

LEARNING PROGRESS REVIEW

Week 7



Citizen Data Scientist: Diaz Jubiary - Hermulia Hadie Putra - Desi Sulistyowati - Farahul Jannah

Contents

01

Advanced Dataframe

- Indexing Dataframe
- Menghapus Variable/kolom
- Menggabungkan Dataframe
- Concatenate & Append Dataframe
- Pivot Table Dataframe
- Melting Dataframe
- Fungsi Lambda dalam Dataframe

02

Database Programming

- File Teks di Python
- Database Programming
- Star vs Snowflake Scheme
- Connect PostgreSQL
- Menggabungkan Dataframe
- Data Wrangling using Pandas

03

Application Programming Interface (API)

- What is API
- API Example
- Benefit of API
- API Data Format
- Component of API
- API Data Processing

Advanced Dataframe

Indexing Dataframe

- Indexing dataframe dalam pandas berarti memilih baris dan kolom data tertentu dari DataFrame. Indexing dapat berarti memilih semua baris dan beberapa kolom, beberapa baris dan semua kolom, atau beberapa dari setiap baris dan kolom.
- Indexing pada dataframe menggunakan Pandas memiliki beberapa pengaplikasian di dalam dataset.
- Contoh:
 - Mengurutkan index
 - Membuat data pada variable tertentu menjadi index



Indexing Dataframe

Syntax

```
## reset index starting from 0  
random_.reset_index(drop=True)
```

	age	sex	bmi	children	smoker	region	charges
1308	25	female	30.20	0	yes	southwest	33900.65300
221	53	female	33.25	0	no	northeast	10564.88450
989	24	female	20.52	0	yes	northeast	14571.89080
839	59	female	31.35	0	no	northwest	12622.17950
354	18	female	38.28	0	no	southeast	14133.03775



	age	sex	bmi	children	smoker	region	charges
0	25	female	30.200	0	yes	southwest	33900.65300
1	53	female	33.250	0	no	northeast	10564.88450
2	24	female	20.520	0	yes	northeast	14571.89080
3	59	female	31.350	0	no	northwest	12622.17950
4	18	female	38.280	0	no	southeast	14133.03775
5	57	male	40.370	0	no	southeast	10982.50130

Indexing Dataframe (Set Column as Index)

Index juga dapat dibuat dari kolom yang ada di dalam dataset

```
## set column as index  
random_.set_index('age')
```

	age	sex	bmi	children	smoker	region	charges
0	25	female	30.200	0	yes	southwest	33900.65300
1	53	female	33.250	0	no	northeast	10564.88450
2	24	female	20.520	0	yes	northeast	14571.89080
3	59	female	31.350	0	no	northwest	12622.17950
4	18	female	38.280	0	no	southeast	14133.03775
5	57	male	40.370	0	no	southeast	10982.50130

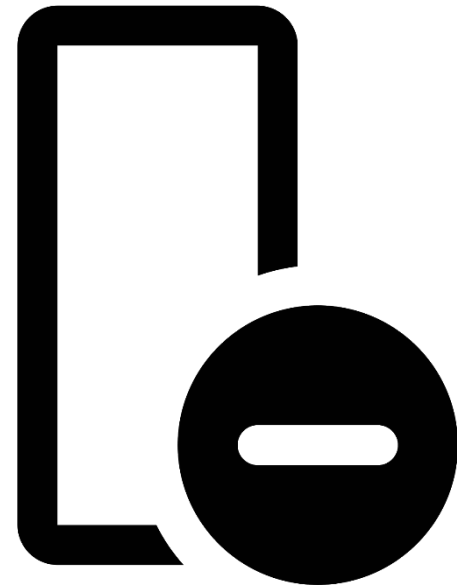


	sex	bmi	children	smoker	region	charges
age						
25	female	30.200	0	yes	southwest	33900.65300
53	female	33.250	0	no	northeast	10564.88450
24	female	20.520	0	yes	northeast	14571.89080
59	female	31.350	0	no	northwest	12622.17950
18	female	38.280	0	no	southeast	14133.03775

Menghapus Variable/kolom

Pandas dapat menghapus kolom-kolom yang tidak diinginkan. Adapun tujuan menghapus kolom adalah:

- Untuk memilih kolom yang akan dianalisa
- Untuk memilih kolom yang digunakan dalam machine learning model



Dropping Column (beberapa kolom)

```
# dropping column  
data.drop(['bmi','children'], axis=1).head()
```

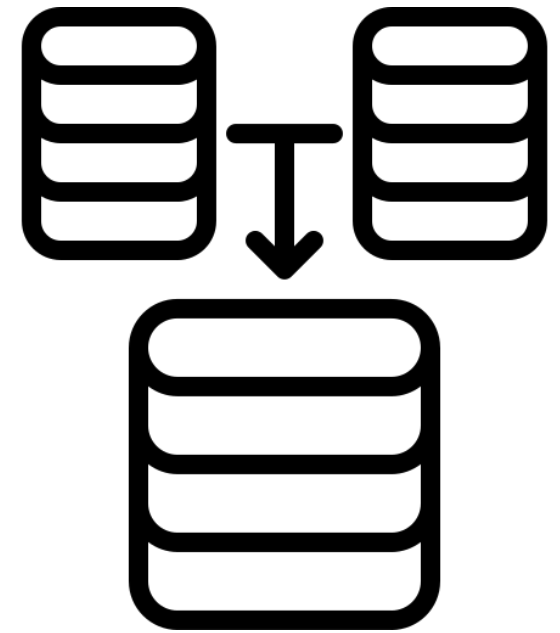
	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520



	age	sex	smoker	region	charges
0	19	female	yes	southwest	16884.92400
1	18	male	no	southeast	1725.55230
2	28	male	no	southeast	4449.46200
3	33	male	no	northwest	21984.47061
4	32	male	no	northwest	3866.85520

