

Solusi Esai

SOAL - Perhatikan kode program berikut terkait aplikasi sederhana pencatatan riwayat cashflow pada dompet digital.

```
1 def isi_dompet(uang, dompet):
2     dompet.append(uang)
3
4 def ambil_dari_dompet(uang, dompet):
5     dompet.append(-uang)
6
7 def cetak_riwayat_dompet(dompet):
8     print(dompet)
9
10 def cek_balance_dompet(dompet):
11     return sum(dompet)
12
13 dompet_rumah = []
14 isi_dompet(50000, dompet_rumah)
15 isi_dompet(100000, dompet_rumah)
16 ambil_dari_dompet(120000, dompet_rumah)
17 cetak_riwayat_dompet(dompet_rumah)
18 print(cek_balance_dompet(dompet_rumah))
19
20 dompet_sedekah = []
21 isi_dompet(30000, dompet_sedekah)
22 isi_dompet(200000, dompet_sedekah)
23 cetak_riwayat_dompet(dompet_sedekah)
24 print(cek_balance_dompet(dompet_sedekah))
```

Jawablah pertanyaan berikut:

a) Apa **output** dari program tersebut?

```
[50000, 100000, -120000]
30000
[30000, 200000]
230000
```

Penjelasan:

- **Dompets Rumah (dompet_rumah):**

1. 50,000 dimasukkan ke dalam dompet_rumah.
2. 100,000 juga ditambahkan.
3. 120,000 diambil dari dompet_rumah.
4. Riwayat transaksi (cetak_riwayat_dompet(dompet_rumah)) mencetak isi dari dompet_rumah yang mencakup [50000, 100000, -120000].
5. cek_balance_dompet(dompet_rumah) menghasilkan saldo terakhir, yaitu 30,000.

- **Dompets Sedekah (dompet_sedekah):**

1. penambahan 30,000 ke dalam dompet_sedekah.
2. 200,000 juga dimasukkan.
3. Riwayat transaksi (cetak_riwayat_dompet(dompet_sedekah)) mencetak isi dari dompet_sedekah yang mencakup [30000, 200000].
4. cek_balance_dompet(dompet_sedekah) menghasilkan saldo terakhir, yaitu 230,000.

b) Jelaskan yang dilakukan oleh masing-masing fungsi pada kode program tersebut?

Terdapat 4 fungsi pada kode program tersebut, yaitu:

- **isi_dompet(uang, dompet):** Fungsi ini menambahkan jumlah uang ke dalam dompet dengan menggunakan metode append. Ini menunjukkan penambahan uang ke dalam dompet.
- **ambil_dari_dompet(uang, dompet):** Fungsi ini mengurangi jumlah uang dari dompet dengan cara menambahkan nilai negatif dari uang ke dalam dompet. Ini menunjukkan pengurangan uang dari dompet.
- **cetak_riwayat_dompet(dompet):** Fungsi ini mencetak riwayat transaksi yang ada dalam dompet. Saat ini, ia hanya mencetak isi dari dompet.
- **cek_balance_dompet(dompet):** Fungsi ini mengembalikan saldo atau total jumlah uang yang ada dalam dompet dengan menggunakan fungsi sum() untuk menjumlahkan semua nilai di dalamnya

SOAL - Buatlah kode program Python tentang Robot menggunakan paradigma object-oriented programming (OOP) dengan syarat berikut:

- a) Robot memiliki nama dan baterai yang terisi 100 (by default).
- b) Robot dapat menyapa dengan sapaan "Hello, human!". Tiap aksi menyapa yang dilakukan akan mengurangi baterai sebanyak 20.
- c) Robot dapat melakukan charge yang akan menambah baterai sebanyak nilai tambahan baterai saat charge.
- d) Robot dapat mencetak status baterai.
- e) Sebagai contoh, jika instruksi di bawah ini dijalankan (dengan catatan Anda sudah membuat kode Robot Anda sebelumnya),

```
r1 = Robot("Ray")
r1.sapa()
r1.print_baterai_status()
r1.charge(100)
r1.print_baterai_status()
```

maka program akan mencetak output:

```
RoboRay: Hello, human!
Status baterai: 80.0
Status baterai: 180.0
```

Pastikan program yang Anda buat dapat dijalankan (tidak terdapat syntax error).
Anda **wajib menambahkan penjelasan/komentar** pada program yang Anda buat.

