

# Week 6 - API & Database

# Presensi Clock IN



**SMA/MA/SMK**

<https://bit.ly/AbsensiHSC>



**Universitas**

<https://bit.ly/AbsensiUni>



**Guru/Dosen**

<https://bit.ly/AbsensiTeach>

# SIC6 Checklist - Stage 2

<b>Week 3</b>	Clock in	Individual	CLOSED
	Pre-test	Individual	CLOSED
	Clock out	Individual	CLOSED
<b>Week 4</b>	Clock in	Individual	CLOSED
	Clock out	Individual	CLOSED
	Challenge 1	Individual	CLOSED
<b>Week 5</b>	Clock in	Individual	CLOSED
	Clock out	Individual	CLOSED
	Assignment 1: Idea Proposal	Team	CLOSED
<b>Week 6</b>	Clock in	Individual	15 Feb (2x)
	Clock out	Individual	15 Feb (2x)
	Challenge 2	Individual	15 Feb (23:59 WIB)
<b>Week 7</b>	Clock in	Individual	Not Open Yet
	Post-test	Individual	Not Open Yet
	Clock out	Individual	Not Open Yet
<b>Mentoring &amp; Final Project</b>	Submit Mentoring Questions	Individual / Team	Not Open Yet
	Assignment 2: IoT Prototype project	Team	Not Open Yet
	<b>Photo Challenge on Threads [NEW]</b>	<b>Team</b>	<b>Not Open Yet</b>
	Survey Stage 2	Individual	Not Open Yet



