LAPORAN TUGAS BESAR
ALGORITMA PEMROGRAMAN

Anggota Kelompok 8 :

Christian Arvianus Nathanael Biran : 121450112
Kanaya Dea Thalita Akhmad : 121450001
Sella Dianka Fitri : 121450000
Ayu Erlinawati : 121450025
Ibnu Farhan Al-Ghifari : 121450121

SAINS DATA
INSTITUT TEKNOLOGI SUMATERA
Tahun Ajaran 2022/2023

ABSTRAK

Data adalah sekumpulan keterangan ataupun fakta yang dibuat dengan kata-kata, kalimat, simbol, angka, dan lainnya. Data-data yang kemudian diolah akan menghasilkan informasi, informasi inilah yang akan digunakan untuk pengambilan keputusan, untuk mendapatkan informasi ini, dapat dilakukan operasi data yang memuat data explore, data cleaning, data sorting dan lain-lain, kita juga bisa melakukan pengolahan data agar menjadi visual sehingga memudahkan para pengamat untuk memahami data-data tersebut.

Pada tugas besar Algoritma Pemrograman ini, kami bertujuan untuk membangun code menggunakan Jupyter Notebook (ipynb) yang kemudian digunakan untuk melakukan pengolahan lebih lanjut terhadap dataset yang sebelumnya telah dipilih dari sumber data publik yaitu "Kaggle", dataset ini bernama winemag-data-130k-v2 yang berbentuk csv, data yang kami gunakan ini memiliki jumlah 129971 baris dan 13 kolom yang terdiri atas tipe data string, float dan integer..

Beberapa pengolahan yang kami lakukan diantaranya adalah Exploring dataset dimana pengguna dapat mengeksplorasi data yang telah dipilih dengan tujuan untuk lebih memahami dataset, Cleaning dataset yang bertujuan untuk membersihkan data-data yang sudah tidak dibutuhkan, Searching yang digunakan agar pengguna dapat melakukan pencarian dengan mudah terhadap data yang diinginkan, Sorting yaitu proses pengurutan data yang sebelumnya disusun secara acak sehingga dapat tersusun secara teratur menurut aturan tertentu dan Visualization yang berfungsi untuk mengolah data yang dimiliki untuk kemudian dibuat menjadi data yang tervisualisasi.

Selain itu digunakan juga berbagai library yang bertujuan untuk mempermudah operasi data, diantaranya adalah Pandas yang digunakan untuk data processing, Numpy yang digunakan untuk melakukan operasi linear algebra, Seaborn yang berfungsi untuk operasi visualisasi pada data dan yang terakhir Matplotlib.

Dari percobaan yang kami lakukan dalam pembuatan code operasi dataset ini, dapat disimpulkan bahwa data yang kami gunakan telah berhasil diproses melewati data cleaning, sorting, visualization dan lain-lain sehingga dataset tersebut menghasilkan informasi berbentuk visualisasi data dalam seperti piechart, subplots dan boxplot.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Seiring dengan perkembangan ilmu sains data, pengolahan dataset dapat dilakukan dengan lebih cepat dan juga efisien. Pengolahan data dapat dilakukan dengan berbagai macam bahasa pemrograman seperti Python dan juga Studio R, selain itu kita juga dapat memanfaatkan banyak library yang ada untuk lebih memaksimalkan pengolahan data yang kita lakukan. Dengan latar belakang ini maka kami melakukan praktik pengolahan data yang mencakup data explore, data cleaning, data sorting dan lain-lain, dengan menggunakan bantuan dari library Pandas, Numpy, Seaborn dan juga Matplotlib.

1.2 Rumusan Masalah

1. Bagaimana aplikasi pengolahan data menggunakan bahasa pemrograman python agar didapat informasi yang berguna dari dataset yang ada?
2. Bagaimana proses operasi dataset yang ada dengan mengaplikasikan library Pandas, Numpy, Seaborn dan Matplotlib?
3. Bagaimana cara membentuk code yang dapat digunakan untuk data cleaning, sorting, exploring, searching dan visualization dengan efisien?

