Pengembangan Web Praktik

# Flask View Template

**Endang Anggiratih, S.T., M.Cs.**

September, 2024

# Materi perkuliahan

1. Perkenalan, Silabus, Pengenalan Konsep web dan server

2. Dasar UI/UX Design, Instalasi Flask

3. Integrasi, HTML & CSS Dasar

4. Integrasi Flask View Template

5. Integrasi Flask Bootsrap dan Mysql CRUD

6. Penerapan Session pada Flask

7. Responsi

8. Konsep microservices dan Rest API pada Flask

9. Migration Model CRUD Rest API pada Flask

10. Penerapan API JWT pada Flask

11. Integrasi Flask dan Javascript

12. JavaScript Custom Components

13. Presentasi Projek Besar

14. Presentasi Projek Besar

# Jinja2 Template Engine

Flask uses Jinja2 as its template engine by default. Jinja2 allows you to **write Python-like expressions inside HTML** which makes it easier to generate dynamic HTML based on data provided by your Flask application.

# Standard Jinja Template

```
from jinja2 import Template

variable_name = "dunia"
template = Template("Halo {{ name }}!")    ◄ Template
print(template.render(name=variable_name))
# Halo dunia!    ▲
```

Output ►

Render value

# Jinja Conditions

```python
from jinja2 import Template

variable_name = "dunia"
template = Template("""
{%- if name == "dunia" -%}       ◀ Condition
    Halo {{ name | upper }}!
{%- elif name == "world" -%}
    Hello {{ name | upper }}!
{%- else -%}
    Hello {{ name }}!               ◀ Statement
{%- endif -%}
""")
print(template.render(name=variable_name))
# Halo DUNIA!          ◀ Output
```

## Try it also with

**{% ‹cond› %}**

return white space and newline

**{%- ‹cond› %}**

trim space before condition

**{% ‹cond› -%}**

trim space after condition

**{%- ‹cond› -%}**

Neither to return white space nor newline

# Jinja Loop

```
from jinja2 import Template

variable_name = "dunia"
template = Template("""
{% for n in [1,2,3] -%}
    {{ n }}
{% endfor -%}
""")
print(template.render(name=variable_name))
# 1 \n 2 \n 3
```

Iterator

Output

## Try it also with

**{% ‹cond› %}**

return white space and newline

**{%- ‹cond› %}**

trim space before condition

**{% ‹cond› -%}**

trim space after condition

**{%- ‹cond› -%}**

Neither to return white space nor newline

# How to implement Jinja inside Flask?

```python
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def index():
    variable_name = "dunia"
    return render_template('jinja.html',
                           name=variable_name)

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

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-widt
    <title>Document</title>
</head>
<body>
    <h1>Halo {{ name | upper }}</h1>
</body>
</html>
```

127.0.0.1:5000

## Halo DUNIA

# Inheritance Template

Child

Parent

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <title>{% block title %}{% endblock %}</title>
5  </head>
6  <body>
7    {% block body %}     ◀ Template block
8    {% endblock %}
9  </body>
10 </html>
11
```

```
1  {% extends 'base.html' %}
2  {% block title %}Home Page{% endblock %}
3  {% block body %}
4    <h1>Welcome {{ name }}</h1>   ◀ Insert value
5    <p>Welcome to our website!</p>
6  {% endblock %}
7
```

127.0.0.1:5000

## Welcome dunia

Welcome to our website!

Output

# Let's dive into Flask!

Pengembangan Web Praktik