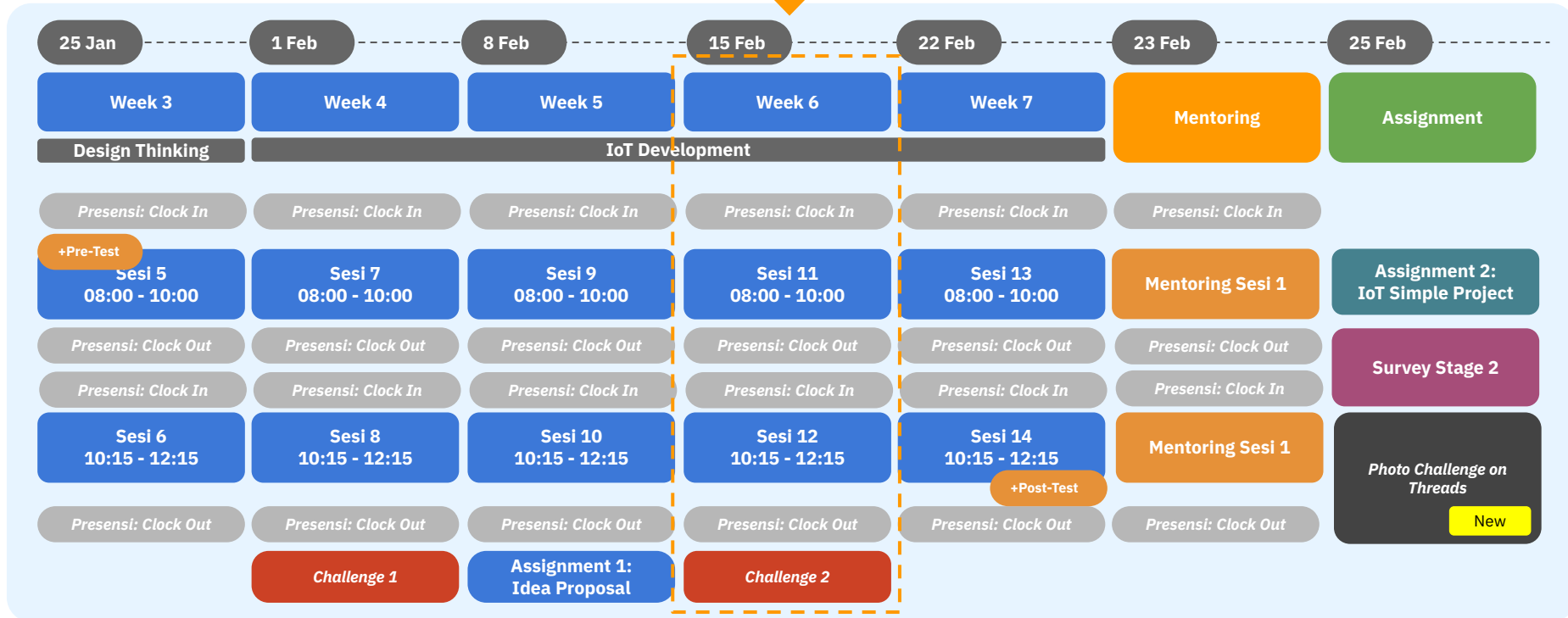
# Stage 2 - Overview



1004  
Participants

Hardware  
Prototyping

- Design Thinking
- Project Planning
- IoT Prototyping using ESP32 & Ubidots



## Stage 2

# Samsung Innovation Campus

Batch 6 - 2024/2025

AI  in Everyday Life



Peserta Samsung  
Innovation Campus  
Batch 6.



15-25 Februari 2025

# SIC6

## Photo Challenge



### Mekanisme

1. Upload foto-foto momen selama SIC Stage 2 di salah satu platform media sosial kamu seperti IG Feed, Tiktok, Linkedin, Facebook dan X.
2. Ceritakan pengalaman, tantangan dan kisah menarik di caption selama mengikuti SIC6 Stage 2.
3. Gunakan hastag:
  - #SIC6Stage2
  - #SIC\_Indonesia\_2025
  - #enabling\_people
  - #samsunginnovationcampus
4. Mention akun @hacktiv8id



E-wallet senilai  
Rp200,000  
untuk 5 pemenang!

Link Google form untuk konfirmasi keikutsertaan  
photo challenge akan diberikan next week



### Ketentuan Lainnya

1. Karya harus orisinal dan tidak melanggar hak cipta.
2. Panitia berhak menggunakan foto dan cerita kamu untuk promosi program.
3. Pemilihan pemenang akan dilakukan secara acak. Keputusan pemilihan pemenang bersifat final.



### Contoh Postingan



peserta\_photochallenge



Dalam mengerjakan challenge ini aku menghadapi berbagai macam tantangan, yang juga menjadi sebuah kesempatan belajar untuk aku!

\*Pemenang diumumkan pada 3 Maret 2025 di akun Instagram @hacktiv8id.

Supported by



**HACKTIV8**

# What will we learn?

1. Intro to API
2. Install postman
3. Intro Flask
4. Flask Routing
5. GET & POST API Flask
6. PyMongo Intro
7. Connect WITH PyMongo
8. Simple Query PyMongo

# Intro ( What is API )



<https://www.youtube.com/watch?v=s7wmiS2mSXY&t=127s>

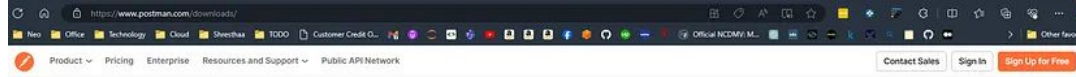
# Intro ( What is API )



<https://www.youtube.com/watch?v=-0MmWEYR2a8>

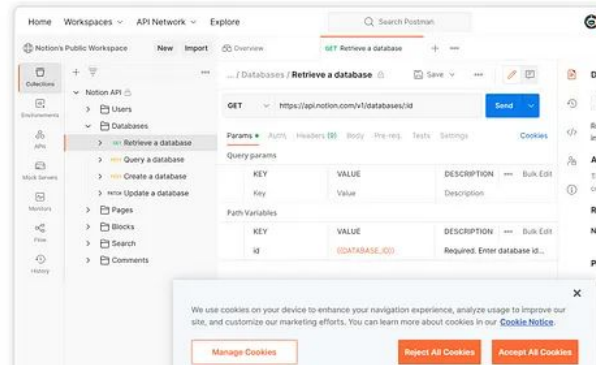
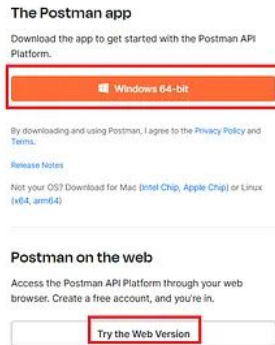