1.3 Deskripsi Data

Dataset yang kami gunakan merupakan kumpulan data-data yang berbentuk file scv, dataset ini terdiri atas tipe data string, float dan juga integer yang terdiri dari 129971 kolom dan juga 13 baris. Data-data ini terdiri dari kolom-kolom country, description, designation, points, price, province, region_1, region_2, taster_name, taster_twitter_handle, title, variety, winery.

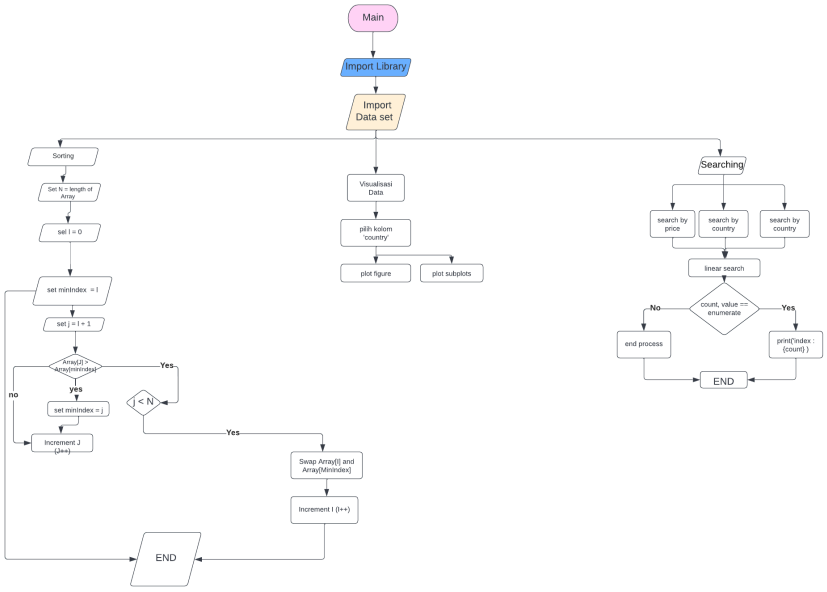
\

BAB II
METODE DAN RANCANGAN SISTEM

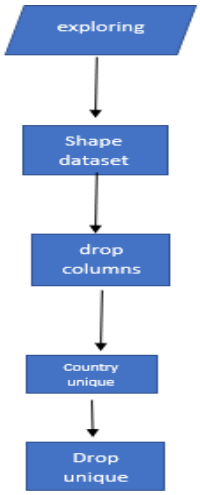
2.1. Gambaran Umum Metode yang digunakan

- Pada Langkah pertama yang digunakan adalah exploring dataset, eksploring dataset ini digunakan sebagai penggambaran umum dari data set yang akan dipakai, digunakan untuk mengecek setiap bagian bagian dari tabel dataset yang terpakai
- Langkah selanjutnya yaitu cleaning data set, pada bagian ini bertujuan untuk memangkas beberapa kolom yang tidak terpakai dalam dataset sehingga analisis dalam penelitian ini dapat terfokus pada beberapa kolom
- Langkah ketiga dibuat fungsi searching. Bertujuan untuk pencarian data dari banyaknya dataset yang kita punya. Mempermudah pencarian index dari data
- Keempat, dibuat sorting. Penelitian ini menambahkan sorting dengan tujuan melakukan pengurutan dari dataset. Salah satu kolom akan kami uji coba untuk pengurutan datanya
- Langkah terakhir yaitu visualization, langkah ini bertujuan untuk menampilkan beberapa grafik yang akan digunakan sebagai deskripsi dataset yang kami punya.

