TESTING DAN QA PERANGKAT

AAB ABDUL BASIT –
20101140016

TESTING

Whitebox testing adalah jenis pengujian perangkat lunak yang dilakukan dengan memeriksa kode sumber atau desain internal suatu program. Tujuan utama dari whitebox testing adalah untuk memastikan bahwa kode program berjalan sesuai dengan spesifikasi dan bahwa semua jalur eksekusi kode telah diuji. Unit testing, di sisi lain, adalah jenis whitebox testing yang fokus pada pengujian unit kecil atau komponen individual dari perangkat lunak, seperti fungsi atau metode.

Teknik – Teknik pengujian White box Testing

	Category 1	Category 2	Category 3	Category 4	Category 5	Category 6
1	Basis Path Testing	Branch coverage	Condition coverage	Loop testing	Multiple condition coverage	Statement coverage

Basis Patch Testing

Pengujian Basis Patch (Patch Testing) adalah jenis pengujian perangkat lunak yang dilakukan untuk menguji perubahan (patch) atau pembaruan kecil dalam perangkat lunak yang ada. Patch adalah kode tambahan atau perubahan yang dirilis untuk memperbaiki masalah atau kerentanan dalam perangkat lunak yang sudah ada. Tujuan dari pengujian basis patch adalah memastikan bahwa patch tersebut tidak memperkenalkan masalah baru atau memengaruhi fungsionalitas yang sudah ada dalam perangkat lunak.

Branch Coverage

Branch coverage adalah metrik dalam pengujian perangkat lunak yang mengukur sejauh mana semua cabang (branch) dalam kode program telah diuji selama pengujian. Ini berfokus pada pemastian bahwa semua percabangan keputusan dalam kode program telah dieksekusi minimal satu kali selama pengujian. Tujuan utama dari branch coverage adalah untuk memastikan bahwa semua jalur eksekusi kode telah diuji, sehingga meminimalkan kemungkinan terjadinya kondisi yang tidak terduga atau bug yang terlewatkan dalam kode program. Branch coverage diukur dalam persentase, yang mengindikasikan berapa banyak cabang yang telah diuji dalam kode dibandingkan dengan jumlah total cabang yang ada. Semakin tinggi persentase branch coverage, semakin baik kualitas pengujian kode program.

Condition Coverage

Condition coverage adalah metrik pengujian perangkat lunak yang mengukur sejauh mana semua kondisi logis dalam kode program telah diuji selama pengujian. Ini fokus pada pemastian bahwa setiap ekspresi logis (misalnya, if, while, atau switch) dalam kode program telah dievaluasi baik sebagai benar (true) maupun salah (false) selama pengujian. Tujuannya adalah untuk memastikan bahwa semua kemungkinan kondisi logis telah diuji, sehingga mengidentifikasi dan mengurangi risiko terjadinya bug terkait kondisi logis dalam program. Condition coverage diukur dalam persentase, yang mengindikasikan berapa banyak kondisi logis yang telah dievaluasi selama pengujian dibandingkan dengan jumlah total kondisi dalam kode. Semakin tinggi persentase condition coverage, semakin baik pengujian kondisi logis dalam kode program.

Loop Testing

Loop testing adalah jenis pengujian perangkat lunak yang difokuskan pada pengujian perulangan (loop) dalam kode program. Tujuannya adalah untuk memastikan bahwa perulangan berfungsi dengan benar dan tidak menyebabkan masalah seperti perulangan tak berujung atau perulangan yang tidak berakhir. Dalam loop testing, Anda menguji berbagai skenario yang melibatkan perulangan, seperti pengujian dengan jumlah iterasi minimum, maksimum, dan nilai-nilai di antaranya. Ini membantu mengidentifikasi masalah potensial yang berkaitan dengan perulangan dalam kode program dan memastikan keandalan dan kestabilan program.

