PJJ Data Analytics 2026

Data Preparation

Riki Akbar
Ibrahim Saleh Siregar

Kenalan Dulu

Riki Akbar
DJSEF

Ibrahim Saleh Siregar
DJP

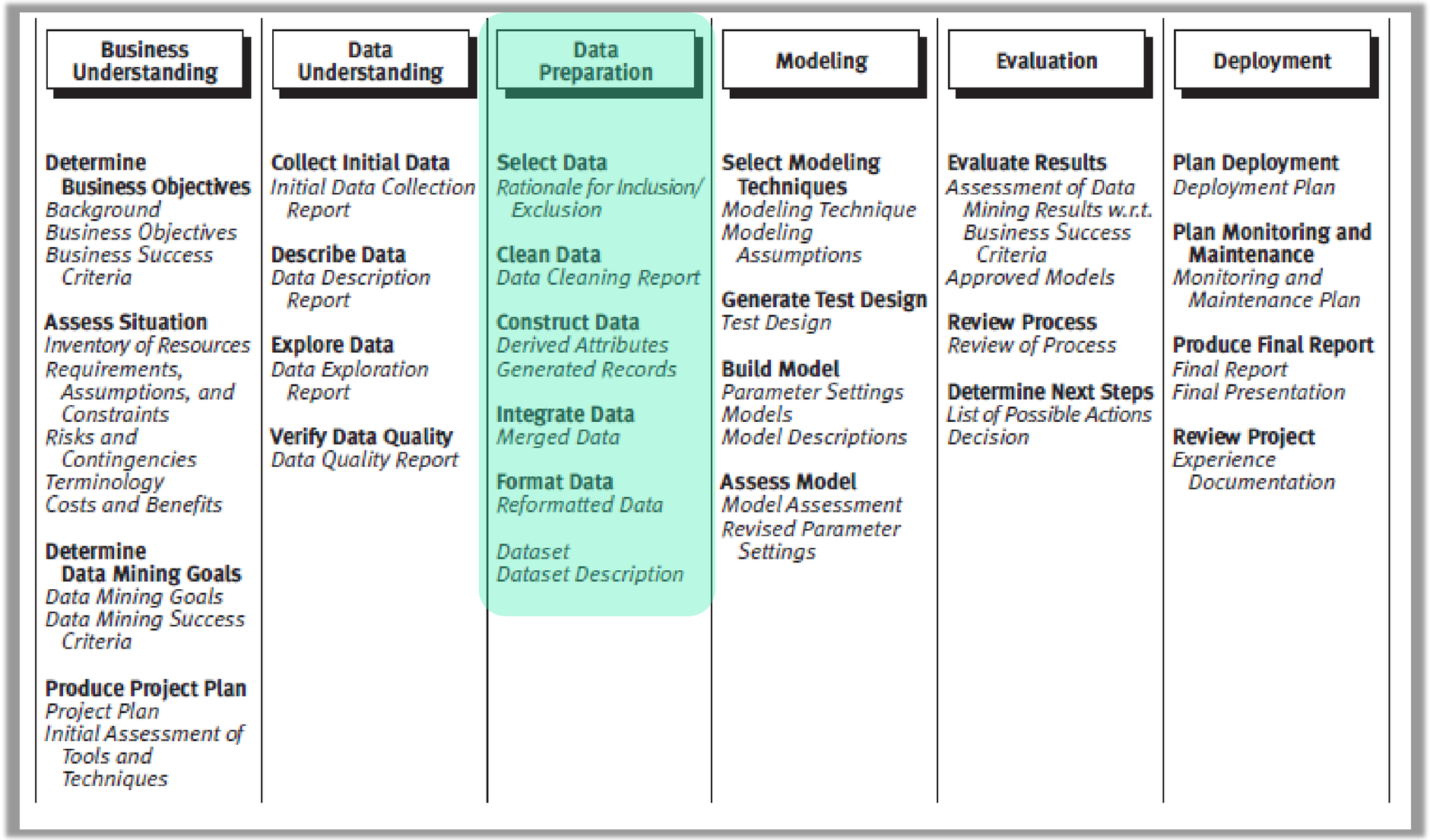
Materi dapat diakses di: https://github.com/IbrahimSalehS/PJJ-Data-Analytics-2026

Sebelum Lanjut, Kita Pop Quiz Dulu

Join at menti.com | use code 8460 9642

Data Preparation

Ingat CRISP-DM?

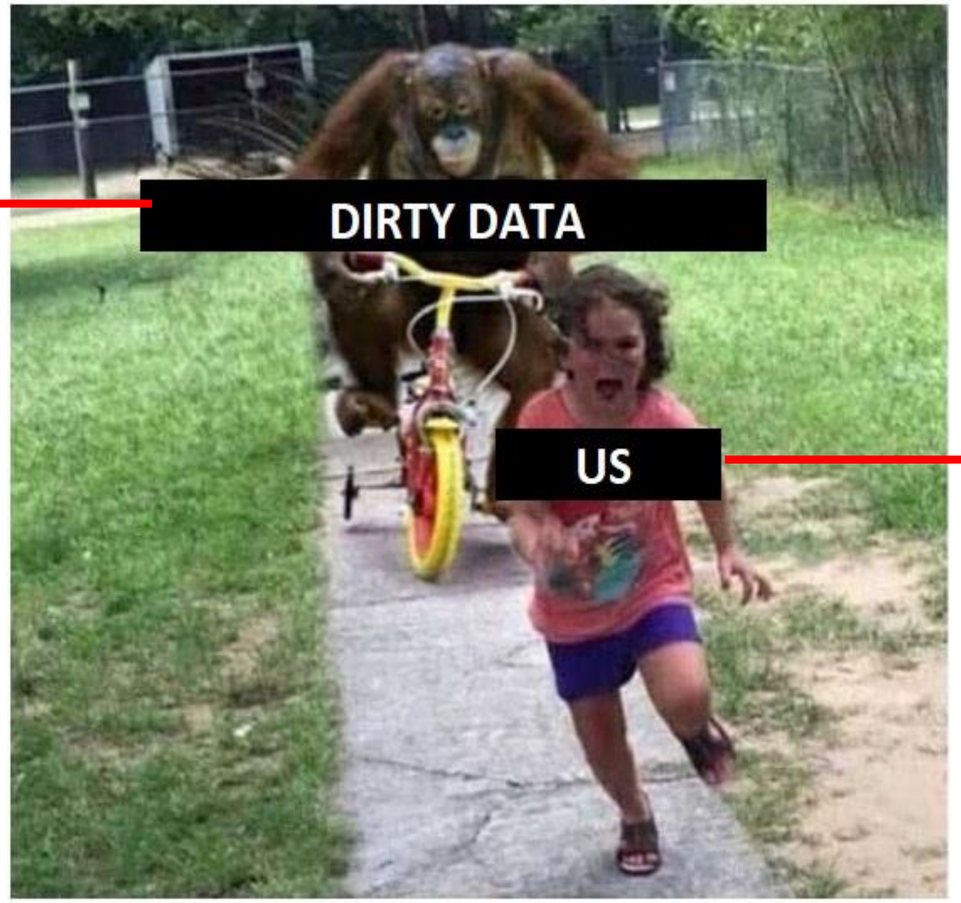


Kenapa Data Harus di-*prepare?*

Data tidak selalu 100% berkualitas

Data Quality Issues

- Ada beberapa atribut yang nilainya kosong (**missing values**)
- Ada beberapa baris data yang duplikat (**duplicates**)
- Beberapa nilai atribut data terlihat seperti anomali (**outliers & noises**)
- Beberapa atribut bisa jadi diformat secara tidak tepat (**formatting errors**)



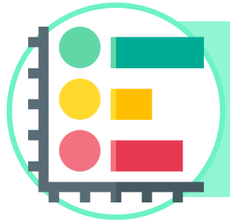
Analysis Perspective

- Ingat **Garbage-in, Garbage-out**
- Pada tahap analisis dan pengembangan model, kita (boleh jadi) **membutuhkan lebih banyak atribut** (turunan)
- Atau bahkan, beberapa **atribut malah tidak relevan** dalam analisis

So, Apa Target Mata Pelatihan Data Preparation?