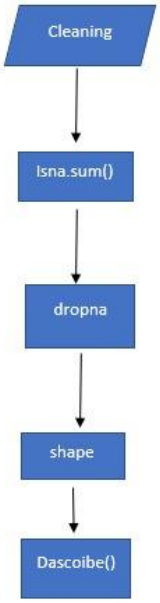
2.2. Flowchart Sistem



- **exploring**



- **Cleaning**



2.3. Dependency / Library yang digunakan

- **Pandas**
  Dalam penelitian ini, program menggunakan library pandas. Library pandas ini digunakan sebagai pengolahan data terlebih di dalam data frame. Pemrosesan data dalam python menggunakan library pandas tentunya.
- **Numpy**
  Penggunaan library numpy berfungsi sebagai pengolah data yang bersifat aljabar linear. Penggunaannya juga untuk mengembalikan dari fungsi array.
- **Seaborn**
  Library seaborn tentu menjadi salah satu yang terpenting dalam program ini. Library ini digunakan sebagai penampil grafik dari dataset. Visualisasi dataset membutuhkan library seaborn.
- **Matplotlib pyplot**
  Penggunaan matplotlib digunakan sebagai penampil grafik dari beberapa kolom pada dataset. Matplotlib itu sendiri sebagai library yang akan menampilkan grafik plot dalam program ini.

2.4. Daftar Fungsionalitas

- **Read Excel**
  Dilakukan untuk membaca dataset yang baru saja di import dalam python. Import ini dilakukan agar library pandas dapat membaca dataset yang berbentuk excel dan akan ditampilkan sebagai dataframe.
- **Eksploring Data Set**
  Memberikan penggambaran secara umum mengenai data set yang akan digunakan kali ini. Dataset yang digunakan dalam penelitian ini adalah dataset yang diambil dari platform kaggle dengan judul ‘winemag data’
    - Data Frame Shape
      Menampilkan ukuran dari data set, panjang kolom dan baris akan diukur menggunakan data frame.shape. Dalam program ini ukuran dari dataset ‘winemag’ adalah 129971 baris dan 13 kolom
    - Data Frame Columns
      Menampilkan index kolom, nama nama dari index kolom dataset akan dikembalikan dengan fungsi ini.
    - Data Frame Drop
      Drop digunakan untuk memangkas beberapa kolom yang tidak digunakan dalam penelitian kali ini. Tujuan dari pemangkasan kolom ini bertujuan agar penganalisisan dikhususkan pada data ‘country’, ‘price’ dan ‘points’
    - Unique ( )
      Digunakan untuk mencari data yang unique, atau bisa disebut data yang sama. Data yang sama akan memberikan hasil analisis yang tidak murni. Oleh karena itu penggunaan unique ( ) untuk menyeleksi data yang duplikat.
    - Drop Duplicates
      Untuk menghapus data yang duplikat setelah diseleksi sebelumnya pada kolom ‘country’.

- **Cleaning Data set**
  Digunakan untuk membersihkan beberapa bagian dari kolom ataupun baris yang tidak terpakai dalam program ini. Sehingga analisis data dapat terfokus pada beberapa kolom saja dalam dataset.
  - Summary
    Menampilkan berapa banyak jumlah data yang ada pada kolom ‘country’, ‘points’ dan ‘price’. Juga menampilkan tipe data dari dataset yang digunakan.
  - Isna
    Mengecek apakah ada nilai NaN pada dataframe. Function ini akan mengembalikan nilai Boolean, True atau False, untuk tiap elemen di dataframe
  - describe()

    digunakan untuk menampilkan deskriptif statistik data. Hanya kolom yang bertipe numerik yang akan ditampilkan statistiknya.

- **Searching**
  Algoritma pencarian untuk data yang terurut. Pencarian dilakukan dengan cara menebak apakah data yang dicari berada ditengah-tengah data, kemudian membandingkan data yang dicari dengan data yang ada ditengah. Bila data yang ditengah sama dengan data yang dicari, berarti data ditemukan.
  - Linear Search

    teknik mencari data dengan cara memplot data satu per satu. Jika kecocokan ditemukan, program akan mengembalikan hasilnya, jika tidak, pencarian akan dilanjutkan hingga akhir array

- **Sorting**

  Selection Sort melakukan pengurutan dengan konsep memilih elemen dengan nilai paling rendah dan menukar elemen tersebut dengan elemen ke i. Nilai dari i dimulai dari 1 ke n, yang dimana n merupakan jumlah total elemen dikurangi satu.

  ```
  def SelectionSort(arr):
      # Traverse through 1 to len(arr)
      for i in range(len(arr)):
      # Find the minimum element in remaining
      # unsorted array
          min_idx = i
          for j in range(i+1, len(arr)):
              if arr[min_idx] > arr[j]:
                  min_idx = j

      # Swap the found minimum element with
      # the first element
      arr[i], arr[min_idx] = arr[min_idx], arr[i]
      return arr
  ```