Multiple condition coverage

Multiple condition coverage adalah metrik pengujian perangkat lunak yang mengukur sejauh mana semua kombinasi kondisi logis dalam kode program telah diuji selama pengujian. Tujuannya adalah memastikan bahwa semua kemungkinan kombinasi dari beberapa kondisi logis (misalnya, dalam ekspresi if-else yang bersarang) telah diuji. Ini membantu mengidentifikasi potensi kerentanan terkait dengan kombinasi kondisi logis dalam kode. Multiple condition coverage membantu meminimalkan risiko terlewatnya kasus uji yang kritis yang hanya terjadi ketika beberapa kondisi bersama-sama memenuhi persyaratan tertentu.

Statement coverage

Statement coverage adalah metrik pengujian perangkat lunak yang mengukur sejauh mana semua pernyataan dalam kode program telah dieksekusi selama pengujian. Tujuannya adalah memastikan bahwa setiap pernyataan (statement) dalam kode telah dijalankan setidaknya sekali selama pengujian. Ini membantu memastikan bahwa seluruh kode telah diuji secara dasar dan dapat membantu mengidentifikasi bagian kode yang tidak dijalankan selama pengujian. Statement coverage diukur dalam persentase, yang mengindikasikan berapa banyak pernyataan yang dieksekusi selama pengujian dibandingkan dengan total pernyataan dalam kode program. Semakin tinggi persentase statement coverage, semakin baik kualitas pengujian kode program.

IMPLEMENTASI

IMPLEMENTASI

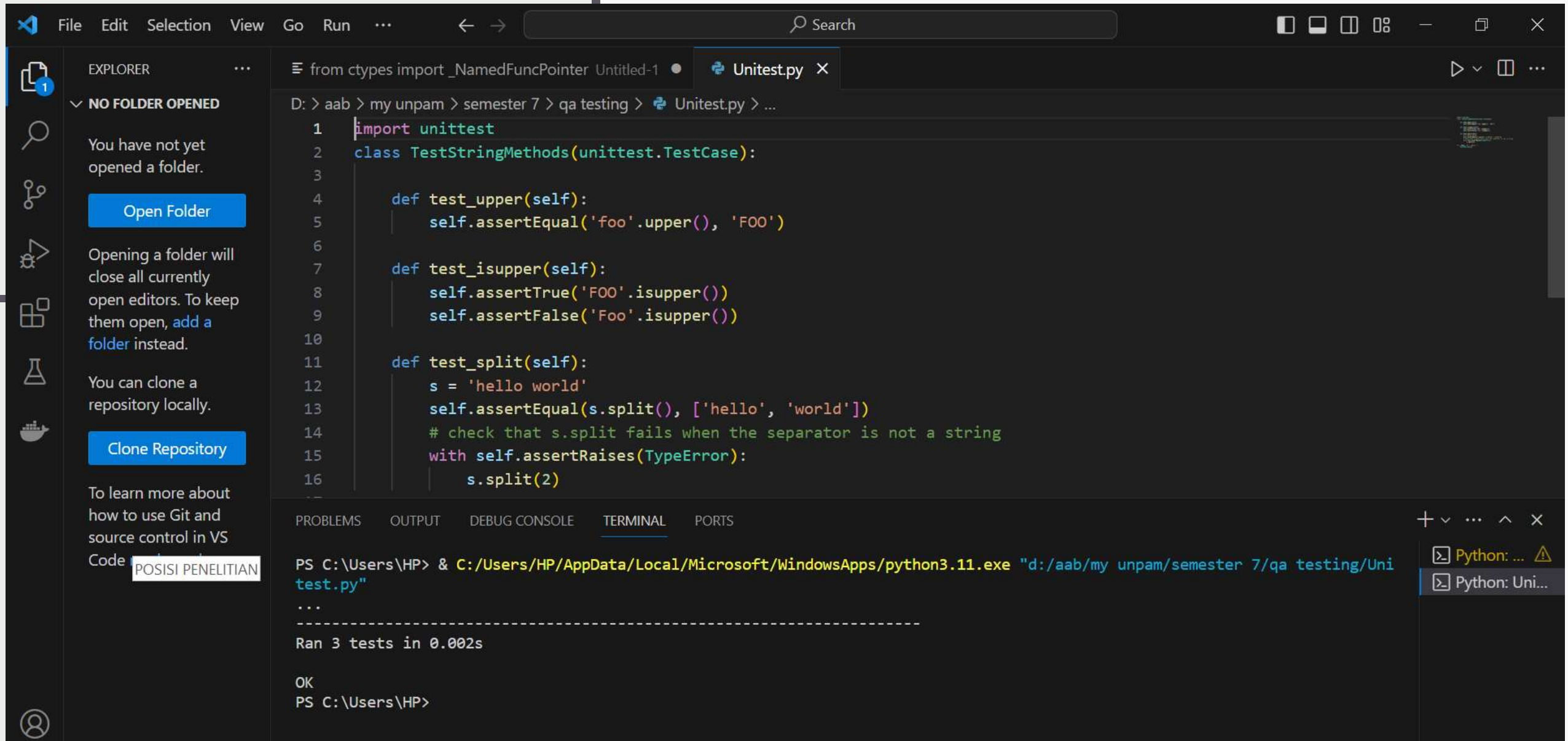
Dalam konteks pengujian perangkat lunak, ada beberapa jenis pengujian yang dapat dilakukan, termasuk pengujian whitebox (seperti statement coverage, branch coverage, condition coverage), pengujian basis patch, loop testing, multiple condition coverage, dan lainnya. Setiap jenis pengujian memiliki tujuan dan fokusnya sendiri dalam memastikan kualitas perangkat lunak.

```python
# my_module.py
def subtract(a, b):
    return a - b

# test_my_module.py
import unittest
import my_module

class TestMyModule(unittest.TestCase):
    def test_subtract(self):
        self.assertEqual(my_module.subtract(5, 3), 2)
        self.assertEqual(my_module.subtract(0, 0), 0)
        self.assertEqual(my_module.subtract(-1, 1), -2)

if __name__ == '__main__':
    unittest.main()
```