Mampu mengenali Data Quality Issues

Mampu melakukan transformasi data numerik



Mampu melakukan transformasi data kategorikal

Mampu melakukan transformasi data tekstual

Data: Berdasarkan Bentuknya

Unstructured data

The university has 5600 students.
John's ID is number 1, he is 18 years old and already holds a B.Sc. degree.
David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

- Tidak ada struktur
- Teks, Audio, File biner, dll

Semi-structured data

```
<University>
  <Student ID="1">
    <Name>John</Name>
    <Age>18</Age>
    <Degree>B.Sc.</Degree>
  </Student>
  <Student ID="2">
    <Name>David</Name>
    <Age>31</Age>
    <Degree>Ph.D.</Degree>
  </Student>
  ....
</University>
```

Key

Value

- Diorganisasikan dengan mekanisme key-value
- Lebih Fleksibel
- JSON, XML, etc

Structured data

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Baris

Kolom/Atribut /Fitur

- Diorganisasikan melalui baris-kolom
- Rigid, cocok untuk tujuan penyimpanan data
- RDBMS e.g., MySQL, Postgre, etc

Cardoso, Jorge. (2005). Developing Dynamic Packaging Applications Using Semantic Web-Based Integration. 10.4018/978-1-59904-192-6.ch001.

Data: Berdasarkan Tipenya

Kategorikal

Mementingkan Urutan

Ordinal

Contoh: Nilai Ujian
A, A-, B, B-

Tidak Mementingkan Urutan

Nominal

Contoh: Jenis Kelamin
Laki-laki, Perempuan

Numerik

Bilangan desimal, tidak-bulat

Kontinu

Contoh: Jarak
27,3 KM, 250,5 Meter

Bilangan Bulat

Diskret

Contoh: Jumlah Pegawai
51 orang

Data Quality Issues

1. Duplicates
2. Outliers & Noises
3. Missing Values
4. Formatting Errors

Data Quality Issues: Duplicates

Apakah baris data ini dapat kita golongkan sebagai baris data duplikat?

```
1 df[df['name'].duplicated(keep=False)].sort_values("name", ascending=True)
```

	name	rating	genre	year	released	score
283	A Nightmare on Elm Street	R	Horror	1984	November 16, 1984 (United States)	7.5
3920	A Nightmare on Elm Street	R	Crime	2010	April 30, 2010 (United States)	5.2
5334	Aladdin	PG	Adventure	2019	May 24, 2019 (United States)	6.9
1171	Aladdin	G	Animation	1992	November 25, 1992 (United States)	8.0

Mengapa kita perlu menghapus baris data duplikat?

Mengidentifikasi lebih dari satu baris data identik berdasarkan kriteria duplikasi yang ditentukan

Hapus duplikat

```
df.drop_duplicates(subset=['name','year'], inplace=True)
```

Data Quality Issues: Outliers and Noises

Seringkali kita kesulitan membedakan noises dengan outliers

Jadi, yang mana outliers? yang mana noises?

Student Name	Grade
Mark	98
Ruffalo	4
Denzel	87
Washington	64
Benedict	75
Cumberbatch	81

Student Name	Grade
Mark	98
Ruffalo	Cat
Denzel	87
Washington	64
Benedict	75
Cumberbatch	81

Outliers

Baris data yang mengandung nilai ekstrem.

Noises

"Unwanted variability" (Kahneman et al.)

Baris data yang tidak seharusnya berada dalam kelompok/distribusi data/dataset

Data Quality Issues: Outliers and Noises

Diskusi

Diketahui nilai yang diberikan kepada para peserta pelatihan harus berada dalam interval 70-90

Student Name	Grade
Mark	98
Ruffalo	4
Denzel	87
Washington	64
Benedict	75
Cumberbatch	81

Outlier? atau Noise?

Outlier? atau Noise?

Data Quality Issues: Mengidentifikasi Outliers

Z-score
Mengukur sejauh apa sebuah data point terdeviasi dari rata-rata kelompok data

$$Z = \frac{x - mean}{std}$$

Outliers menggunakan Z-score
Outliers jika:
Z-score > 3 **atau**
Z-score < -3

Jarak Antar-Kuartil/Inter-Quartile Range (IQR)
Mengukur sebaran data

$$IQR = Q_3 - Q_1$$

Outliers menggunakan IQR
Outliers jika:
$x > Q_3 + 1.5*IQR$ **atau**
$x < Q_1 - 1.5*IQR$

Data Quality Issues : How to Handle Outliers?

Statistik hanya memberikan sinyal, bukan keputusan.

Keputusan untuk menghapus atau tidak menghapus outlier **sangat bergantung pada Business Understanding dan Data Understanding,** nilai ekstrem bisa jadi sampah (*noise*) atau justru "emas" (*golden nugget/anomaly*) bagi bisnis.

- Tindakan untuk *Noises*
  - *Removal*
  - *Imputation*

- Tindakan untuk *Outliers*
  - *Keep & Analyze (*jangan dibuang)
  - Transformasi Data
  - Gunakan Model yang robust
  - Removal (jika kita memang yakin outliers tidak relevan dengan tujuan analisis)

Sumber : Handling Outliers in ML: Best Practices for Robust Data Preprocessing

Data Quality Issues: Missing Values

```
1 df[df.rating.isna()]
```

	name	rating	genre
279	Nausicaä of the Valley of the Wind	NaN	Animation
1031	Y Tu Mamá También	NaN	Drama
1198	Madadayo	NaN	Drama
1229	Return of the Living Dead III	NaN	Horror
1259	Jason Goes to Hell: the Final Friday	NaN	Fantasy
1787	Happy Together	NaN	Drama
2175	Eyes Wide Shut	NaN	Drama
2246	Brother	NaN	Crime

Baris data dengan satu atau lebih kolom yang memuat missing value biasanya ditandai dengan nilai NaN atau NULL

How to handle

Deletion/Penghapusan

- Hapus **semua records/baris data** yang mengandung missing values
- Hapus **semua kolom** yang mengandung missing values
- **Hati-hati!** Menghapus terlalu banyak baris/kolom yang mengandung missing values tanpa alas an yang kuat dapat mengakibatkan **information loss**

Imputation/Pengisian Nilai

- Mengganti missing values **dengan rata-rata (mean), nilai tengah (median), dan atau modus (mode)** untuk atribut **numerik**
  - Hati-hati terhadap potensi bias
  - Ingat, nilai rata-rata bersifat sensitif terhadap keberadaan outliers
- Mengganti missing values dengan **modus** atau **kategori "missing"** untuk atribut **kategorikal**
- Mengganti missing values dengan Teknik **Multiple Imputation By Chained Equation (MICE)**

Data Quality Issues: Formatting Errors

Mengidentifikasi kesalahan format pada dataset

Email	Jenis Transaksi	Nominal	Tanggal
riki_akbar_(at)_gmail.com	Pembayaran Kartu Kredit	2,000,000	23/12/2023
riki_akbar@gmail.com	Transaksi Marketplace	587,000	23/12/2023
riki_akbar@gmail.com	Transaksi Marketplace	238,400	23/12/2023
riki_akbar@gmail.com	Transaksi Marketplace	127,882	23/12/2023
riki_akbar@gmail.com	Pembayaran KPR	6,473,122	23/12/2023
riki_akbar@gmail.com	Transaksi Amazon UK	87	2023/12/23

How to handle?