# Installation Postman



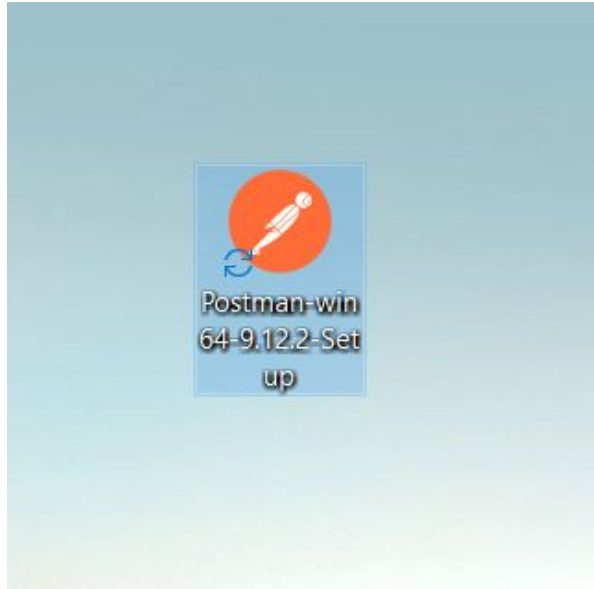
## Download Postman

Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.



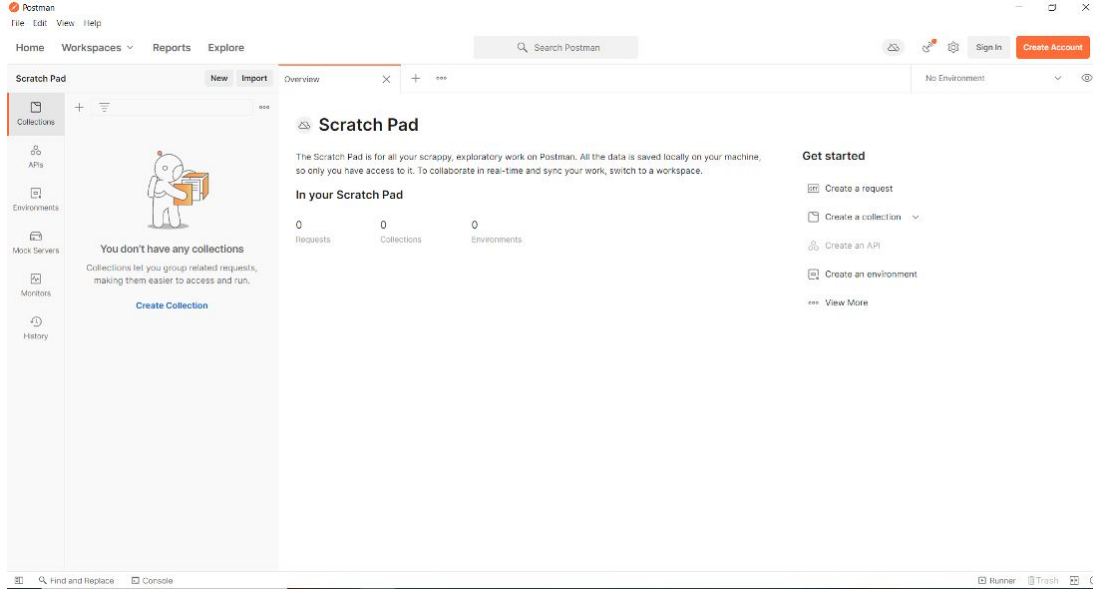
- ❏ Install postman: <https://www.postman.com/downloads/>
- ❏ Click on the “Windows 64-bit” button. This will download the postman exe file which you can then execute on Windows.

# Installation Postman



- ❑ Click on post windows setup
- ❑ Wait until installation process is complete

# Installation Postman



Open the app, and this is how the interface should like

# Intro Flask

**Flask** is a widely adopted Python framework for building web applications. It allows Python developers to use their preferred language with all of its assets while building scalable and fast-to-start Python web applications.