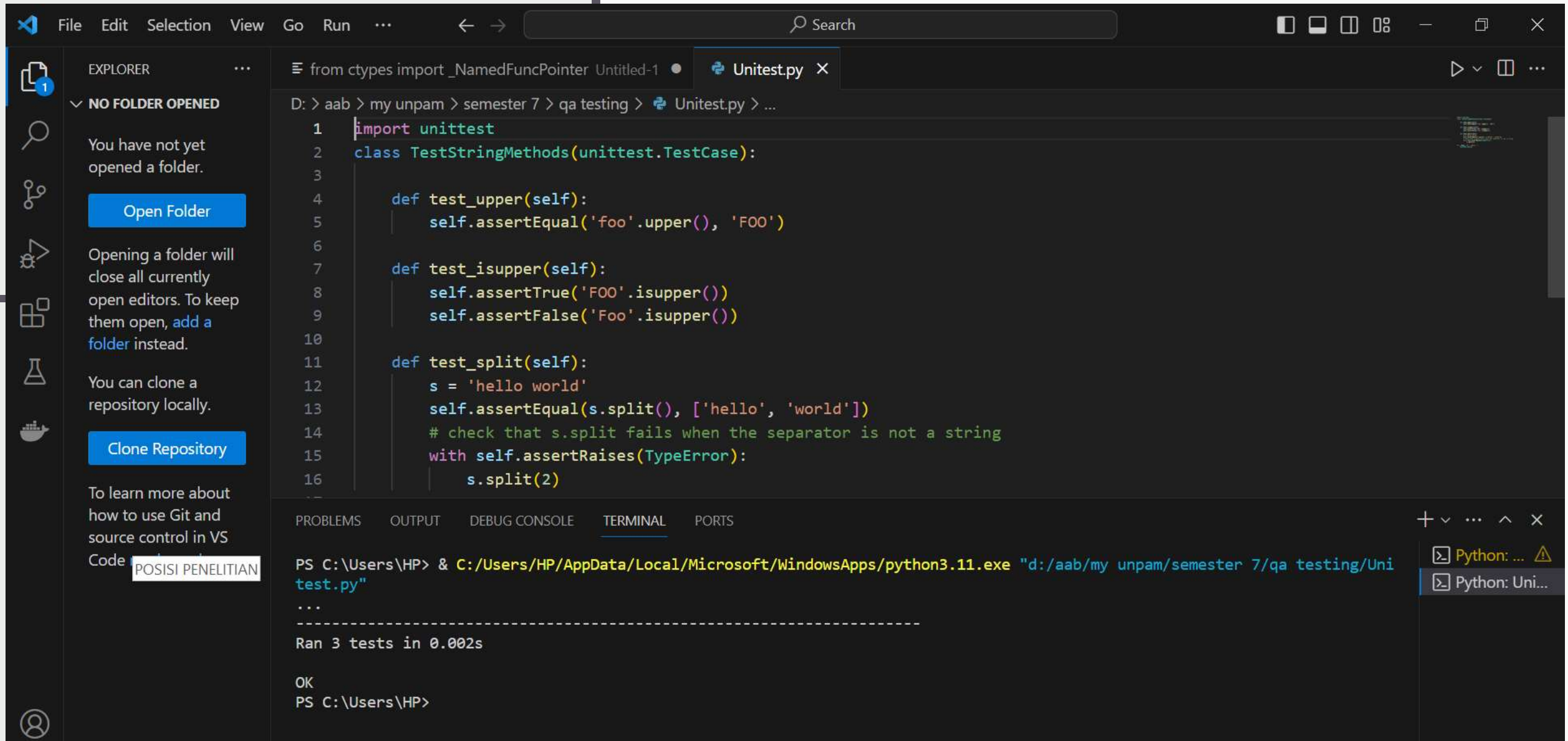
CONTOH

adalah contoh sederhana dari unit testing dengan Python. dapat menggunakan alat pengujian unit lainnya seperti nose atau doctest sesuai dengan preferensi. Unit testing adalah praktik yang penting untuk memastikan keandalan dan kualitas perangkat lunak, dan sebaiknya menguji semua komponen penting dalam program .