Ubah baris/kolom data yang mengandung kesalahan format berdasarkan standar format yang ditetapkan

Contoh: format tanggal, email, mata uang, dll

Feature Engineering

Misal, kita akan membuat sebuah model prediktif yang memprediksi penerimaan pajak komoditas tertentu

$$f(x_1, x_2, \dots, x_n) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$

x_1: Fitur — Kode KPP → Cluster KPP

x_2: Fitur — Jumlah Komoditas → Vol. Transaksi Periodik

x_n: Fitur — Jenis Pajak

Pada beberapa kasus kita perlu mentransformasi fitur yang ada menjadi fitur baru yang lebih *useful*

Transformasi Data Numerik: Scaling & Standardisation

Tujuan: mengubah nilai pada atribut/kolom tertentu menjadi nilai baru dalam skala yang sama

Scaling

Mengubah interval nilai pada atribut/kolom tetapi menjaga agar distribusi nilai pada atribut/kolom tersebut tetap sama

```
1 MinMaxScaler().fit_transform(df[['budget']])[:5]
```

```
array([[0.03369158],
       [0.04492772],
       [0.00756754],
       [0.03930965],
       [0.03369158]])
```

Scale down sehingga interval nilainya menjadi antara 0 dan 1

Data Awal

```
2 np.array(df[['budget']])[:5]
```

```
array([[12000000],
       [16000000],
       [ 2700000],
       [14000000],
       [12000000]])
```

Standardization

Mengubah internval nilai pada atribut/kolom sehingga standar deviasi pada distribusi tersebut bernilai 1

```
1 StandardScaler().fit_transform(df[['budget']])[:5]
```

```
array([[-0.5826946 ],
       [-0.48678786],
       [-0.80567777],
       [-0.53474123],
       [-0.5826946 ]])
```

```
(1.0, 8.484092367284778e-17)
```

Scale down sehingga standar deviasinya bernilai 1

Transformasi Data Numerik: Scaling & Standardisation

DISKUSI

1. Kapan menggunakan teknik Scaling/Standardisation?
2. Bagaimana mengaplikasikan Scaling/Standardisation pada Data Uji/Data Baru?

Transformasi Data Kategorikal: Encoding

"From Categories to Numbers"

Mengapa Encoding Diperlukan?

1. Pada dasarnya, model machine learning adalah model matematis yang hanya dapat memroses representasi numerik dari sebuah data
2. Oleh karena itu, kita perlu mentransformasi seluruh nilai pada atribut/kolom kategorikal menjadi numerik → encoding

Teknik-Teknik Encoding

1. One-Hot Encoding
2. Label Encoding

4 Categorical Encoding Concepts to Know for Data Scientists | by Cornellius Yudha Wijaya | Towards Data Science

Encoding: One-Hot Encoding

Original categorical column

Origin
USA
Japan
Europe
USA
Europe

One-Hot encoded columns

Origin_USA	Origin_Japan	Origin_Europe
1	0	0
0	1	0
0	0	1
1	0	0
0	0	1

Pros

- Cocok untuk memroses kategori **nominal**

Cons

- Rentan terhadap **sparsity**
- Untuk kategori yang memiliki kardinalitas tinggi, dapat berujung pada **Curse of Dimensionality**

4 Categorical Encoding Concepts to Know for Data Scientists | by Cornellius Yudha Wijaya | Towards Data Science

Encoding: Label Encoding

Original categorical column

Education
High School
Primary School
Master Degree
Bachelor Degree
High School

Label encoded column

Education
2
1
4
3
2

Pros

- Cocok untuk memroses **kategori ordinal,** model dapat mengidentifikasi pola urutan bermakna yang berguna pada saat mengembangkan model prediktif

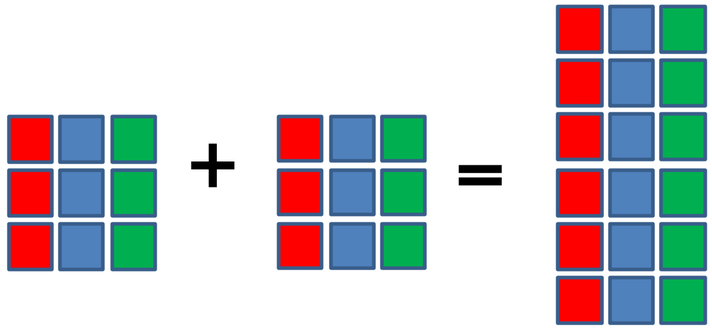
Cons

- Apabila label encoding digunakan pada atribut kategori non-ordinal, model akan keliru menginterpretasi bahwa pola urutan pada atribut/kolom tersebut memiliki makna sehingga berujung pada hasil prediksi yang tidak tepat

4 Categorical Encoding Concepts to Know for Data Scientists | by Cornellius Yudha Wijaya | Towards Data Science

Data Enrichment: Concatenation & Merging (Join)

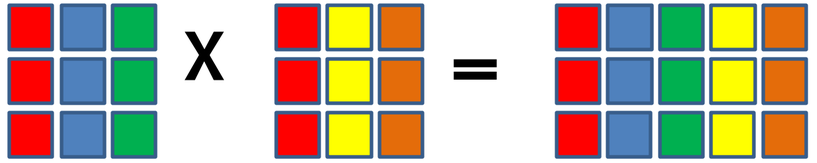
Concatenation



```
df1 = df.iloc[43:45]
df2 = df.iloc[50:52]
pd.concat([df1, df2])
```

Menambah **baris data**

Merging

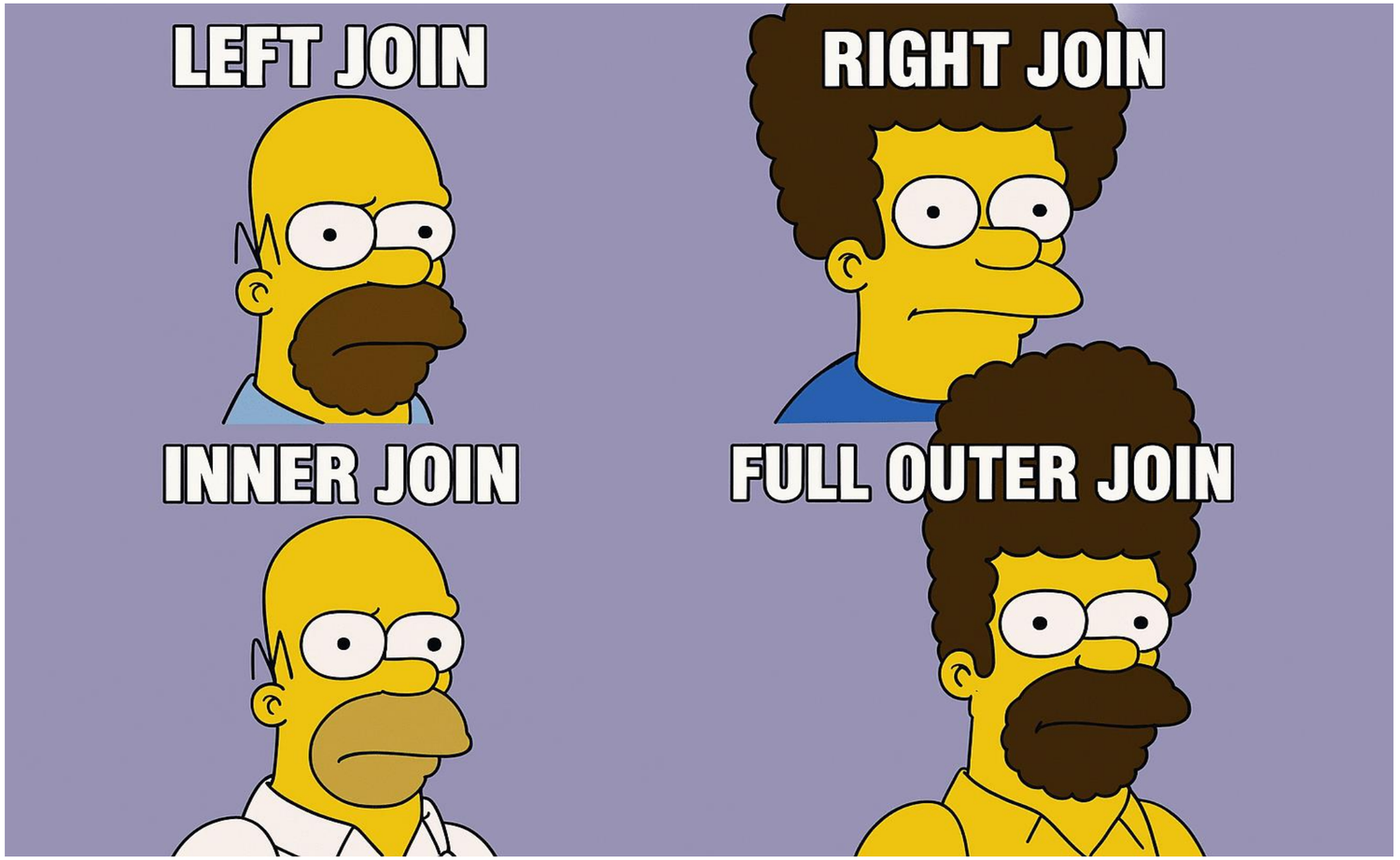


```
df1 = df[['name','rating','genre','year','director','writer','country']]
df2 = df[['name','rating','genre','year','budget','gross']]
pd.merge(df1, df2,
         how="inner",
         right_on=['name','rating','genre','year'],
         left_on=['name','rating','genre','year'])
```