Menggabungkan Dataframe

- Selain menggunakan merge, pandas juga dapat menggabungkan dua dataset menjadi satu menggunakan **join**.
- Terdapat perbedaan antara merge dan join yaitu:
 - ✓ Join: Menggabungkan data berdasarkan index
 - ✓ Merge: Menggabungkan data lebih fleksibel dan memungkinkan untuk menentukan kolom selain index untuk kedua dataframe



Menggabungkan Dataframe

Join

```
data_5.join(data_dummy, lsuffix='_first', rsuffix='_second')
```

data_dummy

	age	sex
0	19	female
1	18	male
2	28	male

+

data_5

	age	bmi
0	19	27.900
1	18	33.770
2	28	33.000
3	33	22.705
4	32	28.880



	age_first	bmi	age_second	sex
0	19	27.900	19.0	female
1	18	33.770	18.0	male
2	28	33.000	28.0	male
3	33	22.705	NaN	NaN
4	32	28.880	NaN	NaN

Concatenate & Append dataframe

Menggabungkan objek dengan Pandas pada spesifik axis baik itu x-axis (horizontal) ataupun y-axis (vertikal)

Concatenante (horizontal)

```
# concatenate data in horizontal  
pd.concat([data_dummy, data_5], axis=1)
```

data_dummy

	age	sex
0	19	female
1	18	male
2	28	male

+

data_5

	age	bmi
0	19	27.900
1	18	33.770
2	28	33.000
3	33	22.705
4	32	28.880



	age	sex	age	bmi
0	19.0	female	19	27.900
1	18.0	male	18	33.770
2	28.0	male	28	33.000
3	NaN	NaN	33	22.705
4	NaN	NaN	32	28.880

Concatenate vertical

```
# concatenate data in vertical  
pd.concat([data_dummy, data_5], axis=0)
```

data_dummy

	age	sex
0	19	female
1	18	male
2	28	male

+

data_5

	age	bmi
0	19	27.900
1	18	33.770
2	28	33.000
3	33	22.705
4	32	28.880



	age	sex	bmi
0	19	female	NaN
1	18	male	NaN
2	28	male	NaN
0	19	NaN	27.900
1	18	NaN	33.770
2	28	NaN	33.000
3	33	NaN	22.705
4	32	NaN	28.880



Append Dataframe

Dalam dataframe, append dapat dilakukan jika terdapat nama kolom pada kedua dataset yang sama

```
# append data  
data_5.append(data_dummy)
```

data_dummy

	age	sex
0	19	female
1	18	male
2	28	male



data_5

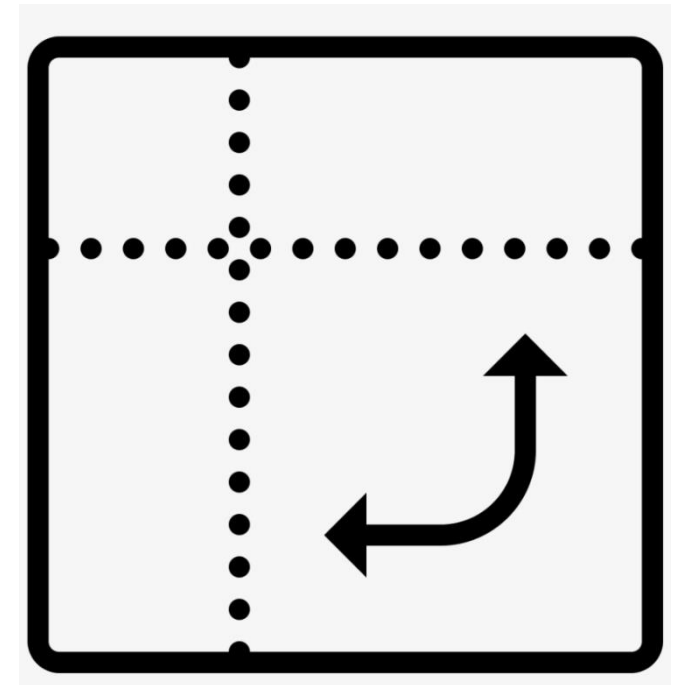
	age	bmi
0	19	27.900
1	18	33.770
2	28	33.000
3	33	22.705
4	32	28.880



	age	bmi	sex
0	19	27.900	NaN
1	18	33.770	NaN
2	28	33.000	NaN
3	33	22.705	NaN
4	32	28.880	NaN
0	19	NaN	female
1	18	NaN	male
2	28	NaN	male

Pivot Table Dataframe

- Pivot table memberikan informasi berupa agregasi suatu data dengan melampirkan isi data pada nama kolom tertentu
- Beberapa karakteristik pivot table menggunakan pandas:
 - Tampilan seperti pivot table yang ada di spreadsheet
 - Nama kolom sebagai level data disimpan dalam bentuk Multi Index





