

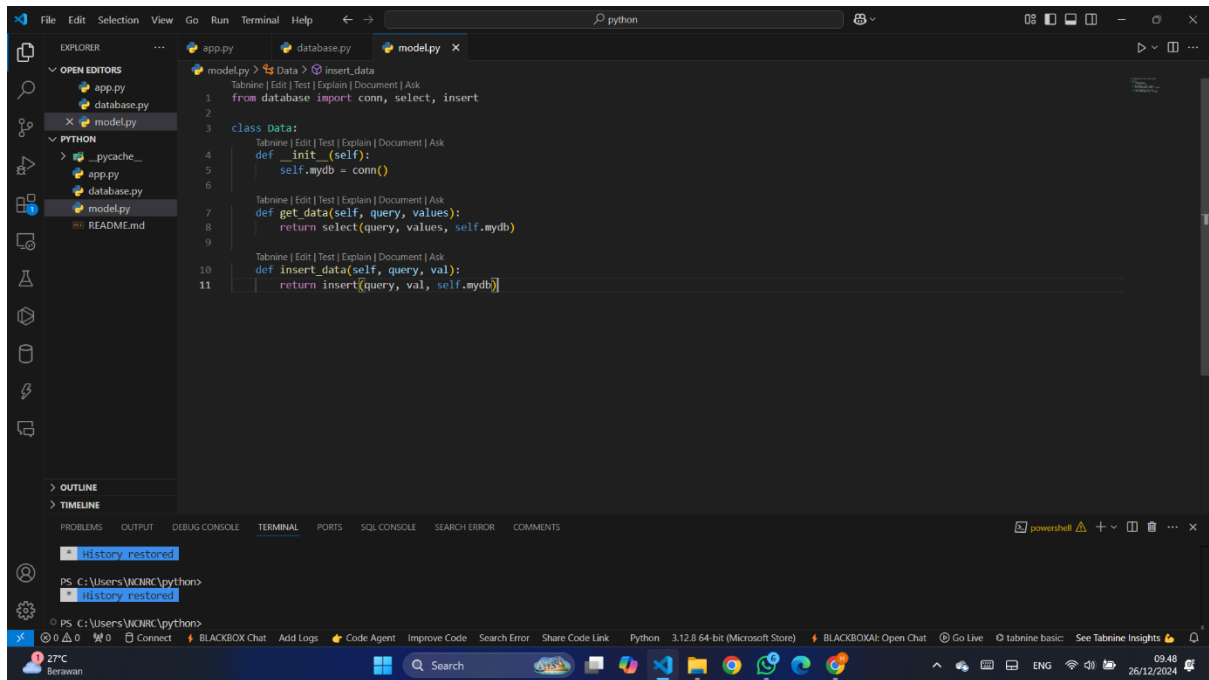
Nama : Haris Azhari Ramadhan

NIM : 230741111

Mata Kuliah : KPT

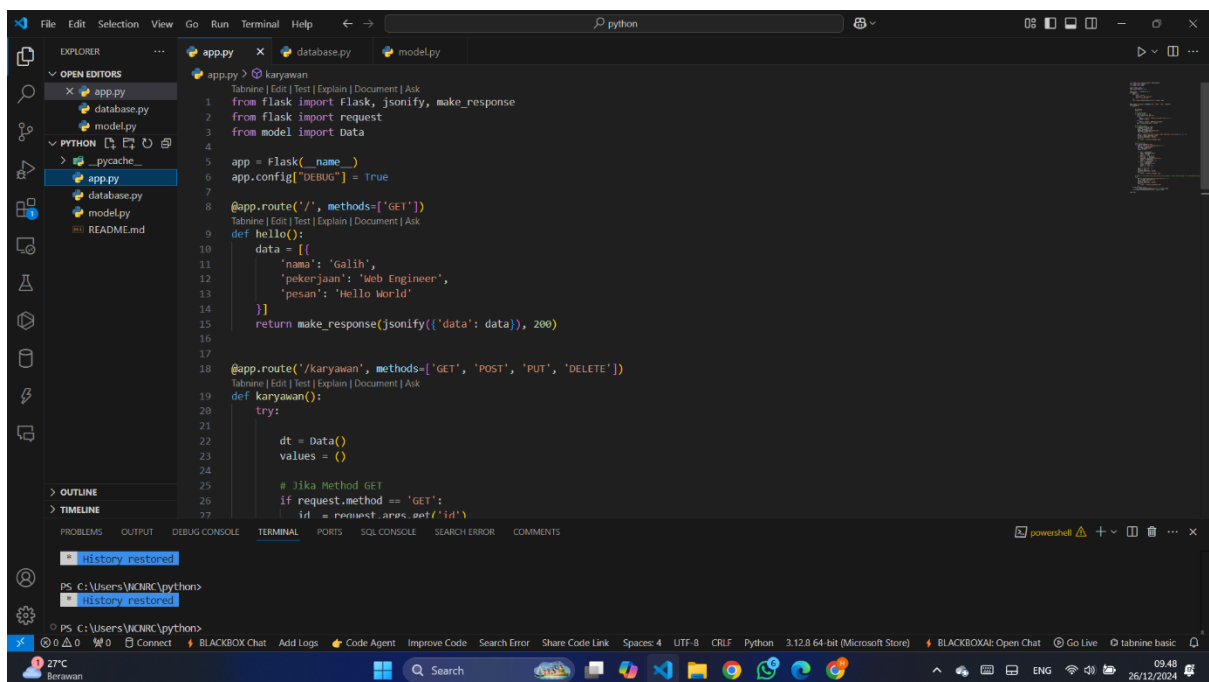
Dokumentasi Hasil Pengerjaan API-Flask menggunakan postman