Menambah **kolom data**

Tipe-Tipe Join

All Left,
Matched-Only Right

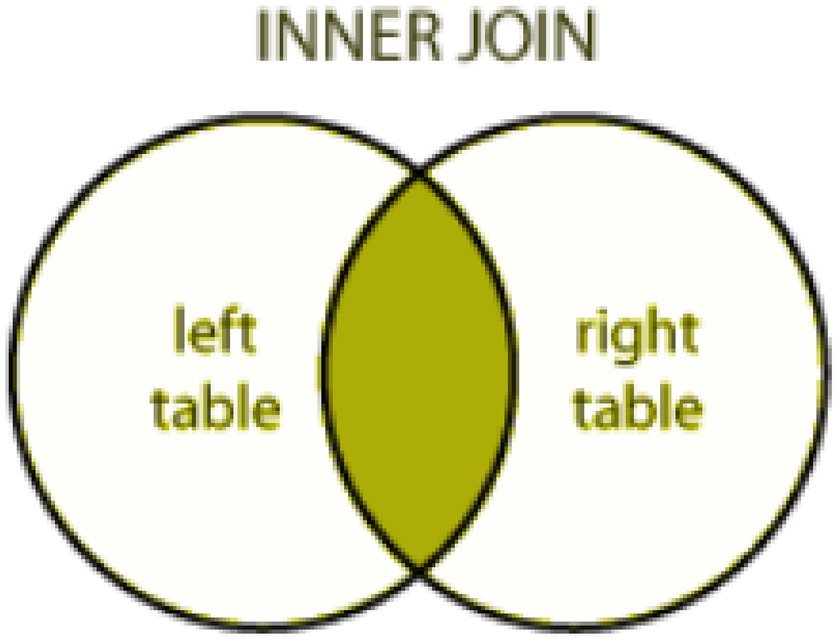


All Right,
Matched-Only Left

All-Matched

All Possible
Combination

Tipe-Tipe Join: Inner Join



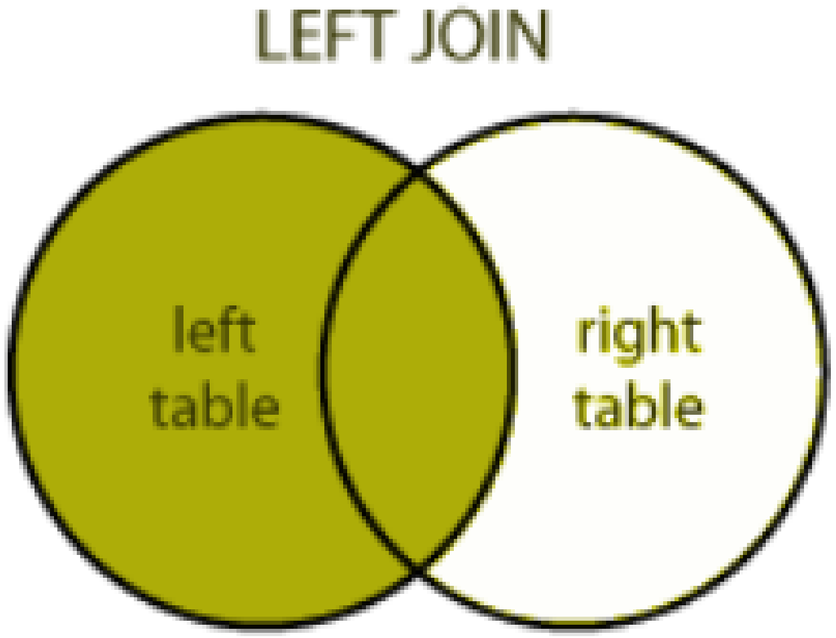
Tabel A		
id	nama	kelas
1	adi	1
2	budi	2
3	charles	1
5	denny	3

X

Tabel B		
id	nama	kelas
2	budi	2
3	charles	1
5	denny	3
7	gerry	1

id	nama	kelas	nama	kelas
2	budi	2	charles	1
3	charles	1	denny	3
5	denny	3	gerry	1

Tipe-Tipe Join: Left Join



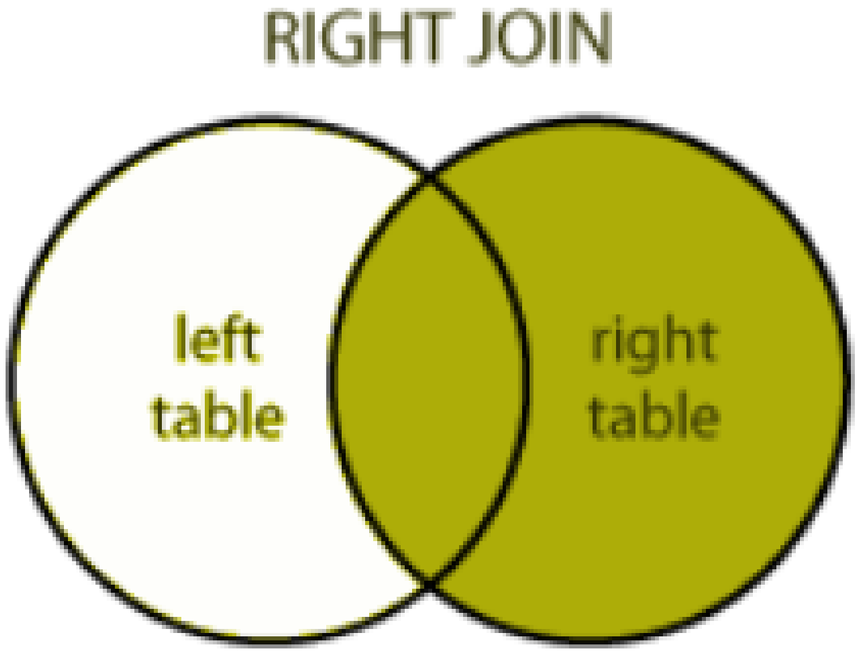
Tabel A		
id	nama	kelas
1	adi	1
2	budi	2
3	charles	1
5	denny	3

