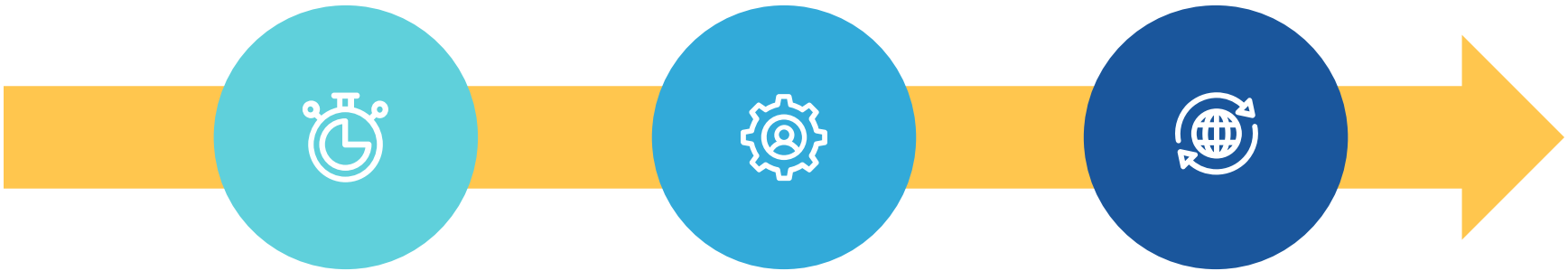
BIG DATA

Big Data

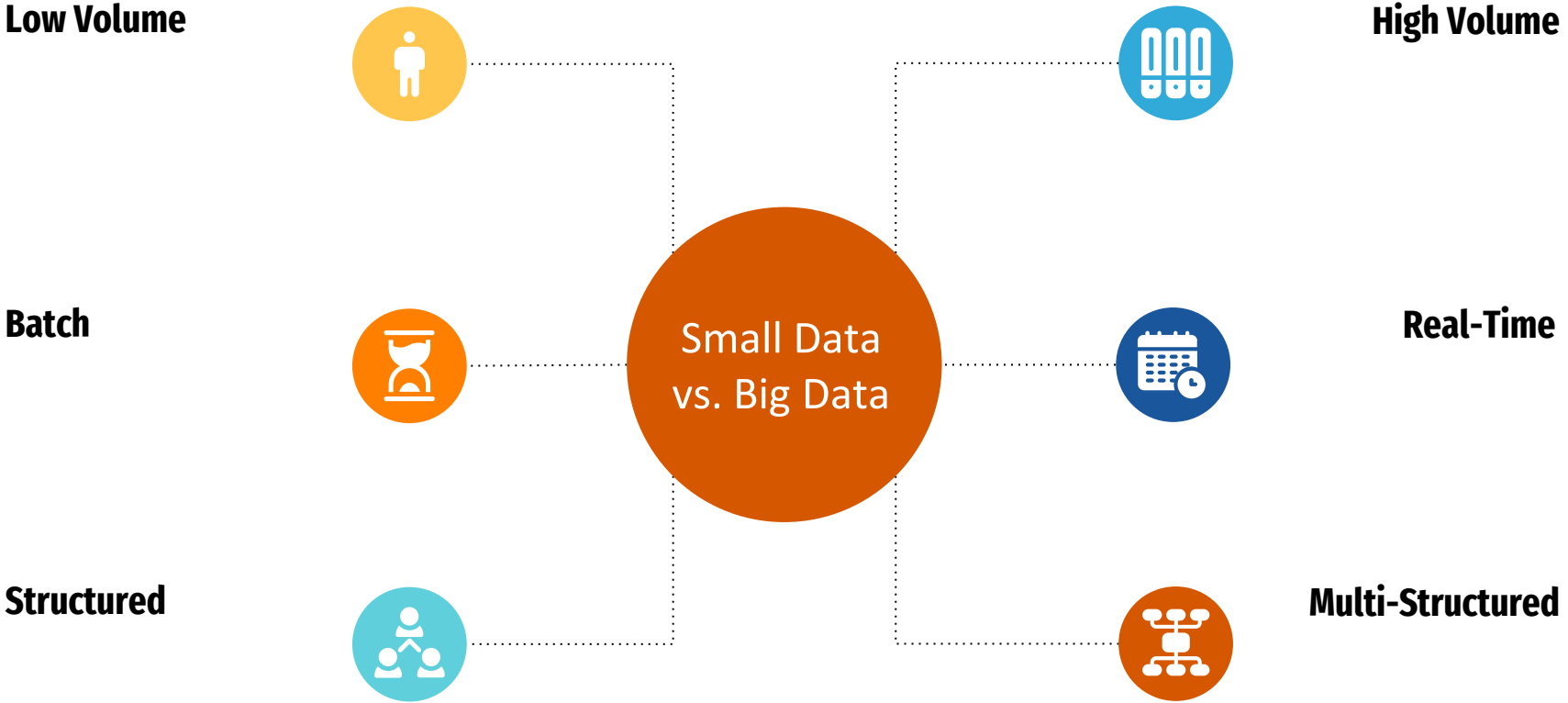


Tren penggunaan Big Data “meledak” di awal dekade abad 21

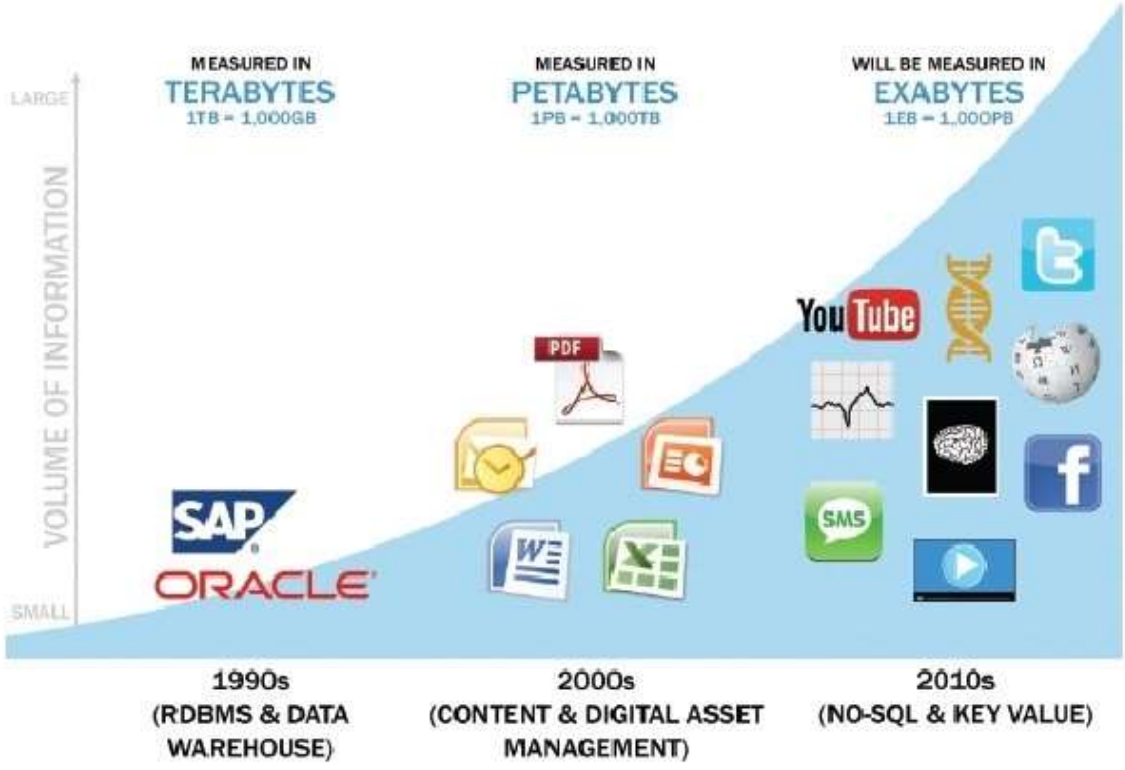
Perusahaan besar online dan startup, misalnya Google, Amazon, eBay, LinkedIn, Facebook, dibangun dari data yang besar

Data yang sangat besar dapat memengaruhi banyak hal, di berbagai bidang: medis, industri, ekonomi dan perbankan, dll

SMALL DATA V BIG DATA



Data Evolution and the Rise of Big Data Sources



Source: EMC Education Services, Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data, Wiley, 2015

Definisi Big Data

McKinsey Global (2011)
“ Big Data dapat didefinisikan dengan data yang memiliki skala (volume), distribusi (velocity), keragaman (variety) yang sangat besar, dan atau abadi, sehingga membutuhkan penggunaan arsitektur teknikal dan metode analitis yang inovatif untuk mendapatkan wawasan yang dapat memberikan nilai bisnis baru (informasi yang bermakna)”

Hurwitz, et al. (2013)
“Big data merupakan istilah untuk sekumpulan data yang begitu besar atau kompleks dimana tidak bisa ditangani lagi dengan sistem teknologi komputer konvensional”

Big Data

Big Data mirip dengan ‘small data’, namun ukurannya jauh lebih besar

Dengan data sangat besar perlu pendekatan berbeda (Teknik, Alat Bantu, Arsitektur)

Big Data memberikan nilai dari penyimpanan dan pemrosesan dari kuantitas sangat besar yang tidak bisa dianalisis (menggunakan) dengan teknik komputasi tradisional

Definisi

Kumpulan proses dari volume data dalam jumlah besar yang terstruktur maupun tidak terstruktur dan digunakan untuk membantu kegiatan bisnis, merupakan pengembangan dari sistem database pada umumnya.

Tujuan

menyelesaikan masalah baru atau masalah lama dengan cara lebih baik

Contoh

Decoding genome manusia awalnya memerlukan pemrosesan selama 1 bulan menjadi 1 minggu

Scientists finally finish decoding entire human genome

Scientists say they have finally assembled the full genetic blueprint for human life, adding the missing pieces to a puzzle nearly completed two decades ago

By LAURA UNGAR AP Science Writer
1 April 2022, 01:28 • 5 min read

Scientists say they have finally assembled the full genetic blueprint for human life, adding the missing pieces to a puzzle nearly completed two decades ago.

An international team described the first-ever sequencing of a complete human genome – the set of instructions to build and sustain a human being – in research published Thursday in the journal Science. The previous effort, celebrated across the world, was incomplete because DNA sequencing technologies of the day weren't able to read certain parts of it. Even after updates, it was missing about 8% of the genome.

