

Nama : Muhammad Riveldo Hermawan Putra

NIM : 122140037

Tugas Tambahan Nilai UTS

Link Github: <https://github.com/MuhammadRiveldo/Tugas-Tambahan-UTS>

01: Single-File Web Applications

The screenshot shows a Windows desktop environment with three main windows open:

- localhost:6543**: A browser window displaying the text "Hello World!".
- localhost:6543**: Another browser window showing a file structure for a "quick_tutorial" project under "01 Single-File Web App".
- Terminal**: A terminal window titled "python" showing pip upgrade logs and a command to run "hello_world.py".

The taskbar at the bottom includes icons for File Explorer, Task View, Start, Search, and several pinned application icons.

02: Python Packages for Pyramid Applications

localhost:6543

Hello World!

__init__.py

```
quick_tutorial > package > tutorial > app.py > ...
```

```
9 if __name__ == '__main__':
10     with Configurator() as config:
11         config.add_route('hello', '/')
12         config.scan(modules=[hello_world])
13     app = config.make_app()
14     app.on_startup(_adjustments_params)
15     serve(app, host='0.0.0.0', port=6543)
```

TERMINAL

```
python > +
```

```
[Errno 2] No such file or directory
(env) PS /D/LARISIP/Semester 7/Pengembangan Aplikasi Web> cd projects/quick_tutorial/package/tutorial> cd
(env) PS /D/LARISIP/Semester 7/Pengembangan Aplikasi Web> projects/quick_tutorial/package python tutorial/app.py
```

```
D:\LARISIP\Semester 7\Pengembangan Aplikasi Web>projects\quick_tutorial\env\lib\site-packages\pkg_resources\__init__.py:21: UserWarning: pkg_resources is deprecated at an API. See https://setuptools.pypa.io/en/latest/pkg_resources.html. The pkg_resources package is slated for removal as early as 2025-11-30. Refrain from using this package or pin to a specific version.
  import pkg_resources
Incoming request
Incoming request
Incoming request
Incoming request
```

main* 0 0 0 0

Ln 15 Col 1 Spacer 4 UTF 8 CRLF ⌂ Python 3.13.3 (env)

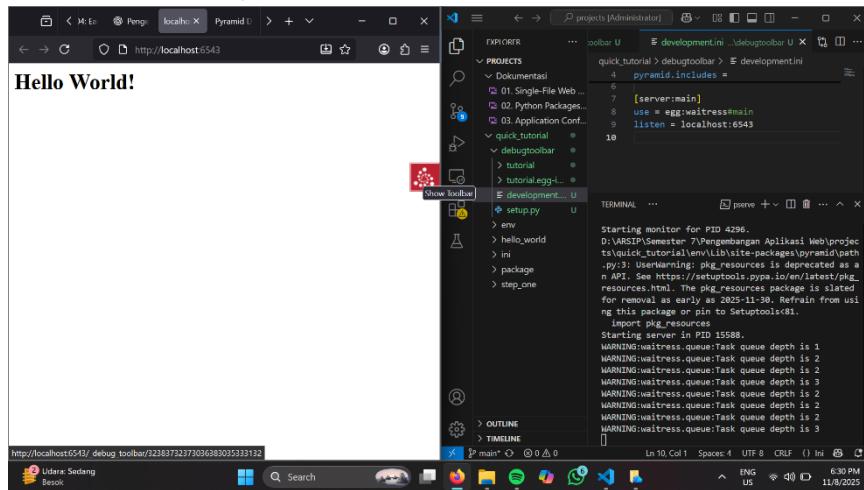
03: Application Configuration with .ini Files

The screenshot shows a Windows desktop environment. In the center, a Microsoft Edge browser window displays the URL `http://localhost:6543`, which shows the text "Hello World!". To the right of the browser is a file explorer window titled "projects [Administrator]". The "PROJECTS" view shows a folder structure for a Python project:

- Dokumentasi
- 01 Single-File Web
- 02 Python Packages
- quick_tutorial
- env
- hello_world
- ini
- tutorial
- __init__.py
- tutorial_egg_info
- development.ini
- setup.py
- > package
- > step_one

The "explorer" tab is selected in the file explorer. Below the file tree, there are sections for "OUTLINE" and "TIMELINE". A status bar at the bottom indicates "Ln 8, Col 1" and "Spaces: 4".