X

Tabel B		
id	nama	kelas
2	budi	2
3	charles	1
5	denny	3
7	gerry	1

id	nama	kelas	nama	kelas
1	adi	1	*NULL*	*NULL*
2	budi	2	budi	2
3	charles	1	charles	1
5	denny	3	denny	3

Tipe-Tipe Join: Right Join



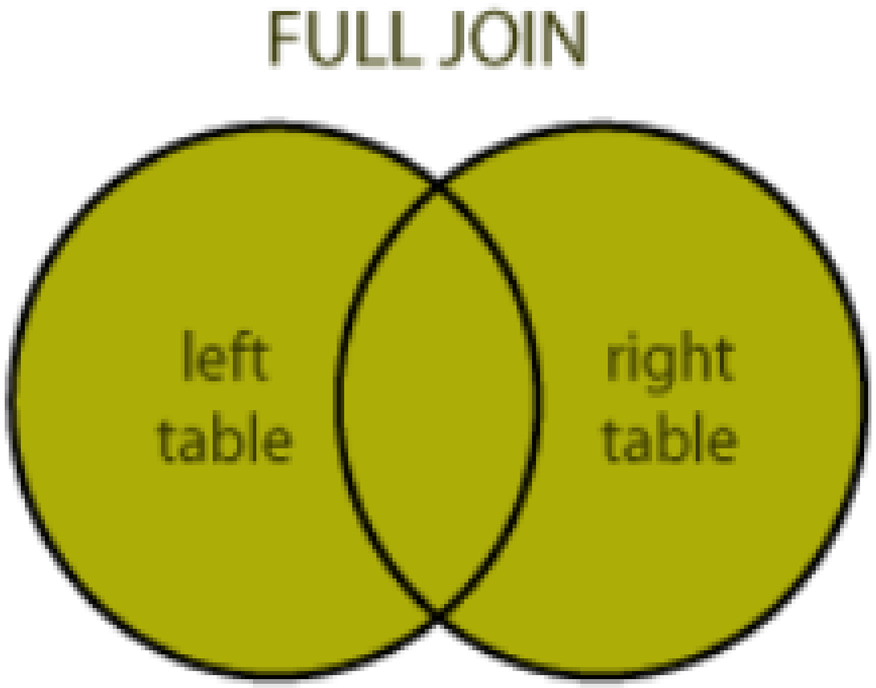
Tabel A		
id	nama	kelas
1	adi	1
2	budi	2
3	charles	1
5	denny	3

X

Tabel B		
id	nama	kelas
2	budi	2
3	charles	1
5	denny	3
7	gerry	1

id	nama	kelas	nama	kelas
2	budi	2	budi	2
3	charles	1	charles	1
5	denny	3	denny	3
7	*NULL*	*NULL*	gerry	1

Tipe-Tipe Join: Full (Outer) Join



Tabel A		
id	nama	kelas
1	adi	1
2	budi	2
3	charles	1
5	denny	3

X

Tabel B		
id	nama	kelas
2	budi	2
3	charles	1
5	denny	3
7	gerry	1

id	nama	kelas	nama	kelas
1	adi	1	*NULL*	*NULL*
2	budi	2	budi	2
3	charles	1	charles	1
5	denny	3	denny	3
7	*NULL*	*NULL*	gerry	1

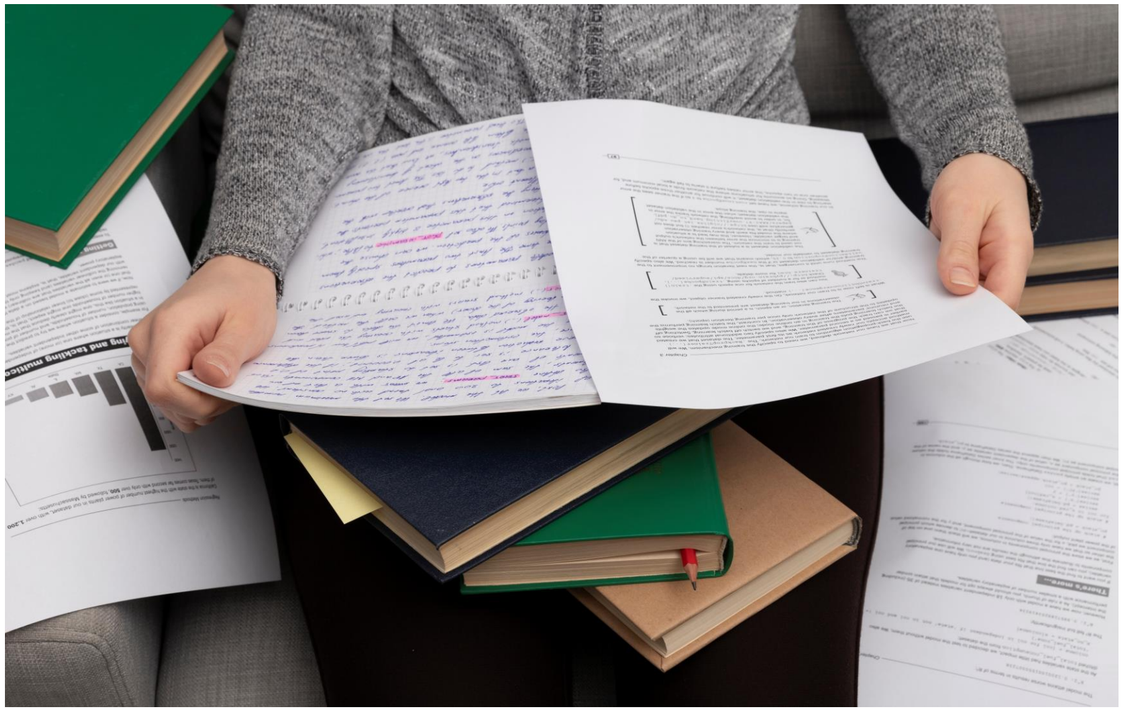
Hands-on Time!

Buka Google Colab

1. Menangani Data Quality Issues
2. Data Transformation
3. Concatenation & Merge (or Join)

Data Tekstual

Berita



Laporan

Mengapa Data Tekstual Perlu Data Preparation?

Karena data tekstual memiliki karakteristik tidak terstruktur (unstructured), kompleks (complex), dan noisy

“

Saya baru aja pulang dari menginap di Hotel X. Hotel ini benar-benar melebihi ekspektasi sayaaa ! Fasilitasnya gila lengkap memuaskan banget, kamarnya bersih-nyaman, serta pelayanannya sangat ramah. Staf hotel sangat membantu dan selalu siap memenuhi kebutuhan saya

”

Mengapa Data Tekstual Perlu Data Preparation?

Karena data tekstual memiliki karakteristik tidak terstruktur (unstructured), kompleks (complex), dan noisy

slang

noise

dua token yang berbeda

imbuhan

*Saya baru **aja** pulang dari menginap di **Hotel** X. Hotel ini benar-benar melebihi ekspektasi **sayaaa !** Fasilitasnya gila lengkap **memuaskan** banget, kamarnya bersih-nyaman, serta pelayanannya sangat ramah. Staf **hotel** sangat membantu dan selalu siap memenuhi kebutuhan saya*

Uppercasing, Lowercasing, dan Penghilangan Tanda Baca

“Fasilitasnya gila lengkap memuaskan banget, kamarnya bersih-nyaman, serta pelayanannya sangat ramah”

lowercasing → fasilitasnya gila lengkap memuaskan banget, kamarnya bersih-nyaman, serta pelayanannya sangat ramah

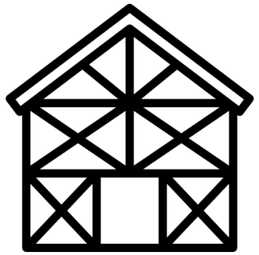
UPPERCASING → FASILITASNYA GILA LENGKAP MEMUASKAN BANGET, KAMARNYA BERSIH-NYAMAN, SERTA PELAYANANNYA SANGAT RAMAH

Penghilangan Tanda Baca → fasilitasnya gila lengkap memuaskan banget kamarnya bersih nyaman serta pelayanannya sangat ramah

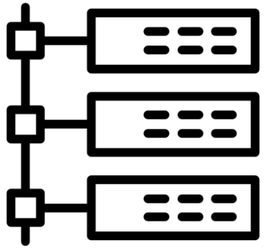
Tokenisasi

Memecah teks menjadi unit-unit yang lebih kecil

Mengapa Perlu Tokenisasi?