BAB III

HASIL PROGRAM (screenshot use case)

- ## Library

```
[97] import pandas as pd # data processing
     import numpy as np # linear algebra
     import seaborn as sns # visualisasi data
     import matplotlib.pyplot as plt
```

```
[98] wine_df = pd.read_csv('/winemag-data-130k-v2.csv', index_col = 0)
     #wine_df.head()
```

- ## Exploring dataset

```
[99] wine_df.shape
```

```
(129971, 13)
```

```
[100] wine_df.columns
```

```
Index(['country', 'description', 'designation', 'points', 'price', 'province',
       'region_1', 'region_2', 'taster_name', 'taster_twitter_handle', 'title',
       'variety', 'winery'],
      dtype='object')
```

```
wine_df = wine_df.drop(columns = ['description', 'designation', 'region_1', 'region_2', 'taster_twitter_handle', 'pr
wine_df.head()
```

	country	points	price
0	Italy	87	NaN
1	Portugal	87	15.0
2	US	87	14.0
3	US	87	13.0
4	US	87	65.0

```
[103] wine_df.country.unique()
```

```
array(['Italy', 'Portugal', 'US', 'Spain', 'France', 'Germany',
       'Argentina', 'Chile', 'Australia', 'Austria', 'South Africa',
       'New Zealand', 'Israel', 'Hungary', 'Greece', 'Romania', 'Mexico',
       'Canada', nan, 'Turkey', 'Czech Republic', 'Slovenia',
       'Luxembourg', 'Croatia', 'Georgia', 'Uruguay', 'England',
       'Lebanon', 'Serbia', 'Brazil', 'Moldova', 'Morocco', 'Peru',
       'India', 'Bulgaria', 'Cyprus', 'Armenia', 'Switzerland',
       'Bosnia and Herzegovina', 'Ukraine', 'Slovakia', 'Macedonia',
       'China', 'Egypt'], dtype=object)
```

```
[104] wine_df.drop_duplicates(inplace = True)
```

- ## Cleaning Dataset

```
[105] wine_df.isna().sum()
```

```
country     45
points       0
price      192
dtype: int64
```

```
[106] wine_df.dropna(inplace = True)
```

```
[107] wine_df.shape
```

```
(10392, 3)
```

```
[108] wine_df['price'].describe()
```

```
count    10392.000000
mean        69.246536
std        106.380201
min          4.000000
25%         22.000000
50%         41.000000
75%         78.000000
max       3300.000000
Name: price, dtype: float64
```

```
[109] reduced_price = np.array(wine_df['price'][:6])
```

- **Searching**

```
[ ]  def search_by_country(name):
         result = wine_df.loc[wine_df['country'] == name]
         return result.head()

     def search_by_price(num):
         result = wine_df['price'] == int(num)
         return wine_df[result].head()

     def search_by_points(num):
         result = wine_df['points'] == int(num)
         return wine_df[result].head()
```

```
[ ]  search_by_country('US')
```

	country	points	price
2	US	87	14.0
3	US	87	13.0
4	US	87	65.0
10	US	87	19.0
12	US	87	34.0

```
[ ]  def LinearSearch(lys, element):
         values = np.array(lys)
         for count, value in enumerate(values):
             if value == element:
                 return f'index : {count+1}'
```

```
[ ]  LinearSearch(wine_df['country'], 'US')
```

```
'index : 2'
```