Pivot Table

```
# pivot table
pd.pivot_table(data, values="bmi", index=["sex","smoker"], columns="region",
               aggfunc=np.max)
```

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520



		region	northeast	northwest	southeast	southwest
sex	smoker					
female	no	48.070	42.940	46.75	46.7	
	yes	42.750	38.950	47.41	47.6	
male	no	42.655	41.325	53.13	45.9	
	yes	41.895	40.565	52.58	39.9	

Melting Dataframe

Melting dataframe digunakan untuk memberikan informasi data dimana nama kolom/variable akan menjadi datapoint dan tetap memberikan informasi nilai dari kolom/variable namun di kolom yang berbeda

```
pd.melt(data_melt, id_vars=["sex"], value_vars=["age"])
```

	age	sex	bmi
0	19	female	27.900
1	18	male	33.770
2	28	male	33.000
3	33	male	22.705
4	32	male	28.880



	sex	variable	value
0	female	age	19
1	male	age	18
2	male	age	28
3	male	age	33
4	male	age	32

Lambda Function

- Lambda digunakan untuk membuat fungsi dalam satu baris ekspresi.
- Sederhananya, fungsi lambda sama seperti fungsi python normal, namun fungsi lambda tidak perlu didefinisikan, dan hanya dalam satu baris kode.
- Lambda bisa memiliki **lebih dari satu argumen** atau parameter, tapi hanya bisa memiliki **satu ekspresi** atau isi.

Kata kunci untuk
membuat fungsi lamda Argumen
atau parameter Isi fungsi

lambda **args** : **expression**

Lambda Function

```
# create new variables/columns with lambda
```

```
data["bmi_category_lambda"] = data['bmi'].apply(lambda x: "Low BMI" if x < 30 else "High BMI")  
data.head()
```

	age	sex	bmi	children	smoker	region	charges	bmi_category_lambda
0	19	female	27.900	0	yes	southwest	16884.92400	Low BMI
1	18	male	33.770	1	no	southeast	1725.55230	High BMI
2	28	male	33.000	3	no	southeast	4449.46200	High BMI
3	33	male	22.705	0	no	northwest	21984.47061	Low BMI
4	32	male	28.880	0	no	northwest	3866.85520	Low BMI

DATABASE PROGRAMMING

File Teks di Python

Beberapa mode mengeksekusi file teks di Python

Character	Meaning
'r'	open for reading (default)
'w'	open for writing, truncating the file first
'x'	open for exclusive creation, failing if the file already exists
'a'	open for writing, appending to the end of file if it exists
'b'	binary mode
't'	text mode (default)
'+'	open for updating (reading and writing)



File Teks di Python

Membuat File Teks di Python

```
with open ("nama_file.txt", "w") as nama_variabel:  
    nama_variabel.write('Teks yang ingin ditulis')
```

Hasilnya dapat ditampilkan dengan syntax:

```
with open ('nama_file.txt', 'r') as teks_variabel:  
    print(teks_variabel.read())
```

```
Teks yang ingin ditulis
```

File Teks di Python

Menambahkan Teks pada File yang Sudah Dibuat

```
with open ('nama_file.txt', 'a') as teks_variabel:  
    teks_variabel.write("\nTeks yang ingin ditambahkan")
```

Hasilnya dapat ditampilkan dengan syntax:

```
with open('nama_file.txt', 'r') as teks_variabel:  
    for line in teks_variabel:  
        print(line, end="")
```

```
Teks yang ingin ditulis  
Teks yang ingin ditambahkan
```

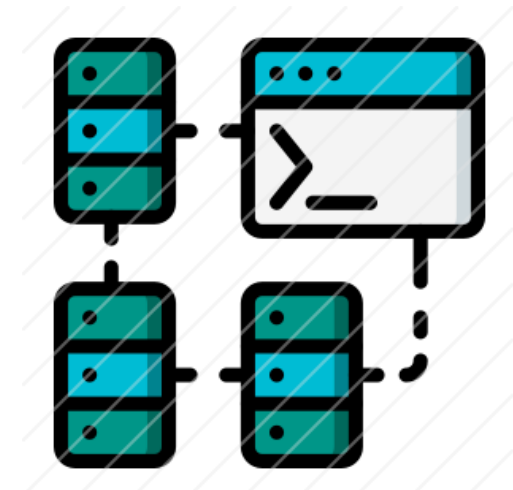
Database Programming

Dalam database biasanya terdapat 2 tabel, yaitu:

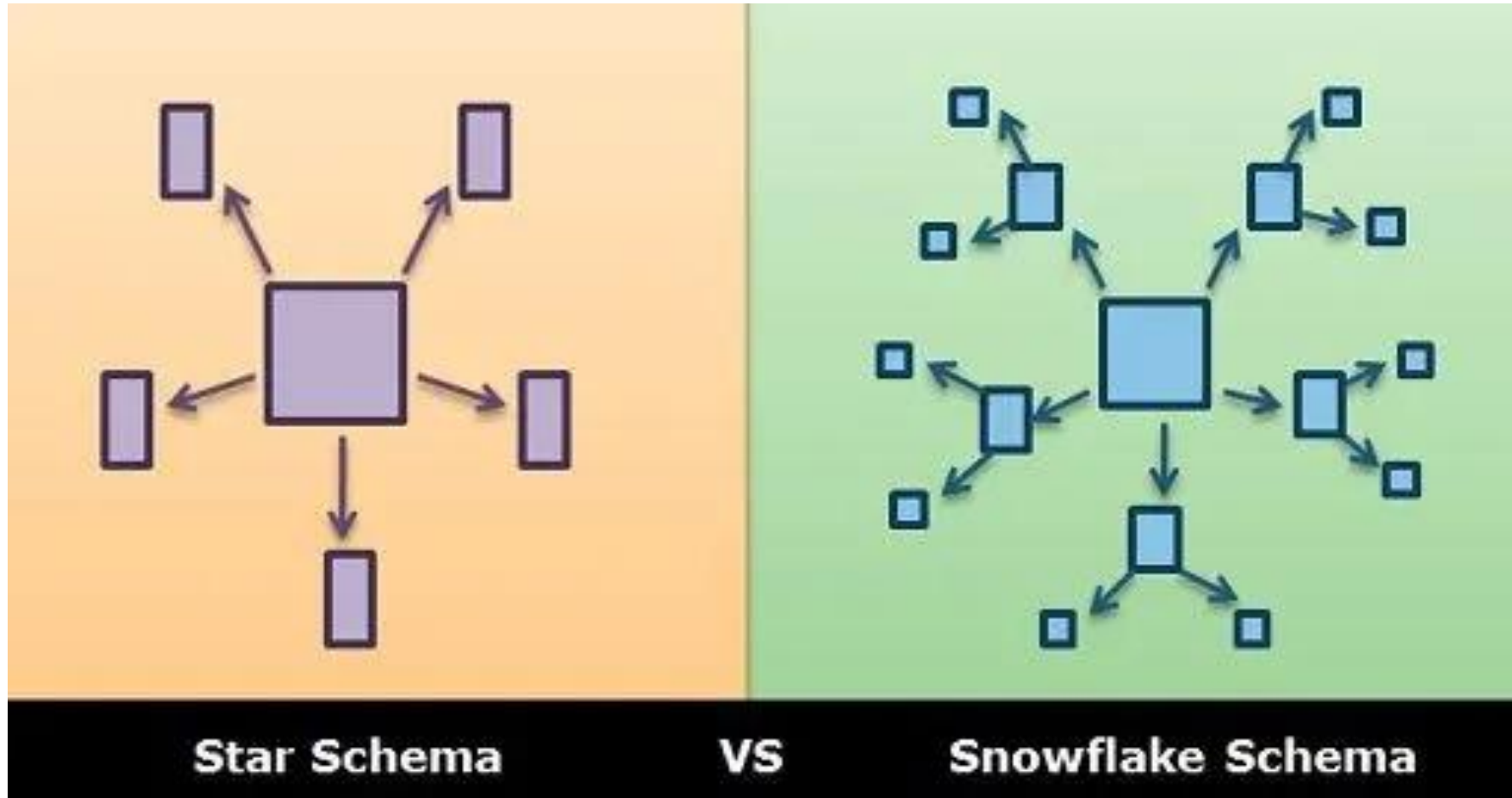
- Fact Table : Tabel yang berisikan foreign key, atau dengan kata lain berisi primary key dari beberapa tabel yang lainnya.
- Dimension Table : tabel yang berisikan berbagai atribut para foreign key yang ada di Fact Table.

Tipe skema pada data modelling

- Star Schema : Pada skema ini yang menjadi pusat "bintang" adalah 1 Fact Table sedangkan Dimension Table menjadi kepanjangan dari Fact Table.
- Snowflake Schema : Pada skema ini sama halnya dengan Star Schema, yaitu Fact Table berada di pusat skema. Perbedaannya adalah Dimension Table di skema ini mempunyai cabang Dimension Table yang lain lagi.



Star vs Snowflake Schema



Connect PostgreSQL ke Python

Pertama, install modul psycopg2

```
!pip install psycopg2
```

Kedua, import psycopg2

```
import psycopg2
```

Ketiga, koneksikan database ke Python

```
conn = psycopg2.connect(  
    host= "alamat server database",  
    database = 'nama database yang ingin dihubungkan',  
    user = 'nama pengguna yang digunakan untuk autentikasi',  
    password = 'password yang digunakan untuk autentikasi'  
)
```

Keempat, store the database table menggunakan cursor (Temporary Memory)

```
cur = conn.cursor()
```

Menjalankan query dan menampilkan hasilnya di Pandas Data Frame

```
sql = """
    select * from batch_11.cb_trips
    """

data2 = pd.read_sql_query(sql, conn)
data2.head()
```

	tripduration	starttime	stoptime	start station id	start_station_name	start_station_latitude	start_station_longitude	end_station_id	end_station_name
0	420	2013-07-08 00:38:09	2013-07-08 00:45:09	403	E 2 St & 2 Ave	40.725029	-73.990697	350	Clinton St & Grand St
1	557	2013-07-08 00:38:22	2013-07-08 00:47:39	247	Perry St & Bleecker St	40.735354	-74.004831	151	Cleveland Pl & Spring St
2	600	2013-07-08 00:38:23	2013-07-08 00:48:23	492	W 33 St & 7 Ave	40.750200	-73.990931	518	E 39 St & 2 Ave
3	1344	2013-07-08 00:38:50	2013-07-08 01:01:14	539	Metropolitan Ave & Bedford Ave	40.715348	-73.960241	301	E 2 St & Avenue B
4	479	2013-07-08 00:39:02	2013-07-08 00:47:01	2022	E 59 St & Sutton Pl	40.758491	-73.959206	482	W 15 St & 7 Ave

