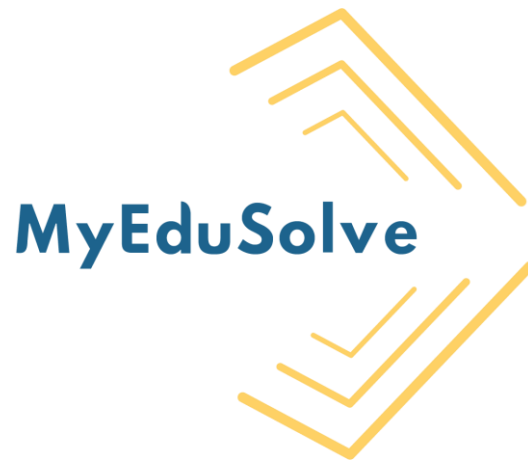


MINI PROJECT PYTHON

MyEduSolve



Disusun Oleh:

Dhina Bastari Rochim

Mentor: Ramdhan Cahyadi

STUDI INDEPENDEN

MERDEKA BELAJAR – KAMPUS MERDEKA

2023

STUDI KASUS 1

1. Bu Inge adalah seorang dosen informatika, saat ini beliau sedang melakukan survey kondisi sosial ekonomi masyarakat di kampung durian runtuh. Bu Inge mengambil sampel berupa 10 keluarga. Bantulah Bu Inge dalam melakukan survey nya
 - a. Lakukan running dengan iterasi 10x untuk membaca input dari user yang berupa nama kepala keluarga dan penghasilan nya
 - i. Contoh setiap baris input: 'Adi alkatiri | 6.7' Yang menunjukkan bahwa kepala keluarga nya adalah Adi alkatiri dan penghasilan bulanan keluarganya adalah 6.7
 - ii. Kemudian ditentukan status ekonomi-nya dengan kriteria sebagai berikut->
#Jika penghasilan diatas 15 juta, maka statusnya crazy rich
#Jika penghasilan antara 5 juta sampai 15 juta, maka statusnya Makmur
#Jika penghasilan antara 2 juta sampai 5 juta, maka statusnya mandiri
#Jika penghasilan antara 1 juta sampai 2 juta, maka statusnya butuh pendampingan
#Jika penghasilan antara 0 sampai 1 juta, maka statusnya rentan
 - b. Untuk mendapatkan status ekonomi, diwajibkan membuat sebuah fungsi yang menerima parameter berupa nilai dan mengembalikan status ekonomi-nya. Contoh pemanggilan fungsi: `status_ekonomi(10)`, hasil pemanggilan nya menghasilkan "makmur"
 - c. Perhatikan bahwa "penghasilan tidak mungkin negatif". Sehingga saat user memberikan input negatif maka. fungsi `status_ekonomi` harus memberikan exception
 - d. Data tersebut dimasukkan pada pandas dataframe dengan kolom sebagai berikut
 - i. Format kolom: Nama kepala keluarga | penghasilan bulanan dalam juta | index
 - ii. Contoh row data: Adi Budi | 10 | makmur
 - e. Setelahnya, silahkan simpan data dengan format csv dengan nama data.csv

PENYELESAIAN

Sebelum melakukan program, kita harus mengimport library yang akan digunakan, di antaranya yaitu library pandas untuk menganalisis data dalam bentuk dataframe, dan library tabulate yang digunakan untuk membuat tampilan dataframe agar lebih rapi dan terstruktur. Sebelum import tabulate, install terlebih dahulu menggunakan code dibawah ini.

```
[ ] pip install tabulate      #sebelum import tabulate, install terlebih dahulu  
Requirement already satisfied: tabulate in /usr/local/lib/python3.10/dist-packages (0.9.0)
```

```
import pandas as pd  
from tabulate import tabulate      #untuk menerapkan tabel pada dataframe yang ada
```

Membuat list yang digunakan untuk menyimpan nilai atau informasi yang akan diinputkan

```
nama_kepala_keluarga = []      # digunakan untuk menyimpan nilai atau informasi nama kepala keluarga  
penghasilan_keluarga = []      # digunakan untuk menyimpan nilai atau informasi penghasilan keluarga
```

