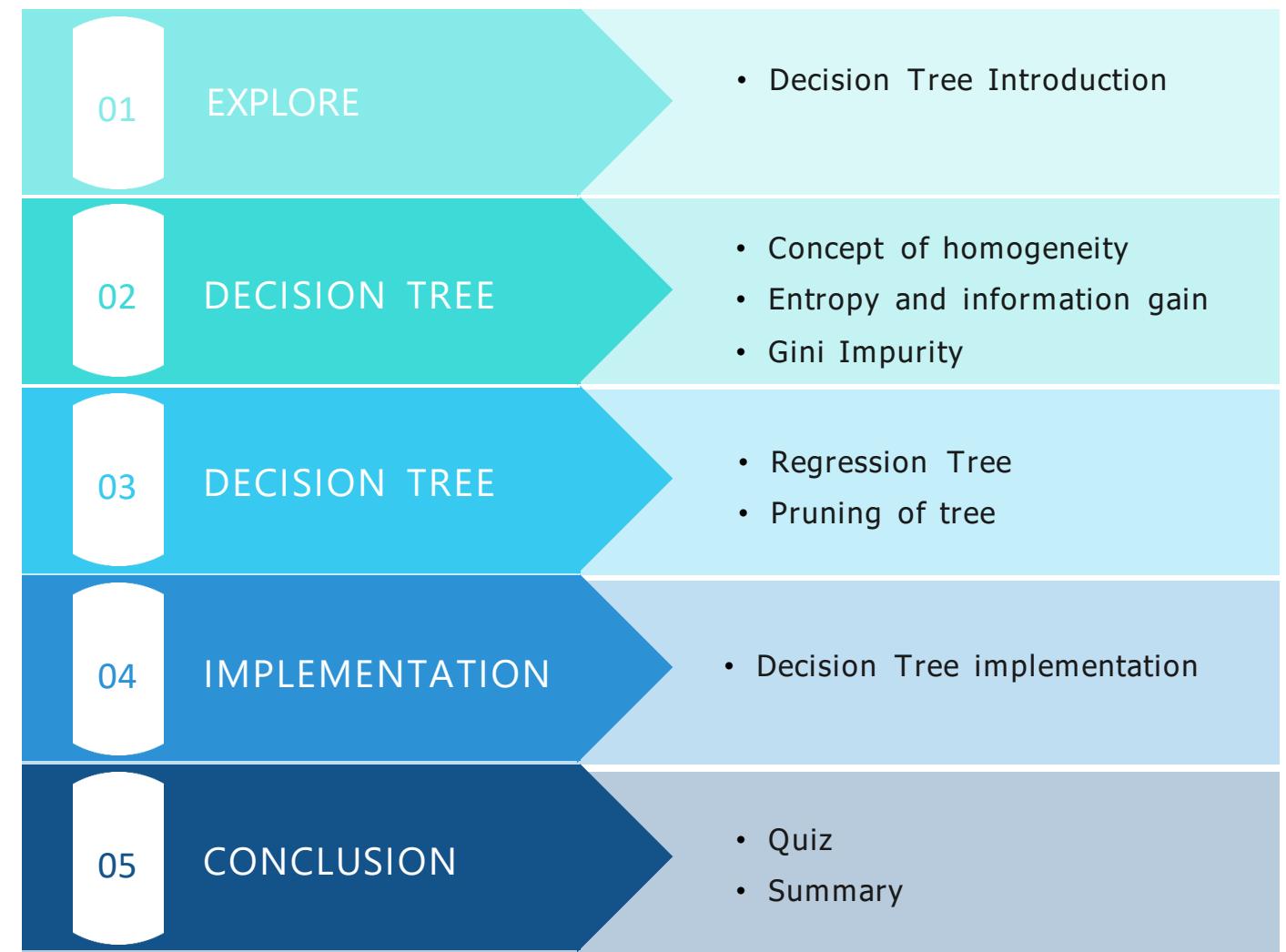
# Learning Objectives

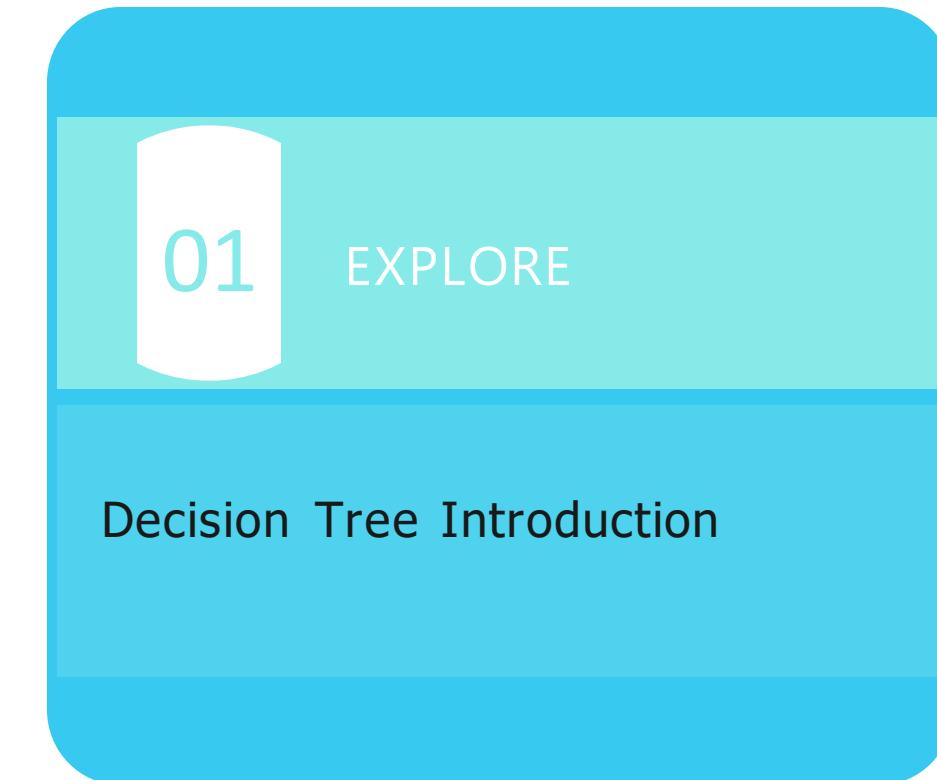
Di akhir modul ini, Anda akan dapat:

- Mendefinisikan konsep-konsep berikut
  - Decision tree
  - Homogeneity
  - Gini index
  - Entropy dan Information Gain
  - Truncation and Pruning
- Memahami implementasi Decision Tree untuk persoalan regresi
- Memahami perbedaan dari beberapa pendekatan untuk mengimplementasikan algoritma Decision Tree



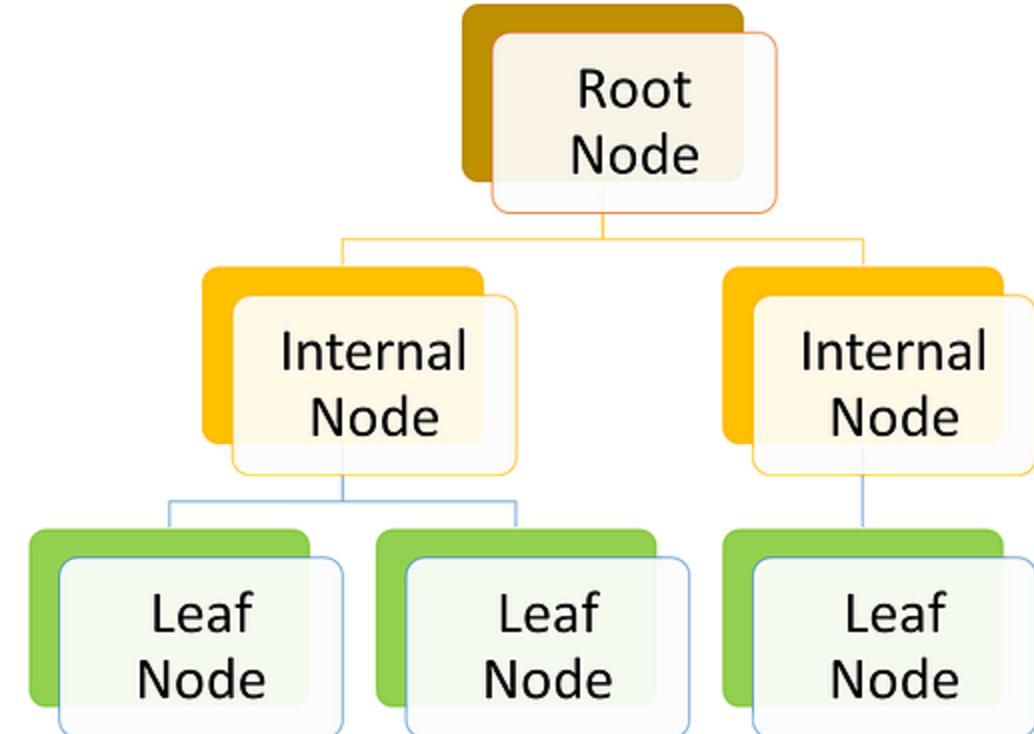
# Agenda





# Decision Tree

- Decision tree merupakan algoritma supervised learning yang bekerja untuk data input dan output yang bersifat categorical (classification tree) dan continuous (regression tree)
- Berupa representasi grafis yang memudahkan dalam melakukan interpretasi dan membantu dalam membuat keputusan (decision making)
- Decision tree juga tampak seperti flow chart yang memiliki struktur if-else condition



# Decision Tree

- **Decision tree membangun model klasifikasi dalam bentuk seperti struktur sebuah pohon**
- **Algoritma Decision Tree membagi dataset ke beberapa bagian kecil**

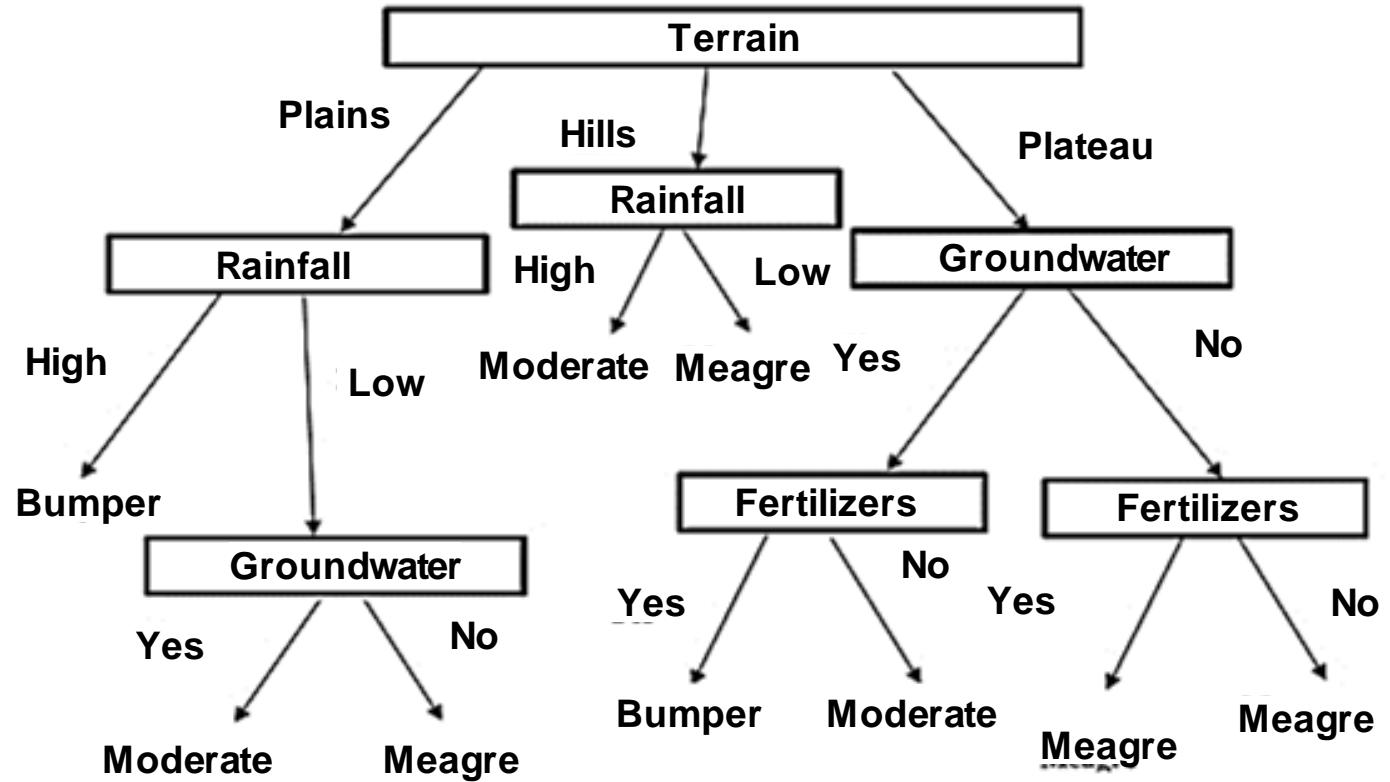
Sebuah inner node merepresentasikan sebuah attribute