04: Easier Development with debugtoolbar



The screenshot shows the PyCharm IDE interface. The top navigation bar has tabs for 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. Below the navigation bar is the project structure tree, which includes 'Dokumentasi' and 'quick_tutorial' (which contains 'debugtoolbar', 'env', 'hello_world', 'ini', 'package', 'step_one', and 'tutorial'). The main code editor window displays a file named 'development.ini' with the following content:

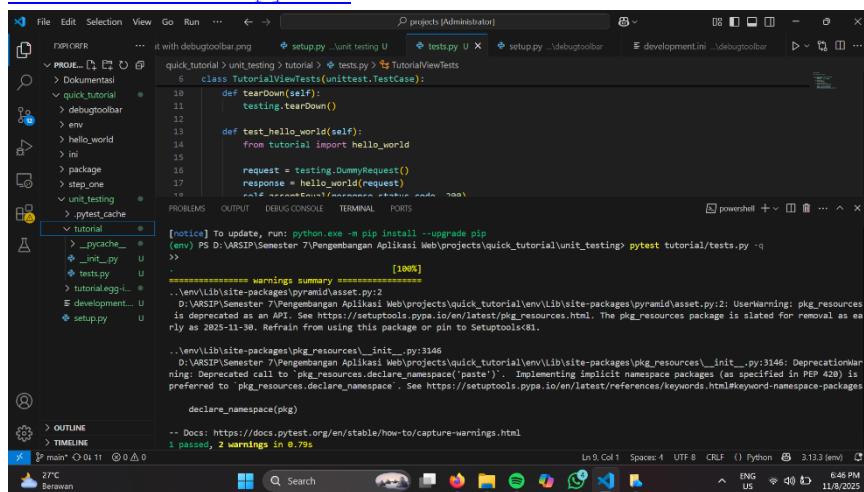
```
development.ini
[server:main]
pyramid.includes =
[server:main]
use = egg:waitress#main
listen = localhost:6543
```

Below the code editor is a terminal window showing the output of a command:

```
Starting monitor for PID 4296.
D:\VARSI\P\Semester 7\Pengembangan Aplikasi Web\projects\quick_tutorial> python -m debugtoolbar
[...]
```

The status bar at the bottom indicates the current file is 'main.py' and the date and time are '11/8/2025 6:30 PM'.

05: Unit Tests and pytest



The screenshot shows the PyCharm IDE interface. The top navigation bar has tabs for 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. Below the navigation bar is the project structure tree, which includes 'Dokumentasi' and 'quick_tutorial' (which contains 'env', 'hello_world', 'ini', 'package', 'step_one', 'unit_testing', and 'tutorial'). The main code editor window displays a file named 'tests.py' with the following content:

```
class TutorialViewTests(unittest.TestCase):
    def tearDown(self):
        testing.tearDown()

    def test_hello_world(self):
        from tutorial import hello_world

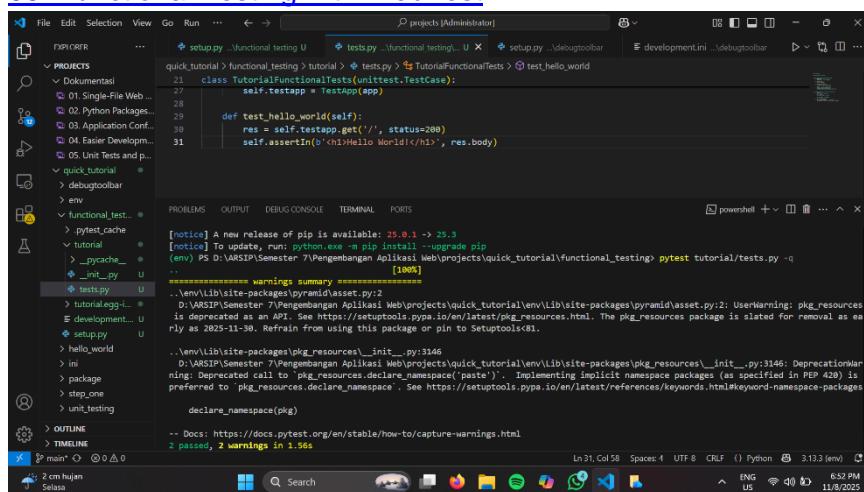
        request = testing.DummyRequest()
        response = hello_world(request)
        self.assertEqual(response['body'], b'Hello World!')
```