Membuat fungsi untuk mengkategorikan status ekonomi keluarga berdasarkan penghasilan yang didapatkan dengan kategori sebagai berikut:

#Jika penghasilan diatas 15 juta, maka statusnya crazy rich

#Jika penghasilan antara 5 juta sampai 15 juta, maka statusnya makmur #Jika penghasilan antara 2 juta sampai 5 juta, maka statusnya mandiri

#Jika penghasilan antara 1 juta sampai 2 juta, maka statusnya butuh pendampingan

#Jika penghasilan antara 0 sampai 1 juta, maka statusnya rentan

```
def kategori(penghasilan):      # 1b, menggunakan function kategori dengan parameter penghasilan  
    if penghasilan > 15_000_000:      # menggunakan percabangan untuk kondisi status atau kategori dari penghasilan keluarga  
        return 'Crazy Rich'          # dimana ada 5 kondisi status ekonomi sesuai dengan penghasilan yang diinput  
    elif penghasilan >= 5_000_000 and penghasilan <= 15_000_000:  
        return 'Makmur'  
    elif penghasilan >= 2_000_000 and penghasilan < 5_000_000:  
        return 'Mandiri'  
    elif penghasilan >= 1_000_000 and penghasilan < 2_000_000:  
        return 'Butuh Pendampingan'  
    else:  
        return 'Rentan'
```

Selanjutnya membuat list survey untuk menyimpan data hasil survey berupa nama kepala keluarga dan penghasilan per bulan. Karena akan dilakukan iterasi sebanyak 10x maka menggunakan perulangan for dengan range(10). Lalu membuat fungsi untuk menginput data dari user berupa nama kepala keluarga dan penghasilan, dengan menggunakan perulangan while true memastikan bahwa inputan yang diterima adalah valid di mana program akan berjalan selama kondisi dalam while bernilai true, dan jika kondisi terpenuhi maka akan berhenti. Untuk mendeteksi apakah inputan dari user sudah benar atau belum maka menggunakan percabangan if else yaitu:

jika inputan user tidak kosong dan bukan digit atau angka, maka break

jika inputan user selain statement if awal, maka akan menghasilkan output berupa keterangan “Nama hanya berupa string atau tulisan”

```
survey = [] # digunakan untuk menyimpan nilai atau informasi survey yaitu nama kepala keluarga dan penghasilannya per bulan
for i in range(10): # menggunakan perulangan for karena Bu Inge ingin input 10 data sebagai sampel

    while True: # diberi perulangan while dan akan berjalan selama kondisi dalam while bernilai true, jika kondisi terpenuhi maka break
        nama = input("Masukkan nama kepala keluarga : ") #inputan dari user untuk nama kepala keluarga
        if nama.strip() != '' and not nama.isdigit(): #1c menggunakan percabangan untuk mendeteksi apakah inputan user sudah benar atau belum,
            break # jika inputan user tidak kosong dan bukan digit atau angka, maka break
        else: #jika inputan user selain statement if awal, maka print sesuai keterangan disamping ini
            print("Nama hanya berupa string atau tulisan")
```

Selanjutnya membuat inputan user untuk penghasilan bulanan dengan menggunakan try dan except. Try digunakan untuk mengubah inputan user menjadi data float kemudian diberi percabangan if else untuk mengatasi beberapa kondisi dimana jika data inputan bernilai negative maka akan memunculkan pesan error dan jika inputan bernilai positif maka akan berhenti. Kemudian except digunakan untuk melakukan pengecualian apabila data yang diinputkan oleh user tidak sesuai (selain float) maka akan menghasilkan output pesan error yaitu “penghasilan hanya berupa angka”. Setelah itu, kita menggunakan kode append untuk mengumpulkan data hasil suvey yang terdiri nama kepala keluarga dan penghasilan, kemudian melakukan perintah print untuk menampilkan data hasil survey.

```
while True: # diberi perulangan while dan akan berjalan selama kondisi dalam while bernilai true, jika kondisi terpenuhi maka break
    penghasilan = input("Masukkan Penghasilan : ") #inputan dari user untuk penghasilan keluarga
    try: # menggunakan try dan except dengan fungsinya untuk mengubah inputan user menjadi float yang nantinya bisa memunculkan pengecualian jika inputan tid
        penghasilan2 = float(penghasilan.replace(".", "", 1)) #di inisialisasi bahwa inputan penghasilan yang ada titiknya akan dirubah ke float dan satu titik desimal ya
        if penghasilan2 < 0: #1c kondisi ini dibuat untuk inputan yang bernilai negatif, maka akan memunculkan pesan error
            print(f"Keluarga {i} dengan nama kepala keluarga {nama} memiliki penghasilan negatif, silahkan masukkan penghasilan yang bernilai positif")
        else: #jika inputan tidak negatif maka break
            break
    except ValueError: #kita ingin mengatasi pengecualian karena jika inputan tidak float maka akan muncul pesan error dibawah ini
        print("Penghasilan hanya berupa angka")
survey.append((nama, penghasilan2)) #code append untuk mengumpulkan data survey yang terdiri dari nama dan penghasilan keluarga
print()
```