# Intro Flask

- Flask installation and getting started

## Install Flask

Within the activated environment, use the following command to install Flask:

```
$ pip install Flask
```

<https://flask.palletsprojects.com/en/2.1.x/installation/>

# Intro Flask

- Flask minimum application

```
#!/usr/bin/python
# coding: utf-8
from flask import Flask
```

Import flask library

```
app = Flask(__name__)
```

Initiate Flask Instances

```
@app.route('/')
def entry_point():
    return 'Hello World!'
```

Flask Routing

```
if __name__ == '__main__':
    app.run(debug=True)
```

Run Flask app

# Flask Routing

- Flask Routing
  - Dengan fungsi `route()` kita dapat membuat beberapa endpoint untuk aplikasi Flask kita. Sehingga kita menentukan sebuah URL akan menuju pada fungsi yang mana

Use the **`route()`** decorator to bind a function to a URL.

```
@app.route('/')
def index():
    return 'Index Page'

@app.route('/hello')
def hello():
    return 'Hello, World'
```

# GET & POST API Flask

- Flask HTTP Method
  - Flask HTTP Method membuat kita bisa menentukan metode HTTP apa yang diperbolehkan dalam routes dan apa yang kita ingin lakukan dengan metode tersebut

## HTTP Methods

Web applications use different HTTP methods when accessing URLs. You should familiarize yourself with the HTTP methods as you work with Flask. By default, a route only answers to `GET` requests. You can use the `methods` argument of the `route()` decorator to handle different HTTP methods.

```
from flask import request

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        return do_the_login()
    else:
        return show_the_login_form()
```



# GET & POST API Flask

- Flask Processing Data
  - Dengan library requests bawaan dari Flask kita dapat memproses data yang dikirimkan melalui API

```
from flask import request
```

Dan untuk mengakses data tersebut kita dapat menggunakan *syntax*:

```
body = request.get_json() # mendapatkan request body dalam JSON  
params = request.args.get('params') # mendapatkan requests parameters dengan key params  
form = request.form.get('form') # mendapatkan requests form data dengan nama form
```

# GET & POST API Flask

- Flask JSON Response
  - Flask menyediakan interface yang mudah untuk kita mengembalikan response JSON, dapat menggunakan function `jsonify()` atau cukup mengembalikannya dalam bentuk dictionary

## APIs with JSON

A common response format when writing an API is JSON. It's easy to get started writing such an API with Flask. If you return a `dict` from a view, it will be converted to a JSON response.

```
@app.route("/me")
def me_api():
    user = get_current_user()
    return {
        "username": user.username,
        "theme": user.theme,
        "image": url_for("user_image", filename=user.image),
    }
```

**STOP, Now it's time for  
Quick DEMO!**



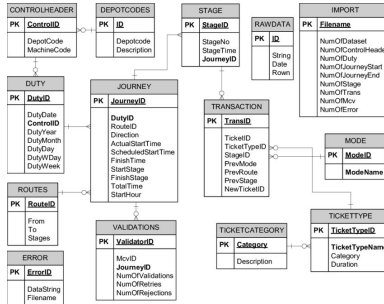
# What is Database(s)?

“A database is an organized collection of structured information, or data, typically stored electronically in a computer system” - Oracle



Source: dataversity.net

# Types of Databases



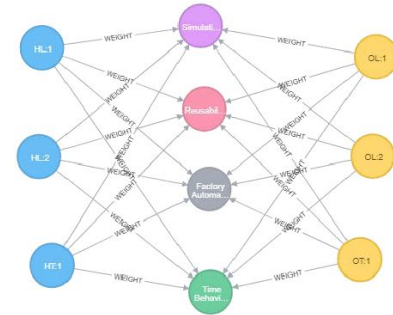
## Relational

- Defines relationship in form of tables
- Data accessed using SQL



## Non-Relational (NoSQL)

- Stores semi-structured and unstructured data
- Horizontally scalable (distributed)



## Graph

- Defines relationship on form of nodes
- Used for highly connected data relationships

# Types of Databases

Database software is called a Database Management System (DBMS)



## Relational



## Non-Relational (NoSQL)



## Graph

# Relational Database

- Table consists of rows and columns
- Tables might have columns in common that have relationship
- Each column in a table have a schema with data type

<i>Product_code</i>	<i>Description</i>	<i>Price</i>
A416	Nails, box	\$0.14
C923	Drawing pins, box	\$0.08