UNITEST

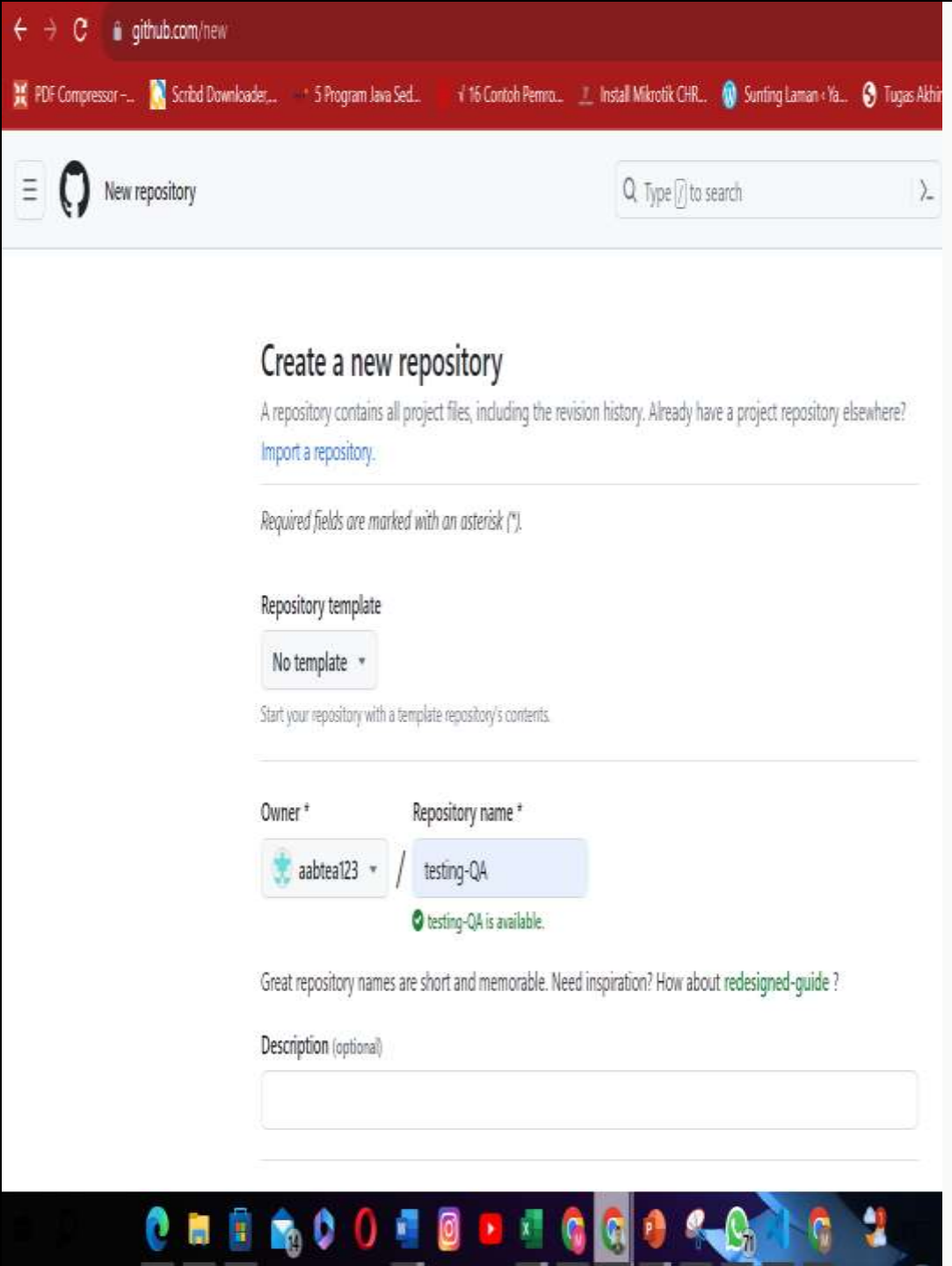

File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use Git and source control in VS Code
POSISI PENELITIAN
from ctypes import _NamedFuncPointer
Untitled-1
Unitest.py
D: > aab > my unpam > semester 7 > qa testing > Unitest.py > ...
import unittest
class TestStringMethods(unittest.TestCase):
def test_upper(self):
self.assertEqual('foo'.upper(), 'FOO')
def test_isupper(self):
self.assertTrue('FOO'.isupper())
self.assertFalse('Foo'.isupper())
def test_split(self):
s = 'hello world'
self.assertEqual(s.split(), ['hello', 'world'])
# check that s.split fails when the separator is not a string
with self.assertRaises(TypeError):
s.split(2)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/aab/my unpam/semester 7/qa testing/Unitest.py"
...
Ran 3 tests in 0.002s
OK
PS C:\Users\HP>
Python: ...
Python: Uni...