Contoh Solusi (FD):

```
class Robot:

    def __init__(self, nama, baterai=100.0):
        # Inisialisasi objek Robot dengan atribut nama dan baterai
        # (default 100.0)
        self.nama = nama
        self.baterai = baterai

    def sapa(self):
        # Method untuk memberi salam, mencetak nama robot dan pesan, dan
        # mengurangi baterai sebesar 20
        print(f"Robo{self.nama}: Hello, human!")
        self.baterai -= 20

    def charge(self, tambahan_baterai):
```

```
        # Method untuk mengisi baterai dengan jumlah tambahan_baterai
        self.baterai += tambahan_baterai

    def print_baterai_status(self):
        # Method untuk mencetak status baterai saat ini
        print(f"Status baterai: {self.baterai}")

# Membuat objek Robot bernama "Ray" dengan baterai awal 100.0
r1 = Robot("Ray")

# Memanggil method sapa() untuk memberi salam dan mengurangi baterai
r1.sapa()

# Mencetak status baterai setelah method sapa() dipanggil
r1.print_baterai_status()

# Mengisi baterai objek dengan 100
r1.charge(100)

# Mencetak status baterai setelah diisi
r1.print_baterai_status()
```

SOAL PG

Question 1

Correct

Mark 0.50 out of 0.50



Edit question

Apa yang dilakukan oleh kode program berikut?

```
def reverse_string(s):  
    if len(s) == 0:  
        return s  
    else:  
        return s[0].upper() + reverse_string(s[1:]) + s[0].upper()
```

Select one:

- ☐ a. Melakukan reverse pada string yang diinput dan mengembalikannya dalam bentuk uppercase pada huruf awal
- ☐ b. Mengembalikan string yang di input dalam bentuk uppercase
- ☒ c. Menambahkan string yang diinput dengan reverse dari string tersebut dan mengembalikannya dalam bentuk uppercase
- ☐ d. Melakukan reverse pada string yang diinput dan mengembalikannya dalam bentuk lowercase

Question 2

Correct

Mark 0.50 out of 0.50



Edit question

Perhatikan kode program di bawah ini. Berapa total akhir penumpang pada bis serta total pendapatan dari bis_besar?

```
class BisTransJakarta:  
    def __init__(self, tipe):  
        self.tipe = tipe  
        self.kapasitas = self.set_kapasitas()  
        self.penumpang = 0  
        self.tarif = 3500  
        self.total_pendapatan = 0  
  
    def set_kapasitas(self):  
        if self.tipe == 'Gandeng':  
            return 60  
        elif self.tipe == 'Tunggal':  
            return 30  
  
    def penumpang_naik(self, penumpang_baru):  
        if self.penumpang + penumpang_baru <= self.kapasitas:  
            self.penumpang += penumpang_baru  
            self.bayar(penumpang_baru)  
        elif self.penumpang + penumpang_baru > self.kapasitas:  
            max_naik = self.kapasitas - self.penumpang  
            self.penumpang += max_naik  
            self.bayar(max_naik)  
  
    def penumpang_turun(self, penumpang_turun):  
        if penumpang_turun > self.penumpang:  
            print("Jumlah penumpang turun melebihi yang ada di dalam  
bis.")  
        else:  
            self.penumpang -= penumpang_turun  
  
    def bayar(self, penumpang_baru):  
        self.total_pendapatan += self.tarif * penumpang_baru  
  
def main():  
    bis_besar = BisTransJakarta('Tunggal')  
  
    # Penumpang naik dan turun  
    bis_besar.penumpang_naik(25)  
    bis_besar.penumpang_turun(5)  
    bis_besar.penumpang_naik(15)  
    bis_besar.penumpang_turun(7)  
    bis_besar.penumpang_turun(10)  
  
if __name__ == "__main__":  
    main()
```