"Some of the genes that make us uniquely human were actually in this 'dark matter of the genome' and they were totally missed," said Evan Eichler, a University of Washington researcher who participated in the current effort and the original Human Genome Project. "It took 20-plus years, but we finally got it done."

Many — including Eichler's own students — thought it had been finished already. "I was teaching them, and they said, 'Wait a minute. Isn't this like the sixth time you guys have declared victory? I said, 'No, this time we really, really did it!"

Scientists said this full picture of the genome will give humanity a greater understanding of our evolution and biology while also opening the door to medical discoveries in areas like aging, neurodegenerative conditions, cancer and heart disease.

"We're just broadening our opportunities to understand human disease," said Karen Miga, an author of one of the six studies published Thursday.

SPECIAL ISSUE RESEARCH ARTICLE HUMAN GENOMICS

The complete sequence of a human genome

+91 authors Authors Info & Affiliations

SCIENCE • 31 Mar 2022 • Vol 376, Issue 6588 • pp. 44-53 • DOI: 10.1126/science.abj6987

RELATED SPECIAL ISSUE RESEARCH ARTICLE
Epigenetic patterns in a complete human genome

RELATED SPECIAL ISSUE RESEARCH ARTICLE
Complete genomic and epigenetic maps of human centromeres

RELATED PERSPECTIVE
A next-generation human genome sequence

Abstract
Cell line and sequencing
Genome assembly
DNA assembly
Assembly validation and polishing
A truly complete genome
Acrocentric chromosomes
Analysis and resources
Future of the human reference genome
Acknowledgments
Supplementary Materials
References and Notes

Abstract

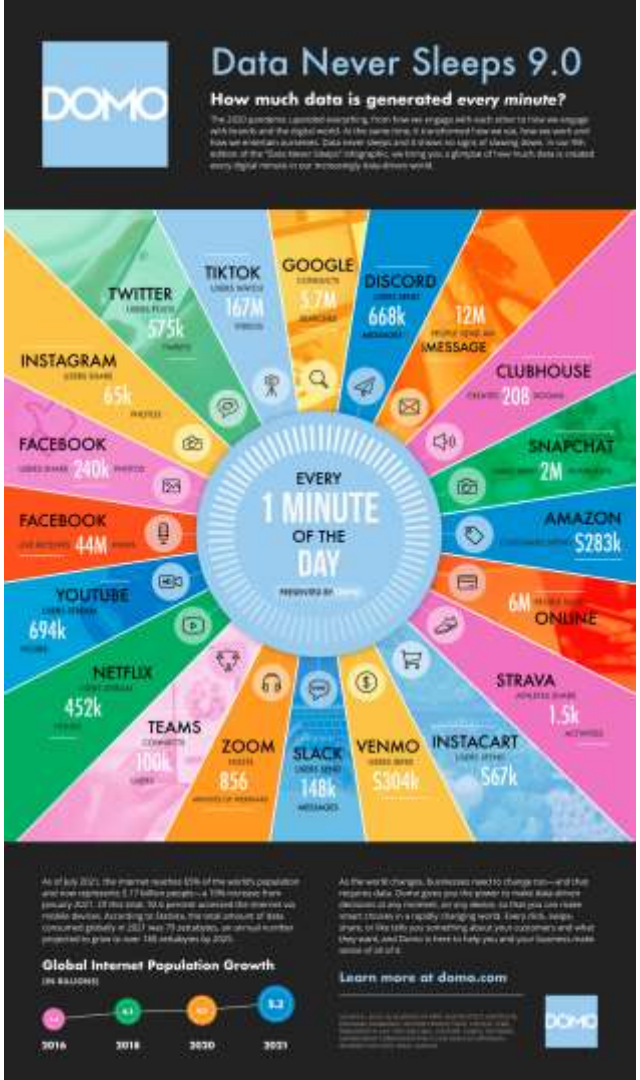
Since its initial release in 2000, the human reference genome has covered only the euchromatic fraction of the genome, leaving important heterochromatic regions unfinished. Addressing the remaining 8% of the genome, the Telomere-to-Telomere (T2T) Consortium presents a complete 3.055 billion–base pair sequence of a human genome, T2T-CHM13, that includes gapless assemblies for all chromosomes except Y, corrects errors in the prior references, and introduces nearly 200 million base pairs of sequence containing 1956 gene predictions, 99 of which are predicted to be protein coding. The completed regions include all centromeric satellite arrays, recent segmental duplications, and the short arms of all five acrocentric chromosomes, unlocking these complex regions of the genome to variational and functional studies.

Apa manfaat belajar analisis Big Data?

- **Kebutuhan untuk analisis yang lebih mendalam dalam industri, akademisi, dan pemerintah, maupun lainnya**

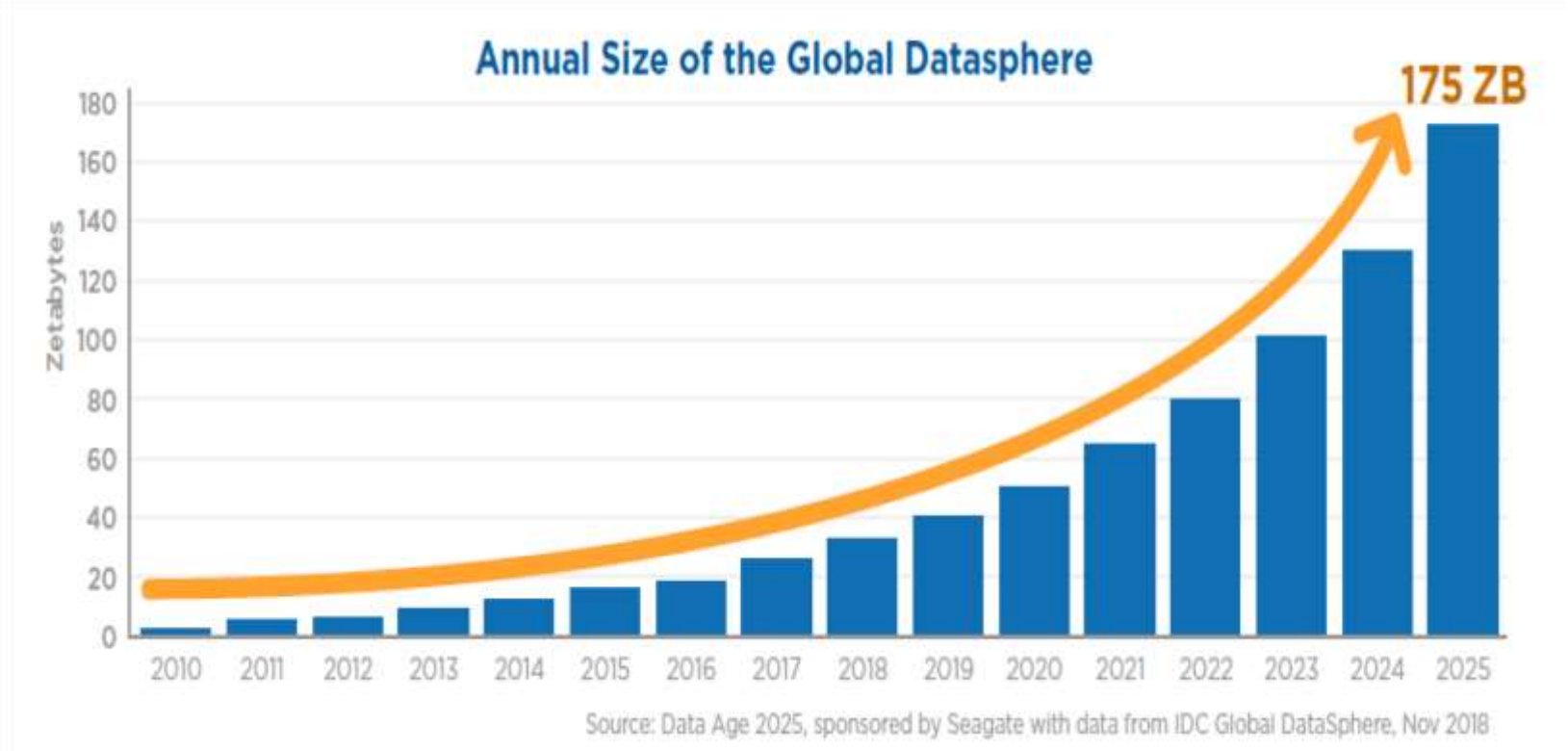
- **Ketersediaan sumber data baru, munculnya peluang analitis yang lebih kompleks menciptakan kebutuhan untuk memikirkan kembali arsitektur data yang ada untuk memungkinkan analisis yang dapat dengan optimal memanfaatkan Big Data**

Big Data Infographic



Big Data Growth

**1 ZB = 1.000.000.000 TB*



BIG DATA dalam detik

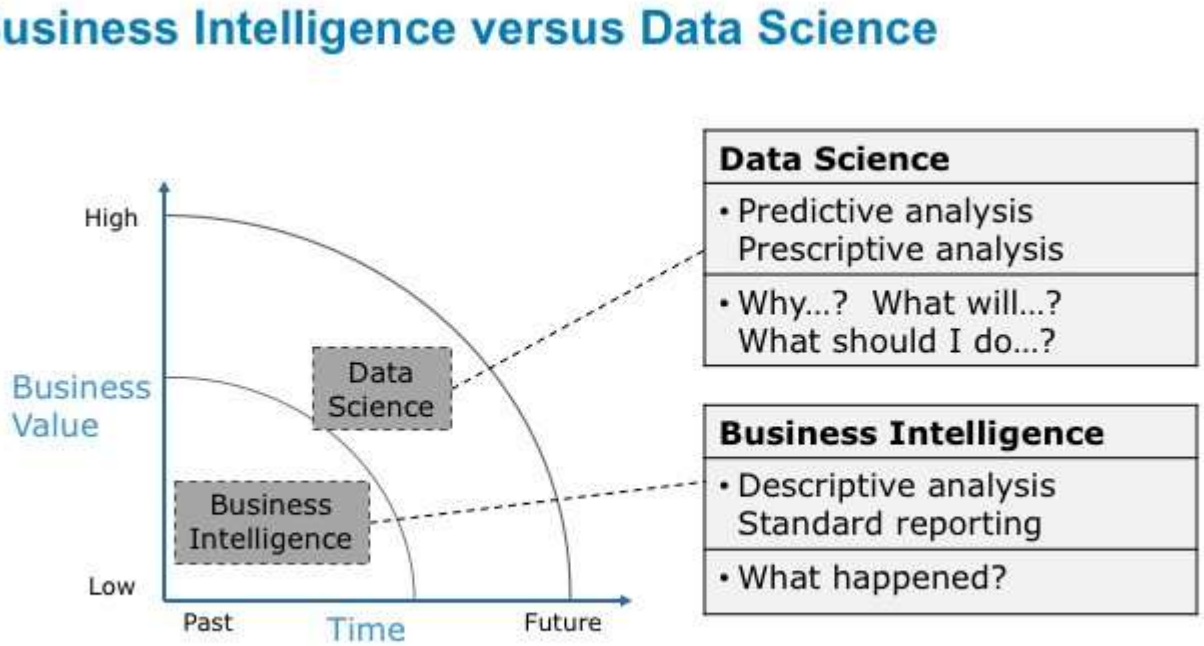
Ada 5.7 juta pencarian di Google dalam satu menit, yang berarti 95 ribu setiap detiknya