Selanjutnya membuat kolom baru untuk nama kepala keluarga dan penghasilan, karena tadi kita sudah menyimpan data ke dalam bentuk tuple maka untuk data kedua kolom ini mengambil dari data inputan yang sudah tersimpan. Kemudian membuat kolom baru untuk status ekonomi dengan menggunakan metode apply untuk mengimplememtasikan fungsi kategori berdasarkan kolom penghasilan bulanan. Setelah itu membuat dataframe untuk menampilkan data ke dalam bentuk table.

```
data = {
    'Kepala Keluarga': [item[0] for item in survey], #1d membuat kolom baru yaitu kepala keluarga dan penghasilan
    'Penghasilan Bulanan': [item[1] for item in survey] #kita punya tuple survey, untuk kolom kepala keluarga, kita mengambil semua elemen pertama dari data survey(
} #kita punya tuple survey, untuk kolom penghasilan bulanan, kita mengambil semua elemen kedua dari data surve

print("    ^^ Hasil Survey Ekonomi Masyarakat ^^")
print("=====")
df = pd.DataFrame(data) #membuat dataframe 'df' dimana dataframe akan memunculkan 2 kolom baru yang telah dibuat sebelumnya
print()
df['Status Ekonomi'] = df['Penghasilan Bulanan'].apply(kategori) #membuat kolom baru bernama status ekonomi dan menggunakan metode apply untuk mengimplementasikan fu
print(tabulate(df, headers='keys', tablefmt='pretty')) #mencetak dataframe dengan output tabel. fungsi tabulate untuk mencetak dataframe 'df', headers='key
```

Setelah itu, data tersebut disimpan ke dalam format csv untuk memudahkan akses data kemudian unduh data ke penyimpanan.

```
df.to_csv('data.csv', index=False) #untuk menyimpan dataframe ke dalam file format csv dengan nama 'data.csv'
```

```
from google.colab import files #mengimpor modul files dari pustaka Google Colab
files.download('data.csv') #unduh file CSV
```

Jika kita ingin mendownload dataframa yang sudah kita simpan ke dalam fromat csv tadi, kita bisa mendownload file tersebut dengan cara import file terlebih dahulu dan download menggunakan code diatas.

Output:

jika nama kepala keluarga tidak string, maka muncul pesan error, jika valid maka akan lanjut meminta inputan penghasilan

```
... Masukkan nama kepala keluarga : 7
Nama hanya berupa string atau tulisan
Masukkan nama kepala keluarga : adi
Masukkan Penghasilan : 0.0
```


jika inputan penghasilan string, maka muncul pesan error

```
Masukkan nama kepala keluarga : budi beriman
Masukkan Penghasilan : budi
Penghasilan hanya berupa angka
Masukkan Penghasilan : 
```

jika inputan penghasilan bernilai negatif, maka muncul pesan error

```
Masukkan nama kepala keluarga : budi beriman
Masukkan Penghasilan : budi
Penghasilan hanya berupa angka
Masukkan Penghasilan : -9.000
Keluarga 1 dengan nama kepala keluarga budi beriman memiliki penghasilan negatif, silahkan masukkan penghasilan yang bernilai positif
Masukkan Penghasilan : 
```

