

1 Pertemuan 2: Latihan Python Dasar 2

Nama :

NIM :

1.1 Petunjuk

- Kerjakan soal-soal dibawah ini dengan menuliskan kode *python* di sel yang sudah disediakan.
- Jawaban yang diminta oleh soal harus tercetak sebagai *output* masing-masing sel.
- Berikan komentar yang cukup pada kode sehingga dapat dipahami oleh orang lain.
- Pada tugas ini tidak diperbolehkan menggunakan *library*!
- Total nilai tersedia adalah 150 (100 dari 3 soal yang harus dijawab + 50 Poin berasal dari pertanyaan bonus/opsional). **Maksimalkan bonus yang tersedia untuk memperbaiki nilai pada aktivitas-aktivitas sebelumnya yang mungkin masih kurang.**
- Jika sudah selesai, kirim file jupyter notebook (.ipynb) melalui LMS.
- Pastikan nama file dan juga Nama dan NIM telah terisi dengan benar.
- Batas waktu pengumpulan tugas adalah 19 Oktober 23:59.

In [21]:

```
1 # jangan ubah kode ini
2 import random
```

executed in 4ms, finished 07:48:57 2022-10-12

1.2 Soal 1: Piutang Jatuh Tempo (20%)

Pada tahun 2022, toko Kalimongso Berdikari memiliki daftar buku piutang dagang yang direpresentasikan dalam variable `piutang_dagang` . Variable ini adalah list of tuples `[(tuple_1, tuple_2, ..., tuple_n)]` dimana setiap tuple berisi 2 elemen yakni (**jumlah piutang, hari ke-h jatuh tempo pada tahun 2022**).

Hitunglah:

1. Jumlah piutang dagang yang jatuh tempo pada hari ke 100 hingga 120.
2. Jumlah piutang dagang yang jatuh tempo pada hari ke 300 hingga 320.

Hint: Gunakan Loop, If, Tuple/List Indexing. Bonus 10% jika bisa mengimplementasikan dalam bentuk fungsi.

In [23]:

```
1 # jangan ubah kode ini
2 random.seed(12)
3 piutang_dagang = [(random.randint(100,7500)*1000, random.randint(1,3
4 piutang_dagang[:5]
```

executed in 6ms, finished 07:49:45 2022-10-12

Out[23]: [(3987000, 138), (5486000, 271), (5559000, 180), (1268000, 196), (188000, 192)]

In []:

```
1 # jawaban
```

1.3 Soal 2: Mencari Supplier Terbaik (30%)

Toko Kalimongso Berdikari ingin membeli barang persediaan dari supplier. Namun mereka bingung karena terdapat banyak sekali supplier barang tersebut dengan harga yang berbeda-beda dan mereka menginginkan tentunya harga yang paling murah, namun mereka juga memikirkan kriteria berikut:

- Kualitas sesuai dengan standar minimal Toko, yakni kualitas minimal tertentu. Kualitas diberikan dengan angka 1-5 semakin tinggi semakin berkualitas.
- Ongkos kirim dari supplier ke toko Kalimongso Berdikari per satu unit barang berbeda-beda sehingga harus diperhitungkan.

Diberikan sebuah data dalam bentuk list `katalog_persediaan` yang isinya adalah list of tuple untuk toko Kalimongso Berdikari. `[tuple_1, tuple_2,, tuple_n]`. Setiap tuple berisi 4 elemen yakni **(Kode Suplier, harga barang per unit, ongkos kirim per unit, kualitas)**.

NB : Tidak ada korelasi antara harga barang dan kualitas barang.

Gunakan iterasi untuk menemukan supplier terbaik dengan kriteria berikut.

1. Supplier dengan harga dan ongkir termurah dengan kelas 2.
2. Supplier dengan harga dan ongkir termurah dengan kualitas 3
3. Supplier dengan harga dan ongkir termurah dengan kualitas minimal 4.

Cetak hasil dari pertanyaan 1,2,3 dengan menampilkan nama suplier, harga, ongkir dan kualitas.

Hint: Gunakan Loop, If, Tuple Indexing atau Tuple Unpacking
(<https://www.geeksforgeeks.org/unpacking-a-tuple-in-python/>
(<https://www.geeksforgeeks.org/unpacking-a-tuple-in-python/>)). Bonus 10% jika bisa mengimplementasikan dalam bentuk fungsi.