Select one:

- ☐ a. 13 Orang, Rp122000
- ☐ b. 10 Orang, Rp122500
- ☐ c. 12 Orang, Rp126000
- ☒ d. 13 Orang, Rp122500

Question 3

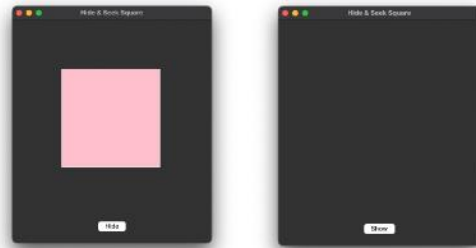
Correct

Mark 0.50 out of 0.50



[Edit question](#)

Potongan kode yang tepat untuk melengkapi kode program dibawah ini agar program dapat menampilkan serta menghilangkan persegi ketika menekan tombol hide/show seperti pada contoh di bawah ini adalah...



```
import tkinter as tk

def toggle_shape():
    if canvas.itemcget(square, "state") == "hidden":
        canvas.itemconfigure(square, state="normal")
        toggle_button.config(text="Hide")
    else:
        canvas.itemconfigure(square, state="hidden")
        toggle_button.config(text="Show")

root = tk.Tk()
root.title("Hide & Seek Square")

canvas = tk.Canvas(root, width=400, height=400)
canvas.pack()

square = # Lengkapi kode ini

toggle_button = tk.Button(root, text="Hide Shape", command=toggle_shape)
toggle_button.pack()
tk.Label(root, text="").pack() # Space added after the button

root.mainloop()
```

Select one:

- ☐ a. canvas.draw(100, 100, 300, 300, fill="pink")
- ☐ b. canvas.config(type="rectangle", size=[200, 200], fill="pink")
- ☒ c. canvas.create_rectangle(100, 100, 300, 300, fill="pink")
- ☐ d. canvas.geometry(200, 200, fill="pink", shape="rectangle")

Question 4

Correct

Mark 0.50 out of 0.50


[Edit question](#)

Kode yang tepat untuk membuat unit test pada function test_clear_cart adalah...

```
import unittest

class ShoppingCart:
    def __init__(self):
        self.items = {}

    def add_item(self, name, price, quantity):
        self.items[name] = {'price': price, 'quantity': quantity}

    def remove_item(self, name, quantity):
        if name in self.items:
            if self.items[name]['quantity'] > quantity:
                self.items[name]['quantity'] -= quantity
            else:
                del self.items[name]

    def clear_cart(self):
        self.items = {}

class TestShoppingCart(unittest.TestCase):

    def setUp(self):
        self.cart = ShoppingCart()
        self.cart.add_item("Apple", 10, 2)
        self.cart.add_item("Orange", 15, 3)

    def test_add_item(self):
        self.cart.add_item("Banana", 8, 4)
        self.assertEqual(self.cart.items["Banana"], {'price': 8,
'quantity': 4})

    def test_remove_item(self):
        self.cart.remove_item("Apple", 2)
        self.assertEqual(self.cart.items, {"Orange": {'price': 15,
'quantity': 3}})

    def test_clear_cart(self):
        self.cart.clear_cart()
        # Lengkapi kode ini

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

Select one:

- ☐ a.self.assertTrue(not self.cart.items)
- ☒ b.self.assertEqual(self.cart.items, {})
- ☐ c.self.assertFalse(self.cart.items)
- ☐ d.self.assertIsNone(self.cart.items)

Question 5

Correct

Mark 0.50 out of 0.50


[Edit question](#)

Apa yang dilakukan oleh kode program berikut?