Sebuah edge merepresentasikan hubungan attribute tersebut dengan father node

Sebuah leaf node merepresentasikan salah satu kelas

# Decision Tree Illustration using Sample Dataset

Place	Rainfall	Terrain	Fertilisers	Groundwater	Harvest
P1	High	Plains	Yes	Yes	Bumper
P2	Low	Hills	No	Yes	Meagre
P3	Low	Plateau	No	Yes	Moderate
P4	High	Plateau	No	Yes	Moderate
P5	High	Plains	Yes	No	Bumper
P6	Low	Hills	No	No	Meagre
P7	Low	Plateau	No	No	Meagre
P8	Mild	Plateau	No	No	Meagre
P9	High	Hills	Yes	Yes	Moderate
P10	Mild	Plateau	Yes	Yes	Bumper
P11	High	Plateau	Yes	No	Bumper
P12	Mild	Plateau	Yes	No	Moderate
P13	High	Hills	Yes	No	Moderate
P14	Low	Plains	Yes	Yes	Moderate
P15	Mild	Plains	Yes	No	Moderate
P16	Low	Plains	No	No	Meagre
P17	Low	Hills	Yes	No	Meagre
P18	Mild	Plateau	No	No	Meagre
P19	High	Plains	No	Yes	Moderate
P20	Mild	Hills	Yes	Yes	Moderate



Rainfall Dataset

A representational decision tree based on the above



02 DECISION TREE

Concept of homogeneity  
Entropy and information gain  
Gini Impurity



# Concept of Homogeneity



# Entropy

$$Entropy(S) = \sum -p_i \log_2 p_i$$

- $P_i$  merupakan proporsi (dari jumlah observasi) dari  $i^{th}$  kategori tertentu dari variabel target.
- Entropy bernilai dari 0 to  $\log_2 c$ , dimana c menjadi jumlah dari kategori yang ada di variabel target.
- Entropy akan bernilai 0 jika observasi yang ada bersifat homogeneous secara sempurna.
- Dan entropy akan mencapai  $\log_2 c$  ketika observasi yang ada bersifat heterogeneous secara sempurna.
- Entropy mencapai nilai maksimumnya ( $\log_2 c$ ) saat semua  $p_i$ 's nya sama.
- Entropy mencapai nilai minimumnya (0) saat salah satu  $p_i$  nya bernilai 1 dan yang lainnya bernilai 0.

# Information Gain

$$information\ Gain(S, V) = S_v - \sum(V_c / V).Entropy(V_c)$$

- $S_v$  adalah total entropy dari node variable  $V$
- $c$  merupakan categories pada node variable
- $V_c$  merupakan jumlah dari total observasi yang ada dengan kategori  $c$  dari node variable
- $V$  merupakan jumlah dari total observasi yang ada
- $Entropy(V_c)$  adalah entropy pada sistem yang memiliki observasi dengan kategori  $c$  dari node variable
- Simbol Summation memuat semua kategori yang ada dari variabel tersebut

# Algorithms to Create a Decision Tree

## ID3 Algorithm

- Hitung nilai initial entropy dari sistem berdasarkan target variable.
- Hitung nilai information gains untuk masing-masing kandidat variable yang akan menjadi node lalu pilih variable yang memiliki nilai information gain paling besar sebagai decision node
- Ulangi langkah 2 untuk masing-masing branch (value) dari node (variable) yang terpilih di langkah 2 dan setiap node baru yang teridentifikasi disebut sebagai leaf nodes.
- Cek apakah leaf node mengklasifikasi seluruh data secara sempurna atau tidak. Jika tidak, ulangi langkah 2, jika ya, maka algoritma berakhir.

# Gini Index

Gini Index mengukur nilai impurity dari sebuah node. Gini index memuat rentang nilai dari 0 hingga  $(1 - 1/n)$  dimana n adalah jumlah kategori dari suatu dependent variable.

$$Gini = 1 - \sum_{i=1}^C (p_i)^2$$

## Important Points :

1. Zero Gini index menunjukkan klasifikasi yang sempurna
2.  $(1 - (1 / \text{No. of classes}))$  menunjukkan klasifikasi dengan kualitas yang paling buruk
3. Kita ingin variable yang di-split memiliki nilai Gini Index yang rendah

# Gini Index

## Gini for each category

$$Gini_{C1} = (0.2)^2 + (0.8)^2 = 0.68$$

$$Gini_{C2} = (0.6)^2 + (0.4)^2 = 0.52$$

## Gini index for a variable

$$Gini\ index = (10 / 20) * 0.68 + (10 / 20) * 0.52 = 0.60$$

# Which is better - Entropy or Gini?

Kedua teknik splitting tersebut memiliki performa yang relatif sama di beberapa kasus. Namun Gini cenderung lebih cepat dari Entropy karena tidak perlu melakukan proses kalkulasi log.



03 DECISION TREE

Regression Tree  
Pruning of tree



# Regression Tree

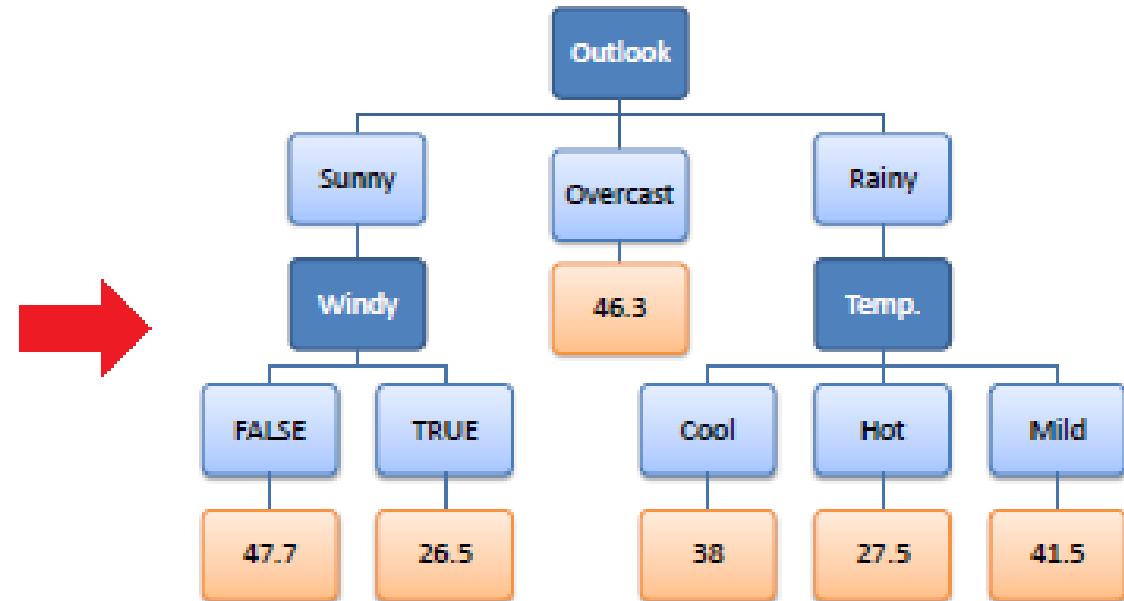
Nilai impurity dari sebuah node diukur dari Least-Squared Deviation (LSD), yang mana secara sederhana memanfaatkan nilai variance dari node.

$$\text{Variance} = \frac{\sum(X - \bar{X})^2}{n}$$

X-bar merupakan rata-rata dari nilai, X merupakan nilai aktual dan n merupakan jumlah nilai.

# Regression Tree

Predictors				Target
Outlook	Temp.	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	62
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	36
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	62
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



# Steps to calculate Variance:

1. Menghitung nilai variance dari setiap node.
2. Menghitung nilai variance untuk setiap split sebagai weighted average dari setiap variance tersebut.

# Decision Trees

Set a metric that evaluates impurity of a split of data. then minimize the metric on each node.

## Classification

### Gini impurity(CART)

$$\sum_{k=1}^K p_k(1 - p_k)$$

### Entropy (C4.5)

$$-\sum_{k=1}^K p_k \ln p_k$$

$p_k$ : probability of an item with label  $k$

$K$  : number of class

## Regression

### Variance

$$\frac{|S_L|}{|S|}Var(S_L) + \frac{|S_R|}{|S|}Var(S_R)$$

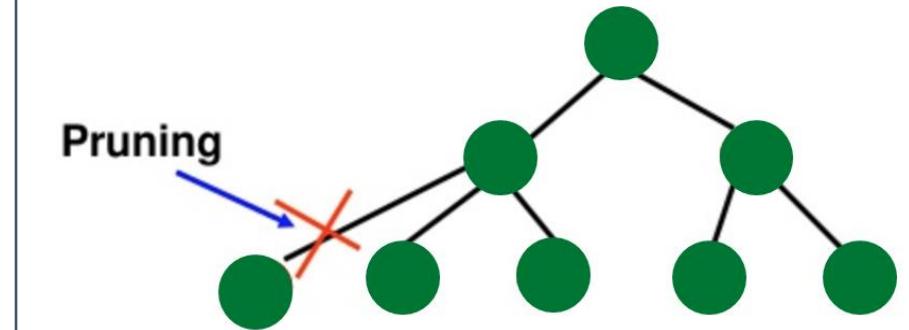
$SD(S)$ : standard variance of set  $S$

$S_L, S_R$  : left and right split of a node

# Truncation and Pruning

Secara umum, membangun decision tree melibatkan 2 langkah:

- Tree construction: Secara rekursif melakukan proses split pada pohon sesuai dengan atribut terpilih (conditions).
- Tree Pruning: Mengidentifikasi dan membuang branch yang tidak relevan (berpotensi sebagai outliers) – meningkatkan performa klasifikasi



# Decision Tree

Dua teknik untuk menerapkan Stopping Criteria

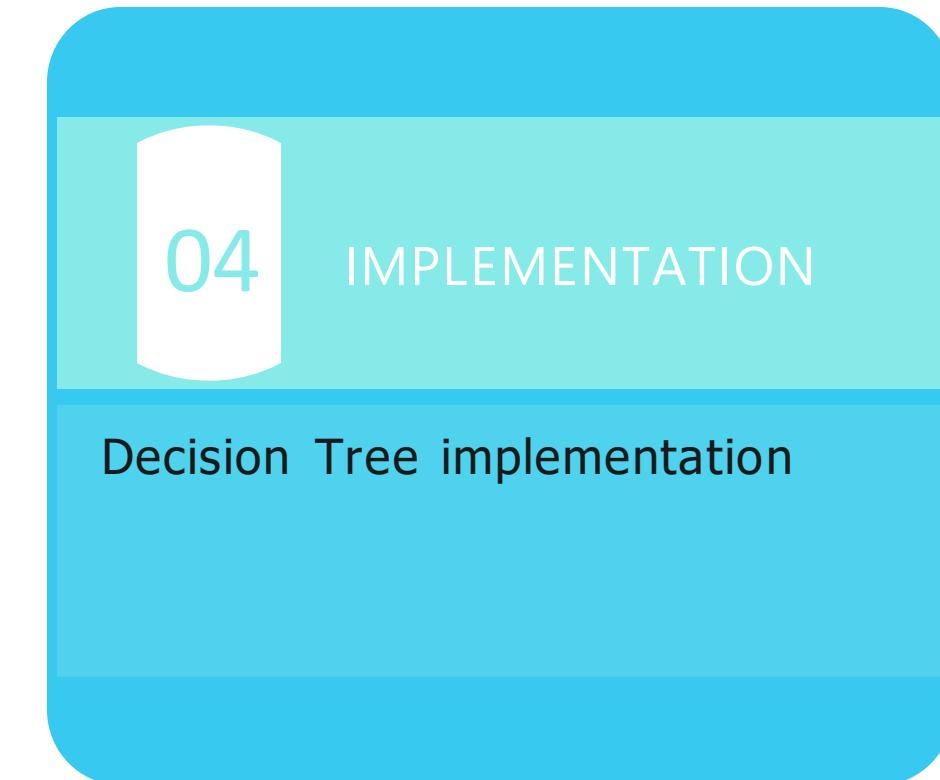
- Maximum depth
- Minimum leaf nodes

Menemukan threshold yang baik untuk data numerik

- observed point of data
- the point that class labels are changed
- percentile of data

# Advantages of Decision Tree

- Decision tree relatif mudah untuk diinterpretasi
- Decision Tree tetap bekerja bahkan jika tidak ada relasi yang linear antara variable (Tidak memerlukan asumsi linear)
- Decision Tree tidak sensitif terhadap outliers.