In [18]:

```
1 # jangan ubah kode ini
2 random.seed(59)
3 katalog_persediaan = [(f"SUP_{x}", random.randint(500,750)*1000, random.randint(2,5)) for x in range(10)]
4
5 # ditampilkan hanya 5 data pertama
6 katalog_persediaan[:5]
```

executed in 7ms, finished 07:44:54 2022-10-12

Out[18]:

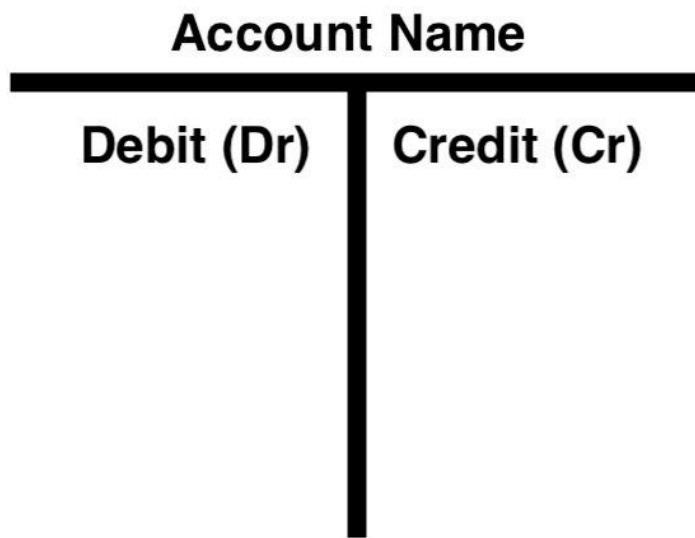
```
[('SUP_0', 557000, 71000, 4),
 ('SUP_1', 505000, 84000, 3),
 ('SUP_2', 682000, 55000, 5),
 ('SUP_3', 502000, 174000, 4),
 ('SUP_4', 624000, 187000, 3)]
```

In []:

```
1 # Jawaban
```

1.4 Soal 3: Pemrograman Berorientasi Objek Untuk Akun (50%)

Kalian tentunya sudah memahami apa itu Account dalam akuntansi.



Definisikan sebuah kelas (class) dalam pemrograman python dengan nama Akun . Kelas ini nantinya digunakan untuk mencatat transaksi debit/kredit.

Anda bebas berkreasi sesuai dengan imajinasi anda, namun hal penting yang harus ada dalam kelas ini adalah:

- Terdapat atribut/properti untuk menyimpan informasi nama akun, kode akun, jenis saldo akun (debit/kredit), dan saldo awal akun
- Terdapat fungsi untuk mencetak informasi akun di atas.
- Terdapat fungsi untuk merekam transaksi (penjurnalan dengan input minimal : nilai debit, nilai kredit)
- Terdapat fungsi untuk mencetak seluruh riwayat transaksi
- Terdapat fungsi untuk mencetak saldo akun terakhir

Hint: Anda bisa mengimplementasikan banyak hal disini selain OOP, seperti list, iterasi dan pernyataan logika.

Bonus : Tersedia 20% jika anda mampu mengimplementasikan akun tersebut **(tidak hanya mendefinisikan)**. Dimana anda menginisiasi akun tersebut dan mencatat contoh transaksi dan mencetak saldo akhir. Anda bisa bebas menggunakan nilai apapun dalam proses implementasi dengan minimal 3 contoh transaksi. Tambahan 20% jika anda bisa menambahkan informasi tambahan pada transaksi saat penjurnalan disamping hanya informasi nilai debit dan kredit (seperti tanggal transaksi atau keterangan).

In [26]:

```
1  # jawaban disini
2  # note: building block ini hanya untuk gambaran, kalian boleh merubah
3
4  class Akun:
5      def __init__(self):
6          pass
7
8      def info(self):
9          pass
10
11     def cetak_saldo(self):
12         pass
13
14     def cetak_transaksi(self):
15         pass
16
17     def rekam_transaksi(self):
18         pass
```

executed in 2ms, finished 08:11:50 2022-10-12

In []:

1

2

3

implementasi (bonus)