

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK(PBO)

PRAKTIKUM 14



2411102441205

Andi Muh Fitrah Andi Kambe

B

FAKULTAS SAINS DAN TEKNOLOGI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR

Latar Belakang

Kualitas sebuah perangkat lunak tidak hanya ditentukan oleh fungsionalitas fiturnya, tetapi juga oleh ketahanan kode terhadap kesalahan logika dan kondisi batas (*boundary conditions*).

Praktikum ini berfokus pada dua aspek krusial dalam siklus pengembangan perangkat lunak, yaitu *Debugging* dan *Unit Testing*. Melalui penggunaan *Python Debugger* (pdb), mahasiswa diajarkan untuk melacak akar masalah (*root cause*) dengan cara memeriksa nilai variabel secara *real-time* saat program berjalan.

Selain itu, penerapan *Unit Testing* menggunakan modul unittest memungkinkan verifikasi kode secara otomatis untuk memastikan bahwa setiap fungsi, seperti perhitungan diskon, memberikan hasil yang akurat dalam berbagai skenario..

Tujuan Praktikum

Melalui praktikum ini, mahasiswa diharapkan dapat:

1. Menjelaskan perbedaan fundamental antara **Debugging** (mencari *mengapa* kode rusak) dan **Testing** (mencari *apakah* kode rusak).
2. Menggunakan *Debugger* bawaan Python (pdb) untuk **menghentikan eksekusi** (*breakpoint*) pada baris kode tertentu dan menelusuri alur logika step-by-step.
3. Menemukan *root cause* dari *bug* logika perhitungan dengan memeriksa nilai variabel pada *runtime*.
4. Mendefinisikan class test yang mewarisi `unittest.TestCase` dan menggunakan metode `setUp` dan `assert` untuk verifikasi otomatis.
5. Menulis *Unit Test* yang secara khusus menguji *Boundary Condition* (Kondisi Batas) seperti diskon 0%, diskon 100%, dan angka nol/negatif.

D. Latihan Mandiri

1. diskon_service_bug.py

```
diskon_service_bug.py > hitung_diskon
1  # diskon_service.py
2
3  def hitung_diskon(harga_awal, persen_diskon):
4      """
5          Fungsi untuk menghitung harga setelah diskon.
6          Sengaja dimasukkan bug PPN 10% untuk latihan debugging.
7      """
8      if harga_awal < 0 or persen_diskon < 0 or persen_diskon > 100:
9          return 0
10
11     harga_setelah_diskon = harga_awal * (1 - persen_diskon / 100)
12
13     # BUG DISENGAJA: Menambahkan PPN 10% secara keliru
14     # Baris inilah yang akan Anda lacak menggunakan pdb
15     #return harga_setelah_diskon * 1.1
16     return harga_setelah_diskon
```

2. Kode Unit Test(test_diskon_advanced.py.)

```
test_diskon_advanced.py > TestDiskonLanjut > test_float_discount
1  import unittest
2  # Import dari file yang sudah diperbaiki nantinya
3  from diskon_service_bug import hitung_diskon
4
5  class TestDiskonLanjut(unittest.TestCase):
6      def test_float_discount(self):
7          # Tes 5: Uji nilai float (diskon 33% pada 999)
8          self.assertAlmostEqual(hitung_diskon(999, 33), 669.33, places=2)
9
10     def test_zero_price(self):
11         # Tes 6: Uji harga awal 0
12         self.assertEqual(hitung_diskon(0, 50), 0)
13
14     if __name__ == '__main__':
15         unittest.main()
```

3. Output Terminal

```
PS C:\Users\Asus GK\OneDrive\Documents\PBO_Praktikum\Praktikum 14> python test_diskon_advanced.py
..
-----
Ran 2 tests in 0.000s

OK
```

4. Markdown

```
↳ DEBUG_REPORT.MD
1   Unit test pada fungsi hitung_diskon di file diskon_service_bug.py mengalami kegagalan
    (Failure). Berdasarkan skenario pengetesan (1000, 10), hasil yang diharapkan adalah 900.
    0, namun sistem mengembalikan nilai 990.0.
2
3   **Langkah langkah debugging (pdb)**
4   Saya menggunakan modul pdb (Python Debugger) untuk menghentikan eksekusi program dan
    memeriksa nilai variabel saat runtime:
5   - Menambahkan import pdb; pdb.set_trace() sebelum baris return pada fungsi hitung_diskon.
6
7   - Menjalankan file tes dan masuk ke mode interaktif (Pdb).
8
9   - Menggunakan perintah p harga_setelah_diskon untuk melihat nilai variabel sebelum
    dikembalikan.
10
11  - Hasil di terminal menunjukkan angka 900.0, yang berarti perhitungan diskon sebenarnya
    sudah benar.
12
13  **Temuan Akar Masalah (Root Cause)**
14  Akar masalah ditemukan pada baris terakhir fungsi hitung_diskon. Melalui perintah p
    harga_setelah_diskon * 1.1 di debugger, terbukti bahwa ada penambahan PPN 10% secara
    tidak sengaja melalui perkalian * 1.1 pada baris return. Inilah yang menyebabkan hasil
    akhir menjadi 990.0 dan memicu kegagalan unit test.
15
16  Solusi Perbaikan Menghapus kode * 1.1 pada baris return sehingga fungsi hanya
    mengembalikan variabel harga_setelah_diskon secara murni.
```

Refleksi SIngkat:

Saya jadi lumayan memahami teknik *debugging* interaktif menggunakan pdb untuk melacak kesalahan logika secara akurat melalui inspeksi variabel *real-time*. Selain itu, saya memahami pentingnya implementasi *unit testing* untuk menangani berbagai skenario.