04 IMPLEMENTATION

Decision Tree implementation



# Decision Tree Working Methodology

## Import Libraries

```
1 # Load libraries
2 import pandas as pd
3 from sklearn.tree import DecisionTreeClassifier # Import Decision Tree Classifier
4 from sklearn.model_selection import train_test_split # Import train_test_split function
5 from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation
```

## Loading Data

```
1 col_names = ['pregnant', 'glucose', 'bp', 'skin', 'insulin', 'bmi', 'pedigree', 'age', 'label']
2 # load dataset
3 pima = pd.read_csv("pima-indians-diabetes.csv", header=None, names=col_names)
```

```
1 pima.head(5)
```

	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

# Decision Tree Working Methodology (Contd.)

## Feature Selection

```

1 #split dataset in features and target variable
2 feature_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']
3 X = pima[feature_cols] # Features
4 y = pima.label # Target variable
  
```

## Feature Selection

```

1 # Split dataset into training set and test set
2 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training and 30% te
  
```

## Feature Selection

```

1 # Create Decision Tree classifier object
2 clf = DecisionTreeClassifier()
3
4 # Train Decision Tree Classifier
5 clf = clf.fit(X_train,y_train)
6
7 #Predict the response for test dataset
8 y_pred = clf.predict(X_test)
  
```

```

1 # Model Accuracy, how often is the classifier correct?
2 print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

('Accuracy:', 0.6796536796536796)
  
```

# Decision Tree Tuning Parameters

## Feature Selection

```
1 # Create Decision Tree classifier object
2 clf = DecisionTreeClassifier(criterion="entropy", max_depth=3)
3
4 # Train Decision Tree Classifier
5 clf = clf.fit(X_train,y_train)
6
7 #Predict the response for test dataset
8 y_pred = clf.predict(X_test)
9
10 # Model Accuracy, how often is the classifier correct?
11 print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

('Accuracy:', 0.7705627705627706)
```



05

## CONCLUSION

- Quiz
- Summary

# Quiz

1. Decision trees bisa digunakan untuk persoalan klasifikasi.



- A. True
- B. False

# Quiz

1. Decision trees bisa digunakan untuk persoalan klasifikasi.



- A. True
- B. False

**Answer: A**

# Summary

- Decision tree berbentuk pohon sebagai representasi grafis dari kemungkinan solusi untuk mengambil keputusan berdasarkan suatu kondisi tertentu
- Decision tree memiliki root nodes, decision nodes dan leaf nodes.
- Entropy memuat jumlah informasi yang diperlukan untuk secara akurat mendeskripsikan beberapa sample.
- Gini Index mengukur nilai impurity pada node.
- Information gain digunakan sebagai ukuran untuk menentukan feature dalam proses splitting.
- Decision Tree tetap bekerja bahkan jika variabel yang ada tidak memiliki relasi yang linear.





# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

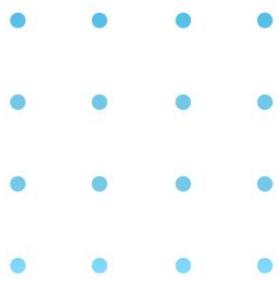
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# AI Mastery Course



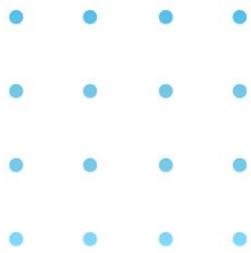
## Module 8

### Deep learning basics

## Section

### Deep learning overview

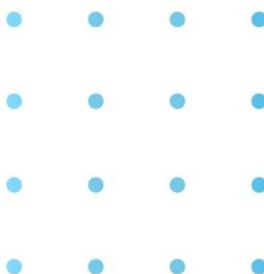




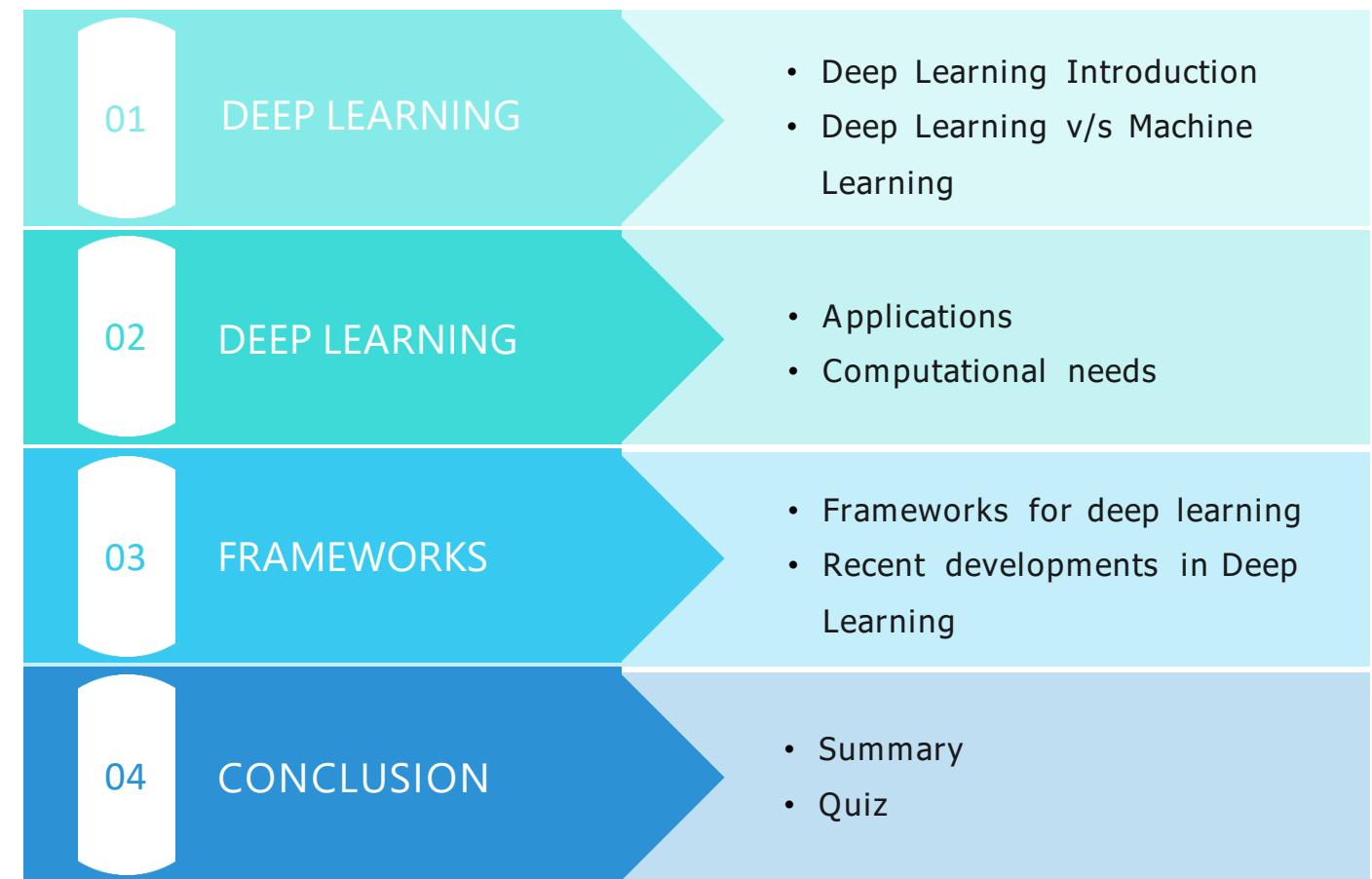
# Learning Objectives

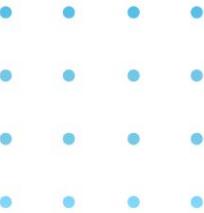
Pada akhir modul ini, Anda akan dapat:

- Memahami konsep Deep Learning dan penerapannya.
- Ketahui apa perbedaan antara deep learning dan machine learning.
- Kenali berbagai opsi komputasi - BERBASIS CPU, GPU, dan cloud.
- Kenali kerangka kerja populer untuk mendesain deep learning solutions



# Agenda





## 01

## DEEP LEARNING

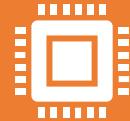
- Deep Learning Introduction
- Deep Learning v/s Machine Learning



# Deep Learning

Deep Learning adalah bagian dari machine learning bidang pembelajaran representasi data.  
Efektif digunakan di pembelajaran pola atau pattern.

# Apa itu Deep Learning?



Deep learning adalah tipe algoritma machine learning di mana model belajar untuk melakukan tugas klasifikasi langsung dari gambar, teks, atau suara.



Deep learning biasanya diimplementasikan dengan menggunakan neural network architecture.

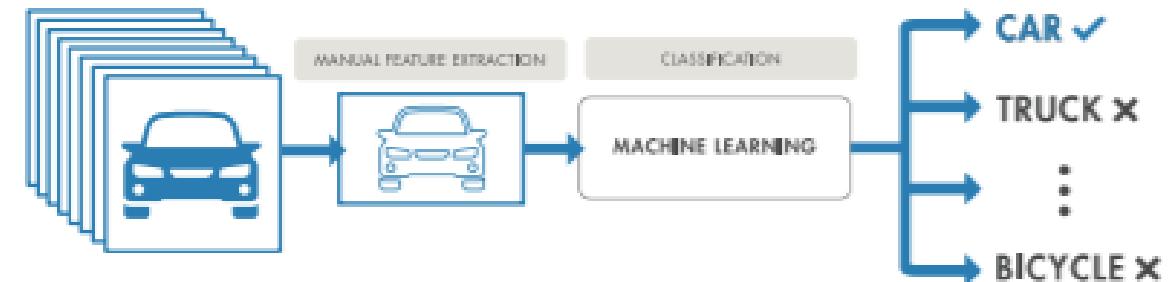


Istilah “deep” Mengacu pada jumlah layers di network—Semakin banyak layers, semakin dalam network. Traditional neural networks hanya mengandung 2 atau 3 layers, sementara deep networks Dapat memiliki ratusan layer.

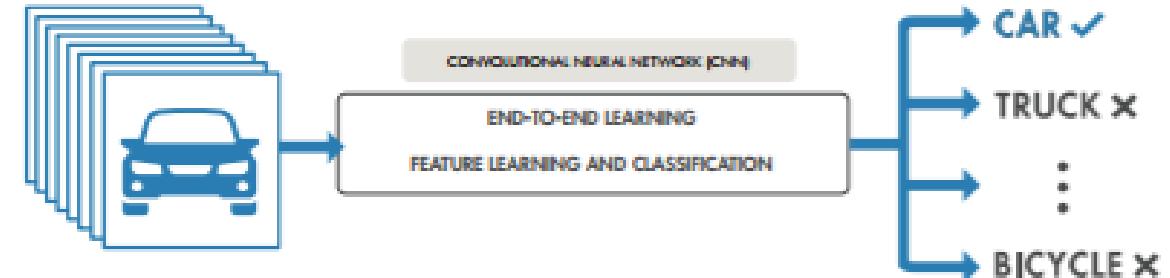
# Perbedaan Antara Deep Learning dan Machine Learning?

- Deep learning Adalah subtipe dari machine learning. Dengan machine learning, Anda mengekstrak secara manual fitur gambar yang relevan.
- Dengan deep learning, Anda menggunakan gambar mentah/image langsung ke deep neural network yang akan mempelajari fitur secara otomatis.

TRADITIONAL MACHINE LEARNING

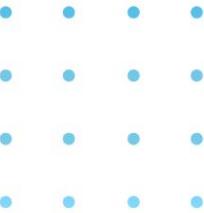


DEEP LEARNING



# Perbedaan Antara Deep Learning dan Machine Learning?

Machine Learning	Deep Learning
+ Hasil yang baik dengan dataset kecil	- Membutuhkan kumpulan data yang sangat besar
+ Cepat untuk melatih model	- Computationally Intensive
- Perlu mencoba berbagai fitur dan pengklasifikasi untuk mencapai hasil terbaik.	+ Mempelajari fitur dan pengklasifikasi secara otomatis
- Akurasi tinggi	+ Akurasi tidak terbatas



02

## DEEP LEARNING

- Applications
- Computational needs

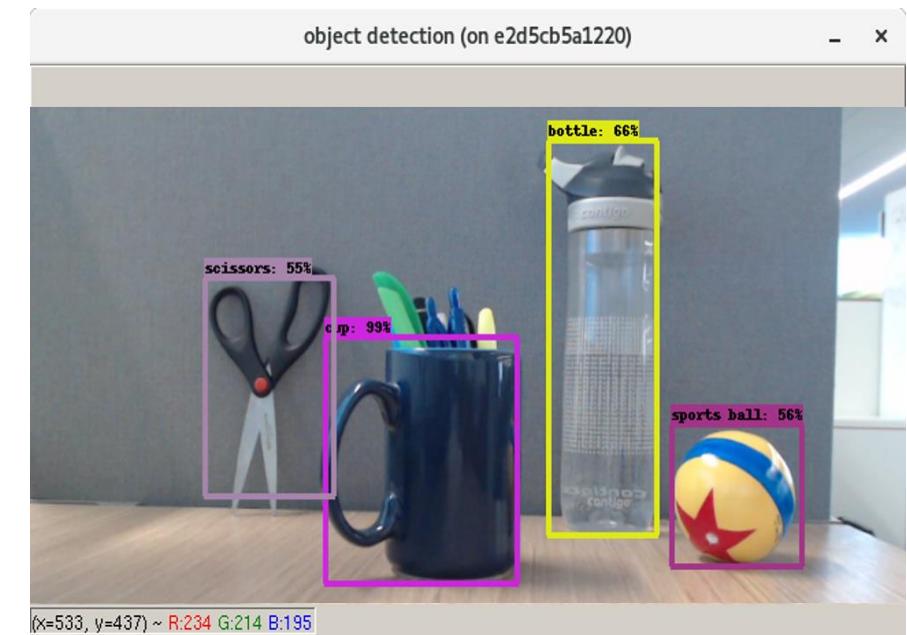


# Deep Learning Applications

Deep learning sangat cocok untuk aplikasi identifikasi seperti face recognition, text translation, voice recognition, dan driver assistance systems, Termasuk, lane classification dan traffic sign recognition.