- **Sorting**

```
[ ]  reduced = np.array(wine_df['price'][:20])
     reduced
```

```
array([15., 14., 13., 65., 15., 16., 24., 12., 27., 19., 30., 34., 12.,
       24., 30., 13., 28., 32., 23., 20.])
```

```
[ ]  def sorting_values_asc(col):
         result = wine_df.sort_values(col, ascending = True)
         return result.head()

     def sorting_values_des(col):
         result = wine_df.sort_values(col, ascending = False)
         return result.head()
```

```
[ ]  def SelectionSort(arr):
         # Traverse through 1 to len(arr)
         for i in range(len(arr)):
         # Find the minimum element in remaining
         # unsorted array
             min_idx = i
             for j in range(i+1, len(arr)):
                 if arr[min_idx] > arr[j]:
                     min_idx = j

         # Swap the found minimum element with
         # the first element
             arr[i], arr[min_idx] = arr[min_idx], arr[i]
         return arr
```

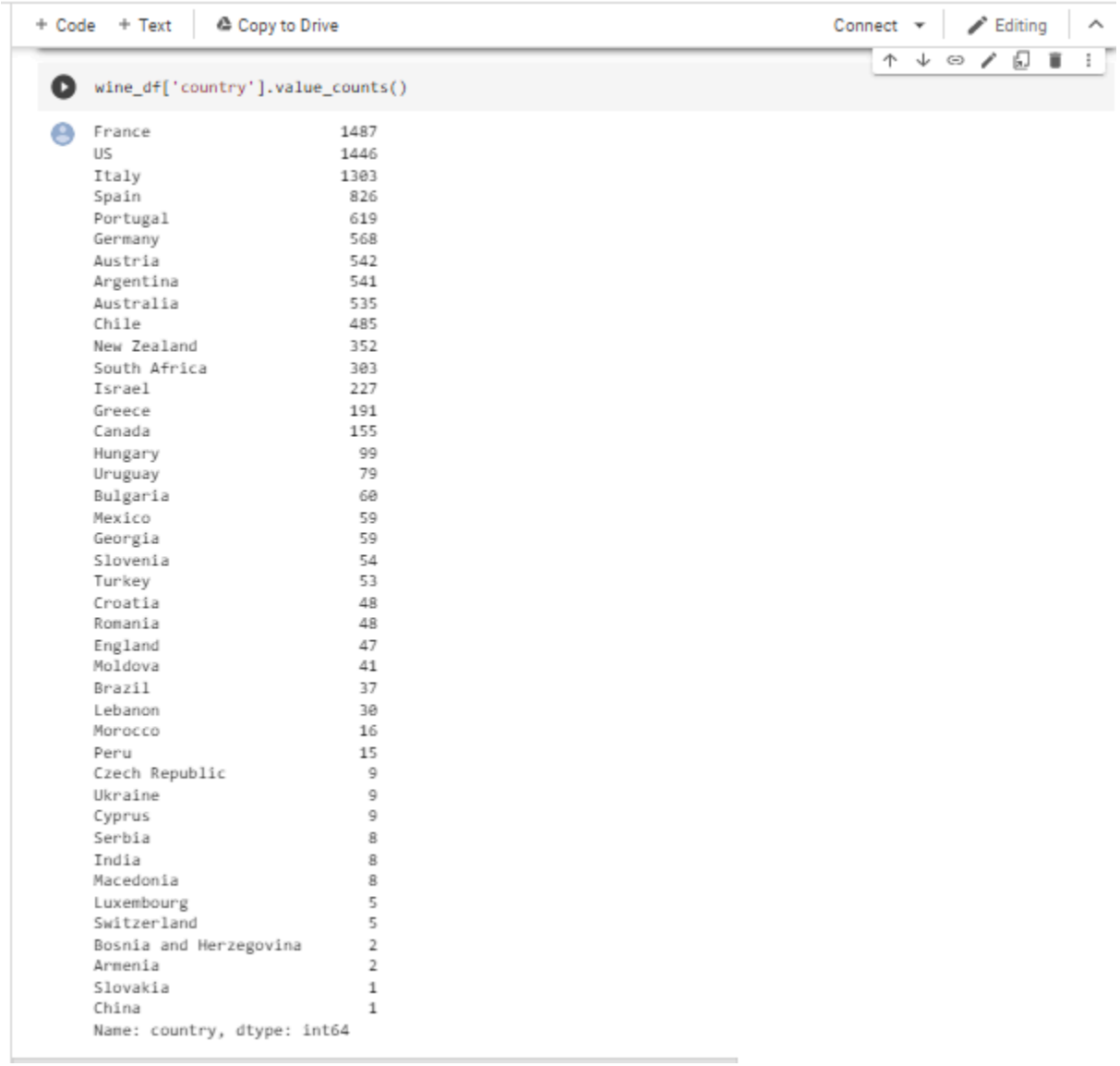
```
[ ]  SelectionSort(reduced)
```

```
array([12., 12., 13., 13., 14., 15., 15., 16., 19., 20., 23., 24., 24.,
       27., 28., 30., 30., 32., 34., 65.])
```

```
[ ]  sorting_values_asc('price')
```