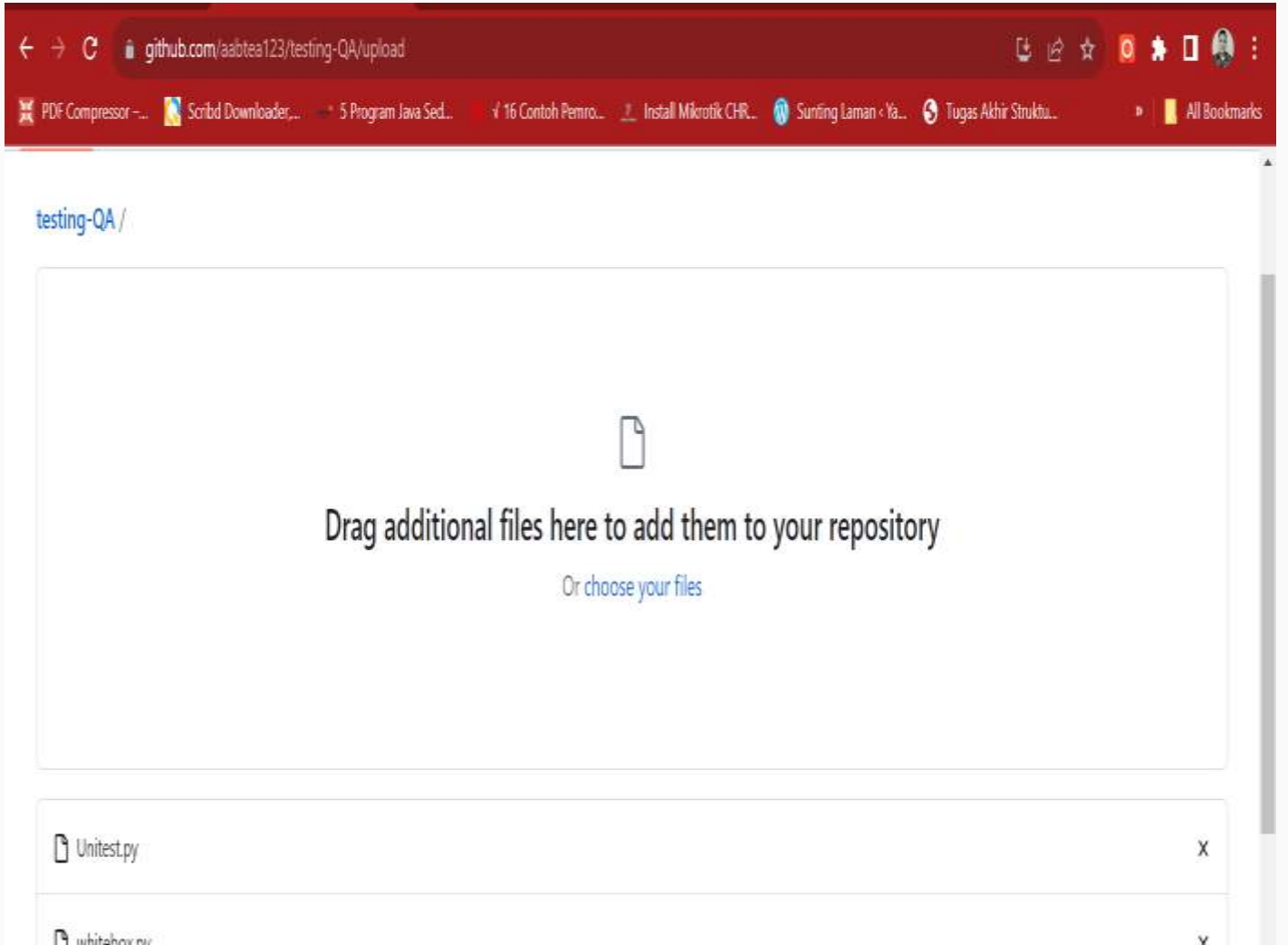
UNITEST


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
You can clone a repository locally.
Clone Repository
To learn more about how to use Git and source control in VS Code
POSISI PENELITIAN
from ctypes import _NamedFuncPointer
Untitled-1
Unitest.py
D: > aab > my unpam > semester 7 > qa testing > Unitest.py > ...
import unittest
class TestStringMethods(unittest.TestCase):
def test_upper(self):
self.assertEqual('foo'.upper(), 'FOO')
def test_isupper(self):
self.assertTrue('FOO'.isupper())
self.assertFalse('Foo'.isupper())
def test_split(self):
s = 'hello world'
self.assertEqual(s.split(), ['hello', 'world'])
# check that s.split fails when the separator is not a string
with self.assertRaises(TypeError):
s.split(2)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Microsoft/WindowsApps/python3.11.exe "d:/aab/my unpam/semester 7/qa testing/Unitest.py"
...
Ran 3 tests in 0.002s
OK
PS C:\Users\HP>
Python: ...
Python: Uni...