Berikut adalah beberapa contoh dari deep learning di tempat kerja:

- Sebuah kendaraan self-driving melambat saat mendekati penyeberangan pejalan kaki.
- ATM menolak uang kertas palsu.
- Aplikasi smartphone memberikan penunjuk jalan.



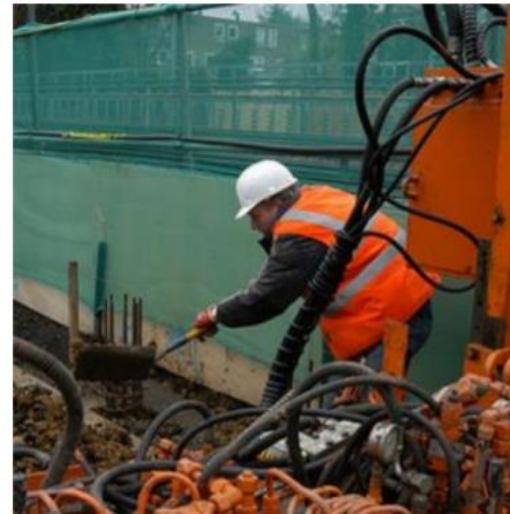
# Image Captioning Menggunakan Deep Learning



"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."



"construction worker in orange safety vest is working on road."



"man in black shirt is playing guitar."

# Apa yang tidak bisa kita lakukan dengan Deep Learning?

- Mirrors
- Sparse information
- 3D Structure
- Physics
- Apa yang ada di pikiran orang-orang?
- Apa yang terjadi selanjutnya?
- Humor



# Computational Resources bagi Deep Learning

Melatih model pembelajaran mendalam dapat memakan waktu berjam-jam, berhari-hari, atau berminggu-minggu, tergantung pada ukuran data dan jumlah daya pemrosesan yang Anda miliki.

Saat ini, ada tiga opsi perhitungan: CPU-based, GPU-based dan cloud-based.

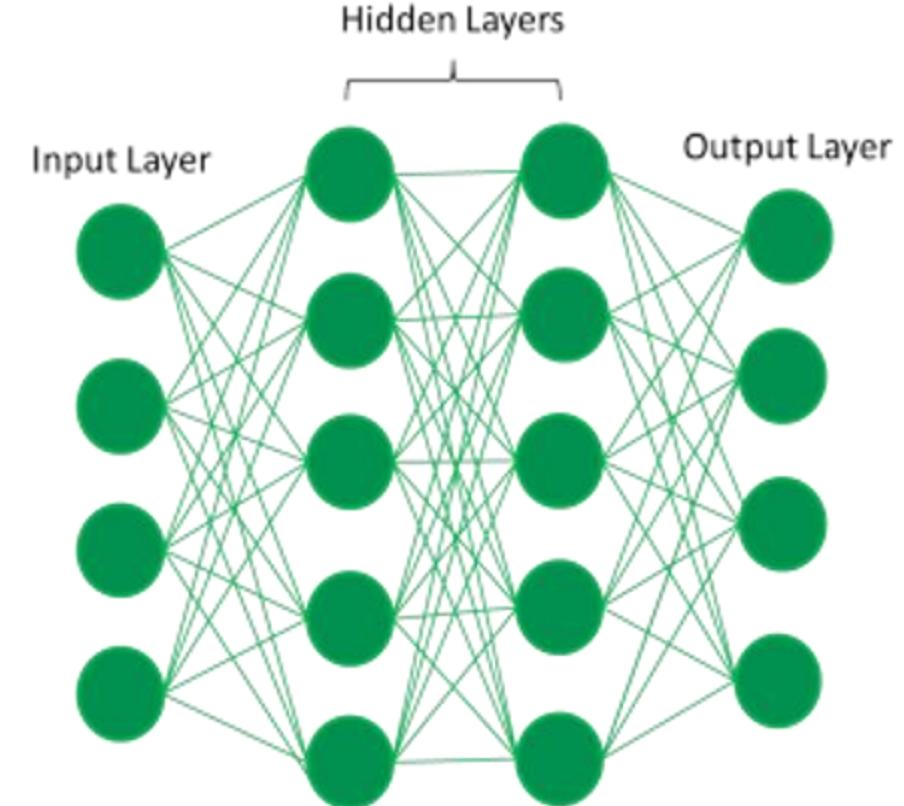
- CPU-based computation Adalah pilihan yang paling sederhana dan paling mudah tersedia. Menggunakan CPU-based computation hanya untuk contoh sederhana menggunakan jaringan yang telah dilatih sebelumnya.
- Menggunakan GPU mengurangi network training waktu dari hari ke jam.
- Cloud-based GPU computation berarti Anda tidak perlu membeli dan mengatur perangkat keras sendiri.

# Understanding deep learning

- Semua deep learning algorithms adalah modifications untuk neural networks.
- Contoh yang umum bisa menjadi deep neural network
- A deep neural network Menggabungkan multiple nonlinear processing layers, menggunakan elemen sederhana yang beroperasi secara paralel dan terinspirasi oleh sistem saraf biologis.
- Ini terdiri dari input layer, beberapa hidden layers, dan output layer. The layers saling berhubungan melalui node, atau neurons, Dengan masing-masing hidden layer Menggunakan output dari previous layer sebagai masukannya.

# Deep Learning Network Structure

- Ini adalah sangat dasar deep learning neural network Memiliki 1 input layer, 2 hidden layers dan 1 output layer.
- Jumlah Neurons dalam input dan output layer Sepenuhnya tergantung pada pernyataan masalah Anda.



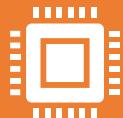


## 03

### FRAMEWORK

- Frameworks for deep learning
- Recent developments in Deep Learning

# Open Frameworks of Pre-Trained Models: Tensorflow



Google TensorFlow, adalah open-source software framework Untuk membangun dan menggunakan machine learning neural networks, sangat mudah diatur dan diperpanjang.



Ini adalah yang paling populer deep learning framework, Dengan jumlah terbesar dari GitHub stars dan Persentase tertinggi kedua dari open-source repositories.

# Open Frameworks of Pre-Trained Models: pyTorch

PyTorch Adalah open source machine learning library Berdasarkan Torch library, digunakan untuk aplikasi seperti computer vision dan natural language processing, Terutama dikembangkan oleh Facebook's AI Research lab.

Dengan TorchScript, PyTorch memberikan kemudahan penggunaan dan fleksibilitas dalam mode bersemangat, sementara transisi mulus ke mode grafik untuk speed, optimization, dan functionality in C++ runtime environments.

PyTorch Didukung dengan baik di cloud platforms, memberikan pengembangan tanpa gesekan dan penskalaan yang mudah melalui gambar bawaan, pelatihan skala besar pada GPU, kemampuan untuk menjalankan model dalam lingkungan skala produksi, Dan banyak lagi.

# Open Frameworks of Pre-Trained Models: Caffe2



Caffe2 Adalah powerful open-source library Yang membuatnya mudah untuk membuat deep learning models Menggunakan PyTorch framework.



Kita dapat dengan cepat mengembangkan scalable models dan singkirkan perhitungan khas yang terlibat dalam model tradisional.



Itulah sebabnya dengan Caffe2, kita dapat memanfaatkan mesin kita sebaik-baiknya dan mendapatkan efisiensi maksimum dari mereka.

# Kemajuan terbaru dari Deep Learning

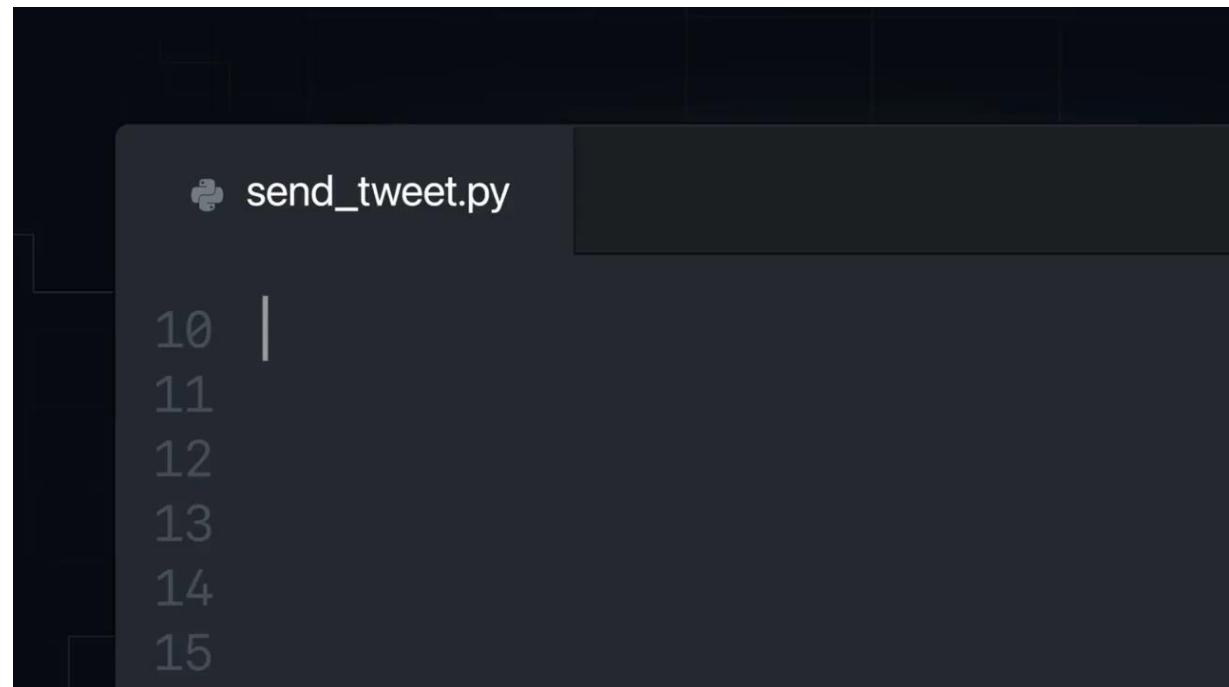
**March 2021:** Meta AI released SEER (SELf-supERvised), satu miliar parameter yang diawasi sendiri dengan computer vision model yang dapat belajar dari setiap kelompok gambar acak di internet. Meta Mengatakan bahwa SEER tidak memerlukan kurasi dan pelabelan yang cermat yang dibutuhkan sebagian besar model training computer vision. SEER mengungguli model yang diawasi canggih di hilir seperti deteksi objek, segmentation, dan image classification, Antara lain.

