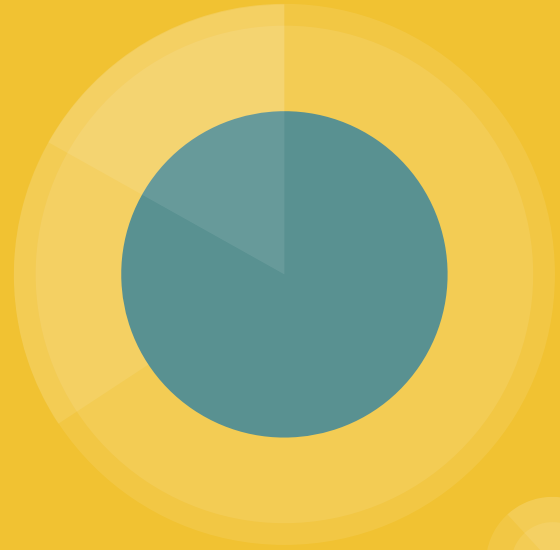


Python Programming

By: Progate





Pengantar

Python adalah bahasa pemrograman *interpreted* (tidak membutuhkan *compiler*) yang mudah dipahami dan dapat digunakan untuk berbagai keperluan, salah satunya untuk “backend development”.

Pada latihan ini, kita akan mencoba untuk mengembangkan sebuah aplikasi sederhana dengan memanfaatkan materi yang telah dipelajari dari Progate. Latihan ini akan mengajak kita untuk belajar berpikir layaknya seorang programmer, yaitu dengan berusaha membuat solusi dari sebuah permasalahan, yang dapat dijelaskan secara mudah, logis, terstruktur sehingga dapat diterjemahkan ke dalam bahasa pemrograman.

Latihan:

Membuat Aplikasi ATM

Bagaimana caranya?

1. **Jangan langsung coding*.. :)**
2. **Pahami bagaimana alur aplikasi bekerja**
3. **Tentukan fitur-fitur yang akan dibuat**
4. **Pahami logika/urutan kerja dari setiap fitur**
5. **Kerjakan bagian yang mudah dahulu**

*Dalam proses pengembangan software, terdapat siklus yang disebut dengan istilah SDLC (Software Development Life Cycle). Programmer yang baik, akan menggunakan pendekatan ini sebelum mulai mengerjakan program.

2. Alur Aplikasi



Nasabah

Masukkan kartu ATM



Mesin ATM

- Cek saldo
- Transaksi
- Ganti Pin
- dll.



Penjelasan

Dengan mengetahui alur dari sebuah program, kita bisa memiliki gambaran lebih jelas mengenai bagaimana program nantinya akan bekerja.

Dari sini kita bisa melakukan proses analisa kebutuhan yang benar dan sesuai, sehingga pekerjaan programming nantinya lebih jelas.



3. Menentukan Fitur

Menentukan fitur yang akan dikerjakan ini akan berdampak pada seorang programmer pada saat proses coding, seperti misalnya pengaruh pada estimasi waktu pengerjaan, cost dan resource yang dibutuhkan.

Pada latihan ini, kita akan coba tentukan beberapa fitur dasar yang dapat dikerjakan secara sederhana.



Daftar Fitur

1. Periksa Pin
2. Ganti Pin
3. Cek Saldo
4. Debet / Withdraw
5. Simpan / Deposit
6. Keluar
7. Cetak ringkasan transaksi




4. Logika dan Urutan Kerja

Berdasarkan daftar fitur yang ada tadi, kita akan coba membuat program Python dengan memanfaatkan sebanyak mungkin materi yang telah dipelajari melalui Progate.

Pada latihan kali ini, program Python akan kita coba kerjakan dengan menggunakan pendekatan berbasis “Class”.