Hasil output survey 10 sampel

```
✓ 2m  Masukkan nama kepala keluarga : adi
Masukkan Penghasilan : 0.0

Masukkan nama kepala keluarga : budi beriman
Masukkan Penghasilan : 2000000

Masukkan nama kepala keluarga : cyntia
Masukkan Penghasilan : 3000000

Masukkan nama kepala keluarga : deni alfarisi
Masukkan Penghasilan : 4000000

Masukkan nama kepala keluarga : eko fahrudin
Masukkan Penghasilan : 6200000

Masukkan nama kepala keluarga : fani veronica
Masukkan Penghasilan : 7200000

Masukkan nama kepala keluarga : ghani berwawasan
Masukkan Penghasilan : 89000000

Masukkan nama kepala keluarga : hendra setiawan
Masukkan Penghasilan : 19000000

Masukkan nama kepala keluarga : indira
Masukkan Penghasilan : 5000000

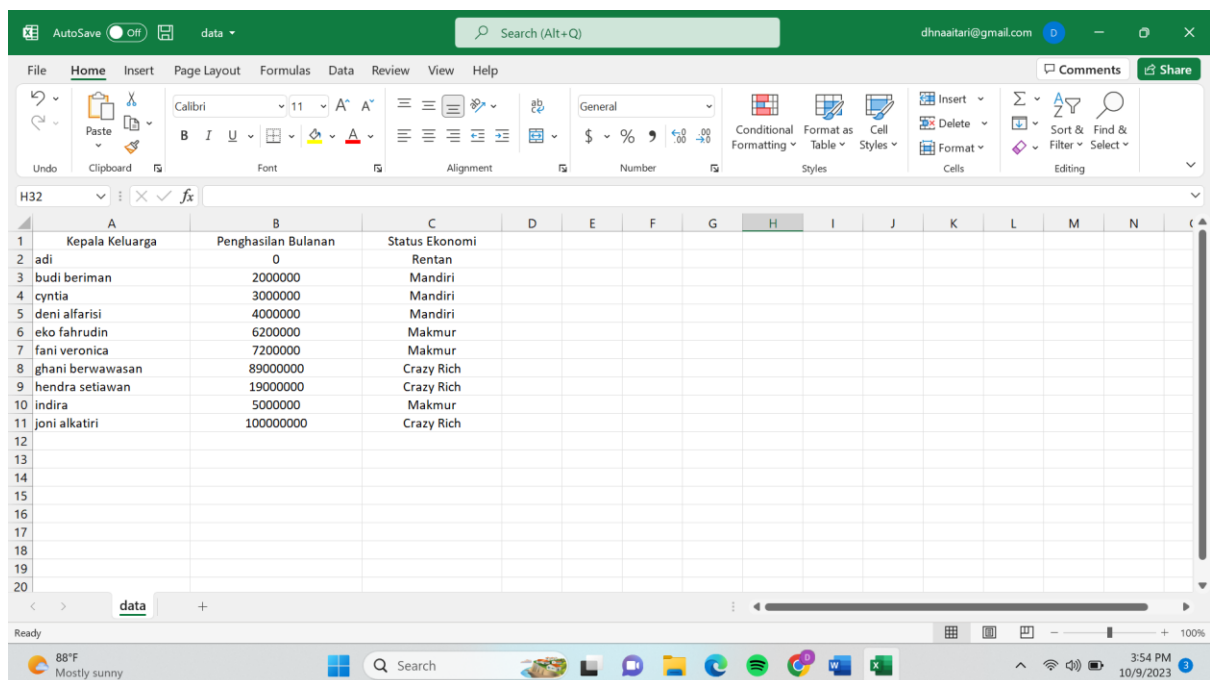
Masukkan nama kepala keluarga : joni alkatiri
Masukkan Penghasilan : 100000000
```

```
# Output dataframe
```

^^ Hasil Survey Ekonomi Masyarakat ^^

=====			
+-----+-----+-----+-----+			
	Kepala Keluarga	Penghasilan Bulanan	Status Ekonomi
+-----+-----+-----+-----+			
0	adi	0.0	Rentan
1	budi beriman	2000000.0	Mandiri
2	cyntia	3000000.0	Mandiri
3	deni alfarisi	4000000.0	Mandiri
4	eko fahrudin	6200000.0	Makmur
5	fani veronica	7200000.0	Makmur
6	ghani berwawasan	89000000.0	Crazy Rich
7	hendra setiawan	19000000.0	Crazy Rich
8	indira	5000000.0	Makmur
9	joni alkatiri	100000000.0	Crazy Rich
+-----+-----+-----+-----+			

```
# File csv data hasil survey:
```