Model.py



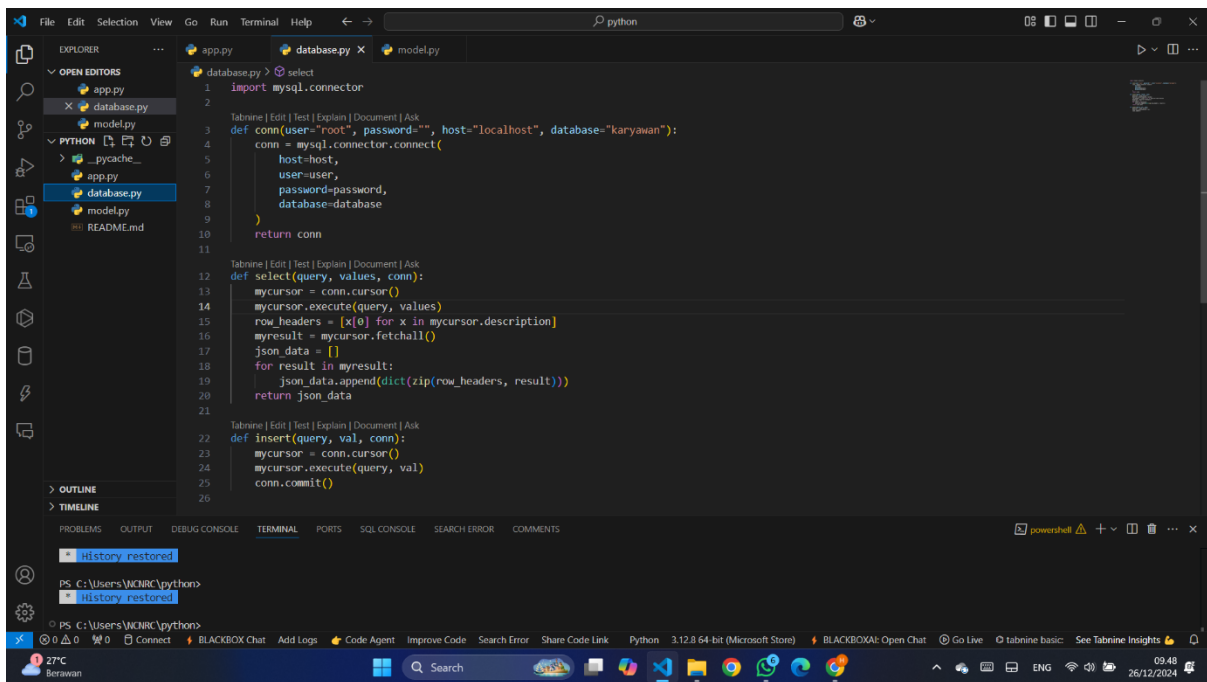
```
1 from database import conn, select, insert
2
3 class Data:
4     def __init__(self):
5         self.mydb = conn()
6
7     def get_data(self, query, values):
8         return select(query, values, self.mydb)
9
10    def insert_data(self, query, val):
11        return insert(query, val, self.mydb)
```