5. Mulai Berlatih



Pertama-tama, kita akan membuat tiga buah file Python yang merepresentasikan setiap objek yang terlibat dalam alur aplikasi ATM, yaitu

1. Kartu ATM => atm_card.py
2. Nasabah => customer.py
3. Mesin ATM => atm_program.py

Latihan 1:

Buatlah setiap file ini pada komputer kalian..



atm_card.py

Logikanya, pada saat nasabah pertama kali membuat kartu ATM, dia akan memiliki nomor pin bawaan dan nominal saldo awal pada kartu tersebut.

Dari sini, kita akan membuat class yang memiliki function `__init__` yang menyimpan nomor pin bawaan dan saldo awal pada kartu tersebut. Class ini nantinya juga memiliki method untuk memeriksa pin dan saldo tersebut.



atm_card.py

Latihan 2:

Buatlah class `ATMCard` yang memiliki function `__init__` dan bisa menerima argumen nilai untuk pin awal dan saldo awal. Beri nama parameter-nya dengan `defaultPin` dan `defaultBalance`.

Ayo coba dikerjakan sendiri atau tanya dengan teman seangkatan sebelum melihat jawabannya.



atm_card.py

Jawaban:

```
class ATMCARD:  
    def __init__(self, defaultPin, defaultBalance):  
        self.defaultPin = defaultPin  
        self.defaultBalance = defaultBalance
```



atm_card.py

Latihan 3:

Masih pada class yang sama, buatlah dua buah method yang masing-masing mengembalikan nilai dari *defaultPin* dan *defaultBalance*.

Ayo dicoba sendiri atau dengan teman seangkatan sebelum melihat jawabannya.



atm_card.py

Jawaban:

```
def cekPinAwal(self):  
    return self.defaultPin
```

```
def cekSaldoAwal(self):  
    return self.defaultBalance
```




customer.py

Nasabah yang telah memiliki kartu ATM, dia akan dapat menggunakan mesin ATM. Pada saat pertama kali menggunakan mesin ATM, nomor pin dan nominal saldo nilainya akan sama sebagaimana yang diberikan oleh pihak bank, yaitu yang tersimpan pada kartu ATM.

Nasabah nantinya bisa mengganti nomor pin ini dan melakukan transaksi yang akan merubah nilai saldo.



customer.py

Latihan 4:

Lakukan import class `ATMCard` dan buatlah class `Customer` yang memiliki function `__init__` dan bisa menerima argumen berupa `id`, `pin` nasabah = 1234 dan saldo nasabah = 10000. Beri nama parameternya dengan `id`, `custPin`, `custBalance`.

Silakan coba mengerjakan sendiri terlebih dahulu.



customer.py

Jawaban:

```
from atm_card import ATMCARD
```

```
class Customer:
```

```
    def __init__(self, id, custPin = 1234, custBalance = 10000):
```

```
        self.id = id
```

```
        self.pin = custPin
```

```
        self.balance = custBalance
```



customer.py

Latihan 5:

Pada class yang sama, buatlah method yang mengembalikan nilai dari *id*, *custPin* dan *custBalance*.

Mirip seperti latihan sebelumnya, silahkan coba mengerjakan sendiri sebelum melihat jawabannya.



customer.py

Jawaban:

```
def checkId(self):
```

```
    return self.id
```

```
def checkPin(self):
```

```
    return self.pin
```

```
def checkBalance(self):
```

```
    return self.balance
```



customer.py

Kita akan coba mengerjakan fitur debet dan simpan. Fitur debet, artinya saldo nasabah akan dikurangi dengan “nominal” yang nantinya ditentukan. Fitur simpan, artinya saldo nasabah akan di tambah dengan “nominal” yang nantinya ditentukan. Rumusnya,

Debet = saldo - nominal

Simpan = saldo + nominal



customer.py

Latihan 6:

Buatlah dua buah method untuk melakukan proses debet dan simpan. Masing-masing method ini memiliki parameter tambahan berupa **nominal**, yang argumennya nanti akan ditentukan ketika program berjalan.