Di Facebook ada 240 ribu pengguna membagikan foto dan 44 juta pengguna menonton tayangan langsung, yang artinya tiap detik ada 4000 foto dibagikan dan 733 ribu orang menonton secara bersamaan

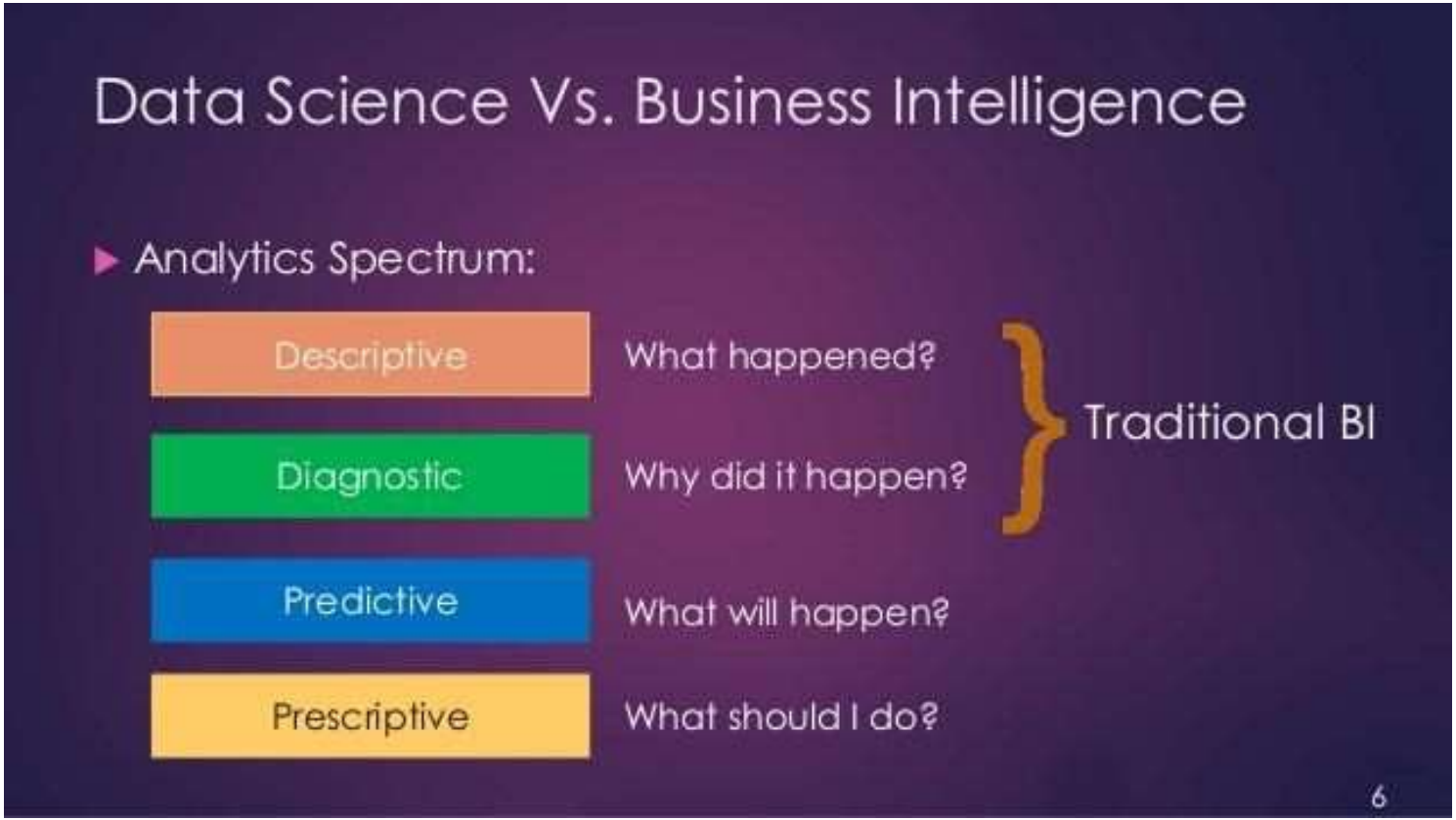
Pada Twitter ada 575 ribu tweet/menit yang artinya ada +- 9600 tweet/detik

Ada 167 juta pengguna menonton video di Tiktok setiap menit, artinya ada +- 2,7 juta pengguna menonton tiap detiknya

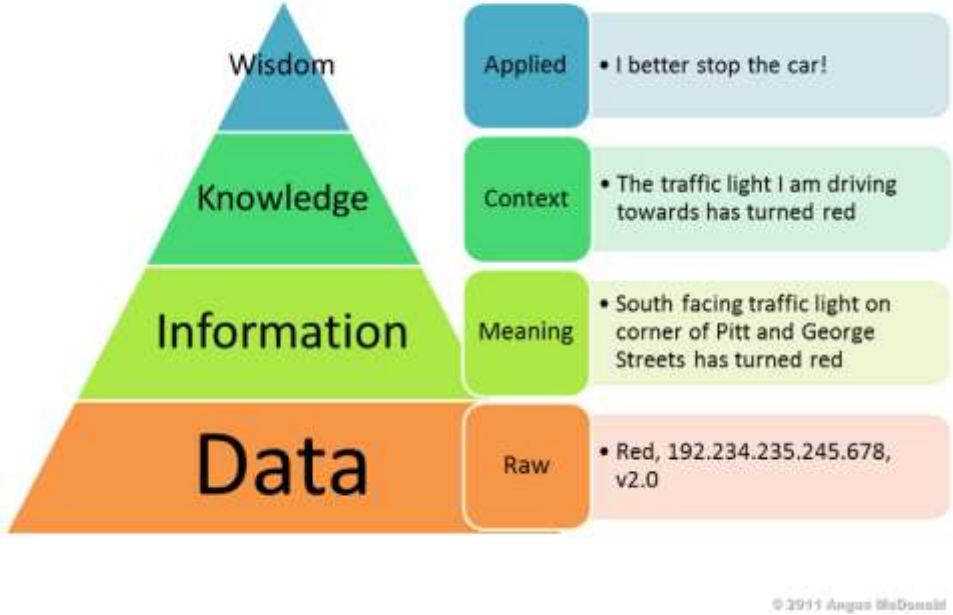
Big Data analisis, Data Science v Business Intelligence



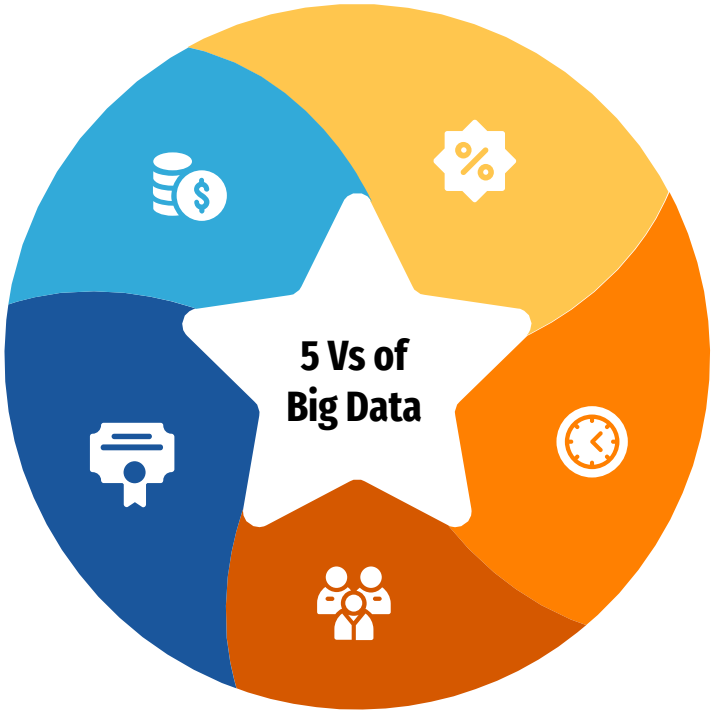
Big Data analisis, Data Science v Business Intelligence



Big Data Pyramid



Kapan data disebut “Big Data”