Token menunjukkan **struktur teks**



Token berperan sebagai **fitur** dalam pemrosesan teks

Bagaimana Tokenisasi dilakukan?

> “Fasilitasnya gila lengkap memuaskan banget, kamarnya bersih-nyaman, serta pelayanannya sangat ramah”

Word Tokenisation

Fasilitasnya	gila	lengkap	memuaskan	banget	
kamarnya	bersih	nyaman	serta	pelayanannya	
sangat	ramah				

Subword Tokenisation

Fasilitas##	nya	lengkap	mem	##uas	##kan
banget	Kamar##	bersih	nyaman	serta	
Pelayanan##	sangat	ramah			

Stopwords

The ‘too frequent’ words

George Kingsley Zipf

CONTOH:

“menghilangkan kata **yang** kemunculannya tinggi, **atau yang disebut dengan** stopwords, **dapat** berdampak **pada** peningkatan akurasi”

↓

menghilangkan kata kemunculannya tinggi, stopwords, berdampak peningkatan akurasi

Stemming dan Lematisasi

Stemming

Membuang awalan dan **akhiran** yang muncul pada sebuah token

Lemmatisation

Mengubah token menjadi **bentuk dasar**nya

Text Representation – Why?

Kenapa teks perlu direpresentasikan ke bentuk numerik/angka?

1. Representasi teks adalah jembatan antara bahasa manusia dengan perhitungan matematis yang dilakukan komputer
2. Representasi numerik memungkinkan teks dianalisis dan diukur secara kuantitatif.

Teks A

"wajib pajak terlambat lapor spt"

Teks B

"kereta api terlambat tiba di stasiun"

Teks C

"setiap wajib pajak aktif harus lapor spt tahunan"

Mana yang paling mirip dengan Teks A?

Text Representation

Teks:
"Fasilitasnya gila lengkap memuaskan banget, kamarnya bersih-nyaman ..."

Preprocessing dan Tokenization

Tokens:
['fasilitasnya', 'gila', 'lengkap', 'memuaskan', 'banget', 'kamarnya', 'bersih','nyaman', ...]

Vectorization

Representasi Numerik:

fasilitasnya	0.8	0.1	...	0.2
gila	0.2	0.3	...	0.1
lengkap	0.3	0.5	...	0.7
...				

Text Representation : Bag of Words (BoW)

Setiap cell adalah jumlah kemunculan token/kata pada setiap dokumen

	D1	D2	D3	D4	D5
fasilitasnya	0	1	5	10	8
gila	0	0	14	14	13
lengkap	5	7	0	9	8
memuaskan	12	13	0	5	4
banget	12	5	4	10	12

Representasi numerik untuk dokumen D4 ? [10, 14, 9, 5, 10]

Kelemahan?

- Bias ke token/kata yang umum karena hanya mempertimbangkan frekuensi kemunculan token/kata saja.
- tidak bisa membedakan mana token/kata yang penting dan mana token/kata yang tidak penting.

Text Representation : TF-IDF

TF-IDF adalah penyempurnaan dari BoW, dimana TF-IDF memberikan nilai kepada kata berdasarkan 2 prinsip:

- **Seberapa sering** kata muncul di dokumen ini (TF)
- **Seberapa jarang** kata muncul di seluruh koleksi dokumen lain (IDF)

Tujuannya adalah menonjolkan kata yang sering muncul di satu dokumen tertentu, tapi jarang muncul di dokumen lain.

$$W_{t,d} = \mathrm{TF}(t,d) \times \mathrm{IDF}(t)$$

Dimana:

1. **Term Frequency (TF)**: Frekuensi kata dalam dokumen tertentu.

$$\mathrm{TF}(t,d) = \frac{\text{Jumlah kemunculan kata } t \text{ di dokumen } d}{\text{Total kata dalam dokumen } d}$$

2. **Inverse Document Frequency (IDF)**: Mengukur seberapa informatif sebuah kata.

$$\mathrm{IDF}(t) = \log\left(\frac{\text{Total Dokumen } (N)}{\text{Jumlah Dokumen yang mengandung kata } t}\right)$$

Text Representation : TF-IDF

Misalkan kita memiliki 3 dokumen:

A = "pemilik kendaraan bermotor harus bayar pajak setiap tahun"

B = "pemilik dan pembeli bertemu di kantor notaris "

C = "setiap pemilik npwp harus bayar pajak dan lapor spt"