STUDI KASUS 2

2. Anda adalah seorang Dosen di kelas Dasar Pemrograman yang memiliki 40 mahasiswa.

Ada 3 nilai berikut:

np.random.normal(mean, var, n)

tugas ~ (85, 7, 40)

UTS ~ (60, 15, 40)

UAS ~ (75, 10, 40)

Simpan data dalam bentuk Data Frame.

Kemudian buat kolom baru (Nilai Akhir) yang dihitung dengan formula:

tugas: 20%

UTS: 40%

UAS: 40%

Buat kolom baru dan lakukan binning dengan kriteria berikut:

A : 85-100

A- : 80-85

B+ : 75-80

B : 70-75

B- : 65-70

C+ : 60-65

C : 55-60

C- : 50-55

D : 20-50

E : 0-20

Hasil akhir yang diharapkan seperti format di bawah ini, sedangkan angka nya bisa berbeda karena random.

	tugas	uts	uas	nilai_akhir	predikat
0	84.174601	33.012306	85.430154	64.211904	C+
1	83.794759	34.141719	68.315071	57.741668	C
2	86.532528	89.259938	75.949682	83.390354	A-

PENYELESAIAN

Pada kasus di atas, kita akan menyimpan data dalam bentuk Data Frame, kemudian akan dibuat kolom baru (Nilai Akhir) yang dihitung dengan formula tugas sebesar 20%, UTS sebesar 40%, dan UAS sebesar 40%. Maka dari itu, untuk menyelesaikan kasus di atas syntax yang akan digunakan adalah sebagai berikut:

```
[ ] pip install tabulate      #sebelum import tabulate, install terlebih dahulu  
  
Requirement already satisfied: tabulate in /usr/local/lib/python3.10/dist-packages (0.9.0)
```

```
▶ #SOAL NO 2  
import pandas as pd          #untuk membuat dataframe  
import numpy as np          #untuk komputasi numerik  
from tabulate import tabulate #membuat tabel yang terstruktural
```

Pada awal syntax kita diharuskan mengimport modul pandas as pd yang akan digunakan untuk memanipulasi dan menganalisis data dalam bentuk dataframe. Selain itu, kita juga diharuskan mengimport modul numpy as np yang akan digunakan untuk melakukan komputasi numerik. Selanjutnya, kita akan mengimpor fungsi dengan syntax from tabulate import tabulate yang akan digunakan untuk membuat tabel pada dataframe agar rapi dan terstruktur dengan catatan sebelum melakukan import tabulate, kita harus sudah menginstall tabulate.

```
data = {  
    #membuat 3 kolom baru  
    'Nilai Tugas': np.random.normal(85, 7, 40), #perintah numpy disamping untuk menghasilkan 40 angka acak dengan rata-rata 85 dan standar deviasinya 7.  
    'Nilai Uts': np.random.normal(60, 15, 40), #perintah numpy disamping untuk menghasilkan 40 angka acak dengan rata-rata 60 dan standar deviasinya 15.  
    'Nilai Uas': np.random.normal(75, 10, 40) #perintah numpy disamping untuk menghasilkan 40 angka acak dengan rata-rata 75 dan standar deviasinya 10.  
}
```

Setelah tahap pengimport-an, langkah selanjutnya adalah membuat 3 kolom baru sesuai dengan nilai yang disediakan pada soal, yaitu nilai tugas, UTS, dan UAS. Dengan menggunakan perintah np.random.normal (Perintah Numpy), maka akan dihasilkan 40 angka acak beserta rata-rata dan standar deviasi dari ke tiga nilai tersebut, yakni sebagai berikut:

- 'Nilai Tugas': 40 nilai acak yang diambil dari distribusi normal dengan rata-rata 85 dan deviasi standar 7.
- 'Nilai Uts': 40 nilai acak yang diambil dari distribusi normal dengan rata-rata 60 dan deviasi standar 15.
- 'Nilai Uas': 40 nilai acak yang diambil dari distribusi normal dengan rata-rata 75 dan deviasi standar 10.