01
Volume

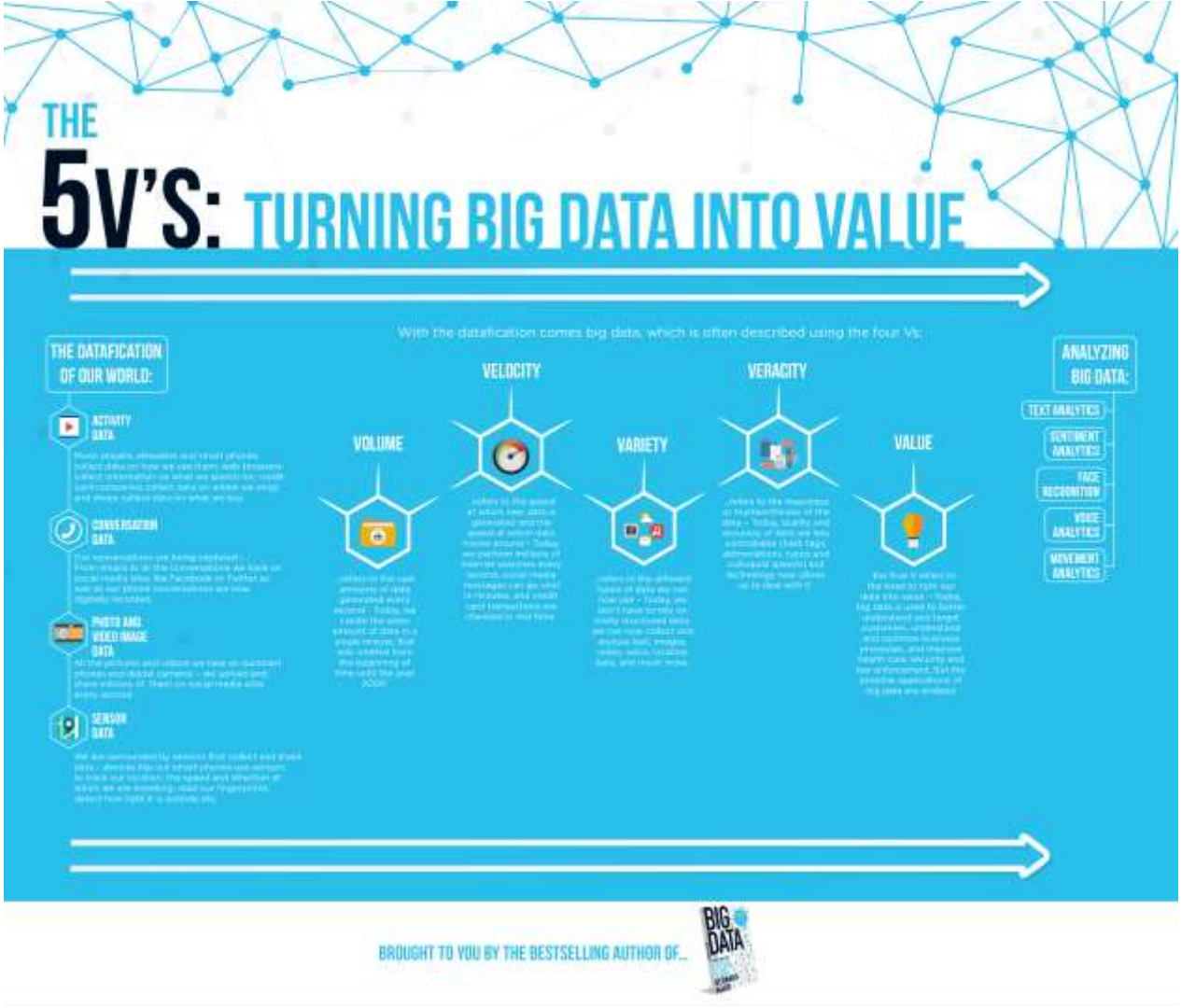
02
Velocity

03
Variety

04
Veracity

05
Value

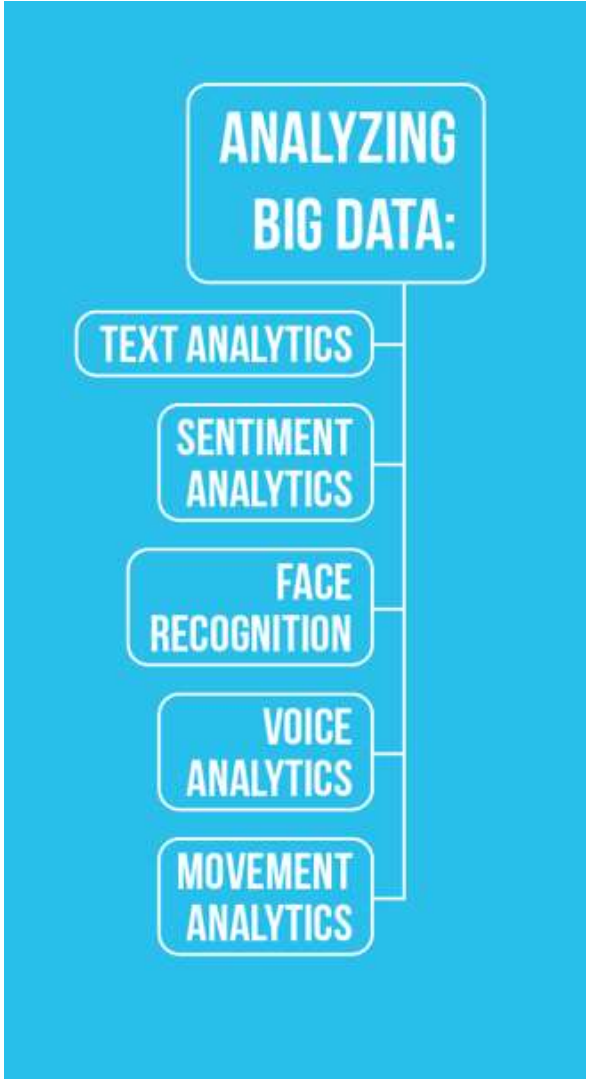
Big Data: Using SMART Big Data, Analytics and Metrics To Make Better Decisions and Improve Performance ***(Bernard Marr)***



Big Data


THE DATAFICATION OF OUR WORLD:
ACTIVITY DATA
Music players, eReaders and smart phones collect data on how we use them; web browsers collect information on what we search for; credit card companies collect data on where we shop; and shops collect data on what we buy.
CONVERSATION DATA
Our conversations are being captured – From emails to all the conversations we have on social media sites like Facebook or Twitter as well as our phone conversations are now digitally recorded.
PHOTO AND VIDEO IMAGE DATA
All the pictures and videos we take on oursmart phones and digital cameras – we upload and share millions of them on social media sites every second.
SENSOR DATA
We are surrounded by sensors that collect and share data – devices like our smart phones use sensors to track our location, the speed and direction at which we are travelling, read our fingerprints, detect how light it is outside, etc.

Big Data


ANALYZING BIG DATA:
TEXT ANALYTICS
SENTIMENT ANALYTICS
FACE RECOGNITION
VOICE ANALYTICS
MOVEMENT ANALYTICS

VOLUME

...refers to the vast amounts of data generated every second - Today, we create the same amount of data in a single minute, that was created from the beginning of time until the year 2000.

VELOCITY

...refers to the speed at which new data is generated and the speed at which data moves around - Today, we perform millions of Internet searches every second, social media messages can go viral in minutes, and credit card transactions are checked in real-time.

VARIETY

...refers to the different types of data we can now use – Today, we don't have to rely on nicely structured data, we can now collect and analyse text, images, video, voice, location data, and much more.

VERACITY

...refers to the messiness or trustworthiness of the data – Today, quality and accuracy of data are less controllable (hash tags, abbreviations, typos and colloquial speech) but technology now allows us to deal with it.

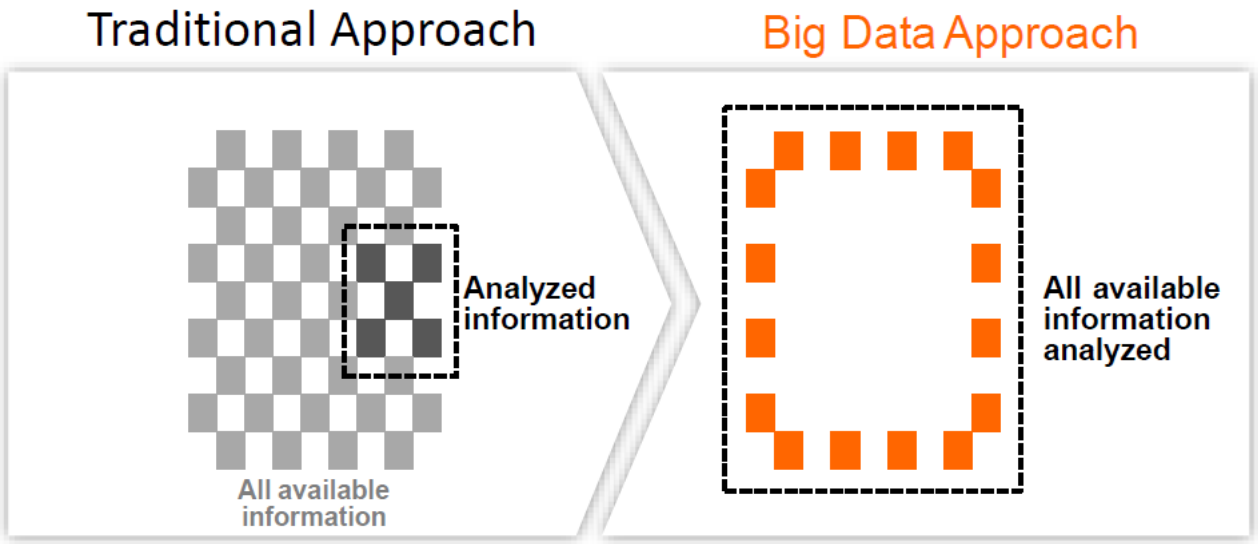
VALUE

...the final V refers to the need to turn our data into value – Today, big data is used to better understand and target customers, understand and optimize business processes, and improve health care, security and law enforcement. But the possible applications of big data are endless!