Anda dapat membaca lagi penjelasan logika di atas atau materi dari Progate jika merasa bingung..



customer.py

Jawaban:

```
def withdrawBalance(self, nominal):
```

```
    self.balance -= nominal
```

```
def depositBalance(self, nominal):
```

```
    self.balance += nominal
```


Checkpoint

Coba jalankan file `atm_card.py` dan `customer.py` melalui terminal/command line. Pastikan tidak ada error pada saat dijalankan. Untuk cara menjalankan program Python melalui terminal/command line, silakan merujuk pada artikel Progate mengenai pengaturan environment Python.



atm_program.py

Inti dari program yang ada pada mesin ATM adalah *looping* dan *control flow*. Dan karena pendekatan pengembangan program kita berbasis class, maka kita juga akan coba menggunakan modul bawaan Python dan mengaplikasikan *instance object* untuk mengerjakan seluruh fitur yang tersisa.



atm_program.py

Latihan 7:

Lakukanlah import modul Python untuk modul `random` dan `datetime`. Import juga class `Customer` agar kita dapat menggunakan method dan properties yang dimilikinya. Kemudian buat *instance* object dari `Customer` dengan parameter `id`.



atm_program.py

Jawaban:

```
import random
```

```
import datetime
```

```
from customer import Customer
```

```
atm = Customer(id)
```



atm_program.py

Program kita akan dimulai dengan sebuah loop yang bernilai benar. Kemudian program dimulai dengan meminta input dari user berupa nomor pin dan membuat variable yang akan membatasi percobaan dari user ketika nomor pin yang dimasukkan ini salah.

Nomer pin yang benar ini akan menjadi identitas nasabah. Program akan lanjut berjalan jika inputpin benar, dan berhenti jika salah memasukkan pin 3 kali.



atm_program.py

Latihan 8:

Buatlah loop menggunakan `while` yang bernilai benar. Kemudian buatlah `inputan` untuk user agar bisa memasukkan nomer pin. Rubah inputan nilai pin ini ke dalam `integer` dan jadikan variable bagi parameter `id`. Siapkan juga sebuah variable yang nantinya digunakan untuk membatasi percobaan user ketika memasukkan pin. Selamat mengerjakan.



atm_program.py

Jawaban:

```
while True:
```

```
    id = int(input("Masukkan pin anda: "))
```

```
    trial = 0
```



atm_program.py

Kita akan menyebut looping ini sebagai “looping pemeriksaan” untuk memudahkan, karena nantinya masih akan ada looping lagi. Selanjutnya, di dalam looping pemeriksaan ini, kita akan membuat “looping verifikasi”. Looping verifikasi ini akan mencocokkan pin yang di input oleh user dengan pin yang dimiliki oleh nasabah. Input yang salah lebih dari tiga kali akan menyebabkan program berhenti dan user harus mengulang lagi dari awal.



atm_program.py

Latihan 9:

Buatlah looping verifikasi menggunakan `while`, yang bernilai benar ketika kondisinya adalah, pin yang dimasukkan user tidak sama seperti pin nasabah dan ketika percobaan masih kurang dari tiga kali.

Jika kondisi terpenuhi, maka user akan di minta untuk memasukkan pin lagi dan di catat sebagai satu kali percobaan.



atm_program.py

Jawaban:

```
while (id != int(atm.checkPin())) and trial < 3):  
    id = int(input("Pin anda salah. Silakan  
Masukkan lagi: "))  
    trial += 1
```



atm_program.py

Tidak masalah, jika anda bisa menjawab latihan di atas dengan benar meskipun sintaksnya berbeda.