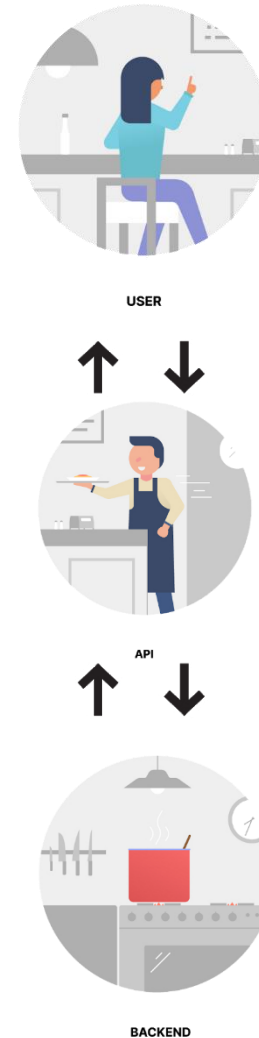
Application Programming Interface (API)

API analogy

Bayangkan Anda sedang duduk di meja restoran dengan menu pilihan untuk dipesan dan dapur yang akan menyiapkan pesanan Anda. Namun ada satu hal yang kurang, siapa yang akan menerima pesanan Anda dan bagaimana makanan akan diantarkan kepada Anda ?

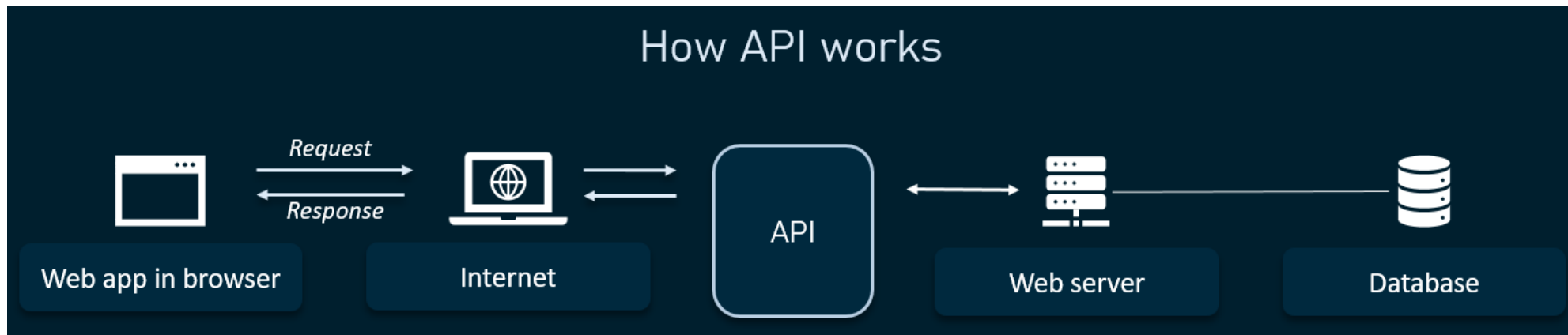
Disitulah fungsi pelayan. Pelayan adalah pembawa pesan pesanan Anda dan mengantarkan permintaan Anda ke dapur. Ketika makanan favorit Anda sudah siap, pelayan mengirimkannya kembali kepada Anda.

Situasi ini kurang lebih sama dengan cara kerja API.



What is API?