Kata "pemilik" dokumen A

- TF("pemilik", dokumen A) = 1/8 = 0.125
- IDF("pemilik") = log(3/3) = 0

TF-IDF("pemilik, dokumen A) = 0.125 x 0 = **0**

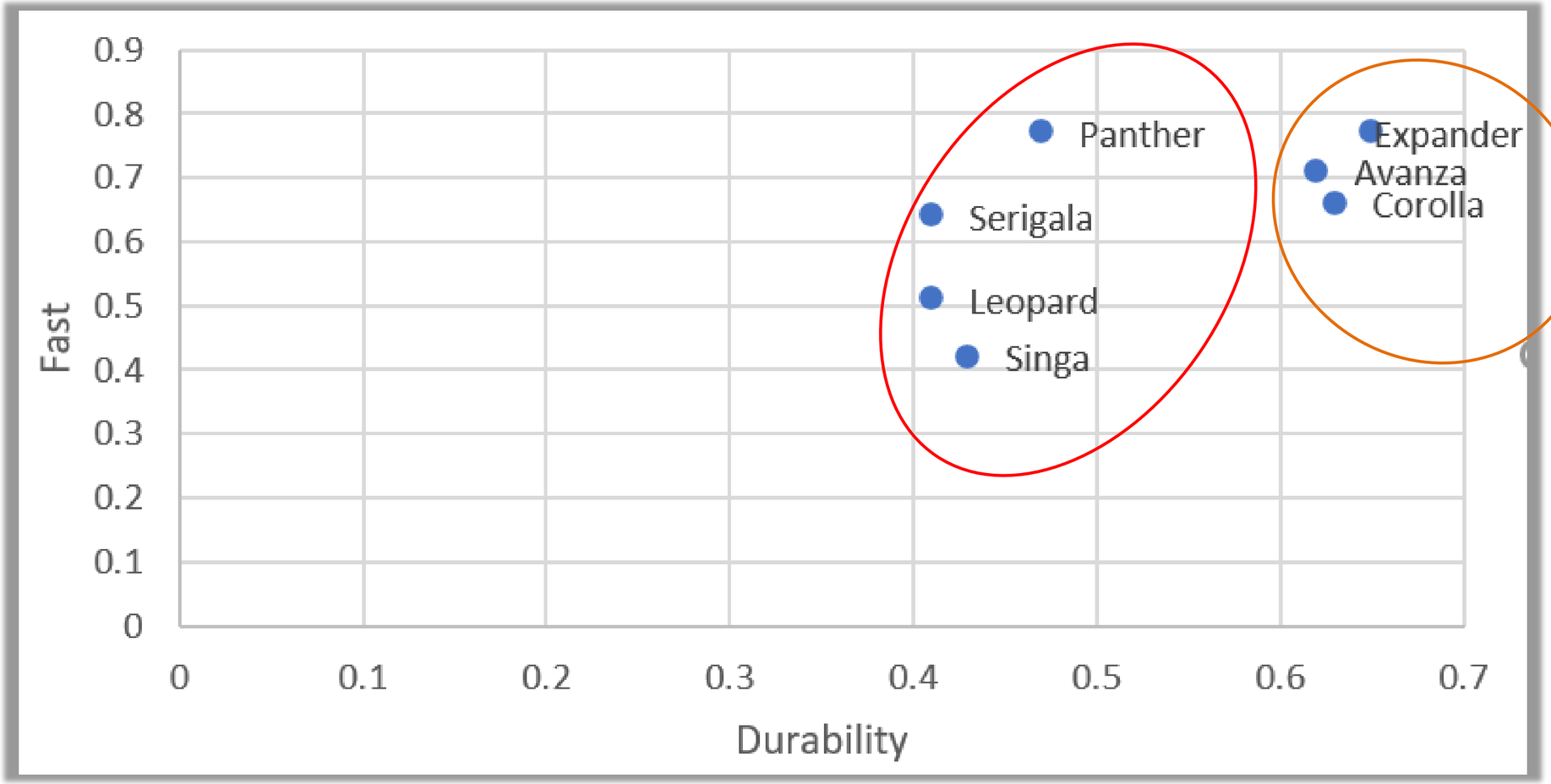
Kata "notaris" di dokumen B

- TF("notaris", dokumen A) = 1/7 = 0.14
- IDF("notaris") = log(3/1) = 0.47

TF-IDF("notaris, dokumen B) = 0.14 x 0.47 = **0.06**

Text Representation: Word Embedding

Mengubah kata menjadi angka



Kata dengan makna yang mirip seharusnya berada pada koordinat yang berdekatan pada semantic space

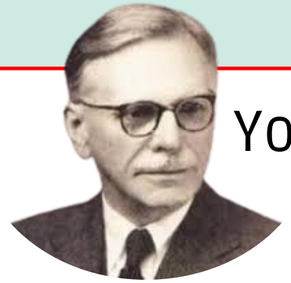
Text Representation: Sentence Embedding

What's the problem with word embedding?

Rudi bersimpati kepada **Adi** yang tidak lulus ujian matematika

Vs.

Adi bersimpati kepada **Rudi** yang tidak lulus ujian matematika



You shall know a word by the company it keeps

(John Rupert Firth, British Linguist)

Hence, Sentence Embedding

Rudi bersimpati kepada **Adi** yang tidak lulus ujian matematika → [0.35 0.49 0.88 -1.3 0.36 0.17 0.88]

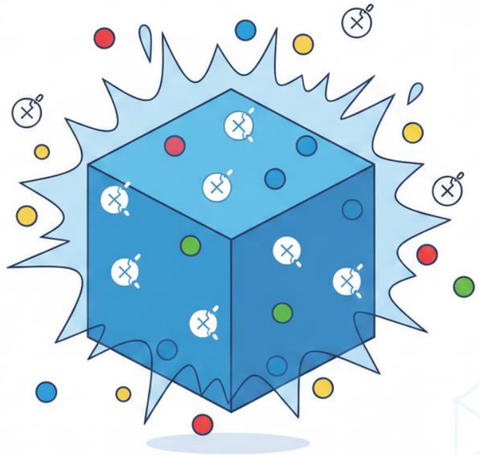
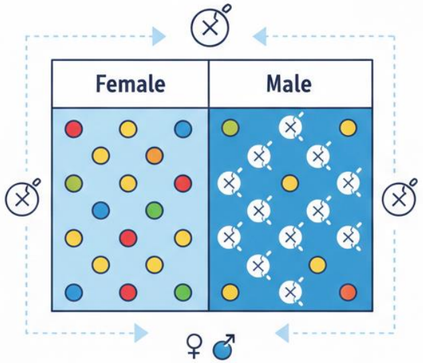
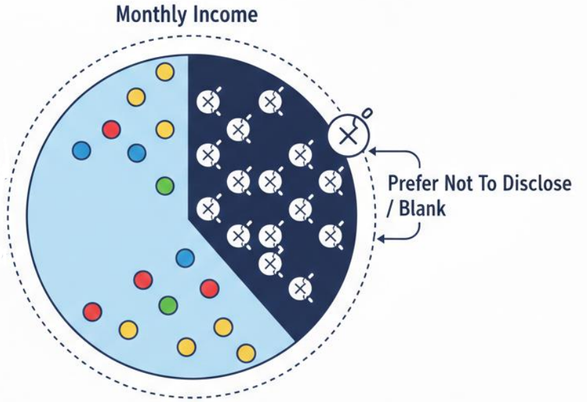
Adi bersimpati kepada **Rudi** yang tidak lulus ujian matematika → [0.85 0.77 0.21 -1.5 0.41 0.77 0.29]

Hands-on Time!

Buka Google Colab 

Transformasi Data Tekstual

Data Quality Issues : Jenis Missing Values

MISSING COMPLETELY AT RANDOM (MCAR)	MISSING AT RANDOM (MAR)	MISSING NOT AT RANDOM (MNAR)
Data loss with no hidden pattern. 	Data loss related to known variables  Example: Men are less likely to report weight in a health survey	Data loss related to the missing value itself 
Data hilang secara acak, bisa karena ketidaksengajaan misalnya karena sistem error	Data hilang karena dipengaruhi karakteristik atau perilaku dari variabel lain atau kelompok tertentu.	Data hilang karena nilai tersebut cenderung ingin disembunyikan atau sengaja untuk tidak diisi.

Data Quality Issues : Jenis Missing Values

Complete Data

Complete data	
Age	IQ score
25	133
26	121
29	91
51	116
54	97
31	98
44	118
46	93
48	141
51	104
30	105
30	110

VS

MCAR

Incomplete data	
Age	IQ score
25	
26	121
29	91
30	
30	110
31	
44	118
46	93
48	
51	
51	116
54	

- Missing values benar-benar muncul secara acak
- Biasanya diindikasikan dengan frekuensi kemunculan missing values yang relatif rendah

MAR

Incomplete data	
Age	IQ score
25	
26	
29	
30	
30	
31	
44	118
46	93
48	141
51	104
51	116
54	97

- **Terindikasi ada hubungan** antara missing values dengan atribut lain yang ada pada data

MNAR

Incomplete data	
Age	IQ score
25	133
26	121
29	
30	
30	110
31	
44	118
46	
48	141
51	
51	116
54	

- **Dipastikan ada hubungan** antara missing values dan non-missing values pada atribut yang sama
- Biasanya diindikasikan dengan frekuensi kemunculan missing values yang tinggi (lebih tinggi jika dibandingkan dengan MAR)

How to Identify Missingness Types With Missingno (MCAR, MAR, MNAR) | Towards Dev

Data Quality Issues : Jenis Missing Values

Diskusi

Sebuah bank digital memiliki dataset berisi nasabah yang mengajukan pinjaman. Ditemukan tiga kolom dengan *missing values* sebagai berikut:

1. Kolom **"Pekerjaan":** Banyak data kosong karena ada **bug pada aplikasi** versi Android yang menyebabkan menu *dropdown* pekerjaan tidak muncul, sehingga nasabah tidak bisa mengisinya. Pengguna iPhone tidak mengalami masalah ini.
2. Kolom **"Pendapatan Tahunan":** Nasabah yang memiliki **skor kredit rendah** (berdasarkan pengecekan BI Checking/OJK) cenderung tidak mengisi kolom ini karena mereka merasa jika mereka jujur, pinjaman mereka akan ditolak.
3. Kolom **"Tujuan Pinjaman":** Data kosong ditemukan secara acak pada berbagai nasabah tanpa pola tertentu. Setelah dicek, ternyata ini terjadi karena **gangguan server** selama 5 menit yang menyebabkan beberapa data transaksi tidak tersimpan sempurna.