VOLUME

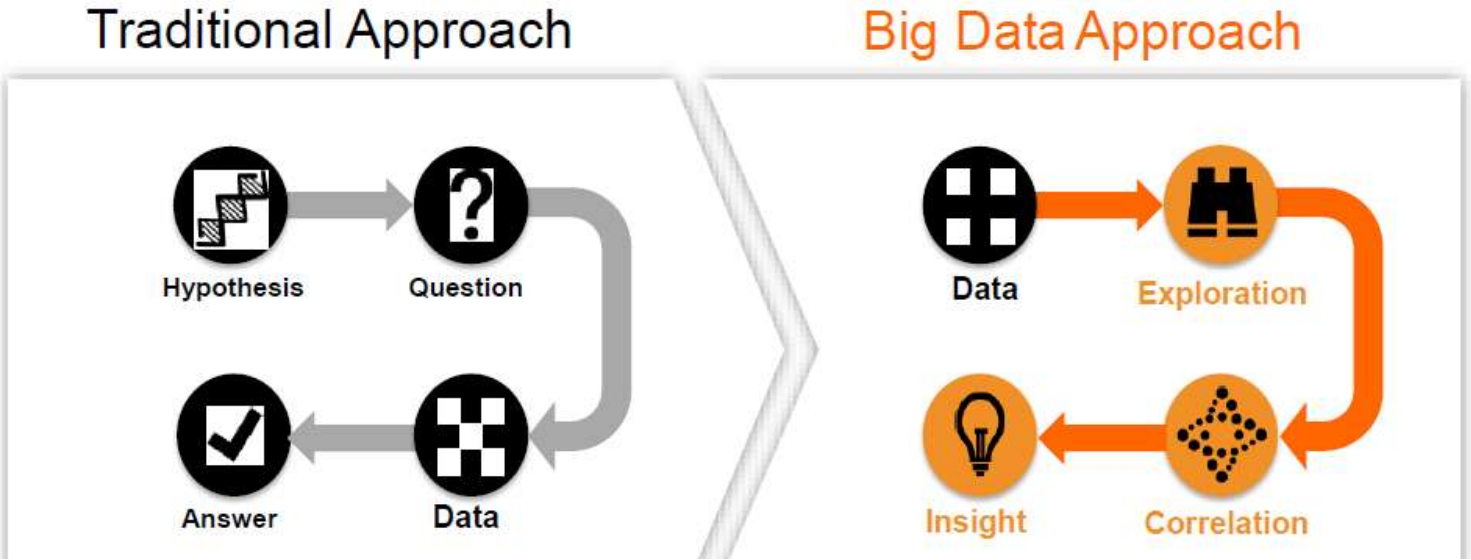
- Facebook menghasilkan 10TB data baru setiap hari, Twitter 7TB
- Sebuah Boeing 737 menghasilkan 240 terabyte data penerbangan selama penerbangan dari satu wilayah bagian AS ke wilayah yang lain
- Microsoft kini memiliki satu juta server, kurang dari Google, tetapi lebih dari Amazon, kata Ballmer (2013).

VOLUME


Traditional Approach
Big Data Approach
Analyzed information
All available information
All available information analyzed

VELOCITY

- Kecepatan data yang masuk (per jam, per detik, etc). *Clickstreams* (*web log*) dan transfer data *asynchronous* yang dapat menangkap apa saja yang dilakukan oleh jutaan atau lebih pengguna yang lakukan saat ini.



VARIETY

- Kumpulan dari berbagai macam data, baik data yang terstruktur, semi terstruktur maupun data tidak terstruktur (bisa dipastikan lebih mendominasi).
- Tampilan data semakin komprehensif (lengkap dan menyeluruh).

VERACITY

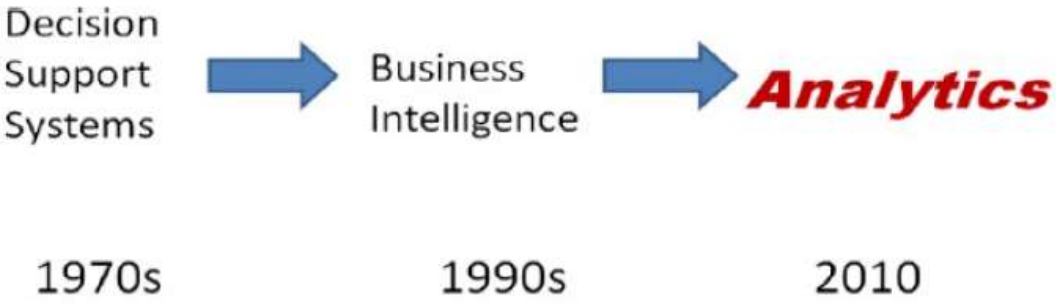
- Ketidakpastian akan data.
- Business process rawan akan kesalahan, tergantung datanya
- Bagaimana suatu data dapat dipercaya mengingat keandalan sumbernya
- Bagaimana mengelola, mengolah data mana yang benar dan mana yang salah

VALUE

- Data yang besar seharusnya berdampak (secara moneter) terhadap suatu perusahaan yang menggunakan komputasi Big Data
- Akan sia-sia bila memiliki data yang sangat besar tapi tidak tahu bagaimana cara mengolah dan menganalisisnya, hanya akan buang-buang *resource*

BIG DATA ANALISIS

- Apa yang dimasud dengan Analytics? Sebuah titik awal untuk memahami Analytics adalah Cara untuk mengeksplorasi/menyelidiki/ memahami secara mendalam suatu objek sampai ke akar-akarnya
- Hasil analytics biasa tidak menyebabkan banyak kebingungan, karena konteksnya biasanya membuat makna yang jelas



Gambar: Dari DSS berkembang menjadi BI kemudian menjadi Analytics.

BUSINESS INTELLIGENCE BIG DATA ANALYTICS

- BI dapat dilihat sebagai istilah umum untuk semua aplikasi yang mendukung DSS, dan bagaimana hal itu ditafsirkan dalam industri dan semakin meluas sampai di kalangan akademisi.
- BI berevolusi dari DSS, dan orang dapat berargumentasi bahwa Analytics berevolusi dari BI (setidaknya dalam hal peristilahan). Dengan demikian, Analytics merupakan istilah umum untuk aplikasi analisis data.
- Big Data Analytics: Alat dan teknik analisis yang akan sangat membantu dalam memahami big data dengan syarat algoritma yang menjadi bagian dari alat-alat ini harus mampu bekerja dengan jumlah besar pada kondisi real-time dan pada data yang berbeda-beda.

CONTOH BIG DATA ANALISIS

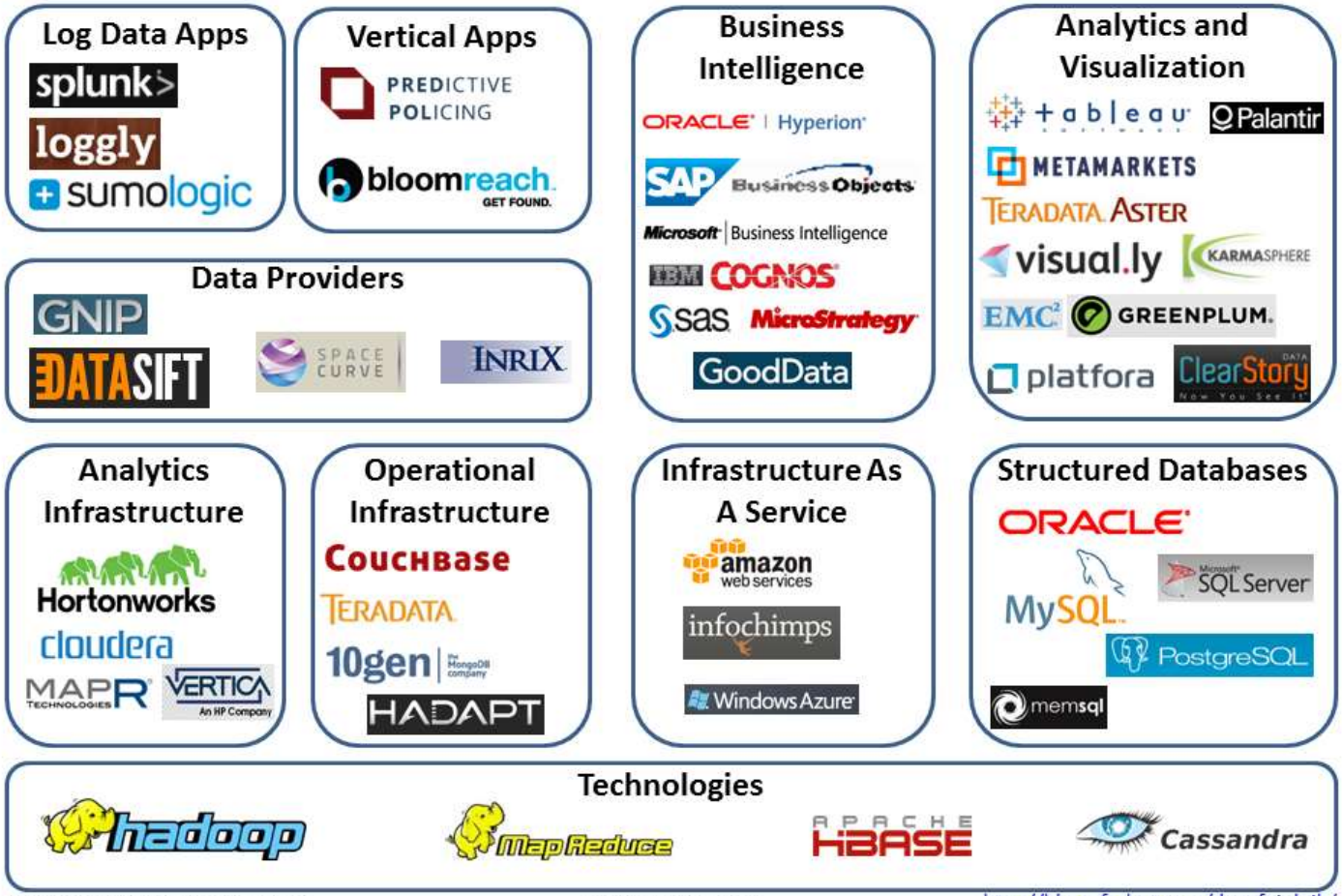
- Contoh perusahaan yang menggunakan analisis Big Data

  Starbucks (Memperkenalkan Produk Coffee Baru). Pagi itu kopi itu mulai dipasarkan, pihak **Starbucks memantau** melalui **blog, Twitter**, dan **kelompok forum diskusi kopi** lainnya untuk menilai reaksi pelanggan. Pada pertengahan-pagi, Starbucks menemukan **hasil dari analisis Big Data** bahwa meskipun orang menyukai rasa kopi tersebut, tetapi mereka berpikir bahwa harga kopi tersebut terlalu mahal. Maka dengan segera pihak Starbucks menurunkan harga, dan menjelang akhir hari semua komentar negatif telah menghilang. Bagaimana jika menggunakan **analisis tradisional**?

- Contoh tersebut menggambarkan penggunaan sumber data yang berbeda dari Big Data dan berbagai jenis analisis yang dapat dilakukan dengan **respon sangat cepat oleh pihak Starbucks**.