<i>Invoice_code</i>	<i>Invoice_line</i>	<i>Product_code</i>	<i>Quantity</i>
3804	1	A416	10
3804	2	C923	15

# Non-Relational Database

- Document-oriented
- JSON like



Key	Document
1001	<pre>{   "CustomerID": 99,   "OrderItems": [     { "ProductID": 2010,       "Quantity": 2,       "Cost": 520     },     { "ProductID": 4365,       "Quantity": 1,       "Cost": 18     }   ],   "OrderDate": "04/01/2017" }</pre>
1002	<pre>{   "CustomerID": 220,   "OrderItems": [     { "ProductID": 1285,       "Quantity": 1,       "Cost": 120     }   ],   "OrderDate": "05/08/2017" }</pre>



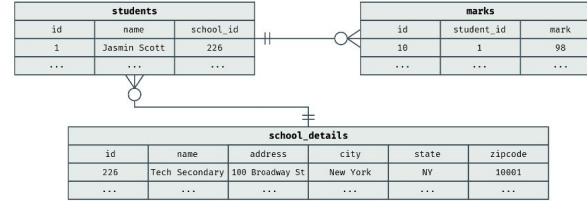
## MongoDB

```
{
  "_id": 1,
  "student_name": "Jasmin Scott",
  "school": {
    "school_id": 226,
    "name": "Tech Secondary",
    "address": "100 Broadway St",
    "city": "New York",
    "state": "NY",
    "zipcode": "10001"
  },
  "marks": [98, 93, 95, 88, 100],
}
```

mongo

```
> db.students.find({"student_name":
  "Jasmin Scott"})
```

## SQL

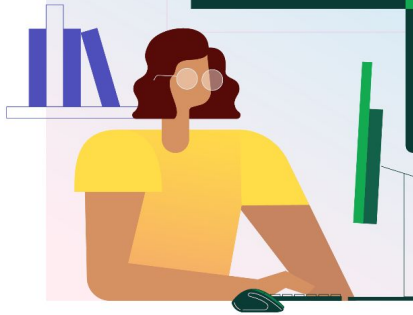


## Results

name	mark	school_name	city
Jasmin Scott	98	Tech Secondary	New York
...	...	...	...

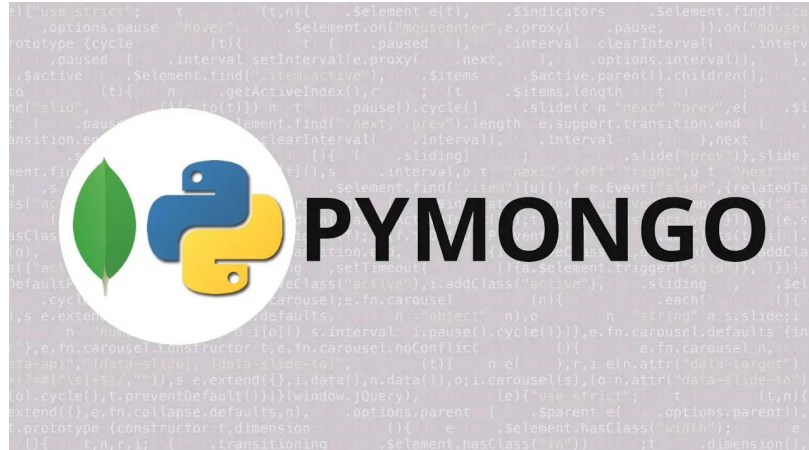
sql

```
SELECT s.name, m.mark, d.name as "school name",
  d.city
FROM students s
INNER JOIN marks m ON s.id = m.student_id
INNER JOIN school_details d ON s.school_id = d.id
WHERE s.name = "Jasmin Scott";
```



# Intro PyMongo

**PyMongo** is a Python distribution containing tools for working with [MongoDB](#), and is the recommended way to work with MongoDB from Python



# Intro PyMongo

- Installing PyMongo

Standard Installation:

```
$ python3 -m pip install pymongo
```

With MongoDB Atlas Installation:

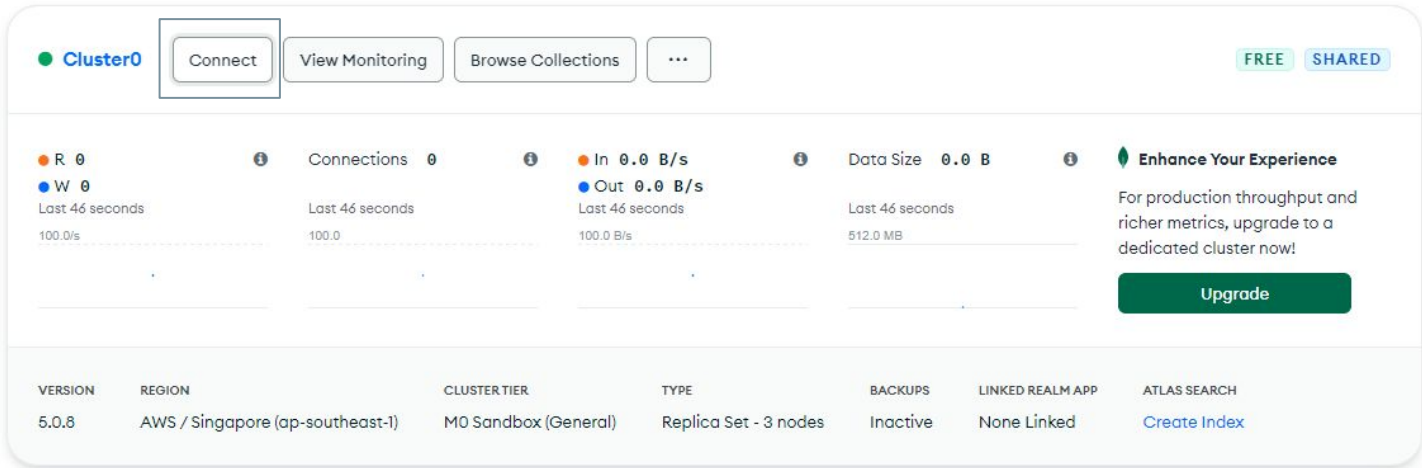
Support for mongodb+srv:// URIs requires **dnspython**:

```
$ python3 -m pip install "pymongo[srv]"
```

<https://pymongo.readthedocs.io/en/stable/installation.html>

# Connect with PyMongo

- Connect with Atlas Cluster



# Connect with PyMongo

- **Connect with Atlas Cluster**

Connect to GettingStarted

✓ Setup connection security

Choose a connection method

Connect

Choose a connection method [View documentation](#)

Get your pre-formatted connection string by selecting your tool below.



**Connect with the MongoDB Shell**

Interact with your cluster using MongoDB's interactive Javascript interface



**Connect your application**

Connect your application to your cluster using MongoDB's native drivers



**Connect using MongoDB Compass**

Explore, modify, and visualize your data with MongoDB's GUI



Go Back

Close

# Connect with PyMongo

- Connect with Atlas Cluster

## Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

### 1 Select your driver and version

DRIVER

Python

VERSION

3.6 or later

### 2 Add your connection string into your application code

☒ Include full driver code example

```
client = pymongo.MongoClient("mongodb+srv://thirafiwan:
<password>@cluster0.rjj49ep.mongodb.net/?retryWrites=true&w=majority")
db = client.test
```

Replace **<password>** with the password for the **thirafiwan** user. Ensure any option params are **URL encoded**.

Having trouble connecting? [View our troubleshooting documentation](#)

# Connect with PyMongo

- **Connect with Atlas Cluster**

```
import pymongo # meng-import library pymongo yang sudah kita install
client =
pymongo.MongoClient("mongodb+srv://jphartogi:BhdvAX9DPH9kjsTx@gettingstarted
.zo2se.mongodb.net/GettingStarted?retryWrites=true&w=majority")
db = client.test
print(db)
```

# Query with PyMongo

- **Insert Data**

- Jika kita ingin membuat sebuah *document* dalam MongoDB, maka kita harus membuat sebuah *dictionary* dimana *keys* adalah *column headers* dan *values* adalah *attribute* dari data yang kita ingin simpan dalam *database*.
- Kita dapat menggunakan *function* `collection.insert_many()` untuk menyimpan beberapa dokumen sekaligus, atau `collection.insert_one()` untuk menyimpan satu data saja. Disini kita akan menggunakan *database* contoh yang telah kita buat sebelumnya.