App.py



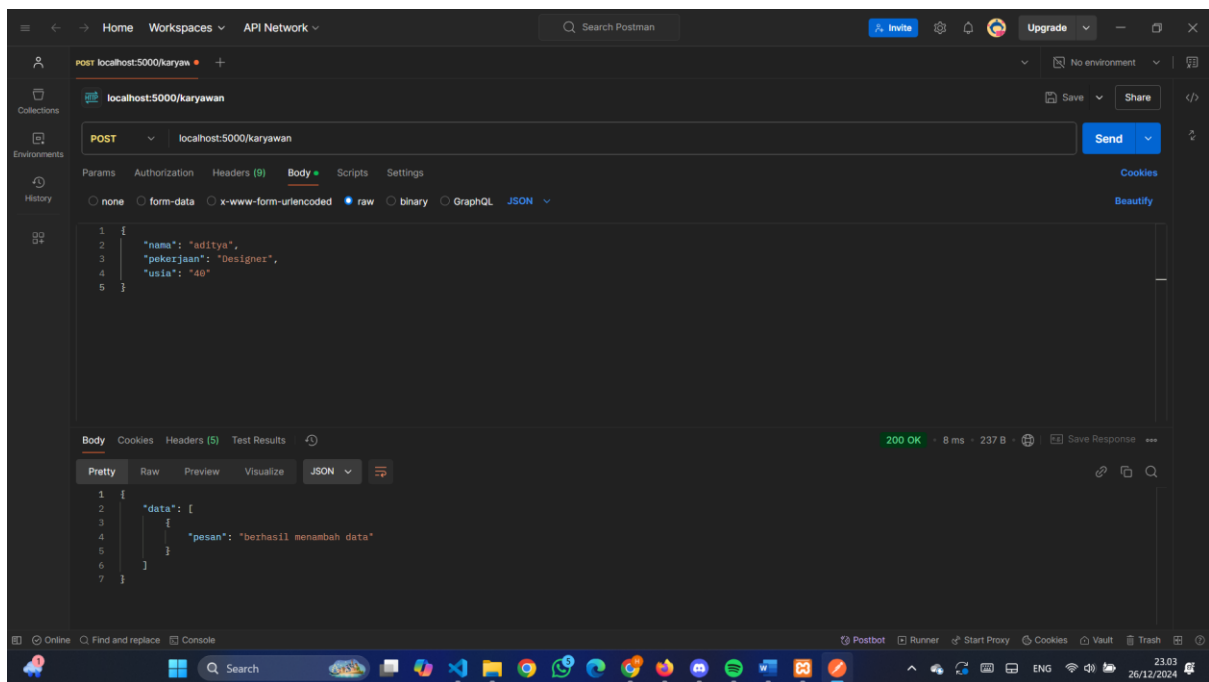
```
1 from flask import Flask, jsonify, make_response
2 from flask import request
3 from model import Data
4
5 app = Flask(__name__)
6 app.config["DEBUG"] = True
7
8 @app.route('/', methods=['GET'])
9 def hello():
10     data = [{
11         'nama': 'Galih',
12         'pekerjaan': 'Web Engineer',
13         'pesan': 'Hello World'
14     }]
15     return make_response(jsonify({'data': data}), 200)
16
17 @app.route('/karyawan', methods=['GET', 'POST', 'PUT', 'DELETE'])
18 def karyawan():
19     try:
20         dt = Data()
21         values = ()
22
23         # Jika Method GET
24         if request.method == 'GET':
25             id = request.args.get('id')
```