	country	points	price
20484	US	85	4.0
1987	Spain	85	4.0
64590	US	86	4.0
31530	US	84	4.0
59507	US	83	4.0

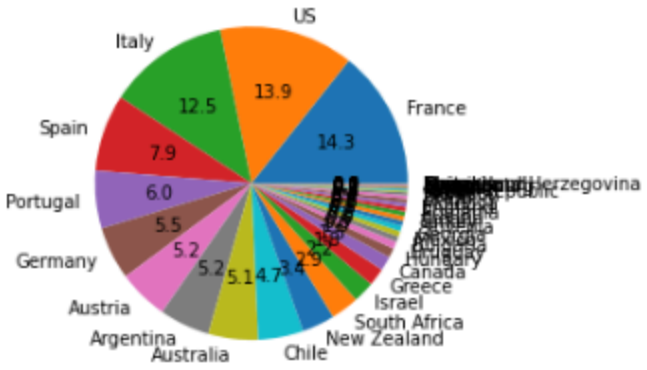
- **Visualization**


+ Code
+ Text
Copy to Drive
Connect
Editing
wine_df['country'].value_counts()
France 1487
US 1446
Italy 1303
Spain 826
Portugal 619
Germany 568
Austria 542
Argentina 541
Australia 535
Chile 485
New Zealand 352
South Africa 303
Israel 227
Greece 191
Canada 155
Hungary 99
Uruguay 79
Bulgaria 60
Mexico 59
Georgia 59
Slovenia 54
Turkey 53
Croatia 48
Romania 48
England 47
Moldova 41
Brazil 37
Lebanon 30
Morocco 16
Peru 15
Czech Republic 9
Ukraine 9
Cyprus 9
Serbia 8
India 8
Macedonia 8
Luxembourg 5
Switzerland 5
Bosnia and Herzegovina 2
Armenia 2
Slovakia 1
China 1
Name: country, dtype: int64

```
fig = plt.figure()
plt.pie(wine_df['country'].value_counts(), labels = wine_df['country'].value_counts().keys(),
```

```
([<matplotlib.patches.Wedge at 0x7fd4a9b8d850>,
  <matplotlib.patches.Wedge at 0x7fd4a9b8dd00>,
  <matplotlib.patches.Wedge at 0x7fd4a9b643d0>,
  <matplotlib.patches.Wedge at 0x7fd4a9b64a60>,
  <matplotlib.patches.Wedge at 0x7fd4a9b61130>,
  <matplotlib.patches.Wedge at 0x7fd4a9b617c0>,
  <matplotlib.patches.Wedge at 0x7fd4a9b61e50>,
  <matplotlib.patches.Wedge at 0x7fd4a9b51520>,
  <matplotlib.patches.Wedge at 0x7fd4a9b51bb0>,
  <matplotlib.patches.Wedge at 0x7fd4a9b6f280>,
  <matplotlib.patches.Wedge at 0x7fd4a9b8d820>,
  <matplotlib.patches.Wedge at 0x7fd4a9b6ff70>,
  <matplotlib.patches.Wedge at 0x7fd4a9b52640>,
  <matplotlib.patches.Wedge at 0x7fd4a9b52cd0>,
  <matplotlib.patches.Wedge at 0x7fd4a9c4a3a0>,
  <matplotlib.patches.Wedge at 0x7fd4a9c4aa30>,
  <matplotlib.patches.Wedge at 0x7fd4a9c51100>,
  <matplotlib.patches.Wedge at 0x7fd4a9c51790>,
  <matplotlib.patches.Wedge at 0x7fd4a9c51e20>,
  <matplotlib.patches.Wedge at 0x7fd4a9c644f0>,
  <matplotlib.patches.Wedge at 0x7fd4a9c64b80>,
  <matplotlib.patches.Wedge at 0x7fd4a9c4c250>,