<https://ai.facebook.com/blog/seer-the-start-of-a-more-powerful-flexible-and-accessible-era-for-computer-vision/>

Can we learn with Self-Supervised Learning on random images

# Kemajuan terbaru dari Deep Learning

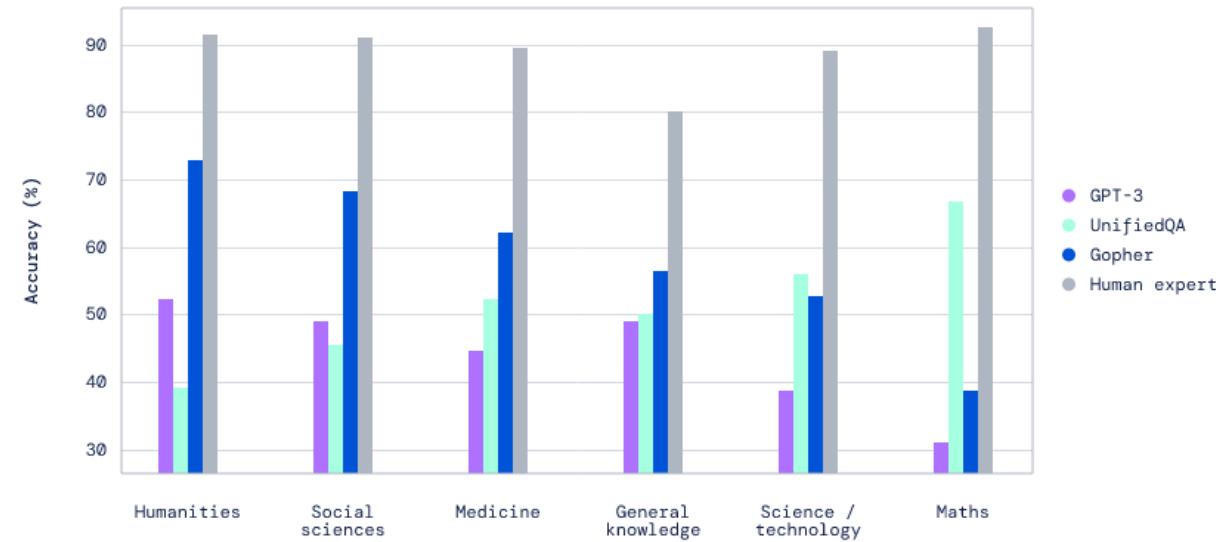
**JUN 2021:** OpenAI dan Microsoft's GitHub Copilot Adalah sebuah AI pair programmer untuk writing better code. Ini memiliki kemampuan untuk bekerja dengan berbagai bahasa seperti Python, JavaScript, TypeScript, Ruby, Java, dan lain-lain. Ini berfungsi sebagai ekstensi di desktop atau di cloud di GitHub Codespaces. Programmer dapat melihat saran alternatif, edit kode yang disarankan secara manual, dan pilih apa yang harus diterima dan tolak dengan bantuannya.



# Kemajuan terbaru dari Deep Learning

**Dec 2021:** DeepMind Memperkenalkan 280 miliar model bahasa transformator parameter yang disebut Gopher. DeepMind's Penelitian melanjutkan dengan mengatakan bahwa Gopher hampir membagi dua kesenjangan akurasi dari GPT-3 ke human expert performance dan melebihi ekspektasi peramal.

<https://deepmind.com/blog/article/language-modelling-at-scale>





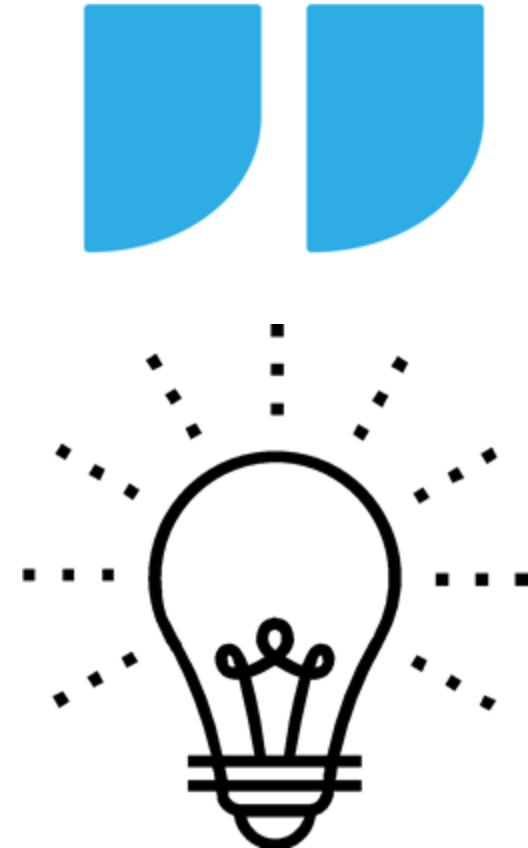
04

## CONCLUSION

- Summary
- Quiz

# Summary

- Deep learning berarti mesin belajar klasifikasi dari teks atau gambar dengan sendirinya.
- Ada 3 sumber daya komputasi untuk pembelajaran mendalam - berbasis CPU, berbasis GPU, dan berbasis cloud.
- Deep learning diperlukan karena ukuran data telah meningkat dan Kita perlu menyegmentasikan data dari banyak resources seperti gambar, teks, dll.



# Quiz

## Question

Yang mana dari berikut ini adalah / adalah aplikasi dari deep learning?

- A. Visual Question answering
- B. Video summarization
- C. Image processing
- D. All of the above



# Quiz

## Question

Yang mana dari berikut ini adalah / adalah aplikasi dari deep learning?

- A. Visual Question answering
- B. Video summarization
- C. Image processing
- D. All of the above



**Answer: D**



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

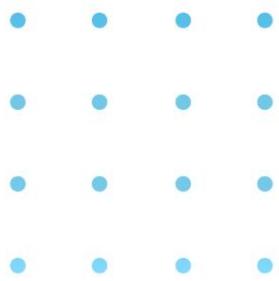
## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)

# AI Mastery Course

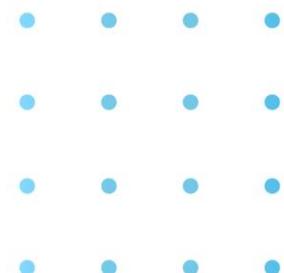


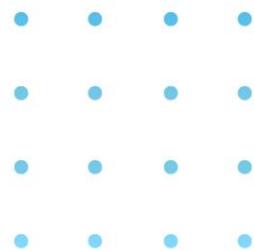
## Module 8

### Deep learning basics

#### Section

Artificial neural networks:  
backpropagation

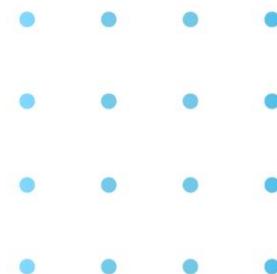




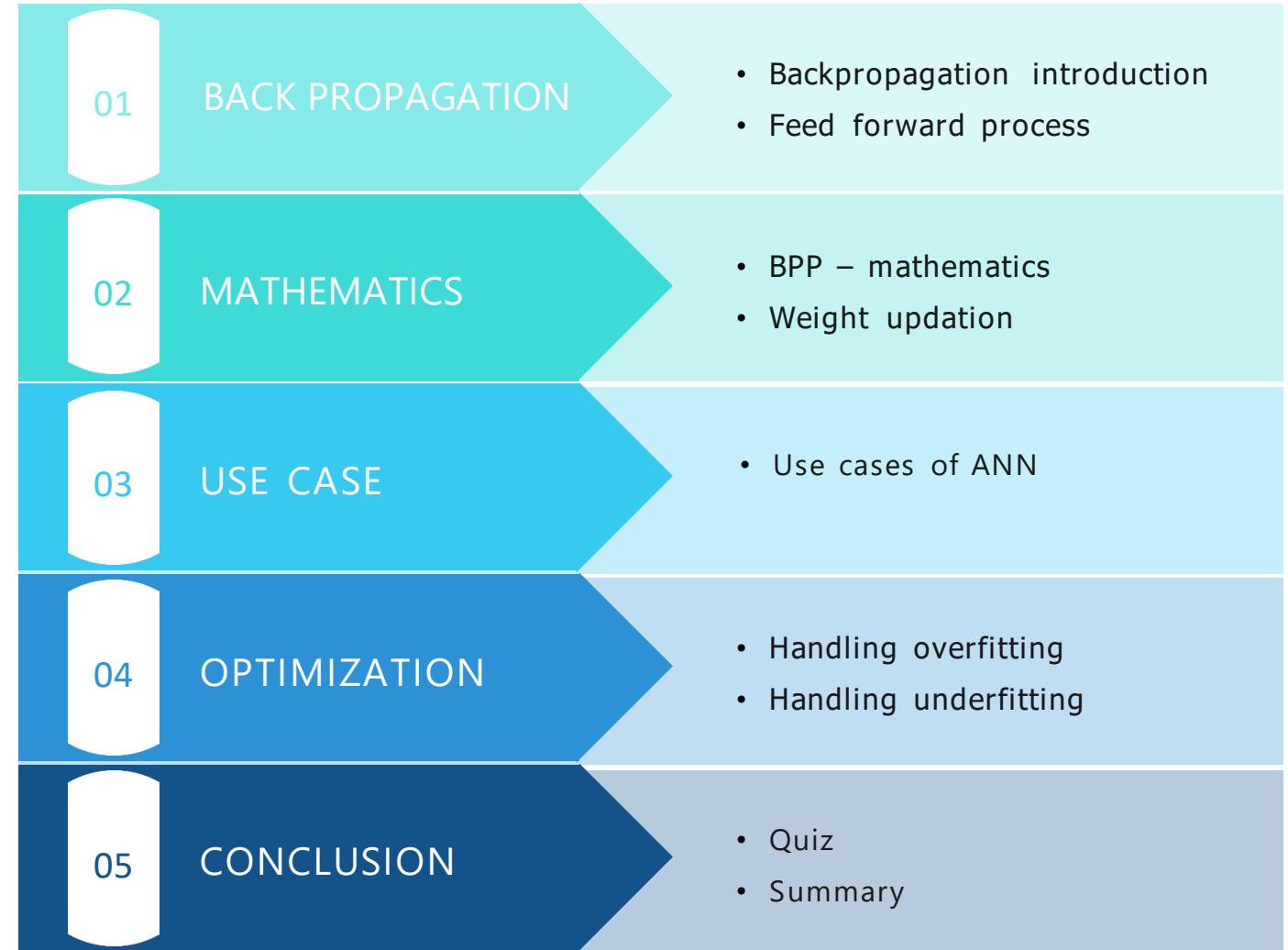
# Learning Objectives

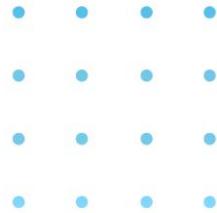
At the end of this module, you will be able to:

- Understand the concept of back propagation and mathematics behind back propagation algorithm
- Learn the use cases of ANN
- Learn how to handle overfitting and underfitting in context to neural networks



# Agenda





01

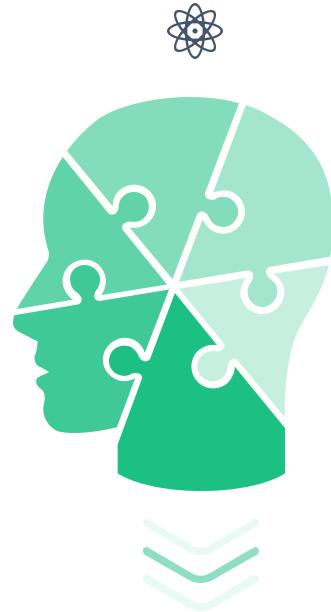
## BACK PROPAGATION

- Backpropagation introduction
- Feed forward process



# Backpropagation

Following are the key details about backpropagation:



Weights are determined by training

- Back-propagation:
  - ↳ On a given input, compare the actual output to desired outputs
  - ↳ Adjust weights to output nodes
  - ↳ Work backwards through the various layers



- Start out with initial random weights
  - ↳ Best to keep weights close to zero (<<10)

# Back Propagation (Gradient computation)

## Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
  
2. Backward propagation of the propagation's output activations through the neural network using the training pattern target in order to generate the deltas of all output and hidden neurons.