Database.py

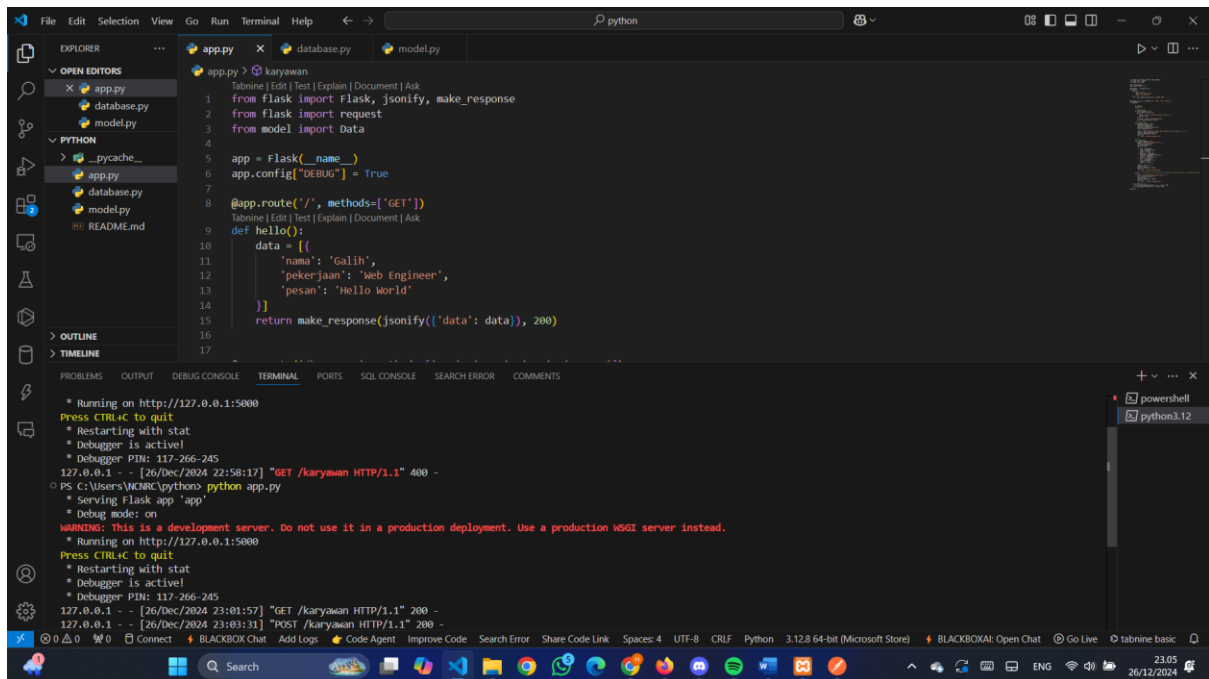


```
1 import mysql.connector
2
3 def conn(user="root", password="", host="localhost", database="karyawan"):
4     conn = mysql.connector.connect(
5         host=host,
6         user=user,
7         password=password,
8         database=database
9     )
10    return conn
11
12 def select(query, values, conn):
13     mycursor = conn.cursor()
14     mycursor.execute(query, values)
15     row_headers = [x[0] for x in mycursor.description]
16     myresult = mycursor.fetchall()
17     json_data = []
18     for result in myresult:
19         json_data.append(dict(zip(row_headers, result)))
20    return json_data
21
22 def insert(query, val, conn):
23     mycursor = conn.cursor()
24     mycursor.execute(query, val)
25     conn.commit()
26
```

Postman



Run python



The screenshot shows a Visual Studio Code editor window with a Python project. The Explorer sidebar on the left shows a file structure with `app.py`, `database.py`, `model.py`, and `README.md`. The main editor displays the `app.py` file, which is a Flask application. The code imports `Flask`, `jsonify`, `make_response`, and `request` from the `flask` module, and `Data` from the `model` module. It creates a Flask app, configures debug mode, and defines a `hello()` endpoint that returns a JSON response with employee data. The terminal at the bottom shows the command `python app.py` being executed, and the application running on `http://127.0.0.1:5000`. It displays the debug mode status and the response for a GET request to `/karyawan`, which is a 400 status code. A warning message is also visible in the terminal.

```
File Edit Selection View Go Run Terminal Help python
EXPLORER
  app.py x database.py model.py
  OPEN EDITORS
    app.py
    database.py
    model.py
  PYTHON
    _pycache_
    app.py
    database.py
    model.py
    README.md
  OUTLINE
  TIMELINE

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE SEARCH ERROR COMMENTS
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-266-245
127.0.0.1 - - [26/Dec/2024 22:58:17] "GET /karyawan HTTP/1.1" 400 -
PS C:\Users\WICNIC\python> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 117-266-245
127.0.0.1 - - [26/Dec/2024 23:01:57] "GET /karyawan HTTP/1.1" 200 -
127.0.0.1 - - [26/Dec/2024 23:03:31] "POST /karyawan HTTP/1.1" 200 -
Spaces: 4 UTF-8 CRLF Python 3.12.8 64-bit (Microsoft Store) BLACKBOXAI: Open Chat Go Live tabnine basic
23.05 26/12/2024
```