```
n = int(input("Enter an integer: "))

i = 1
while i <= n:
    a = i
    i += 1
    print(a)
```

Select one:

- ☐ a.Program mencetak angka dari 1 hingga 10 yang merupakan kelipatan dari bilangan bulat yang dimasukkan oleh pengguna.
- ☐ b.Program mencetak hasil total pertambahan dari 1 hingga n-1 yang merupakan input dari pengguna.
- ☐ c.Program mengalikan bilangan bulat yang dimasukkan oleh pengguna dengan angka 1 hingga n, kemudian mencetak hasilnya.
- ☒ d.Program mencetak angka dari 1 hingga n yang merupakan input dari pengguna. ✓

Question 6

Correct

Mark 0.50 out of 0.50

[Edit question](#)

Apa yang dilakukan oleh kode program berikut?

```
def secret_list_function(arr):
    for i in range(len(arr)):
        if arr[i] % 2 != 0:
            num = arr[i] * 2
            arr[i] = num

    check = len(arr) != len(set(arr))

    return arr, check

result, check = secret_list_function([2,5,3,5,9,12,11,6,10])
print(result)
print(check)
```

Select one:

- ☐ a. Menghapus semua angka ganjil pada list yang diinput dan mengubahnya dalam bentuk tipe data set
- ☒ b. Mengalikan 2 pada element list yang bernilai ganjil dan memeriksa apakah list dengan elemen yang baru mengandung angka yang sama
- ☐ c. Mengembalikan list yang sama seperti di input dan mencetak panjang dari list
- ☐ d. Mengembalikan daftar baru yang hanya berisi angka genap dari list yang di input

Question 7

Correct

Mark 0.50 out of 0.50

[Edit question](#)

Perhatikan kode program di bawah ini. Apa yang dicetak ketika program dijalankan dan diberikan input "T" serta dilanjutkan dengan input "Y"?

```
suka_pedas = input("Suka pedas (Y/T)? ")
tanggal_tua = input("Tanggal tua (Y/I)? ")

if suka_pedas == "Y":
    if tanggal_tua == "Y":
        print("Rekomendasi menu: Nasi sambal")
    else:
        print("Rekomendasi menu: Nasi rica-rica iga sapi")
else:
    if tanggal_tua == "Y":
        print("Rekomendasi menu: Nasi kecap")
    else:
        print("Rekomendasi menu: Nasi ayam kecap")
```

Select one:

- ☐ a. "Rekomendasi menu: Nasi ayam kecap"
- ☐ b. "Rekomendasi menu: Nasi sambal"
- ☐ c. "Rekomendasi menu: Nasi rica-rica iga sapi"
- ☒ d. "Rekomendasi menu: Nasi kecap" ✓

Question 8

Correct

Mark 0.50 out of 0.50



[Edit question](#)

Perhatikan kode program di bawah ini. Apa yang dicetak ketika pengguna ingin meminjam buku dengan judul buku, 'The Midnight Library' pada library?

```
def borrow_book(library, bookshelf, book):
    try:
        if book in library and library[book] > 0:
            library[book] -= 1
            bookshelf[book] = 1
            return 'Book borrowed successfully!'
        elif book in library and library[book] == 0:
            return 'No copies available for borrowing.'
        else:
            return 'Book not available in the library.'
    except KeyError:
        return 'Book not available in the library.'

def main():
    library = {
        'Python Crash Course': 3,
        'Deep Learning': 2,
        'Clean Code': 0,
    }

    my_bookshelf = {
        'The Midnight Library': 1,
        'Battle of The Labyrinth': 1
    }

    book_to_borrow = 'The Midnight Library' # Ganti dengan buku
    yang ingin dipinjam
    result = borrow_book(library, my_bookshelf, book_to_borrow)
    print(result)

if __name__ == "__main__":
    main()
```

Select one:

- ☐ a. "Book borrowed successfully!"
- ☒ b. "Book not available in the library." ✓
- ☐ c. "No copies available for borrowing."
- ☐ d. Tidak ada yang dicetak.