GITHUB

MEMBUAT NEW REPOSITORY PADA GITHUB DAN UPLOAD FILE





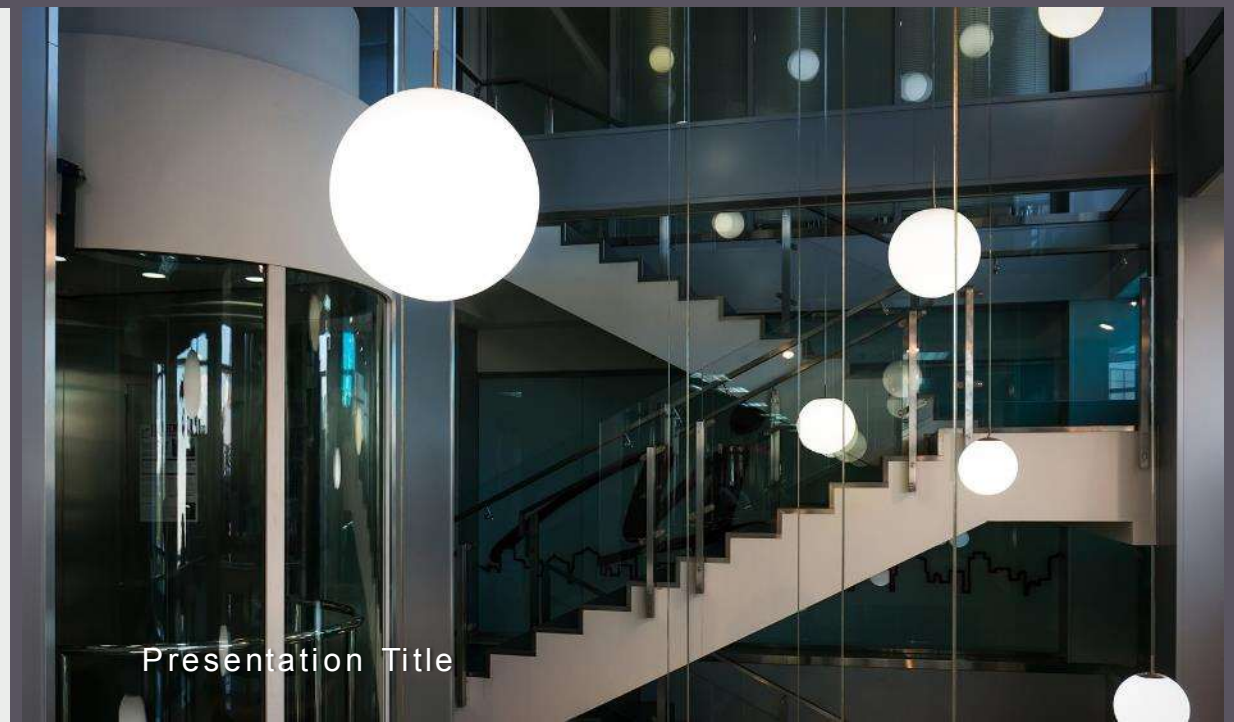
IMPLEMENTASI

Dalam konteks pengujian perangkat lunak, ada beberapa jenis pengujian yang dapat dilakukan, termasuk pengujian whitebox (seperti statement coverage, branch coverage, condition coverage), pengujian basis patch, loop testing, multiple condition coverage, dan lainnya. Setiap jenis pengujian memiliki tujuan dan fokusnya sendiri dalam memastikan kualitas perangkat lunak.

REFERENSI

- https://www.dicoding.com/blog/white-box-testing/
- https://www.youtube.com/watch?v=PsO5dZqBckY

THANK YOU



AAB ABDUL BASIT
201011400161