# Back Propagation (Gradient computation)

Phase 2: Weight update

For each weight-synapse follow the following steps:

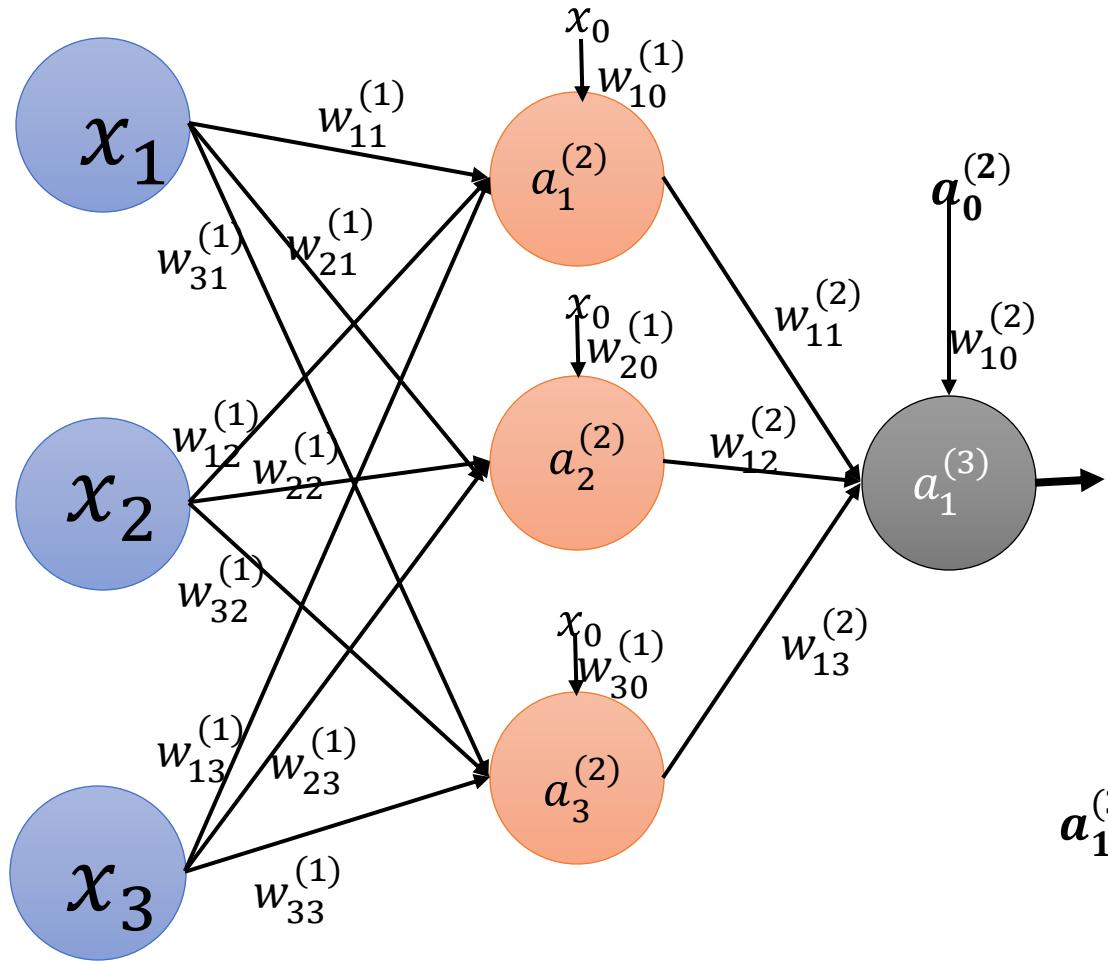
1. Multiply its output delta and input activation to get the gradient of the weight.
2. Subtract a ratio (percentage) of the gradient from the weight.

This ratio (percentage) influences the speed and quality of learning; it is called the learning rate. The greater the ratio, the faster the neuron trains; the lower the ratio, the more accurate the training is. The sign of the gradient of a weight indicates where the error is increasing, this is why the weight must be updated in the opposite direction.

Repeat phase 1 and 2 until the performance of the network is satisfactory.

# Feed forward process

## Mathematical Modelling of Neural Network



$$a_1^{(2)} = g\{w_{10}^{(1)}x_0 + w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + w_{13}^{(1)}x_3\}$$

$$a_2^{(2)} = g\{w_{20}^{(1)}x_0 + w_{21}^{(1)}x_1 + w_{22}^{(1)}x_2 + w_{23}^{(1)}x_3\}$$

$$a_3^{(2)} = g\{w_{30}^{(1)}x_0 + w_{31}^{(1)}x_1 + w_{32}^{(1)}x_2 + w_{33}^{(1)}x_3\}$$

$$a_1^{(3)} = g\{w_{10}^{(2)}a_0^{(2)} + w_{11}^{(2)}a_1^{(2)} + w_{12}^{(2)}a_2^{(2)} + w_{13}^{(2)}a_3^{(2)}\}$$

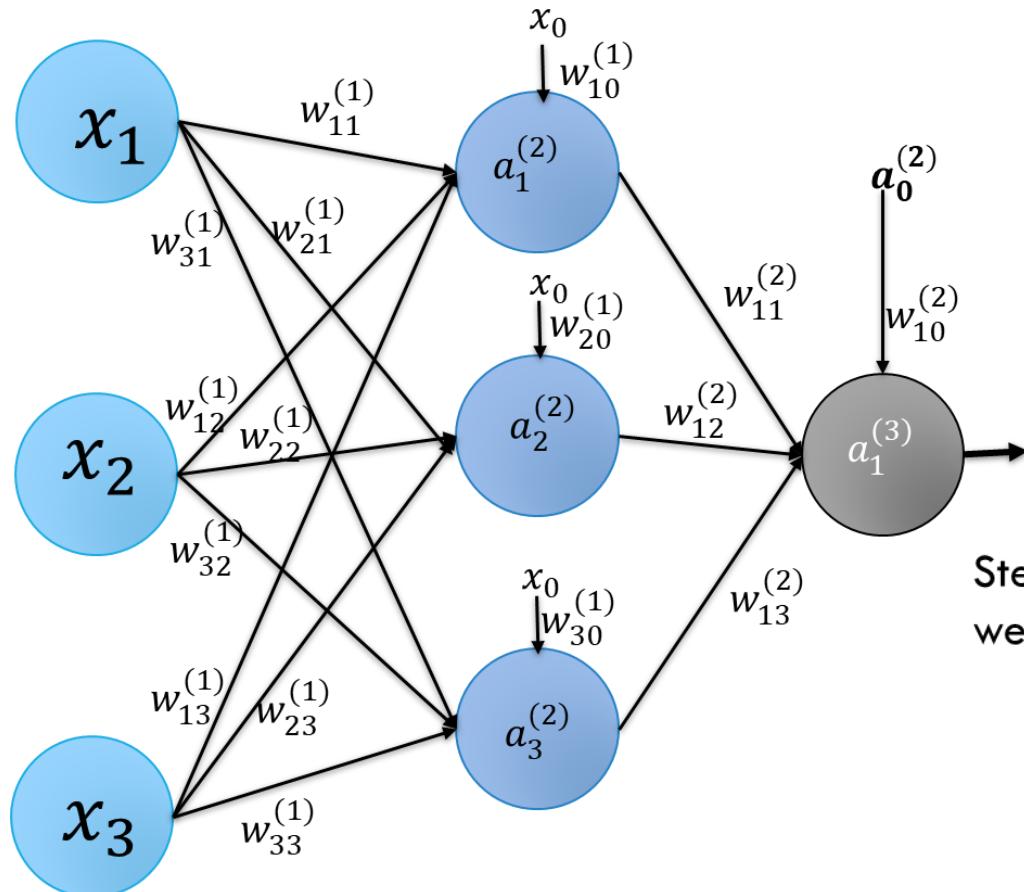


02

## MATHEMATICS

- BPP – mathematics
- Weight updation

# Mathematical Modelling of Neural Network



$$(Error) E = \frac{1}{2} (t - \hat{y})^2$$

E = mean square error

t = target value (Actual Value)

$\hat{y}$  = predicted output of the output neuron

Step 2 finding the derivative of the error with respect to weights

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \text{Gradient of error w.r.t. weight}$$

$$E = \frac{1}{2}(t - y\hat{a})^2 = \frac{1}{2}(t - a_1^{(3)})^2$$

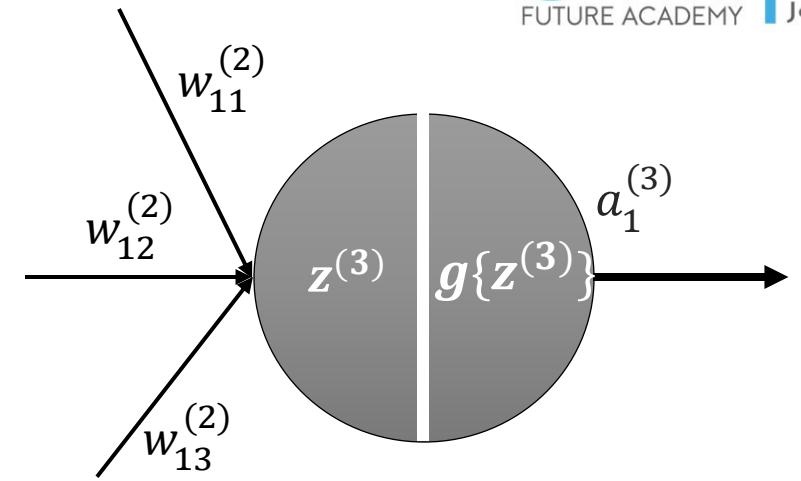
$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \frac{\partial E}{\partial a_1^{(3)}} \frac{\partial a_1^{(3)}}{\partial z^{(3)}} \frac{\partial z^{(3)}}{\partial w_{ij}^{(l)}}$$

$$\frac{\partial z^{(3)}}{\partial w_{ij}^{(l)}} = \frac{\partial}{\partial w_{ij}^{(l)}} \left( \sum_{k=1}^n (w_{kj}^{(l)} * a_k^{(l)}) \right) = a_k^{(l)}$$

$$\frac{\partial a_1^{(3)}}{\partial z^{(3)}} = \frac{\partial g\{z^{(3)}\}}{\partial z^{(3)}} = g\{z^{(3)}\}(1 - g\{z^{(3)}\})$$

$$= a_1^{(3)}(1 - a_1^{(3)})$$

$$\frac{\partial E}{\partial a_1^{(3)}} = \frac{\partial}{\partial a_1^{(3)}} \frac{1}{2} (t - y)^2 = \frac{\partial}{\partial a_1^{(3)}} \frac{1}{2} (t - a_1^{(3)})^2 = (a_1^{(3)} - t)$$



*putting it altogether,*

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = a_1^{(3)}(1 - a_1^{(3)}) * (a_1^{(3)} - t) * a_k^{(2)}$$

$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \delta^{(3)} * a_k^{(2)}$$

*where*  $\delta^{(3)} = a_1^{(3)}(1 - a_1^{(3)}) * (a_1^{(3)} - t)$

# Backpropagation Steps

Step 1: Operate Feed Forward Network to find the  $y$  actual value of network and calculate the Error term on output neuron.

Step 2: Calculate  $\delta$  for output neuron and hidden neurons,  $\delta$  will not be calculated for input neuron

$$\delta^{(3)} = a_1^{(3)}(1-a_1^{(3)}) * (a_1^{(3)} - t)$$

$$\delta^{(2)} = \delta^{(3)} w_{kj}^{(2)} a_1^{(2)}(1-a_1^{(2)})$$

# Backpropagation Steps

Step 3 Update the weights so as to minimize the error term

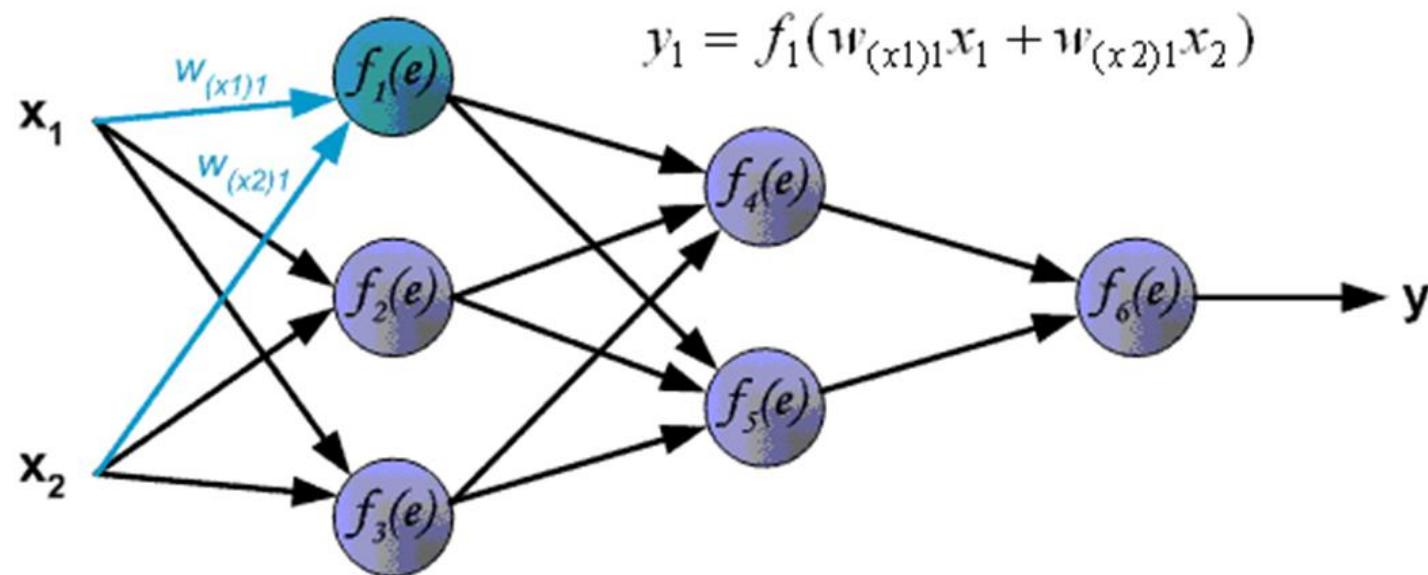
$$\frac{\partial E}{\partial w_{ij}^{(l)}} = \delta^{(3)} * a_k^{(2)}$$

$$\Delta w_{ij}^{(2)} = \alpha \frac{\partial E}{\partial w_{ij}^{(l)}} = \alpha * \delta^{(3)} * a_k^{(2)}$$