The terminal window shows the output of the pytest command:

```
[notice] To update, run: python.exe -m pip install --upgrade pip
[env] PS D:\VARSI\P\Semester 7\Pengembangan Aplikasi Web\projects\quick_tutorial\unit_testing> pytest tutorial/tests.py -q
[...]
1 passed, 2 warnings in 0.79s
```

The status bar at the bottom indicates the current file is 'tests.py' and the date and time are '11/8/2025 6:46 PM'.

06: Functional Testing with WebTest



The screenshot shows the PyCharm IDE interface. The top navigation bar has tabs for 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', etc. Below the navigation bar is the project structure tree, which includes 'Dokumentasi' and 'quick_tutorial' (which contains 'env', 'functional_test', 'ini', 'package', 'step_one', 'tutorial', and 'unit_testing'). The main code editor window displays a file named 'tests.py' with the following content:

```
class TutorialFunctionalTests(unittest.TestCase):
    self.testapp = TestApp(app)

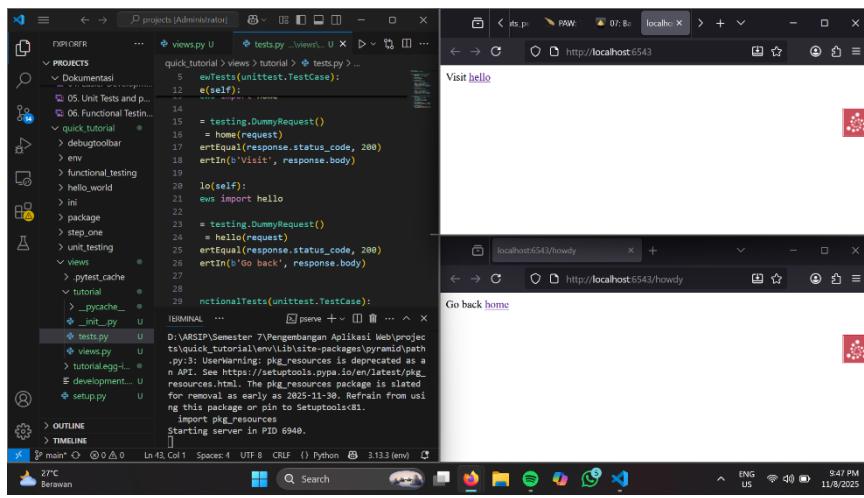
    def test_hello_world(self):
        res = self.testapp.get('/', status=200)
        self.assertIn(b'<h1>Hello World!</h1>', res.body)
```

The terminal window shows the output of the pytest command:

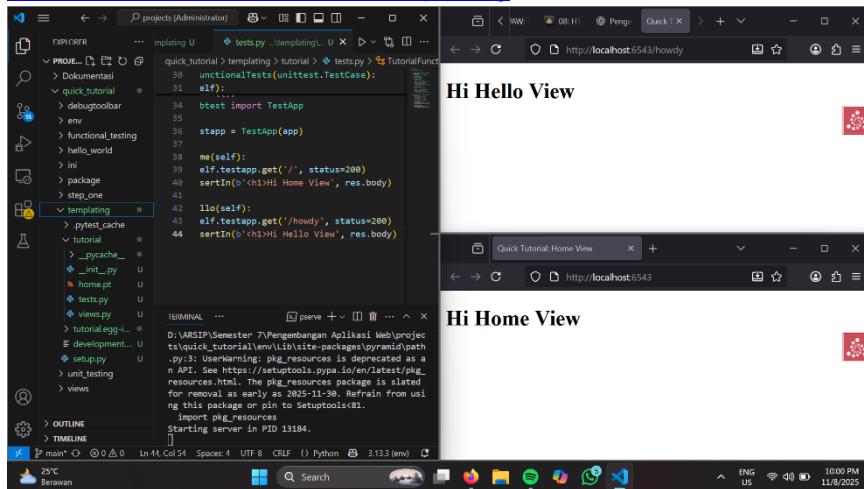
```
[notice] A new release of pip is available: 25.0.1 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
[env] PS D:\VARSI\P\Semester 7\Pengembangan Aplikasi Web\projects\quick_tutorial\functional_testing> pytest tutorial/tests.py -q
[...]
2 passed, 2 warnings in 1.56s
```

The status bar at the bottom indicates the current file is 'tests.py' and the date and time are '11/8/2025 6:52 PM'.

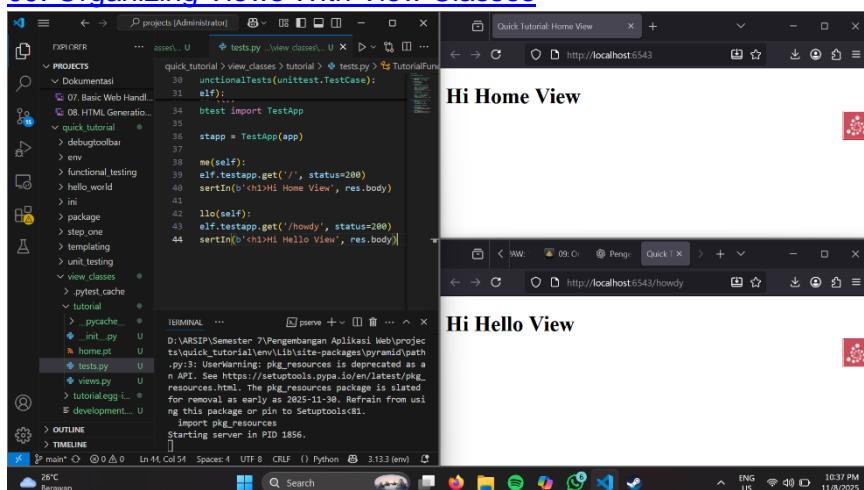
07: Basic Web Handling With Views



08: HTML Generation With Templating



09: Organizing Views With View Classes



10: Handling Web Requests and Responses

The screenshot shows the PyCharm IDE interface. On the left is the project structure for a Python application named 'quick_tutorial'. The 'tests.py' file is open, containing test cases for a 'TutorialViewTests' class. One test, 'test_home', creates a 'DummyRequest' object and checks if the response status is 200. Another test, 'test_plain_without_name', checks if the response status is 400 when no name is provided. The right side of the screen shows two browser windows. The top window displays the URL 'http://localhost:6543/plain' with the message 'URL http://localhost:6543/plain with name: No Name Provided'. The bottom window displays the URL 'http://localhost:6543/plain?name=alice' with the message 'URL http://localhost:6543/plain?name=alice with name: alice'. The status bar at the bottom indicates the server is running in PID 9448.