  <matplotlib.patches.Wedge at 0x7fd4a9c4c8e0>,
  <matplotlib.patches.Wedge at 0x7fd4a9c4cf70>,
  <matplotlib.patches.Wedge at 0x7fd4a9c3d640>,
  <matplotlib.patches.Wedge at 0x7fd4a9c3dcd0>,
  <matplotlib.patches.Wedge at 0x7fd4a7ff03a0>,
  <matplotlib.patches.Wedge at 0x7fd4a7ff0a30>,
  <matplotlib.patches.Wedge at 0x7fd4a7fe3100>,
  <matplotlib.patches.Wedge at 0x7fd4a7fe37c0>,
  <matplotlib.patches.Wedge at 0x7fd4a7fe3e50>,
  <matplotlib.patches.Wedge at 0x7fd4a7fd4520>,
  <matplotlib.patches.Wedge at 0x7fd4a7fd4bb0>,
  <matplotlib.patches.Wedge at 0x7fd4a7fc4280>,
  <matplotlib.patches.Wedge at 0x7fd4a7fc4910>,
  <matplotlib.patches.Wedge at 0x7fd4a7fc4fa0>,
  <matplotlib.patches.Wedge at 0x7fd4a7fb7670>,
```

```
Text(0.5962302282406694, -0.0671529219921161, '0.4'),
Text(0.597647802643393, -0.053076397725580336, '0.4'),
Text(0.5986001859339276, -0.04096116941527966, '0.3'),
Text(0.5991119034388714, -0.03263322168974908, '0.2'),
Text(0.5993914148144842, -0.027017250909204604, '0.1'),
Text(0.599571658127737, -0.022667747350717988, '0.1'),
Text(0.5996861285758631, -0.019404823979963775, '0.1'),
Text(0.5997828420305275, -0.016141326023078387, '0.1'),
Text(0.5998578752794624, -0.013058693089616614, '0.1'),
Text(0.5999140220281897, -0.010157075069171326, '0.1'),
Text(0.5999561332224626, -0.0072552194143734585, '0.1'),
Text(0.5999800131039119, -0.00489733354283268, '0.0'),
Text(0.5999920765256828, -0.0030835217526759143, '0.0'),
Text(0.5999972583269254, -0.001813835762331846, '0.0'),
Text(0.5999990130073616, -0.0010882969226432518, '0.0'),
Text(0.5999997532581332, -0.0005441416905271818, '0.0'),
Text(0.5999999725870545, -0.00018137125950768275, '0.0')])
```

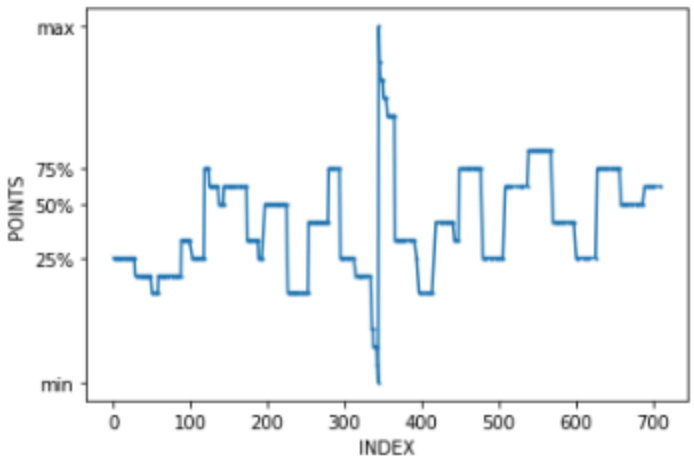



```
[ ] wine_df['points'].describe()
```

```
count    10392.000000
mean        89.404349
std          3.989001
min         80.000000
25%         87.000000
50%         90.000000
75%         92.000000
max        100.000000
Name: points, dtype: float64
```