# Query with PyMongo

- **Insert Data**

```
import pymongo # meng-import library pymongo yang sudah kita install
client = pymongo.MongoClient("MASUKAN ID KALIAN")
db = client['MyDatabase'] # ganti sesuai dengan nama database kalian
my_collections = db['MyCollection'] # ganti sesuai dengan nama collections kalian

# Data yang ingin dimasukkan
murid_1 = {'nama': 'John Doe', 'Jurusan': 'IPS', 'Nilai': 90}
murid_2 = {'nama': 'Jane Doe', 'Jurusan': 'IPA', 'Nilai': 85}

results = my_collections.insert_many([murid_1, murid_2])
print(results.inserted_ids) # akan menghasilkan ID dari data yang kita masukkan
```

# Query with PyMongo

- **Read Data**

- Setelah data sudah masuk ke dalam *database*, maka kita juga bisa membaca data tersebut menggunakan PyMongo. Untuk membaca seluruh data kita dapat menggunakan *function* `collections.find()` untuk membaca seluruh data dalam *collections*.

# Query with PyMongo

- **Read Data**

```
import pymongo # meng-import library pymongo yang sudah kita install
client = pymongo.MongoClient("MASUKAN ID KALIAN")
db = client['MyDatabase'] # ganti sesuai dengan nama database kalian
my_collections = db['MyCollection'] # ganti sesuai dengan nama
collections kalian

for x in my_collections.find():
    print(x)
```

**STOP, Now it's time for  
Quick DEMO!**



# Challenge!

Buatlah sebuah aplikasi Flask yang terkoneksi kedalam MongoDB dan terdiri dari beberapa kondisi berikut

1. API dengan
  - a. route /sensor1, metode POST
  - b. 2 buah data ( buat dummy i.e temperature, kelembapan ) dan timestamp diparsing dari parameter API
  - c. Simpan data pada database

# Challenge!

Buatlah sebuah aplikasi Flask yang terkoneksi kedalam MongoDB dan terdiri dari beberapa kondisi berikut

1. API dengan
  - a. route `/sensor1/temperature/avg` dan `/sensor1/kelembapan/avg`, metode GET
  - b. Mengambil seluruh data temperature/kelembapan dari database
  - c. Return nilai rata-rata dari seluruh data tersebut

# Q&A



# Presensi Clock OUT



**SMA/MA/SMK**

<https://bit.ly/AbsensiHSC>



**Universitas**

<https://bit.ly/AbsensiUni>



**Guru/Dosen**

<https://bit.ly/AbsensiTeach>





# BREAK TIME

Kita mulai lagi pukul 10:15 ya~

# Presensi Clock IN



**SMA/MA/SMK**

<https://bit.ly/AbsensiHSC>



**Universitas**

<https://bit.ly/AbsensiUni>



**Guru/Dosen**

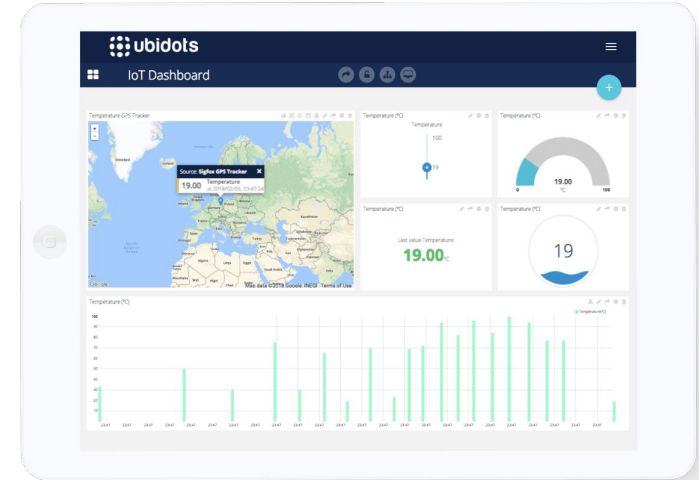
<https://bit.ly/AbsensiTeach>

# Assignment #2 - Stage 2

Buatlah sebuah dashboard di platform ubidots dengan minimal 2 visualisasi ( bisa berupa chart/grafik dll ) dan script micropython untuk mengirimkan data sensor ( boleh menggunakan sensor apa saja )