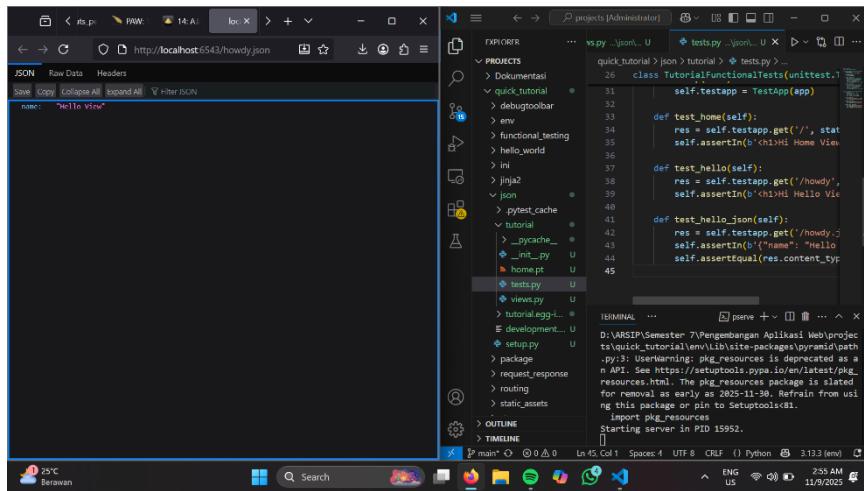
11: Dispatching URLs To Views With Routing

This screenshot shows the PyCharm IDE. The project structure includes a 'views.py' file under the 'routing' directory. The code defines a 'home' view function that returns a dictionary with 'name' and 'body' keys. The right side shows a browser window displaying the URL 'http://localhost:6543/howdy/amy/smith' with the text 'First: amy, Last: smith'. The status bar at the bottom indicates the server is running in PID 17492.

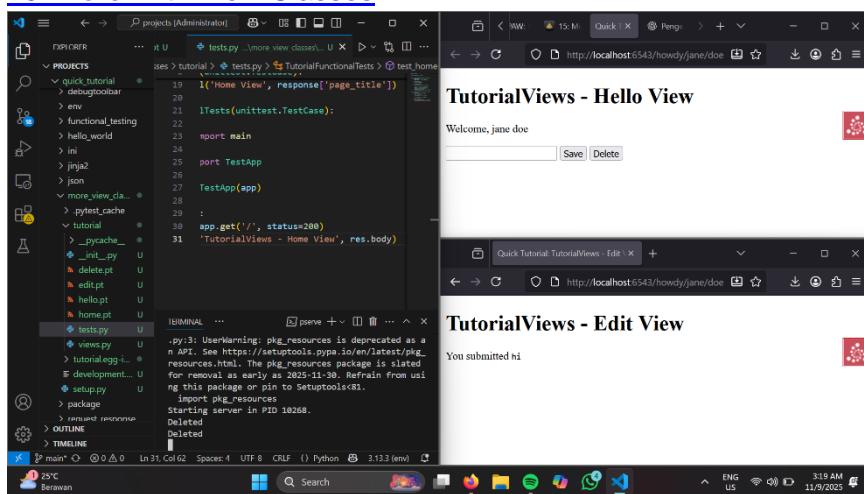
12: Templating With jinja2

The screenshot shows the PyCharm IDE. The project structure includes a 'jinja2' directory. Inside, a 'views.py' file contains Jinja2 template code. The code defines a 'home_jinja2' view that returns a template with a 'name' variable. The right side shows a browser window displaying the URL 'http://localhost:6543' with the text 'Hi Home View'. The status bar at the bottom indicates the server is running in PID 16444.