Selanjutnya, kita akan membuat dataframe df yang kemudian akan dihitung nilai akhir dengan proses sebagai berikut:

```
df = pd.DataFrame(data) #membuat dataframe 'df'

df['Nilai Akhir'] = 0.2 * df['Nilai Tugas'] + 0.4 * df['Nilai Uts'] + 0.4 * df['Nilai Uas'] #suatu proses untuk hitung nilai akhir

angka = [0, 20, 50, 55, 60, 65, 70, 75, 80, 85, 100] #daftar yang berisi rentang nilai ketentuan kategori hurufnya
huruf = ['E', 'D', 'C-', 'C', 'C+', 'B-', 'B', 'B+', 'A-', 'A'] #daftar yang berisi rentang nilai huruf sesuai ketentuan angkanya

df['Nilai Huruf'] = pd.cut(df['Nilai Akhir'], angka, labels=huruf) #membuat kolom baru nilai huruf yang berdasarkan nilai akhir. pd.cut untuk mengelompokkan data ke dalam kategori berdasarkan rentan
table = tabulate(df, headers='keys', tablefmt='pretty', showindex=False) #labels=huruf adalah daftar yang berisi nilai huruf yang sesuai dengan setiap rentang nilai
#table diatas dibuat untuk mencetak dataframe dan dijadikan kedalam bentuk tabel yang terstruktur
print(table) #mencetak tabel yang telah dibuat
```

Proses menghitung nilai akhir diawali dengan menggabungkan nilai tugas, nilai UTS, dan nilai UAS dengan bobot masing-masing 0.2, 0.4, dan 0.4. Hasilnya disimpan dalam kolom 'Nilai Akhir'. Definisi angka dan huruf digunakan untuk menentukan rentang batasan untuk mengkategorikan 'Nilai Akhir' ke dalam kategori nilai huruf. Misalnya, jika 'Nilai Akhir' berada dalam rentang 85-100, akan diberi label 'A'.

'Nilai Huruf' dihitung menggunakan fungsi pd.cut, yang membagi 'Nilai Akhir' ke dalam kategori berdasarkan rentang yang telah ditentukan. Kemudian, tabulate digunakan untuk menghasilkan tabel dari dataframe df dimana labels=huruf adalah daftar yang berisi nilai huruf yang sesuai dengan setiap rentang nilai. Tabel ini akan mencakup semua kolom dalam dataframe, dan dijadikan ke dalam bentuk table yang terstruktur, dengan kolom 'Nilai Tugas', 'Nilai Uts', 'Nilai Uas', 'Nilai Akhir', dan 'Nilai Huruf' di dalamnya. Selanjutnya, tabel hasilnya dicetak ke layar dengan format yang rapi menggunakan print.

Output:

Nilai Tugas	Nilai Uts	Nilai Uas	Nilai Akhir	Nilai Huruf
93.9212617783016	93.07264258783192	63.42435735734473	81.38305233373099	A-
84.5352968634949	71.90282515421697	76.70495587227599	76.35017178329616	B+
83.58508157871253	49.22013279016743	68.06003684078271	63.62908416812257	C+
91.13028447345764	48.09120692565865	81.34998845172683	70.00253504564571	B
93.20512914840994	76.2902446193313	78.81060678900043	80.68136639301468	A-
79.81808345790861	57.65253248034723	76.64013350459093	69.68068308555699	B-
79.22469546537891	82.74892744145903	80.9532763348111	81.32582060358385	A-
71.6870116094061	50.976036968322816	80.50967552040534	66.93168731737248	B-
85.01154699381091	63.997613539542954	83.46059313994674	75.98559207055806	B+
86.90748471936756	69.05915214218503	68.50321686188367	72.40644454550099	B
79.70004226116879	62.81269742918268	63.50157712515834	66.46571827397017	B-
90.76125470668532	73.63207934250882	77.16752677496946	78.47209338832837	B+
84.75665968692431	30.21314188118921	76.6670974759552	59.703427680242626	C
80.27352134762728	53.799459406436554	93.04837821244651	74.79383931707869	B
83.54464313656821	58.71007044204173	77.5562823679897	71.21546975132621	B
77.01390135898279	56.24270856105521	73.57664013930827	67.33051975194195	B-
87.41186274000603	73.71670628137204	67.81123892880073	74.09355063207032	B
88.13179168423628	52.45367037539285	79.29770663060444	70.32690913924618	B
81.94399427894328	58.50357594782953	80.69499931982556	72.0682289628507	B
77.29648459874977	67.86390838173078	108.56396581845539	86.03044659982442	A
90.79518455235151	56.17347630035058	70.1520467379365	68.68924612578513	B-
92.79852996819973	77.88585239121139	85.37125629406678	83.86254946775122	A-
82.16034120640649	55.878738403509395	79.42104575129665	70.55198190320371	B
87.62660212306994	49.32589119344869	65.47086529005769	63.444023018016544	C+
78.14101546734244	64.87658527861363	76.94316913289926	72.35610485807365	B
79.54298205145126	57.2043515877664	77.41246826546612	69.75532435158327	B-
90.76125470668532	73.63207934250882	77.16752677496946	78.47209338832837	B+
84.75665968692431	30.21314188118921	76.6670974759552	59.703427680242626	C
80.27352134762728	53.799459406436554	93.04837821244651	74.79383931707869	B
83.54464313656821	58.71007044204173	77.5562823679897	71.21546975132621	B
77.01390135898279	56.24270856105521	73.57664013930827	67.33051975194195	B-
87.41186274000603	73.71670628137204	67.81123892880073	74.09355063207032	B
88.13179168423628	52.45367037539285	79.29770663060444	70.32690913924618	B
81.94399427894328	58.50357594782953	80.69499931982556	72.0682289628507	B
77.29648459874977	67.86390838173078	108.56396581845539	86.03044659982442	A
90.79518455235151	56.17347630035058	70.1520467379365	68.68924612578513	B-
92.79852996819973	77.88585239121139	85.37125629406678	83.86254946775122	A-
82.16034120640649	55.878738403509395	79.42104575129665	70.55198190320371	B
87.62660212306994	49.32589119344869	65.47086529005769	63.444023018016544	C+
78.14101546734244	64.87658527861363	76.94316913289926	72.35610485807365	B
79.54298205145126	57.2043515877664	77.41246826546612	69.75532435158327	B-
86.70303690500832	57.65754192630206	57.01174256931072	63.20832117924678	C+
93.73690398209108	59.868759527501226	59.93012472571508	66.66693449770474	B-
81.02592867083327	61.54715167954598	85.85746112004914	75.16703085400471	B+
81.76909399763431	52.238568650457296	58.53923601958395	60.66494066754336	C+
92.96379222472517	68.66561886546167	57.98066197059936	69.25127077936945	B-
85.1144519169392	57.394019688775906	77.36815522695693	70.92776034968098	B
96.07747064451338	95.66809092067436	80.12068208094084	89.53100332954875	A
71.42373128004807	45.51043012766215	83.56293407797449	65.91409193826428	B-
91.51808600623949	77.01791535649758	89.90500171231118	85.0727840287714	A
94.78995671960243	56.12220557143831	77.18575456523058	72.28117539858805	B
83.36500577689144	55.17917823993831	79.4257021195869	70.51495329918836	B
92.6403770482314	47.15484369160748	71.2487581527293	65.88951614738099	B-
78.30283008711922	60.96139937462171	70.48439997587981	68.23888575762446	B-
85.34881727285627	77.35210292551685	65.35804107187558	74.15382105352823	B

Berdasarkan output diatas, 'Nilai Tugas', 'Nilai Uts', dan 'Nilai Uas' adalah nilai-nilai tugas, UTS, dan UAS yang dihasilkan secara acak untuk 40 mahasiswa. 'Nilai Akhir' adalah nilai akhir yang dihitung berdasarkan bobot tertentu dari nilai tugas, UTS, dan UAS. 'Nilai Huruf' adalah hasil kategori dari 'Nilai Akhir' yang menggambarkan kinerja mahasiswa berdasarkan skala nilai huruf (misalnya, A, B, C, dll.).

Link Google Colab:

https://colab.research.google.com/drive/1Kv3uzh0Md-kvDAeRaUQn_7RpIiZkNryu?usp=sharing