- Dapatkan data sensor ultrasonic/menggunakan sensor yang ada/yang kalian pakai untuk project kalian
- Kirimkan data tersebut ke ubidots dashboard melalui REST API/MQTT
- Tampilkan data sensor dalam sebuah dashboard (minimal 2 buah visualisasi, contohnya grafik & gauge)!

Upload screenshot hasil dashboard kalian beserta dengan code python ke dalam repo github kalian!



Read more : <https://ubidots.com/>

## Challenge 2 (Stage 2)

Deadline: 15 Feb 23:59 WIB

SMA/SMK/MA



<https://bit.ly/Challenge2HSC>

Universitas



<https://bit.ly/Challenge2UNI>

# Photo Challenge

**Deadline: 25 Feb 23:59 WIB**

**SMA/SMK/MA**



**Universitas**



<https://bit.ly/PhotoChallenge2HSC>

<https://bit.ly/PhotoChallenge2UNI>

# SIC6 Checklist - Stage 2

<b>Week 3</b>	Clock in	Individual	CLOSED
	Pre-test	Individual	CLOSED
	Clock out	Individual	CLOSED
<b>Week 4</b>	Clock in	Individual	CLOSED
	Clock out	Individual	CLOSED
	Challenge 1	Individual	CLOSED
<b>Week 5</b>	Clock in	Individual	CLOSED
	Clock out	Individual	CLOSED
	Assignment 1: Idea Proposal	Team	CLOSED
<b>Week 6</b>	Clock in	Individual	15 Feb (2x)
	Clock out	Individual	15 Feb (2x)
	Challenge 2	Individual	15 Feb (23:59 WIB)
<b>Week 7</b>	Clock in	Individual	Not Open Yet
	Post-test	Individual	Not Open Yet
	Clock out	Individual	Not Open Yet
<b>Mentoring &amp; Final Project</b>	Submit Mentoring Questions	Individual / Team	Not Open Yet
	Assignment 2: IoT Prototype project	Team	Not Open Yet
	Photo Challenge	Individual / Team	Not Open Yet
	Survey Stage 2	Individual	Not Open Yet



# Presensi Clock OUT



**SMA/MA/SMK**

<https://bit.ly/AbsensiHSC>



**Universitas**

<https://bit.ly/AbsensiUni>



**Guru/Dosen**

<https://bit.ly/AbsensiTeach>

## Stage 2

# Samsung Innovation Campus

Batch 6 - 2024/2025

AI  in Everyday Life



Peserta Samsung  
Innovation Campus  
Batch 6.



15-25 Februari 2025

# SIC6

## Photo Challenge



### Mekanisme

1. Upload foto-foto momen selama SIC Stage 2 di salah satu platform media sosial kamu seperti IG Feed, Tiktok, LinkedIn, Facebook dan X.
2. Ceritakan pengalaman, tantangan dan kisah menarik di caption selama mengikuti SIC6 Stage 2.
3. Gunakan hastag:
  - #SIC6Stage2
  - #SIC\_Indonesia\_2025
  - #enabling\_people
  - #samsunginnovationcampus
4. Mention akun @haktiv8id



E-wallet senilai  
Rp200,000  
untuk 5 pemenang!



### Ketentuan Lainnya

1. Karya harus orisinal dan tidak melanggar hak cipta.
2. Panitia berhak menggunakan foto dan cerita kamu untuk promosi program.
3. Pemilihan pemenang akan dilakukan secara acak. Keputusan pemilihan pemenang bersifat final.



### Contoh Postingan



peserta\_photochallenge



Dalam mengerjakan challenge ini aku menghadapi berbagai macam tantangan, yang juga menjadi sebuah kesempatan belajar untuk aku!

\*Pemenang diumumkan pada 3 Maret 2025 di akun Instagram @haktiv8id.

Supported by



**HAKTIV8**