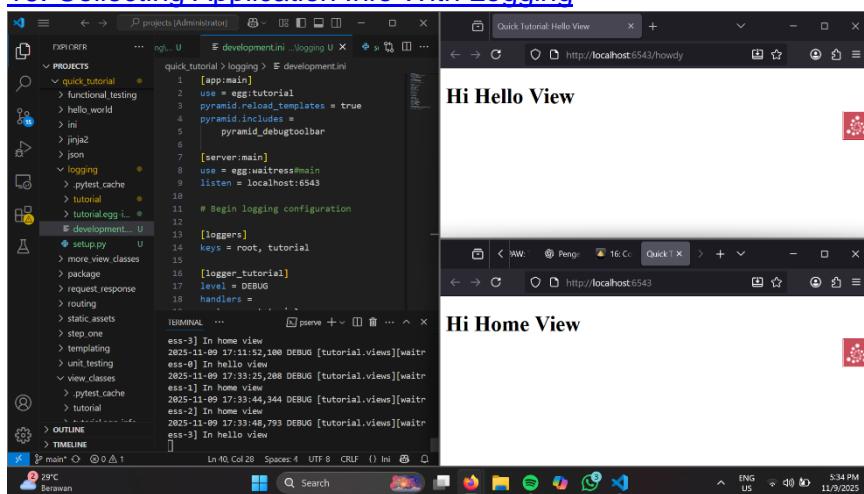
14: AJAX Development With JSON Renderers



15: More With View Classes



16: Collecting Application Info With Logging



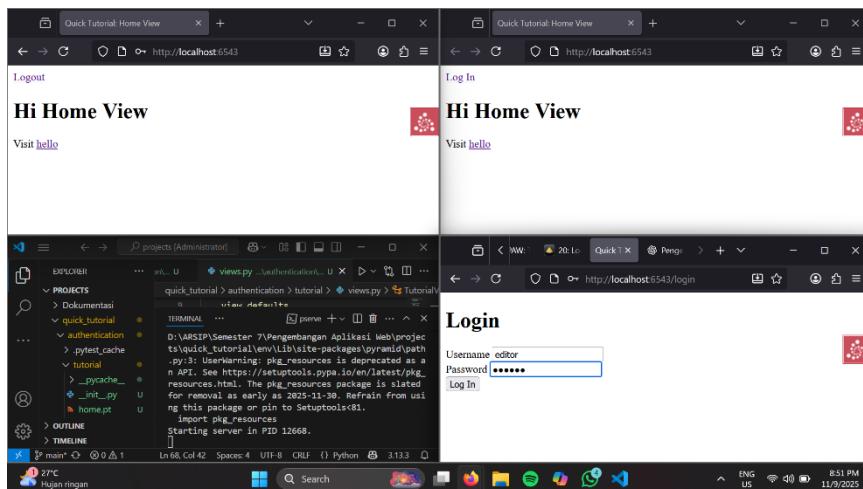
17: Transient Data Using Sessions

18: Forms and Validation with Deform

The screenshot shows a Windows desktop environment. In the center, a Microsoft Edge browser window displays a local host page at <http://localhost:6543>. The page content includes a 'Wiki' header, a 'Add WikiPage' button, and a list of pages: 'Page 100', 'Page 101', and 'Page 102'. To the right of the browser is a File Explorer window showing a project structure under 'PROJECTS'. The structure includes 'Documentasi', 'quick_tutorial', 'debug_toolbar', 'env', 'forms', 'pytest_cache', 'tutorial', '_pycache__', '_init_.py', 'home.pyt', 'tests.py', 'views.py', 'wiki_view.py', 'wikipage_add.py', 'wikipage_edit.py', 'wikipage_view.py', 'tutorial_egg_info', 'development_egg_info', 'setup.py', 'functional_testing', 'hello_world', 'ini', 'jinja2', 'json', and 'OUTLINE'. Below the browser and File Explorer is a terminal window titled 'BIRMINAL'. The terminal output shows Python code for testing a 'New Title' feature, followed by a warning message about the 'pkg_resources' package being slated for removal. The terminal also shows the start of a development server on port 16664 and several 'WARNING:watress:queue' messages indicating task queue depth. At the bottom of the screen is a taskbar with icons for various applications like File Explorer, Task Manager, and a search bar.

19: Databases Using SQLAlchemy

20: Logins with authentication



21: Protecting Resources With Authorization