KATEGORI TEKNOLOGI BIG DATA

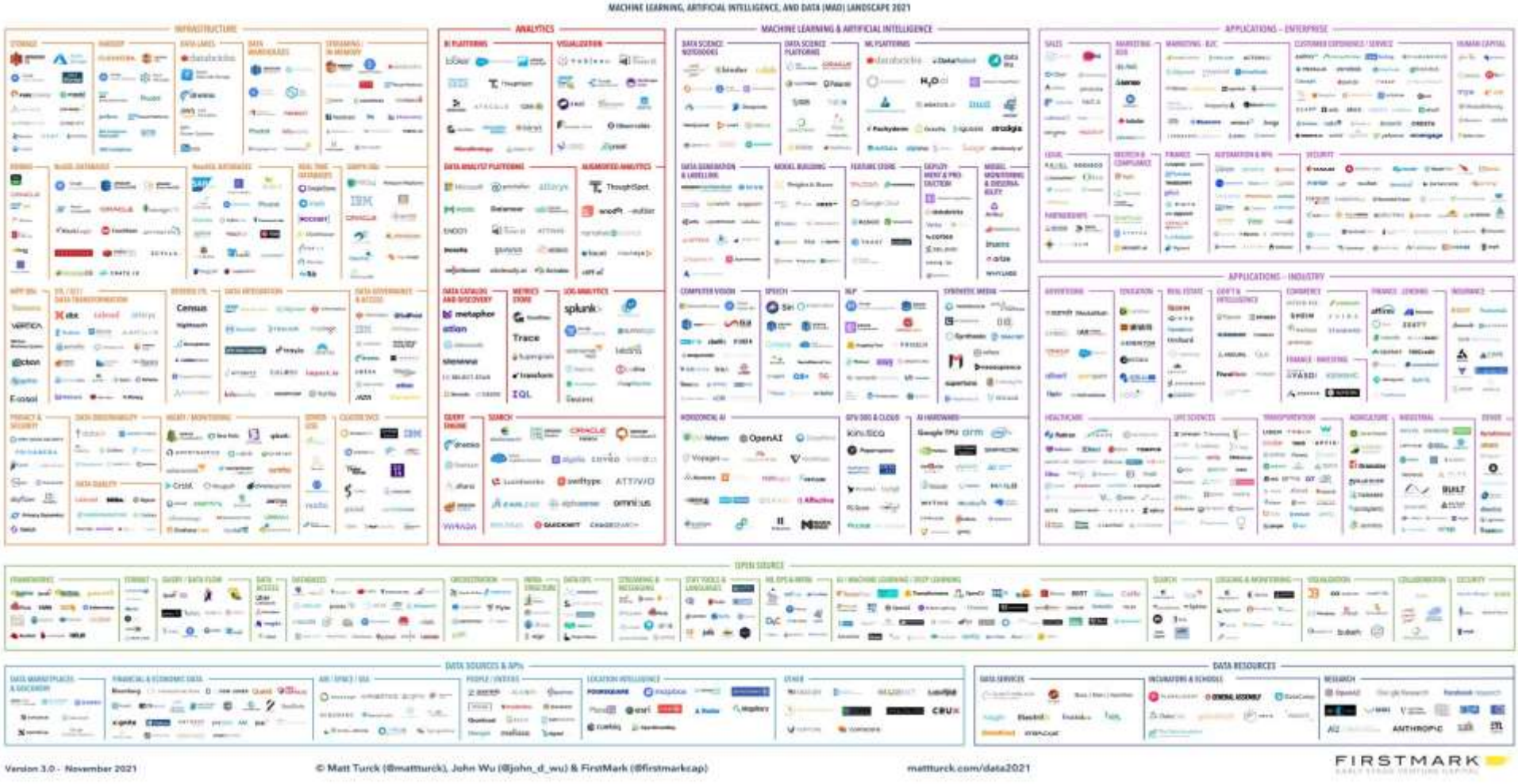
Big Data Landscape



dave@vcdave.com
http://blogs.forbes.com/davefeinleib/

TOOLS & TEKNOLOGI BIG DATA

TOOLS & TEKNOLOGI BIG DATA



DATA CARRIER TRACK

Data Scienctist

Data Analyst

Data Engineer

Data Workflow

What is Data Engineer

- build an effective data architecture
- streamline data processing
- maintain large-scale data systems
- Working with python (combine with other: shell, SQL, Scala, etc)
    - create data engineering pipelines
    - automate common file system tasks
    - build a high-performance database

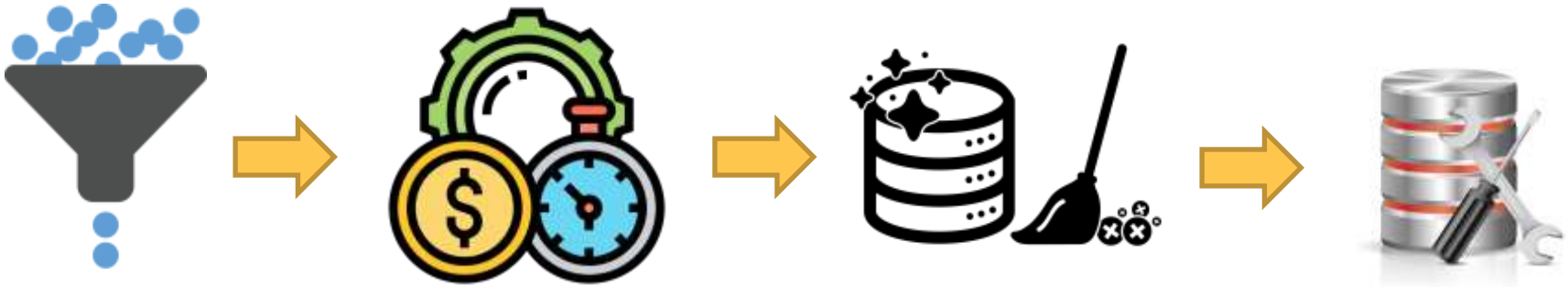
Data Engineer Task

“Responsible for the **first step** of the process:
ingesting **collected data** and **storing it**”

Data Engineer Deliver

Data Engineer Responsibility

- Ingest data from different resources
- Optimized databases for analysis
- Manage/remove corrupted data
- Develop, construct, test, and maintain data architectures



Data Engineer & Big Data

- **Data engineer more and more needed in big data era**
- **Big Data:**
  - **Have to think how to deal with it's size**
  - **So large traditional methods don't work anymore**

Data Analyst Responsible

“import, clean, manipulate, and visualize data”

Data Analyst Task to Define the problem

- **Determine the clients needs**
    - **Dashboard, Reports, Product Analyst**

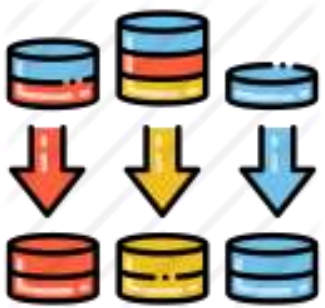
- **Create a plan of action**
- **Communicate the plan to team**

Data Analyst Task Collect the data

- **Data comes from multiple source**
- **Work with programmers to create Extract Transform Load (ETL) process**
- **Aggregate data**

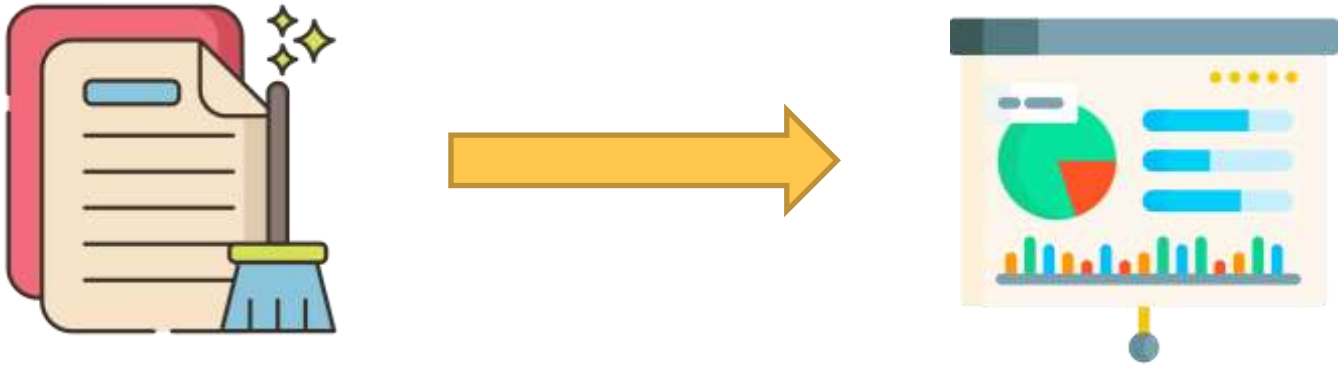
Data Analyst Task Clean the data

- **Data is always messy, clean data makes it more useable**
- **Normalize & standardize data**
- **Data validation**



Data Analyst Task Set up data for report/visualization

- **Create view**
- **Format data chart for specific purpose**
- **Connect data to data visualization tools**
- **Make sure your report/visualization solve the define problem**

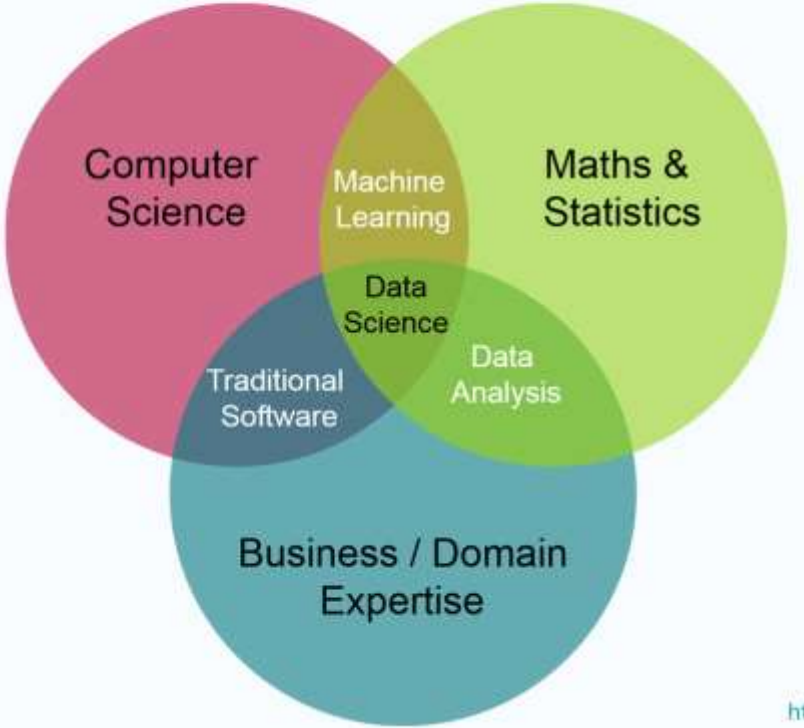


Data Scientist v Data Analyst

- Data Scientist
  - Use current data to discover opportunity
  - Develop analytical method & Machine Learning model
  - Tuning/optimizing hyper parameter