$$w_{ij}^{(2)} = w_{ij}^{(2)} + \Delta w_{ij}^{(2)}$$

# Backpropagation

FP



# The back-propagation training algorithm

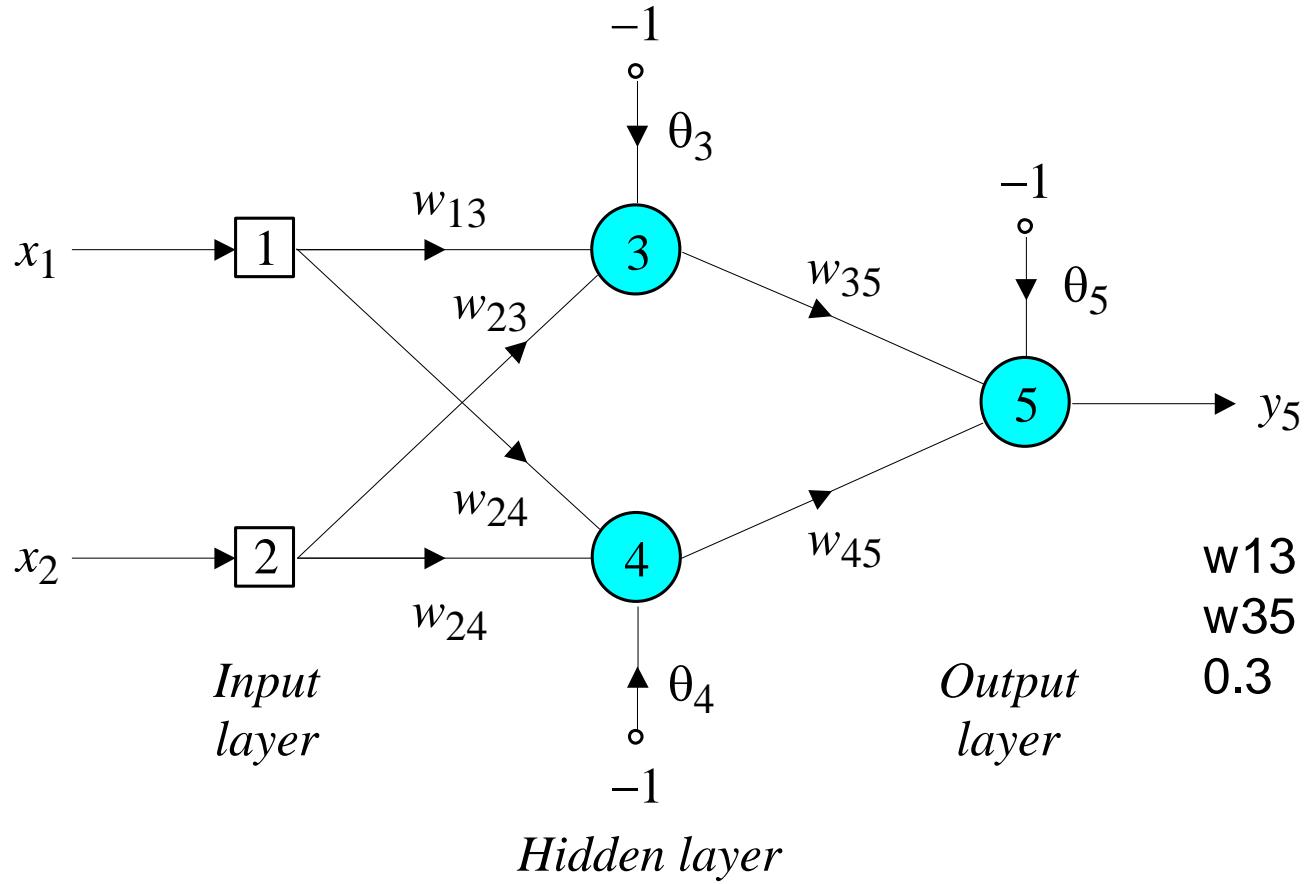
## Step 1: Initialization

Set all the weights and threshold levels of the network to random numbers uniformly distributed inside a small range:

$$\left( -\frac{2.4}{F_i}, +\frac{2.4}{F_i} \right)$$

where  $F_i$  is the total number of inputs of neuron  $i$  in the network. The weight initialization is done on a neuron-by-neuron basis.

# Three-layer network

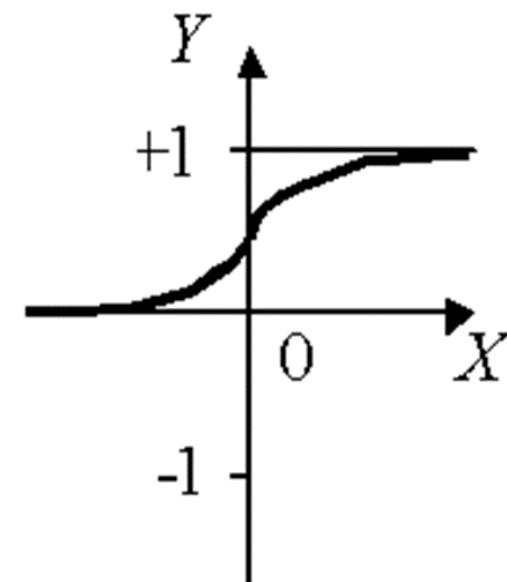


$w_{13} = 0.5, w_{14} = 0.9, w_{23} = 0.4, w_{24} = 1.0,$   
 $w_{35} = 1.2, w_{45} = 1.1, \theta_3 = 0.8, \theta_4 = 0.1$  and  $\theta_5 = 0.3$

# The back-propagation training algorithm

- The effect of the threshold applied to a neuron in the hidden or output layer is represented by its weight,  $\theta$ , connected to a fixed input equal to -1.
- The initial weights and threshold levels are set randomly e.g., as follows:  
 $w_{13} = 0.5, w_{14} = 0.9, w_{23} = 0.4, w_{24} = 1.0, w_{35} = -1.2, w_{45} = 1.1, \theta_3 = 0.8, \theta_4 = -0.1$  and  $\theta_5 = 0.3$ .

Assuming the sigmoid activation Function



$$y^{sigmoid} = \frac{1}{1+e^{-X}}$$

# The back-propagation training algorithm

## Step 2: Activation

Activate the back-propagation neural network by applying inputs  $x_1(p), x_2(p), \dots, x_n(p)$  and desired outputs  $y_{d,1}(p), y_{d,2}(p), \dots, y_{d,n}(p)$ .

- (a) Calculate the actual outputs of the neurons in the hidden layer:

$$y_j(p) = \text{sigmoid} \left[ \sum_{i=1}^n x_i(p) \cdot w_{ij}(p) - \theta_j \right]$$

- where  $n$  is the number of inputs of neuron  $j$  in the hidden layer, and sigmoid is the sigmoid activation function.

# The back-propagation training algorithm

Step 2: Activation (continued)

- (b) Calculate the actual outputs of the neurons in the output layer:

$$y_k(p) = \text{sigmoid} \left[ \sum_{j=1}^m x_{jk}(p) \cdot w_{jk}(p) - \theta_k \right]$$

where m is the number of inputs of neuron k in the output layer.

# The back-propagation training algorithm

If the sigmoid activation function is used the output of the hidden layer is

$$y_3 = \text{sigmoid}(x_1 w_{13} + x_2 w_{23} - \theta_3) = 1 / \left[ 1 + e^{-(1 \cdot 0.5 + 1 \cdot 0.4 - 1 \cdot 0.8)} \right] = 0.5250$$

$$y_4 = \text{sigmoid}(x_1 w_{14} + x_2 w_{24} - \theta_4) = 1 / \left[ 1 + e^{-(1 \cdot 0.9 + 1 \cdot 1.0 + 1 \cdot 0.1)} \right] = 0.8808$$

And the actual output of neuron 5 in the output layer is

$$y_5 = \text{sigmoid}(y_3 w_{35} + y_4 w_{45} - \theta_5) = 1 / \left[ 1 + e^{-(0.5250 \cdot 1.2 + 0.8808 \cdot 1.1 - 1 \cdot 0.3)} \right] = 0.5097$$

And the error is

$$e = y_{d,5} - y_5 = 0 - 0.5097 = -0.5097$$

# The back-propagation training algorithm

What learning law applies  
in a multilayer neural network?

$$\Delta w_{jk}(p) = \alpha \cdot y_j(p) \cdot \delta_k(p)$$

## Step 3: Weight training output layer

Update the weights in the back-propagation network propagating backward the errors associated with output neurons.

(a) Calculate the error

$$e_k(p) = y_{d,k}(p) - y_k(p)$$

and then the error gradient for the neurons in the output layer:

$$\delta_k(p) = y_k(p) \cdot [1 - y_k(p)] \cdot e_k(p)$$

Then the weight corrections:

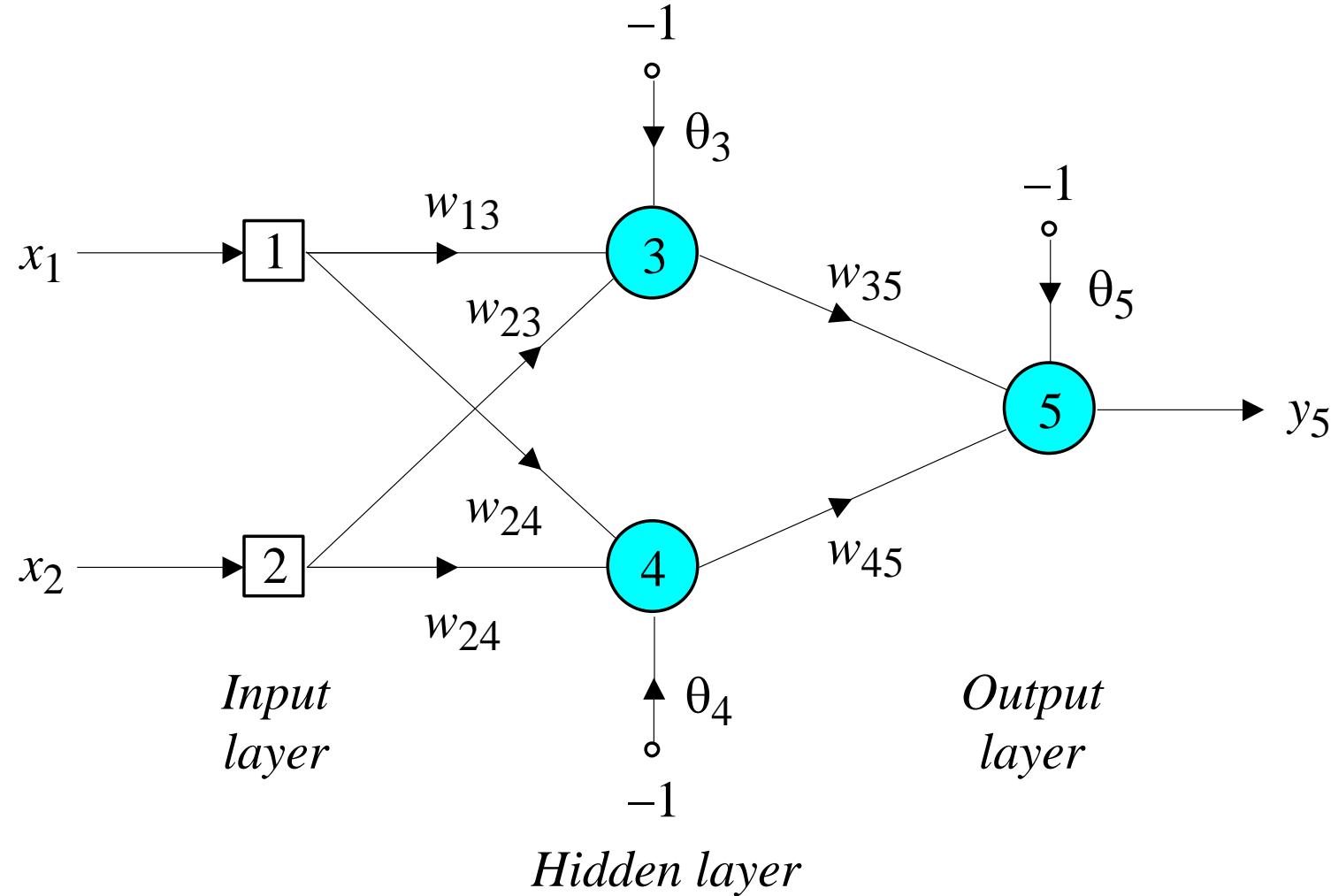
$$\Delta w_{jk}(p) = \alpha \cdot y_j(p) \cdot \delta_k(p)$$

Then the new weights at the output neurons:

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$$

# The back-propagation training algorithm

A three-layer network for solving the Exclusive-OR operation



# The back-propagation training algorithm

- The error gradient for neuron 5 in the output layer:

$$\delta_5 = y_5(1-y_5)e = 0.5097 \cdot (1-0.5097) \cdot (-0.5097) = -0.1274$$

- Determine the weight corrections assuming that the learning rate parameter,  $\alpha$ , is equal to 0.1:

$$\Delta w_{35} = \alpha \cdot y_3 \cdot \delta_5 = 0.1 \cdot 0.5250 \cdot (-0.1274) = -0.0067$$

$$\Delta w_{45} = \alpha \cdot y_4 \cdot \delta_5 = 0.1 \cdot 0.8808 \cdot (-0.1274) = -0.0112$$

$$\Delta \theta_5 = \alpha \cdot (-1) \cdot \delta_5 = 0.1 \cdot (-1) \cdot (-0.1274) = -0.0127$$

# The back-propagation training algorithm

Step 3: Weight training hidden layer

(b) Calculate the error gradient for the neurons in the hidden layer:

$$\delta_j(p) = y_j(p) \cdot [1 - y_j(p)] \cdot \sum_{k=1}^l \delta_k(p) w_{jk}(p)$$

Calculate the weight corrections:

$$\Delta w_{ij}(p) = \alpha \cdot x_i(p) \cdot \delta_j(p)$$

Update the weights at the hidden neurons:

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

# The back-propagation training algorithm

- The error gradients for neurons 3 and 4 in the hidden layer:

$$\delta_3 = y_3(1-y_3) \cdot \delta_5 \cdot w_{35} = 0.5250 \cdot (1-0.5250) \cdot (-0.1274) \cdot (-1.2) = 0.0381$$

$$\delta_4 = y_4(1-y_4) \cdot \delta_5 \cdot w_{45} = 0.8808 \cdot (1-0.8808) \cdot (-0.1274) \cdot 1.1 = -0.0147$$

- Determine the weight corrections:

$$\Delta w_{13} = \alpha \cdot x_1 \cdot \delta_3 = 0.1 \cdot 1 \cdot 0.0381 = 0.0038$$

$$\Delta w_{23} = \alpha \cdot x_2 \cdot \delta_3 = 0.1 \cdot 1 \cdot 0.0381 = 0.0038$$

$$\Delta \theta_3 = \alpha \cdot (-1) \cdot \delta_3 = 0.1 \cdot (-1) \cdot 0.0381 = -0.0038$$

$$\Delta w_{14} = \alpha \cdot x_1 \cdot \delta_4 = 0.1 \cdot 1 \cdot (-0.0147) = -0.0015$$

$$\Delta w_{24} = \alpha \cdot x_2 \cdot \delta_4 = 0.1 \cdot 1 \cdot (-0.0147) = -0.0015$$

$$\Delta \theta_4 = \alpha \cdot (-1) \cdot \delta_4 = 0.1 \cdot (-1) \cdot (-0.0147) = 0.0015$$

# The back-propagation training algorithm

- At last, we update all weights and threshold:

$$w_{13} = w_{13} + \Delta w_{13} = 0.5 + 0.0038 = 0.5038$$

$$w_{14} = w_{14} + \Delta w_{14} = 0.9 - 0.0015 = 0.8985$$

$$w_{23} = w_{23} + \Delta w_{23} = 0.4 + 0.0038 = 0.4038$$

$$w_{24} = w_{24} + \Delta w_{24} = 1.0 - 0.0015 = 0.9985$$

$$w_{35} = w_{35} + \Delta w_{35} = -1.2 - 0.0067 = -1.2067$$

$$w_{45} = w_{45} + \Delta w_{45} = 1.1 - 0.0112 = 1.0888$$

$$\theta_3 = \theta_3 + \Delta \theta_3 = 0.8 - 0.0038 = 0.7962$$

$$\theta_4 = \theta_4 + \Delta \theta_4 = -0.1 + 0.0015 = -0.0985$$

$$\theta_5 = \theta_5 + \Delta \theta_5 = 0.3 + 0.0127 = 0.3127$$

- The training process is repeated until the sum of squared errors is less than 0.001.

# The back-propagation training algorithm

## Step 4: Iteration

Increase iteration  $p$  by one, go back to Step 2 and repeat the process until the selected error criterion is satisfied.

As an example, we may consider the three-layer back-propagation network. Suppose that the network is required to perform logical operation Exclusive-OR. Recall that a single-layer perceptron could not do this operation. Now we will apply the three-layer net.



03

## APPLICATIONS

- Use cases of ANN

# Use Cases of ANN

## Artificial Neural Networks are Improving Marketing Strategies

- By adopting Artificial Neural Networks businesses are able to optimize their marketing strategy.
- Systems powered by Artificial Neural Networks are capable of processing masses of information. This includes customers personal details, shopping patterns as well as any other information relevant to your business.
- Once processed this information can be sorted and presented in a useful and accessible way. This is generally known as market segmentation.

# Use Cases of ANN

## Developing Targeted Marketing Campaigns

- Through unsupervised learning, Artificial Neural Networks are able to identify customers with a similar characteristic.
- This allows businesses to group together customers with similarities, such as economic status or preferring vinyl records to downloaded music.
- Supervised learning systems allow Artificial Neural Networks to set out a clear aim for your marketing strategy.
- Like unsupervised systems, they can also segment customers into similar groupings.

# Use Cases of ANN

## Improving Search Engine Functionality

- During 2015 Google I/O keynote address in San Francisco, Google revealed they were working on improving their search engine.
- These improvements are powered by a 30-layer deep Artificial Neural Network.
- This depth of layers, Google believes, allows the search engine to process complicated searches such as shapes and colors.
- Using an Artificial Neural Network allows the system to constantly learn and improve.



03

## OPTIMIZATION

- Handling overfitting
- Handling underfitting

# Handling overfitting

## 1. Simplifying The Model

- The first step when dealing with overfitting is to decrease the complexity of the model. To decrease the complexity, we can simply remove layers or reduce the number of neurons to make the network smaller.
- While doing this, it is important to calculate the input and output dimensions of the various layers involved in the neural network.
- There is no general rule on how much to remove or how large your network should be. But, if your neural network is overfitting, try making it smaller.

# Handling overfitting

## 2. Early Stopping

- Early stopping is a form of regularization while training a model with an iterative method, such as gradient descent.
- Since all the neural networks learn exclusively by using gradient descent, early stopping is a technique applicable to all the problems.

## 3. Use Data Augmentation

- In the case of neural networks, data augmentation simply means increasing size of the data that is increasing the number of images present in the dataset.
- Some of the popular image augmentation techniques are flipping, translation, rotation, scaling, changing brightness, adding noise etcetera.

# Handling overfitting

## 4. Use Regularization

Regularization is a technique to reduce the complexity of the model. It does so by adding a penalty term to the loss function. The most common techniques are known as L1 and L2 regularization:

- The L1 penalty aims to minimize the absolute value of the weights. This is mathematically shown in the below formula.

$$L(x, y) \equiv \sum_{i=1}^n (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^n |\theta_i|$$

# Handling overfitting

- The L2 penalty aims to minimize the squared magnitude of the weights. This is mathematically shown in the below formula.

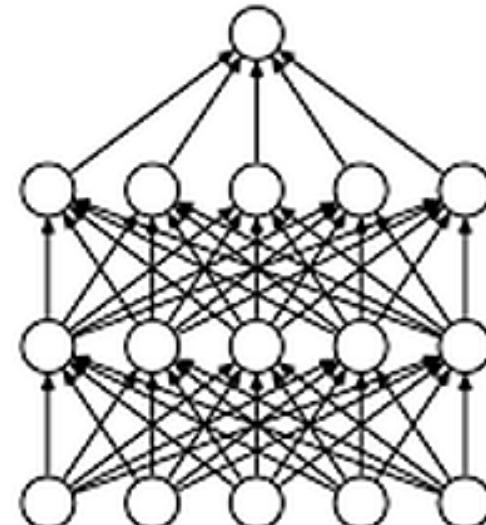
$$L(x, y) \equiv \sum_{i=1}^n (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^n \theta_i^2$$

L2 Regularization

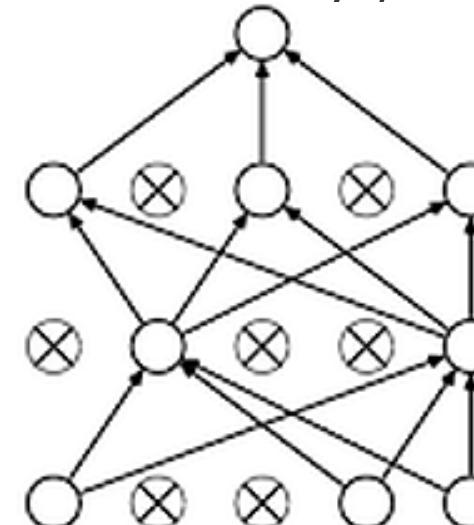
# Handling overfitting

## 5. Use Dropouts

- Dropout is a regularization technique that prevents neural networks from overfitting. Regularization methods like L1 and L2 reduce overfitting by modifying the cost function.
- Dropout on the other hand, modify the network itself. It randomly drops neurons from the neural network during training in each iteration.
- When we drop different sets of neurons, it's equivalent to training different neural networks. The different networks will overfit in different ways, so the net effect of dropout will be to



(a) Standard Neural Net



(b) After applying dropout.

# Handling underfitting

Let's understand few techniques to handle Underfitting

## 1. Increase The Complexity Of The Model

- Increasing the number of layers in the model.
- Increasing the number of neurons in each layer.
- Changing what type of layers we're using and where.

# Handling underfitting

## 2. Add More Features To The Input Samples

For example, say we have a model that is attempting to predict the price of a stock based on the last three closing prices of this stock. So our input would consist of three features:

- day 1 close
- day 2 close
- day 3 close

# Handling underfitting

## 3. Reduce Dropout

- The last tip we'll discuss about reducing underfitting is to reduce dropout. Again, this is exactly opposite of a technique we use in a reducing overfitting.
- Dropout is a regularization technique that randomly ignores a subset of nodes in a given layer. It essentially prevents these dropped out nodes from participating in producing a prediction on the data



04

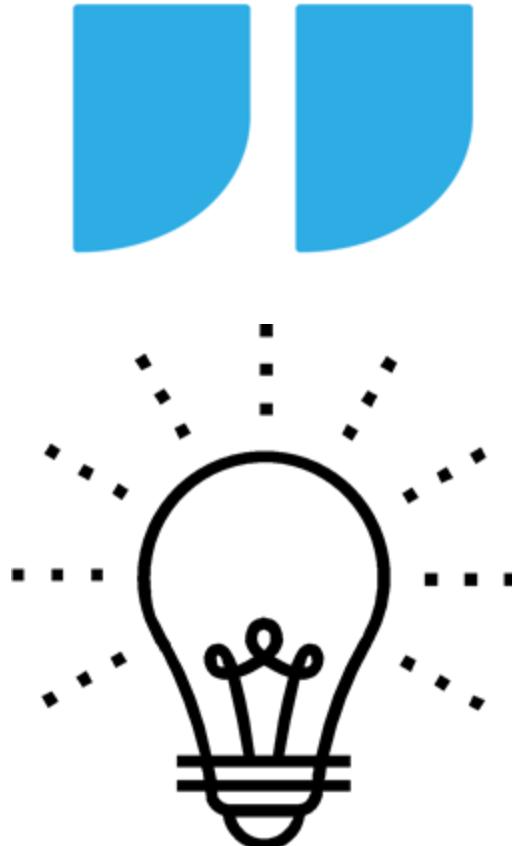
## CONCLUSION

- Summary
- Quiz



# Summary

- Weights in neural networks are updated using backpropagation algorithm
- Neural networks work very well on unstructured and structured data
- Overfitting can be handled by using regularization, dropout optimization, data augmentation and simplifying model
- Underfitting can be handled by collecting more features, reduce dropout and regularization



# Quiz

## Question

Which of the following technique is used to handle overfitting?

- A. Dropout optimization
- B. Regularization
- C. Early stopping
- D. All of above



# Quiz

## Question

Which of the following technique is used to handle overfitting?

- A. Dropout optimization
- B. Regularization
- C. Early stopping
- D. All of above



Answer: D



# TERIMA KASIH

## Orbit Future Academy

PT Orbit Ventura Indonesia  
Center of Excellence (Jakarta Selatan)  
Gedung Veteran RI, Lt.15  
Unit Z15-002, Plaza Semanggi  
Jl. Jenderal Sudirman Kav.50, Jakarta  
12930, Indonesia

- Jakarta Selatan/Pusat
- Jakarta Barat/BSD
- Kota Bandung
- Kab. Bandung
- Jawa Barat

## Hubungi Kami

Director of Sales & Partnership  
[ira@orbitventura.com](mailto:ira@orbitventura.com)  
+62 858-9187-7388

## Social Media

-  [Orbit Future Academy](#)
-  [@OrbitFutureAcademyIn1](#)
-  [OrbitFutureAcademy](#)
-  [Orbit Future Academy](#)