Data Quality Issues : Key Takeaways

1. Data duplikat ditentukan oleh defenisi entitas dan konteks bisnis, bukan hanya karena baris data terlihat sama,
2. Nilai ekstrem tidak selalu salah, bisa merupakan *noise* atau justru outlier yang penting, tergantung tujuan analisis,
3. Data kosong/*Missing Values* tidak selalu berarti datanya hilang, bagaimana dan alasan kenapa data bisa hilang sangat menentukan *treatment* dan cara menyikapinya.
4. Kesalahan format data bukan berarti datanya salah, tetapi representasi datanya tidak standar sehingga perlu dilakukan perbaikan.

Hati-hati: Data Leakage!

Menggunakan informasi dari masa depan untuk memprediksi masa depan (a.k.a *cheating*)

Target/Feature Leakage

Menggunakan fitur yang berkorelasi tinggi dengan kelas target untuk memprediksi target

Contoh:

1. Prediksi Putusan Pengadilan Pajak atas Upaya Banding WP menggunakan amar putusan sebagai salah satu fitur
2. Prediksi churn-rate dari pelanggan komunikasi pascabayar dengan menggunakan fitur "jumlah_hari_setelah_kontrak_berakhir"

Train-Test Leakage

Melakukan preprocessing atau melatih model dengan menggunakan data populasi (data train + data test)

Contoh:
Melakukan scaling atribut numerik dengan berdasarkan distribusi data keseluruhan;

Temporal Leakage

Melatih model *time-series* menggunakan interval waktu yang lebih Panjang tetapi menguji model pada interval waktu yang merupakan subset dari interval waktu yang digunakan pada pelatihan

Contoh:
Training model pada interval Jan-Des tetapi menguji model pada data bulan Juni dalam pada rentang tahun yang sama

Sumber : [Data Leakage in Preprocessing | TDS Archive]

Large Scale Data Preparation: Chunking in Pandas

Tanpa Chunking

```python
import pandas as pd
import time

start_time = time.time()
df = pd.read_csv(data_path)
end_time = time.time()
elapsed = end_time - start_time

print(f"Size data: {len(df)}")
print(f"Waktu: {round(elapsed,3)} detik")
```

```
Size data: 4565000
Waktu: 3.947 detik
```

Dengan Chunking

```python
start_time = time.time()
total = 0

for chunk in pd.read_csv(data_path, chunksize=100000):
    total += len(chunk)
    pass

end_time = time.time()
elapsed = end_time - start_time

print(f"Size data: {total}")
print(f"Waktu: {round(elapsed,3)} detik")
```

```
Size data: 4565000
Waktu: 3.726 detik
```

```python
start_time = time.time()
total = 0

for chunk in pd.read_csv(data_path, chunksize=500000):
    total += len(chunk)
    pass

end_time = time.time()
elapsed = end_time - start_time

print(f"Size data: {total}")
print(f"Waktu: {round(elapsed,3)} detik")
```

```
Size data: 4565000
Waktu: 2.877 detik
```

```python
start_time = time.time()
total = 0

for chunk in pd.read_csv(data_path, chunksize=1000000):
    total += len(chunk)
    pass

end_time = time.time()
elapsed = end_time - start_time

print(f"Size data: {total}")
print(f"Waktu: {round(elapsed,3)} detik")
```

```
Size data: 4565000
Waktu: 3.518 detik
```

Large Scale Data Preparation: or Better Yet, Use Polars

Pandas

```python
import pandas as pd
import time

start_time = time.time()
df = pd.read_csv(data_path)
end_time = time.time()
elapsed = end_time - start_time

print(f"Size data: {len(df)}")
print(f"Waktu: {round(elapsed,3)} detik")
```

```
Size data: 4565000
Waktu: 3.947 detik
```

Polars

```python
import polars as pl

start_time = time.time()
df = pl.read_csv(data_path)
end_time = time.time()
elapsed = end_time - start_time

print(f"Size data: {len(df)}")
print(f"Waktu: {round(elapsed,3)} detik")
```

```
Size data: 4565000
Waktu: 1.164 detik
```

Latihan!

Buka Google Colab 

Pandas: Revisited

Series (one-dimensional)

```
1 df['total_bedrooms']
```

```
0        1283.0
1        1901.0
2         174.0
3         337.0
4         326.0
          ...
16995     394.0
16996     528.0
16997     531.0
16998     552.0
16999     300.0
Name: total_bedrooms, Length: 17000, dtype: float64
```

Dataframe (two-dimensional)

```
1 df[['total_bedrooms','population','median_income']]
```

	total_bedrooms	population	median_income
0	1283.0	1015.0	1.4936
1	1901.0	1129.0	1.8200
2	174.0	333.0	1.6509
3	337.0	515.0	3.1917
4	326.0	624.0	1.9250
...	...	...	...
16995	394.0	907.0	2.3571
16996	528.0	1194.0	2.5179
16997	531.0	1244.0	3.0313
16998	552.0	1298.0	1.9797
16999	300.0	806.0	3.0147

17000 rows × 3 columns

Pandas: Revisited

Access Data by Index

Dataframe Name

Row Index

```
df.iloc[0:2, 0:2]
```

Column Index

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-114.31	34.19	15.0	5612.0	1283.0	1015.0	472.0	1.4936	66900.0

1

```
1 df.iloc[0]
```

```
longitude            -114.3100
latitude               34.1900
housing_median_age     15.0000
total_rooms          5612.0000
total_bedrooms       1283.0000
population           1015.0000
households            472.0000
median_income           1.4936
median_house_value  66900.0000
Name: 0, dtype: float64
```

2

```
1 df.iloc[0,0:]
```

```
longitude            -114.3100
latitude               34.1900
housing_median_age     15.0000
total_rooms          5612.0000
total_bedrooms       1283.0000
population           1015.0000
households            472.0000
median_income           1.4936
median_house_value  66900.0000
Name: 0, dtype: float64
```

3

```
1 df.iloc[0:1,0:1]
```

	longitude
0	-114.31

4

```
1 df.iloc[0,0]
```

```
-114.31
```

Pandas: Revisited

Access Data by Label

Dataframe Name

Row Index/Label

```
df.loc['a':'c', 'x':'z']
```

Column Label

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-114.31	34.19	15.0	5612.0	1283.0	1015.0	472.0	1.4936	66900.0

1

```
1 df.loc[0]
```

```
longitude            -114.3100
latitude               34.1900
housing_median_age     15.0000
total_rooms          5612.0000
total_bedrooms       1283.0000
population           1015.0000
households            472.0000
median_income           1.4936
median_house_value  66900.0000
Name: 0, dtype: float64
```

2

```
1 df.loc[0, 'longitude':]
```

```
longitude            -114.3100
latitude               34.1900
housing_median_age     15.0000
total_rooms          5612.0000
total_bedrooms       1283.0000
population           1015.0000
households            472.0000
median_income           1.4936
median_house_value  66900.0000
Name: 0, dtype: float64
```

3

```
1 df.loc[0:1, 'longitude':'latitude']
```

	longitude	latitude
0	-114.31	34.19
1	-114.47	34.40

Pandas: Revisited

Filtering

```
1 df[df['latitude'] > 35]
```

	longitude	latitude
119	-115.93	35.55
157	-116.22	36.00
264	-116.57	35.43
568	-117.02	36.40
1863	-117.28	35.13

```
1 df[(df['latitude'] > 35) & df['housing_median_age'].isin([18,19])]
```

	longitude	latitude	housing_median_age	total_rooms	total_bed
119	-115.93	35.55	18.0	1321.0	
568	-117.02	36.40	19.0	619.0	
2638	-117.67	35.65	18.0	2737.0	
2745	-117.70	35.62	18.0	2657.0	
3054	-117.81	35.65	19.0	1124.0	

PJJ Data Analytics 2026

Data Preparation

Cheers!

Riki Akbar
Ibrahim Saleh Siregar