

**LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (PBO)**

**PRAKTIKUM 14**



**2411102441215**

**Nikon Valent Sakaesa**

**C**

**FAKULTAS SAINS DAN TEKNOLOGI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR**

## Laporan Praktikum

### A. Kode Program (`test_diskon.py`)

Program `test_diskon.py` dibuat menggunakan modul `unittest` untuk menguji fungsi `hitung_diskon()` pada class `DiskonCalculator`. Pengujian dilakukan untuk memastikan perhitungan diskon berjalan dengan benar, termasuk pengujian kondisi normal, boundary case, dan edge case seperti diskon 0%, 100%, lebih dari 100%, serta input float.

Pengujian juga mencakup validasi bahwa bug perhitungan PPN/diskon ganda telah diperbaiki sehingga hasil perhitungan sesuai dengan yang diharapkan. Setelah perbaikan kode pada `diskon_services.py`, seluruh test case berhasil dijalankan tanpa error.

```
# test_diskon.py
import unittest
from diskon_services import DiskonCalculator

class TestDiskonCalculator(unittest.TestCase):
    def setUp(self):
        """Menyiapkan instance Calculator."""
        self.calc = DiskonCalculator()

    def test_diskon_standar_10_persen(self):
        """Tes 1: Diskon 10% dari 1000."""
        hasil = self.calc.hitung_diskon(1000, 10)
        self.assertEqual(hasil, 900.0)

    def test_diskon_nol(self):
        """Tes 2: Diskon 0%."""
        hasil = self.calc.hitung_diskon(500, 0)
        self.assertEqual(hasil, 500.0)

    def test_diskon_batas_atas(self):
        """Tes 3: Diskon 100%."""
        hasil = self.calc.hitung_diskon(750, 100)
        self.assertEqual(hasil, 0.0)

    def test_input_negatif(self):
        """Tes 4: Diskon negatif tidak menurunkan harga."""
        hasil = self.calc.hitung_diskon(500, -5)
        self.assertGreaterEqual(hasil, 500.0)

class TestDiskonLanjut(unittest.TestCase):
    def setUp(self):
```

```
self.calc = DiskonCalculator()

def test_diskon_float(self):
    """Tes 5: Diskon float (33% dari 999)."""
    hasil = self.calc.hitung_diskon(999, 33)
    self.assertAlmostEqual(hasil, 669.33, places=2)

def test_edge_case_harga_nol(self):
    """Tes 6: Harga awal 0."""
    hasil = self.calc.hitung_diskon(0, 50)
    self.assertEqual(hasil, 0.0)

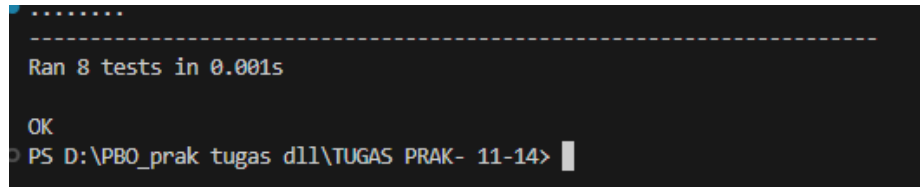
def test_diskon_lebih_dari_100(self):
    """Tes 7: Diskon > 100% tidak menghasilkan harga negatif."""
    hasil = self.calc.hitung_diskon(400, 150)
    self.assertGreaterEqual(hasil, 0.0)

def test_ppn_ganda_setelah_bug_fix(self):
    """Tes 8: Memastikan bug PPN ganda sudah diperbaiki."""
    hasil = self.calc.hitung_diskon(1000, 10)
    self.assertEqual(hasil, 900.0)

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

**B. Hasil Pengujian (Screenshot Output)**

Setelah dilakukan perbaikan pada bug perhitungan diskon/PPN, seluruh unit test berhasil dijalankan dengan hasil **OK**. Output terminal menunjukkan bahwa semua test case lulus tanpa kegagalan, yang menandakan bahwa logika perhitungan sudah benar dan stabil.

A screenshot of a terminal window with a dark background. It shows the output of a test run: "Ran 8 tests in 0.001s" followed by "OK" on the next line. Below that, a command prompt shows the path "PS D:\PBO\_prak tugas dll\TUGAS PRAK- 11-14>".

```
.....  
-----  
Ran 8 tests in 0.001s  
  
OK  
PS D:\PBO_prak tugas dll\TUGAS PRAK- 11-14>
```

**C. Debug Log (DEBUG\_REPORT.md)**

Proses debugging dilakukan menggunakan modul pdb untuk menelusuri nilai variabel pada fungsi `hitung_diskon()`. Dengan menggunakan perintah `p [variabel]`, ditemukan bahwa nilai diskon dihitung tanpa membagi persentase dengan 100 sehingga menyebabkan hasil yang tidak wajar dan bernilai negatif. Setelah rumus perhitungan diperbaiki, nilai variabel kembali sesuai dan bug PPN/diskon ganda berhasil diatasi.

**Link GitHub :**

<https://github.com/2411102441215-Valent/TUGAS-DLL>

**Kesimpulan**

Dalam praktikum ini menunjukkan pentingnya unit testing dan debugging dalam pengembangan perangkat lunak. Dengan adanya unittest dan bantuan `pdb`, kesalahan logika dapat kita temukan dan diperbaiki secara baik sehingga menghasilkan program yang lebih sesuai dan mudah dikembangkan lagi lebih lanjut.