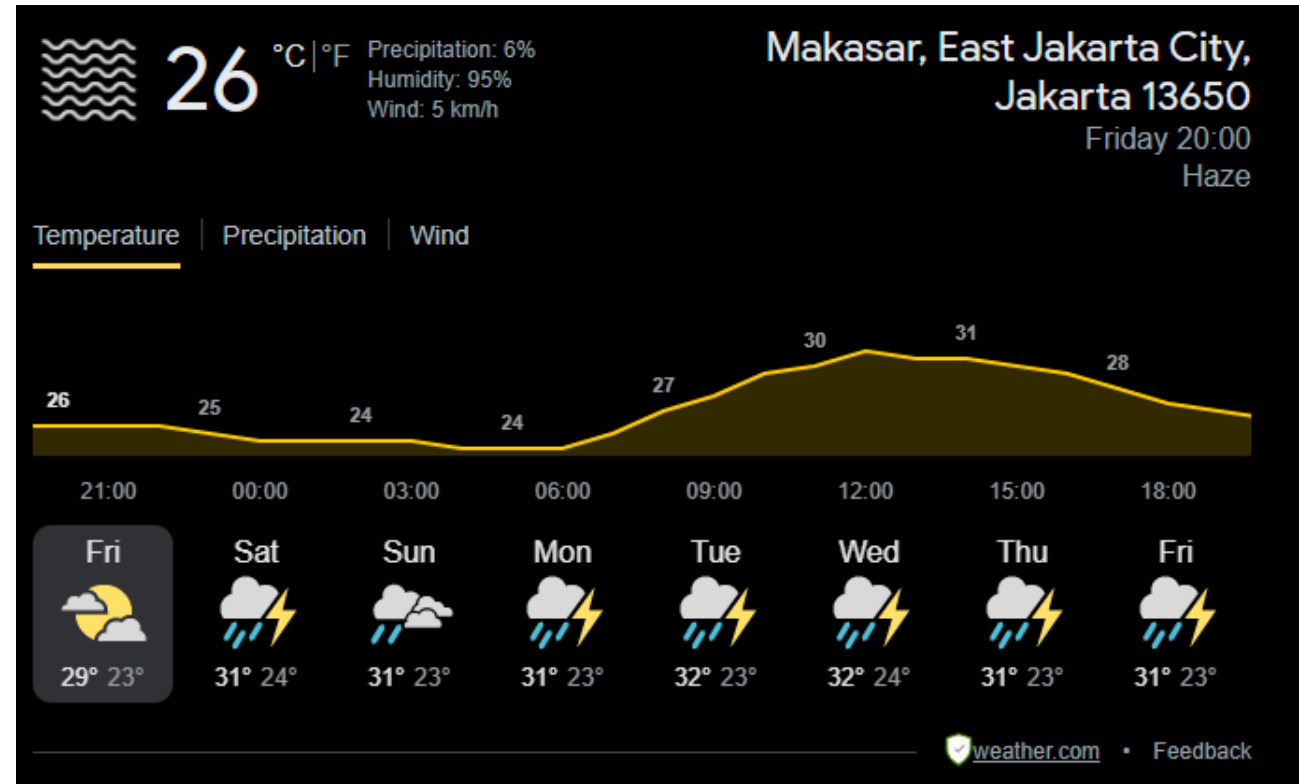
API adalah singkatan dari Application Programming Interface, yang merupakan perantara perangkat lunak yang memungkinkan dua aplikasi untuk berbicara satu sama lain. API adalah pembawa pesan yang menerima permintaan dan memberi tahu sistem apa yang ingin Anda lakukan, lalu mengembalikan responsnya kepada Anda. Setiap kali Anda menggunakan aplikasi seperti LinkedIn, mengirim pesan instan, atau memeriksa cuaca di ponsel, Anda menggunakan API.



API Example

1. Weather Snippets

Salah satu contoh penggunaan API umum yang kita temui setiap hari adalah data cuaca. Cuplikan data cuaca yang tampaknya biasa, dapat ditemukan di semua platform, seperti Penelusuran Google, aplikasi Cuaca Apple, atau bahkan dari perangkat rumah pintar Anda.





API Example

2. Log-in Using XYZ

Contoh lain dari penggunaan API adalah fitur

“login menggunakan Facebook/Twitter/Google/Github”

yang kita lihat di banyak situs web.

Ini sangat memudahkan, alih-alih benar-benar masuk ke akun media sosial pengguna (yang akan menimbulkan masalah keamanan yang serius), aplikasi dengan fungsi ini memanfaatkan API platform ini untuk mengautentikasi pengguna dengan setiap login.

Log in to your account



Log in with Twitter



Log in with Facebook



Log in with LinkedIn

API Example

3. Travel Booking

- API sangat bagus untuk layanan perjalanan karena memudahkan mesin untuk bertukar data dan permintaan dengan cepat dan mandiri — dalam hal ini, ketersediaan perjalanan dan permintaan reservasi.
- Tanpa menggunakan API, karyawan layanan pemesanan harus mengirim email secara manual ke maskapai penerbangan atau hotel untuk mengetahui ketersediaannya.
- Kemudian, setelah email kembali dari penyedia, mereka harus mengonfirmasinya dengan pelancong. Pada saat broker perjalanan mengirim email lagi ke penyedia, mengonfirmasi perjalanan, itu mungkin tidak lagi tersedia!

The screenshot displays a mobile application interface for finding flights and hotels. The left panel shows search filters: 'PESAWAT' (Flights) selected, 'Kota Asal' (Origin) as Yogyakarta (JOG), 'Kota Tujuan' (Destination) as Jakarta (JKTA), and 'Pilih tanggal berangkat' (Select departure date) as 19 October 2015. The right panel shows search results for flights from Yogyakarta to Jakarta on October 19, 2015, for 1 passenger. The results list flights from Lion and AirAsia with their respective times and prices. At the bottom, there is a 'CARI TIKET' (Find Ticket) button and options to 'URUTKAN' (Sort) and 'FILTER'.

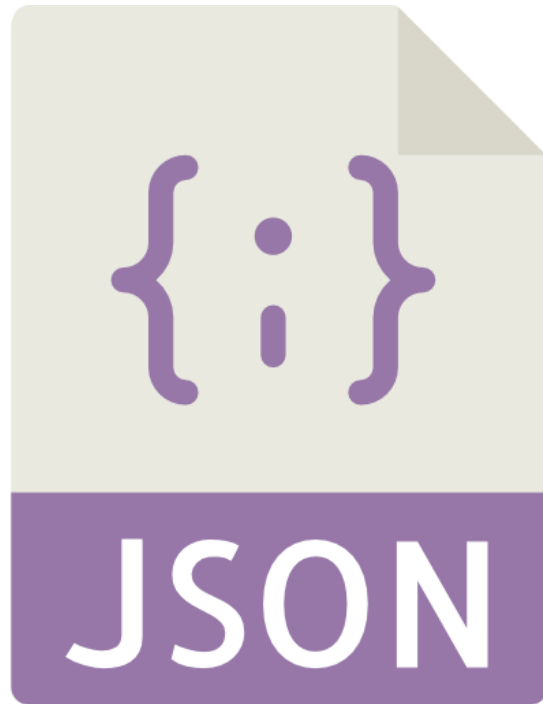
Airline	Flight Time	Price (Rp)
Lion	18:50 - 20:05 1j15m, Langsung	Rp431.100
Lion	20:00 - 21:15 1j15m, Langsung	Rp431.100
AirAsia	16:10 - 17:10 1j0m, Langsung	Rp440.900
AirAsia	12:10 - 13:15 1j5m, Langsung	Rp440.900
AirAsia	12:55 - 14:00 1j5m, Langsung	Rp440.900