- Data Analyst
  - Use existing data to solve a problem
  - Create report
  - Create dashboard

Drew Conway Venn Diagram



Computer Science

Math & Statistics

Domain Expertise

https://thedatascientist.com/data-science-without-programming/

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$

Mengenal *library* Pandas

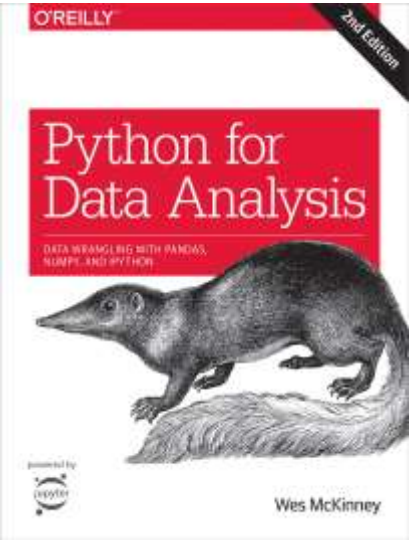
- **Pandas adalah salah satu *library* python yang popular digunakan untuk mengolah data, karena menyediakan *tools* yang powerful dan produktif untuk analisis data**
- **Dapat digunakan untuk mengolah data mentah menjadi lebih terstruktur dan siap untuk dianalisis**
- **Untuk memanipulasi data antara lain: menggabungkan, membandingkan, menangani data yang hilang, menggurutkan data, dll**
- **Mempersiapkan dan membersihkan data**

Kelebihan *library* Pandas

- ***Open source library***
- **Memiliki performa yang sangat baik untuk manipulasi data**
- **Struktur data yang mudah dipahami untuk digunakan**

Sejarah Pandas

- **Pandas diciptakan oleh Wes McKinney pada awal 2008**
- **Awalnya ditujukan untuk memenuhi kebutuhan analisis finansial pada perusahaan AQR Capital Management**
- **Pada tahun 2009, pandas dirilis secara publik dan bersifat *open source***
- **Pandas merupakan akronim dari *"Python Data Analysis Library"*, yang terinspirasi dan diturunkan dari *"Panel Data"* sebuah istilah dari *econometrics* untuk struktur data multidimensi terukur dari waktu ke waktu**



Fitur umum pandas

- **Membuat objek dataframe dari data, untuk manipulasi secara efisien**
- **Memuat dan menulis (read/load, write) ke objek data dalam format yang berbeda**
- **Kemudahan untuk normalisasi dan penanganan data hilang secara terintegrasi**
- **Mengatur ulang bentuk dan poros data sesuai kebutuhan**
- **Melakukan pemotongan, index, membagi data dalam jumlah yang ditentukan**
- **Menambah, menghapus kolom dari struktur dasar data**
- **Membuat data dengan *time series***
- **Menggabungkan beberapa data yang terpisah**
- **Membuat kelompok data yang siap untuk diagregasikan dan diintegrasikan**

Install Pandas

1: CMD Windows

1. Buka CMD (*Command Prompt*)
2. Ketikkan **install python -m pip install -U pandas**
3. Tunggu hingga proses selesai

2: Terminal / prompt

```
conda install pandas
```

OR

```
pip install pandas
```

3: (Jupyter notebook cell)

```
!pip install pandas
```

Tipe struktur dasar data pandas

- **Series**
    - Series adalah array berlabel satu dimensi yang mampu menampung semua tipe data (integer (bilangan bulat), strings, floating point numbers, dan objek Python, dll). Label sumbu disebut sebagai indeks.

- **Dataframe**
    - Data frame merupakan array dua dimensi dengan baris dan kolom. Data frame merupakan tabel/data tabular. Setiap kolom pada Data Frame merupakan objek dari Series, dan baris terdiri dari elemen yang ada pada Series.

Sumber: https://rezadilla30.medium.com/data-manipulation-463f89f5ba0d

Series vs Dataframe

Series 1

	Mango
0	4
1	5
2	6
3	3
4	1

+

Series 2

	Apple
0	5
1	4
2	3
3	0
4	2

+

Series 3

	Banana
0	2
1	3
2	5
3	2
4	7

=

DataFrame

	Mango	Apple	Banana
0	4	5	2
1	5	4	3
2	6	3	5
3	3	0	2
4	1	2	7

Menggunakan Pandas Series (List)

```
import pandas as pd
```

```
# LIST

myList= [45,'halo', 0.76, 'hai', True, 125]

s = pd.Series(myList)
print(s)
```

Menggunakan Pandas Series (List)

```
import pandas as pd
```

```
# LIST

myList= [45,'halo', 0.76, 'hai', True, 125]

s = pd.Series(myList)
print(s)
```

```
0      45
1    halo
2    0.76
3     hai
4    True
5     125
dtype: object
```

Menggunakan Pandas Series (List)

```
# LIST dengan menggunakan rename index

sidx = pd.Series(myList, index=[10, 20, 30, 'D', 'E', 'F'])
sidx
```

Menggunakan Pandas Series (List)

```
# LIST dengan menggunakan rename index

sidx = pd.Series(myList, index=[10, 20, 30, 'D', 'E', 'F'])
sidx
```

```
10      45
20    halo
30    0.76
D      hai
E     True
F      125
dtype: object
```

Apakah yang terjadi jika **jumlah index tidak sama dengan data??**

Menggunakan Pandas Series (Tuple)

```
# TUPLE

myTuple = (87, "belajar", 0.55, False, "bersama")

se = pd.Series(myTuple)
se
```

Menggunakan Pandas Series (Tuple)

```
# TUPLE

myTuple = (87, "belajar", 0.55, False, "bersama")

se = pd.Series(myTuple)
se
```

```
0         87
1    belajar
2       0.55
3      False
4    bersama
dtype: object
```

Menggunakan Pandas Series (Tuple)

```
# TUPLE dengan menggunakan index

seidx = pd.Series(myTuple, index=['e', 'F', 'g', 11, 12])
seidx
```

Menggunakan Pandas Series (Dictionary)

```
# Dictionary

myDc = {'a':1, 'b':2, 'c':3, 'd':'belajar', 'E':False}

ser = pd.Series(myDc)
ser
```

Menggunakan Pandas Series (Dictionary)

```
seridx = pd.Series(myDc, index=['x','y','z', 13, 14])
seridx
```

Apa yang terjadi ketika index digunakan pada dictionary?

Menggunakan Pandas Dataframe (List)

```
# Membuat DataFrame dengan LIST

dataList =[['ayam', 10, 0.67, True],
           ['ikan', 9.76, 34],
           [45, 55, 65],
           [False, True, True, False],
           ['kucing', 'musang']]

df = pd.DataFrame(dataList)
df
```

Menggunakan Pandas Dataframe (List)

```
# Membuat DataFrame dengan LIST

dataList =[['ayam', 10, 0.67, True],
           ['ikan', 9.76, 34],
           [45, 55, 65],
           [False, True, True, False],
           ['kucing', 'musang']]

df = pd.DataFrame(dataList)
df
```

	0	1	2	3
0	ayam	10	0.67	True
1	ikan	9.76	34	None
2	45	55	65	None
3	False	True	True	False
4	kucing	musang	None	None

Menggunakan Pandas Dataframe (List, index)

```
# Membuat DataFrame dengan LIST, index

dataList =[['ayam', 10, 0.67, True],
           ['ikan', 9.76, 34],
           [45, 55, 65],
           [False, True, True, False],
           ['kucing', 'musang']]

dfid= pd.DataFrame(dataList, index= ['nama','usia','berat','tempat','keterangan'])
dfid
```

Menggunakan Pandas Dataframe (List, index)

```
# Membuat DataFrame dengan LIST, index

dataList =[['ayam', 10, 0.67, True],
           ['ikan', 9.76, 34],
           [45, 55, 65],
           [False, True, True, False],
           ['kucing', 'musang']]

dfid= pd.DataFrame(dataList, index= ['nama','usia','berat','tempat','keterangan'])
dfid
```

	0	1	2	3
nama	ayam	10	0.67	True
usia	ikan	9.76	34	None
berat	45	55	65	None
tempat	False	True	True	False
keterangan	kucing	musang	None	None

Menggunakan Pandas Dataframe (List, index)

```
# Membuat DataFrame dengan LIST, index

dataList =[['ayam', 10, 0.67, True],
           ['ikan', 9.76, 34],
           [45, 55, 65],
           [False, True, True, False],
           ['kucing', 'musang']]

dfid= pd.DataFrame(dataList, index= ['nama','usia','berat','tempat','keterangan'])
dfid
```

Bagaimanakah jika dilakukan pemilihan index??

```
dfid[1:3]
```

Menggunakan Pandas Dataframe (List, index, kolom)

```
# Membuat DataFrame dengan LIST, index, kolom

dataList =[['ayam', 10, 0.67, True],
           ['ikan', 9.76, 34],
           [45, 55, 65],
           [False, True, True, False],
           ['kucing', 'musang']]

dfidx= pd.DataFrame(dataList, index= ['10',20, 'angka', 40 ,'tambahan'], columns=['nama','usia','berat','keterangan'])
dfidx
```

Menggunakan Pandas Dataframe (List, index, kolom)

```
# Membuat DataFrame dengan LIST, index, kolom

dataList =[['ayam', 10, 0.67, True],
           ['ikan', 9.76, 34],
           [45, 55, 65],
           [False, True, True, False],
           ['kucing', 'musang']]

dfidx= pd.DataFrame(dataList, index= ['10',20, 'angka', 40 ,'tambahan'], columns=['nama','usia','berat','keterangan'])
dfidx
```

	nama	usia	berat	keterangan
10	ayam	10	0.67	True
20	ikan	9.76	34	None
angka	45	55	65	None
40	False	True	True	False
tambahan	kucing	musang	None	None

Menggunakan Pandas Dataframe (Tuple)

```
# Membuat DataFrame dengan Tuple

dataTuple = (('ayam', 10, 0.67, True),
             ('ikan', 9.76, 34),
             (45, 55, 65),
             (False, True, True, False),
             ['kucing', 'musang'])

dft = pd.DataFrame(dataTuple)
dft
```