Selanjutnya, masih pada looping verifikasi, disini juga terdapat sebuah kondisi, di mana ketika percobaan input pin telah mencapai tiga kali, maka user akan mendapatkan pemberitahuan error dan program akan berhenti berjalan.



atm_program.py

Latihan 10:

Buatlah sebuah kondisi, di mana ketika user telah mencoba memasukkan pin salah sebanyak tiga kali, user akan mendapatkan pemberitahuan error dan program akan berhenti berjalan.



atm_program.py


Jawaban:

```
if trial == 3:
```

```
    print("Error. Silakan ambil kartu dan coba  
lagi..")
```

```
    exit()
```

atm_program.py



Pada tahap ini, kita telah menyelesaikan looping pemeriksaan (yang pertama) dan looping verifikasi, di mana ini juga merupakan salah satu fitur program yang diminta, yaitu cek pin.

Selanjutnya kita akan membuat looping lagi di dalam looping pemeriksaan. Ini akan kita sebut sebagai “looping utama”. Looping utama ini akan berisi pilihan menu-menu utama dan kondisi yang terjadi ketika user memilih sebuah menu dari menu-menu yang ada.

Checkpoint



```
1  while True:
2
3      id = int(input("Masukkan pin anda: "))
4      trial = 0
5
6      while (id != int(atm.checkPin()) and trial < 3):
7          id = int(input("Pin anda salah. Silakan Masukkan lagi: "))
8
9          trial += 1
10
11         if trial == 3:
12             print("Error. Silakan ambil kartu dan coba lagi..")
13             exit()
```



atm_program.py

Latihan 11:

Buatlah sebuah looping `while` bernilai `true`, dengan indentasi (spasi) yang sama seperti looping verifikasi. Jika nilainya benar, looping ini akan menampilkan pesan selamat datang dan pilihan menu utama tampilan mesin ATM berupa: cek saldo, debet, simpan, ganti pin, keluar.



atm_program.py

Jawaban:

```
while True:
```

```
    print("Selamat datang di ATM Progate..")
```

```
    print("\n1 - Cek Saldo \t 2 - Debet \t 3 -  
Simpan \t 4 - Ganti Pin \t 5 - Keluar ")
```



atm_program.py

Latihan 12:

Di bawah function print() dari code di atas, buatlah inputan untuk user, yang kemudian diubah menjadi integer dan di simpan ke dalam sebuah variable.



atm_program.py

Jawaban:

```
selectmenu = int(input("\nSilakan pilih menu: "))
```



atm_program.py

Pada tahap ini, user akan memilih menu berdasarkan angka. Setiap angka yang di input oleh user akan menjadi acuan untuk fitur yang akan dijalankan. Sehingga kita perlu membuat control flow yang mampu merespon setiap angka yang di input oleh user dan menampilkan fitur yang dimaksud dengan tepat.



atm_program.py

Latihan 13:

Dengan mengacu pada jawaban dari latihan 12, buatlah sebuah control flow untuk merespon pilihan user terhadap menu pilihan pertama hingga ke lima. Jangan lupa untuk menutup control flow dengan pilihan lain seandainya user memasukkan inputan yang tidak tepat.



atm_program.py

Jawaban:

```
if selectmenu == 1:
```

```
    elif selectmenu == 2:
```

```
    elif selectmenu == 3:
```

```
    elif selectmenu == 4:
```

```
    elif selectmenu == 5:
```

```
else:
```



atm_program.py

Jangan khawatir jika code ini memunculkan error, karena kita nanti akan mengisi setiap kondisi ini dengan statement. Setiap kondisi yang ada ini, akan mewakili satu fitur yang telah didefinisikan di atas. Dan statement yang akan kita tambahkan ini, semuanya akan memanggil setidaknya satu method dari instance object “atm” yang telah kita buat sebelumnya.



atm_program.py

Latihan 14:

Jika user memilih menu pertama, tampilkan jumlah saldo nasabah saat ini dengan cara memanggil method `checkBalance()`.



atm_program.py

Jawaban:

```
if selectmenu == 1:
```

```
    print("\nSaldo anda sekarang: Rp. " +
```

```
str(atm.checkBalance()) + "\n" )
```



atm_program.py

Latihan 15:

Jika user memilih menu kedua, buatlah input untuk menerima nominal saldo yang akan di debet. Nominal ini akan dibandingkan dengan saldo nasabah, jika nominal < saldo, maka method `withdrawBalance()` akan dipanggil dan sisa saldo akan ditampilkan

atm_program.py

Jawaban:

```
elif selectmenu == 2:
    nominal = float(input("Masukkan nominal saldo: "))
    verify_withdraw = input("Konfirmasi: Anda akan melakukan debet dengan nominal
berikut ? y/n " + str(nominal) + " ")

    if verify_withdraw == "y":
        print("Saldo awal anda adalah: Rp. " + str(atm.checkBalance()) + "")
    else:
        break
    if nominal < atm.checkBalance():
        atm.withdrawBalance(nominal)
        print("Transaksi debet berhasil!")
        print("Saldo sisa sekarang: Rp. " + str(atm.checkBalance()) + "")
    else:
        print("Maaf. Saldo anda tidak cukup untuk melakukan debet!")
        print("Silakan lakukan penambahan nominal saldo")
```



atm_program.py

Latihan 16:

Jika user memilih menu ketiga, panggil method `depositBalance()` dan tampilkan saldo saat ini.



atm_program.py

Jawaban:

```
elif selectmenu == 3:
    nominal = float(input("Masukkan nominal saldo: "))
    verify_deposit = input("Konfirmasi: Anda akan melakukan  
penyimpanan dengan nominal berikut ? y/n " + str(nominal) + " ")

    if verify_deposit == "y":
        atm.depositBalance(nominal)
        print("Saldo anda sekarang adalah: Rp." +  
str(atm.checkBalance()) + "\n" )
    else:
        break
```



atm_program.py

Latihan 17:

Jika user memilih menu keempat, buatlah inputan untuk nomor pin baru dan lakukan verifikasi dengan membandingkannya dengan pin nasabah saat ini.

atm_program.py

Jawaban:

```
elif selectmenu == 4:
```

```
    verify_pin = int(input("masukkan pin anda: "))
```

```
    while verify_pin != int(atm.checkPin()):
```

```
        print("pin anda salah, silakan masukkan pin:")
```

```
    updated_pin = int(input("silakan masukkan pin baru: "))
```

```
    print("pin anda berhasil diganti!")
```

```
    verify_newpin = int(input("coba masukkan pin baru: "))
```

```
    if verify_newpin == updated_pin:
```

```
        print("pin baru anda sukses!")
```

```
    else:
```

```
        print("maaf, pin anda salah!")
```



atm_program.py

Latihan 18:

Jika user memilih menu kelima, program akan berakhir. Namun program juga akan membuat resi dengan mencetak nomor record berdasarkan random integer antara 100000-1000000, tanggal saat ini dan sisa saldo terakhir.

Jawaban:

```
elif selectmenu == 5:  
    print("Resi tercetak otomatis saat anda keluar. \n Harap  
simpan tanda terima ini \n sebagai bukti transaksi anda.")  
    print("No. Rekord: ", random.randint(100000, 1000000))  
    print("Tanggal: ", datetime.datetime.now())  
    print("Saldo akhir: ", atm.checkBalance())  
    print("Terima kasih telah menggunakan ATM Progate!")  
    exit()
```



atm_program.py

Latihan 16:

Jika user memilih menu lainnya, cetaklah informasi error yang dapat dibaca oleh user.

atm_program.py

Jawaban:

```
else:
```

```
    print("Error. Maaf, menu tidak tersedia")
```

atm_program.py

Jika sudah berhasil tanpa error sampai sini, maka kita telah selesai membuat program dan berhasil dijalankan.

Jalankan program atm_python.py pada terminal / command line dan lihatlah hasilnya apakah sudah sesuai dengan daftar fitur yang diminta.