Benefit of API

- ✓ Automation: dengan API, komputer bukan orang yang dapat mengelola pekerjaan. Melalui API, agensi dapat memperbarui alur kerja untuk membuatnya lebih cepat dan lebih banyakproduktif.
- ✓ New data available: API memungkinkan semua informasi yang dihasilkan ditingkat pemerintahan untuk tersedia bagi setiap warga negara, bukan hanya beberapa orang terpilih.
- ✓ Integration: API memungkinkan konten untuk disematkan dari situs atau aplikasi apa punlebih mudah. Ini menjamin penyampaian informasi yang lebih lancar dan integrasipengalaman pengguna.
- ✓ Personalization: melalui API, setiap pengguna atau perusahaan dapat menyesuaikan kontendan layanan yang paling sering mereka gunakan.



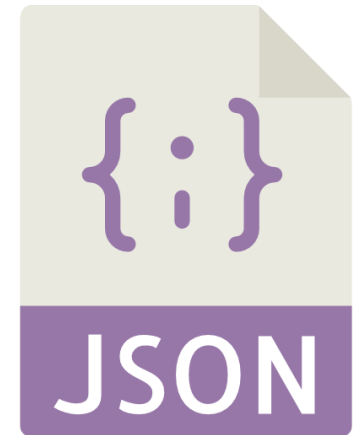
API Data Format



API Data Format

JSON

- Banyak API baru telah mengadopsi JSON sebagai format karena dibangun di atas bahasa pemrograman Javascript yang populer, yang ada di mana-mana di web dan dapat digunakan di bagian depan dan belakang aplikasi atau layanan web.
- JSON adalah format yang sangat sederhana yang memiliki dua bagian: kunci dan nilai. Kunci mewakili atribut tentang objek yang dijelaskan. Pesanan pizza bisa menjadi objek. Ini memiliki atribut (kunci), seperti jenis kerak, topping, dan status pesanan. Atribut ini memiliki nilai yang sesuai (kerak tebal, pepperoni, dan out-for-delivery).



API Data Format

XML

- XML telah ada sejak tahun 1996. Seiring bertambahnya usia, XML telah menjadi format data yang sangat matang dan kuat.
- Seperti JSON, XML menyediakan beberapa blok bangunan sederhana yang digunakan pembuat API untuk menyusun data mereka. Blok utama disebut node. XML selalu dimulai dengan simpul akar, yang dalam contoh pizza kami adalah "pesanan".
- Di dalam urutan ada lebih banyak simpul "anak". Nama setiap node memberi tahu kita atribut pesanan (seperti kunci dalam JSON) dan data di dalamnya adalah detail aktual (seperti nilai dalam JSON).



API in Python

- Python adalah bahasa yang *powerfull* yang dapat digunakan untuk *create* dan *request* API. Ada beberapa library populer python yang dibuat khusus untuk API, misalnya request, flask, fastapi, django, dll.
- Tapi untuk pembelajaran kali ini kita tidak akan membuat API karena membutuhkan pemahaman yang lebih dalam tentang cara kerja backend.
- Kita akan belajar bagaimana mendapatkan data dari API dan library yang akan kita gunakan adalah Requests.



The Roles of HTTP, APIs, and REST

- Application Programming Interface (API) adalah layanan web yang memberikan akses ke data dan metode spesifik yang dapat diakses oleh aplikasi lain – dan terkadang mengedit – melalui protokol HTTP standar, seperti situs web. Kesederhanaan ini memudahkan pengintegrasian API dengan cepat ke dalam berbagai macam aplikasi.
- Representational State Transfer (REST) mungkin adalah gaya arsitektur API untuk layanan web yang paling populer. Ini terdiri dari seperangkat pedoman dirancang untuk menyederhanakan komunikasi klien / server. REST API membuat data akses jauh lebih mudah dan logis.