	0	1	2	3
0	ayam	10	0.67	True
1	ikan	9.76	34	None
2	45	55	65	None
3	False	True	True	False
4	kucing	musang	None	None

Menggunakan Pandas Dataframe (Dictionary)

```
dataDict = {
    'apel': [3, 2, 0, 1, 5],
    'jeruk': [0, 3, 7, 2, 4],
    'mangga':[3, 6, 2, 4, 3],
    'baju': [7, 0, 1, 2, 1],
    'coklat': [2, 6, 8, 9, 0]
    }


dfDict = pd.DataFrame(dataDict)
dfDict
```

Menggunakan Pandas Dataframe (Dictionary)

```
dataDict = {
    'apel': [3, 2, 0, 1, 5],
    'jeruk': [0, 3, 7, 2, 4],
    'mangga':[3, 6, 2, 4, 3],
    'baju': [7, 0, 1, 2, 1],
    'coklat': [2, 6, 8, 9, 0]
    }


dfDict = pd.DataFrame(dataDict)
dfDict
```

	apel	jeruk	mangga	baju	coklat
0	3	0	3	7	2
1	2	3	6	0	6
2	0	7	2	1	8
3	1	2	4	2	9
4	5	4	3	1	0

Menggunakan Pandas Dataframe (Dictionary, index)

```
# Membuat Dataframe dengan Dictionary, index

dataDict = {
    'apel': [3, 2, 0, 1, 5],
    'jeruk': [0, 3, 7, 2, 4],
    'mangga':[3, 6, 2, 4, 3],
    'baju': [7, 0, 1, 2, 1],
    'coklat': [2, 6, 8, 9, 0]
    }


dfDictr = pd.DataFrame(dataDict, index = ['Irish', 'Franco', 'Dora', 'Alda', 'Bruno'])
dfDictr
```

Menggunakan Pandas Dataframe (Dictionary, index)

```
# Membuat Dataframe dengan Dictionary, index

dataDict = {
    'apel': [3, 2, 0, 1, 5],
    'jeruk': [0, 3, 7, 2, 4],
    'mangga':[3, 6, 2, 4, 3],
    'baju': [7, 0, 1, 2, 1],
    'coklat': [2, 6, 8, 9, 0]
    }


dfDictr = pd.DataFrame(dataDict, index = ['Irish', 'Franco', 'Dora', 'Alda', 'Bruno'])
dfDictr
```

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

Menggunakan Pandas Dataframe (Dictionary, index)

```
pelanggan = ['Irish', 'Franco', 'Dora', 'Alda', 'Bruno']

dfDictri = pd.DataFrame(dataDict, index = pelanggan)
dfDictri
```

Menggunakan Pandas Dataframe (loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
# Melihat Data dengan loc

dfDictri.loc['Dora']
```

Menggunakan Pandas Dataframe (loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
apel      0
jeruk     7
mangga    2
baju      1
coklat    8
Name: Dora, dtype: int64
```

```
# Melihat Data dengan loc

dfDictri.loc['Dora']
```

Menggunakan Pandas Dataframe (loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
apel      1
jeruk     2
mangga    4
baju      2
coklat    9
Name: Alda, dtype: int64
```

```
#Melihat data dengan iloc

dfDictr.iloc[3]
```

Menggunakan Pandas Dataframe (kolom loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
Irish     3
Franco    6
Dora      2
Alda      4
Bruno     3
Name: mangga, dtype: int64
```

```
# Melihat Data kolom dengan loc

dfDictri.loc[:,'mangga']
```

Menggunakan Pandas Dataframe (kolom loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
#Melihat Data dengan iloc

dfDictr.iloc[:,0]
```

Menggunakan Pandas Dataframe (kolom loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
Irish     3
Franco    2
Dora      0
Alda      1
Bruno     5
Name: apel, dtype: int64
```

```
#Melihat Data dengan iloc

dfDictr.iloc[:,0]
```

Menggunakan Pandas Dataframe (kolom loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
# Memilih kolom dengan loc

dfDictri.loc[:,['coklat','mangga','apel',]]
```

Menggunakan Pandas Dataframe (kolom loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
# Memilih kolom dengan loc

dfDictri.loc[:,['coklat','mangga','apel',]]
```

	coklat	mangga	apel
Irish	2	3	3
Franco	6	6	2
Dora	8	2	0
Alda	9	4	1
Bruno	0	3	5

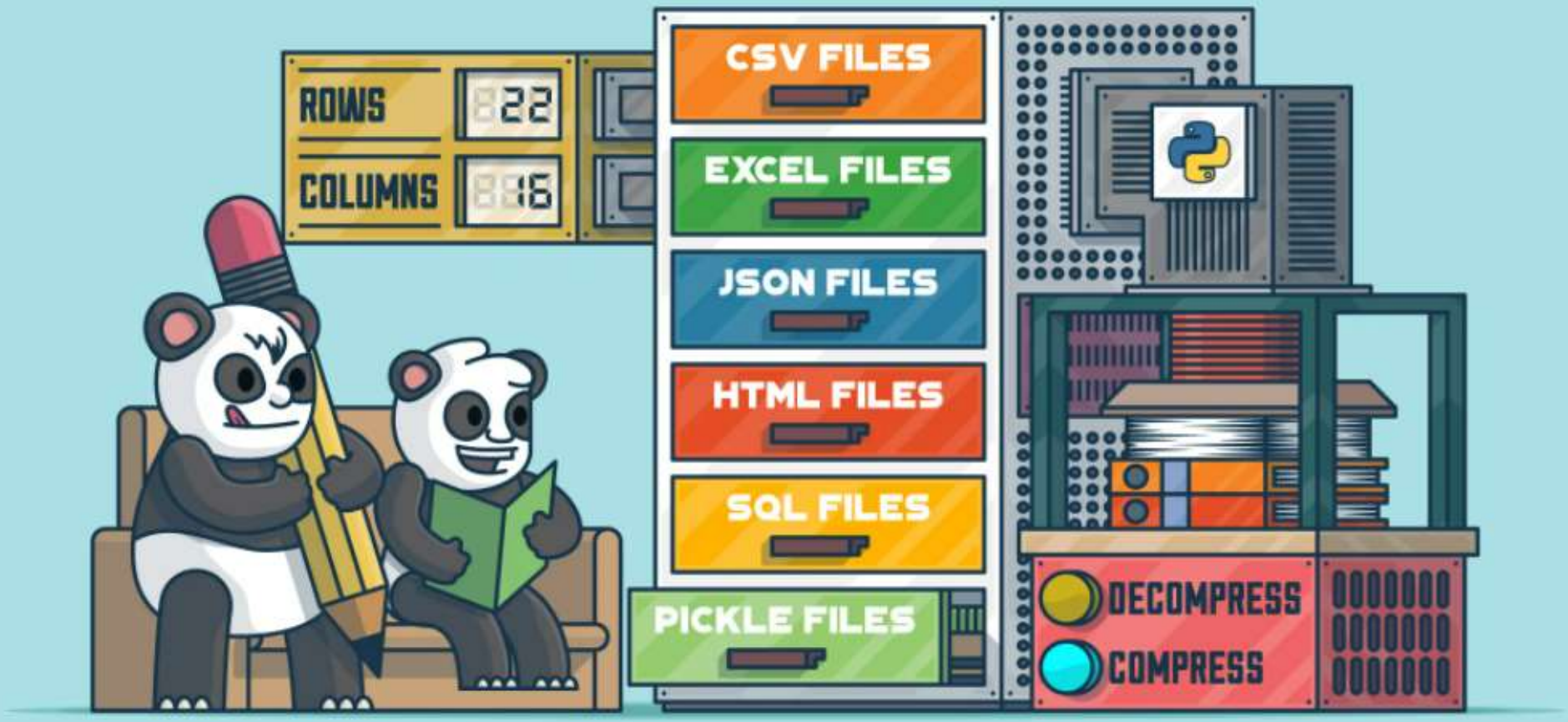
Menggunakan Pandas Dataframe (kolom loc dan iloc)

	apel	jeruk	mangga	baju	coklat
Irish	3	0	3	7	2
Franco	2	3	6	0	6
Dora	0	7	2	1	8
Alda	1	2	4	2	9
Bruno	5	4	3	1	0

```
# Memilih kolom dengan iloc

dfDictr.iloc[:,[4,1,0]]
```

	coklat	jeruk	apel
Irish	2	0	3
Franco	6	3	2
Dora	8	7	0
Alda	9	2	1
Bruno	0	4	5



PANDAS READ-WRITE DATA

Real Python

Read Local Data File

```
import pandas as pd
```

```
dfMovie = pd.read_csv('C:/imdb.csv')
dfMovie
```

Read From Collab Google Drive Data File

```
from google.colab import drive
drive.mount('/content/drive/')
```

```
# Read csv, Excel Colab

dfMovie = pd.read_csv('/content/drive/MyDrive/DTSProA/imdb.csv')
dfMovie
```

Write File Local

```
dfNew = dfMovie[(dfMovie['Year'] >= 2010) & (dfMovie['Genre'].str.contains('Comedy')) & (dfMovie['Rating'] >= 7.5) & (dfMovie['Me
dfNew= dfNew.to_csv('C:/LATIHAN PYTHON/newData.csv')
#dfNewxl= dfNew.to_excel('C:/LATIHAN PYTHON/newData1.xlsx')
```

Write File Google Collab

```
dfNew = dfMovie[(dfMovie['Year'] >= 2010) & (dfMovie['Genre'].str.contains('Comedy')) & (dfMovie['Rating'] >= 7.5) & (dfMovie['Metascore'] >= 80)]
dfNew= dfNew.to_csv('/content/drive/MyDrive/DTSProA/newData.csv')
#dfNewxl= dfNew.to_excel('/content/drive/MyDrive/DTSProA/newData1.xlsx')
```

Introduction DataFrames Manipulation

- **Data manipulation with pandas**
  - **Print(data)**
  - **Data.head()**
  - **Data.info()**
  - **Data.shape**
  - **Data.describe()**
  - **Data.values**
  - **Data.columns**

Join Data with Pandas

- **Select Condition**
- **Group by**
- **Statistic with Pandas**
- **Pivot**
- **Add Column**
- **Index; loc, iloc**
- **Visualization**