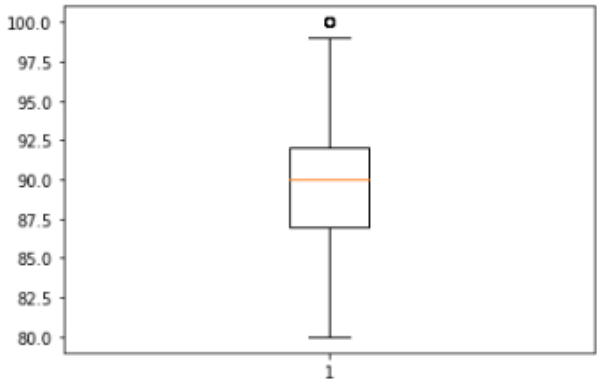
```
[ ]  fig, ax = plt.subplots()
     ax.plot(wine_df['points'][:500], marker = 'o', markersize = 1)
     ax.set_ylabel('POINTS')
     ax.set_xlabel('INDEX')
     ax.set_yticks([80, 87, 90, 92, 100])
     ax.set_yticklabels(['min', '25%', '50%', '75%', 'max'])
```

```
[Text(0, 0, 'min'),
 Text(0, 0, '25%'),
 Text(0, 0, '50%'),
 Text(0, 0, '75%'),
 Text(0, 0, 'max')]
```



```
[ ]  fig = plt.figure()
     plt.boxplot(wine_df['points'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7fd4a8779b50>,
  <matplotlib.lines.Line2D at 0x7fd4a8779eb0>],
 'caps': [<matplotlib.lines.Line2D at 0x7fd4a88b0250>,
  <matplotlib.lines.Line2D at 0x7fd4a88b05b0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7fd4a87797f0>],
 'medians': [<matplotlib.lines.Line2D at 0x7fd4a88b0910>],
 'fliers': [<matplotlib.lines.Line2D at 0x7fd4a88b0c10>],
 'means': []}
```




```
[ ]  def inserting(country, points, price):
       #wine_df = wine_df.append({'country' : country, 'points' : int(points), 'p
       wine_df.loc[len(wine_df)] = [country, points, price]
       return wine_df.tail()
```

```
[ ]  inserting('indo', 90, 980)
```

	country	points	price
129818	France	89	115.0
129886	Argentina	91	88.0
129931	France	91	107.0
129948	Argentina	90	43.0
10392	indo	90	980.0

```
[ ]  wine_df.shape
```

```
(10393, 3)
```

```
[ ]  def deleting_row(i):
       wine_df.drop(wine_df.index[int(i)], inplace = True)
       return wine_df.tail()
```

```
[ ] deleting_row(10392)
```

	country	points	price
129777	Argentina	89	70.0
129818	France	89	115.0
129886	Argentina	91	88.0
129931	France	91	107.0
129948	Argentina	90	43.0

```
[ ] wine_df.shape
```

```
(10392, 3)
```

```
[ ]
```

Kesimpulan

Algoritma dan pemrograman adalah urutan atau langkah langkah yang dilakukan untuk menyelesaikan suatu permasalahan pada program. Pada tugas besar ini kami membuat program dimana kami mengambil dataset public dari Kaggle yaitu winemag-data-130k-v2 dataset yang berbentuk csv. Dalam program tersebut kami menggunakan empat library yang sudah dipelajari yaitu Numpy, Pandas, Seaborn, dan Matplotlib. Library berfungsi untuk membuat pemrograman python menjadi lebih sederhana dan nyaman bagi programmer karena tidak perlu menulis kode yang sama berulang kali untuk program yang berbeda. Pada visualisasi data yang kami lakukan pandas digunakan untuk membaca dan menampilkan data, Numpy untuk mengolah data yang bersifat aljabar linear, Seaborn sebagai penampil grafik dari dataset dan matplotlib untuk membuat grafik plot yang mempresentasikan kolom yang ada di dataset tersebut.