Component of API

- ❖ Endpoint – URL yang menggambarkan data apa yang Anda gunakan untuk berinteraksi. Mirip dengan bagaimana URL halaman web diikat ke halaman tertentu, URL titik akhir diikat ke sumber daya tertentu di dalam API.
- ❖ Data – Jika Anda menggunakan metode yang melibatkan perubahan data dalam REST API, Anda harus menyertakan payload data dengan permintaan yang menyertakan semua data yang akan dibuat atau dimodifikasi.
- ❖ Header – Berisi metadata apa pun yang perlu disertakan dengan permintaan, seperti token autentikasi, tipe konten yang harus dikembalikan, dan kebijakan caching apa pun.
- ❖ Method – Menentukan bagaimana Anda berinteraksi dengan sumber daya yang terletak di titik akhir yang disediakan. REST API dapat menyediakan metode untuk mengaktifkan fungsionalitas penuh. Buat, Baca, Perbarui, dan Hapus (CRUD). Metode umum adalah GET, PUT, POST, DELETE





API Statuses

HTTP response status codes indicate whether a specific HTTP request has been successfully completed. Responses are grouped in five classes:

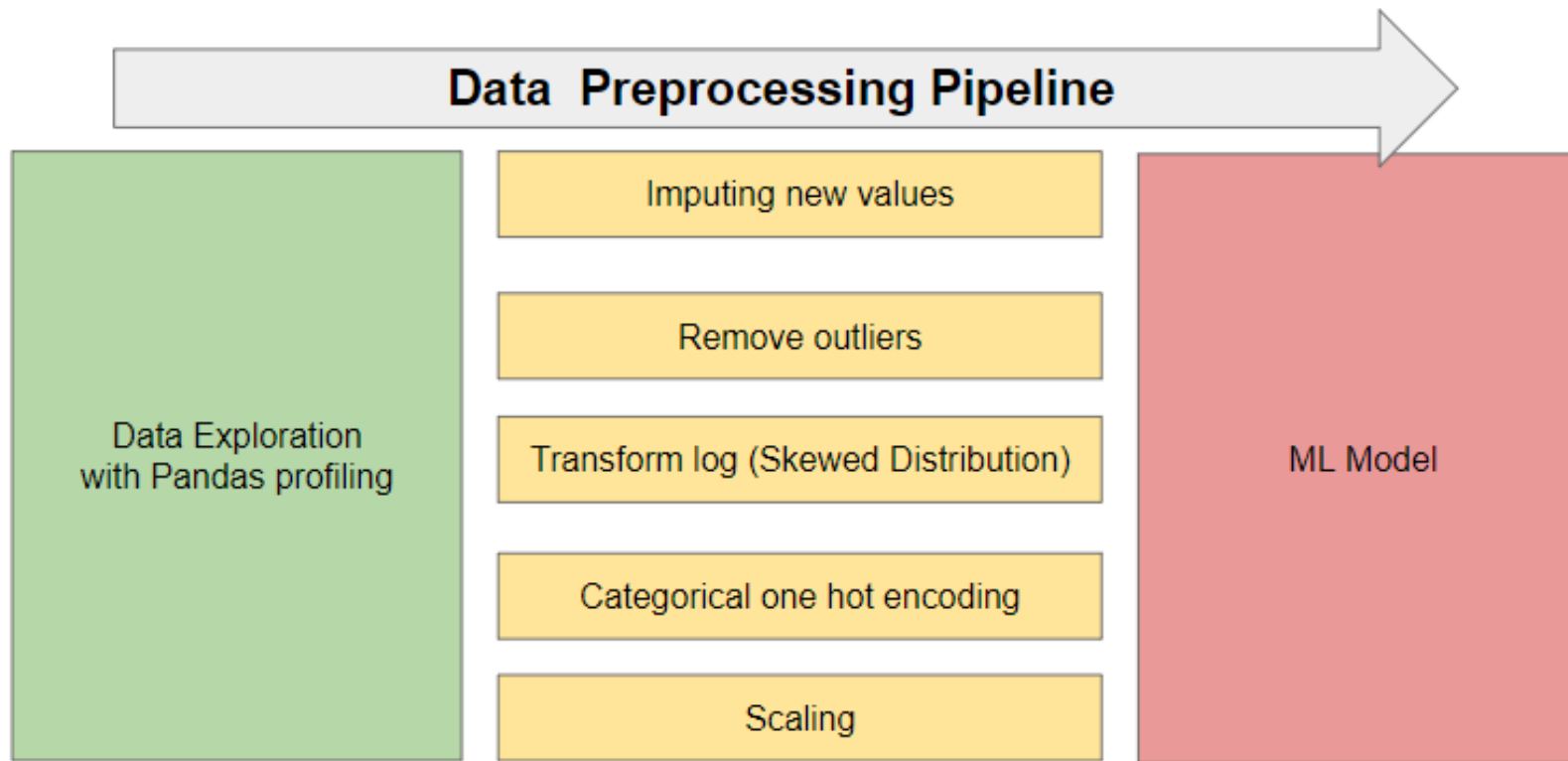
1. Informational responses (`100`–`199`)
2. Successful responses (`200`–`299`)
3. Redirects (`300`–`399`)
4. Client errors (`400`–`499`)
5. Server errors (`500`–`599`)

REST API Methods

HTTP Verb	CRUD
POST	Create
GET	Read
PUT	Update/Replace
PATCH	Update/Modify
DELETE	Delete

API Data Processing

Setelah mendapatkan data dari API, kita perlu mengolah data tersebut, agar berguna untuk analisis kita.



A top-down view of a person with a beard wearing a blue button-down shirt, sitting at a wooden desk and typing on a silver laptop. To the left of the laptop is a white cup of coffee with a spoon. To the right is a small potted basil plant. A large white geometric graphic, resembling a stylized 'X' or a mountain peak, is overlaid on the laptop and plant. The background is a wooden desk, and the bottom of the image shows dark wooden slats.

THANK
YOU