
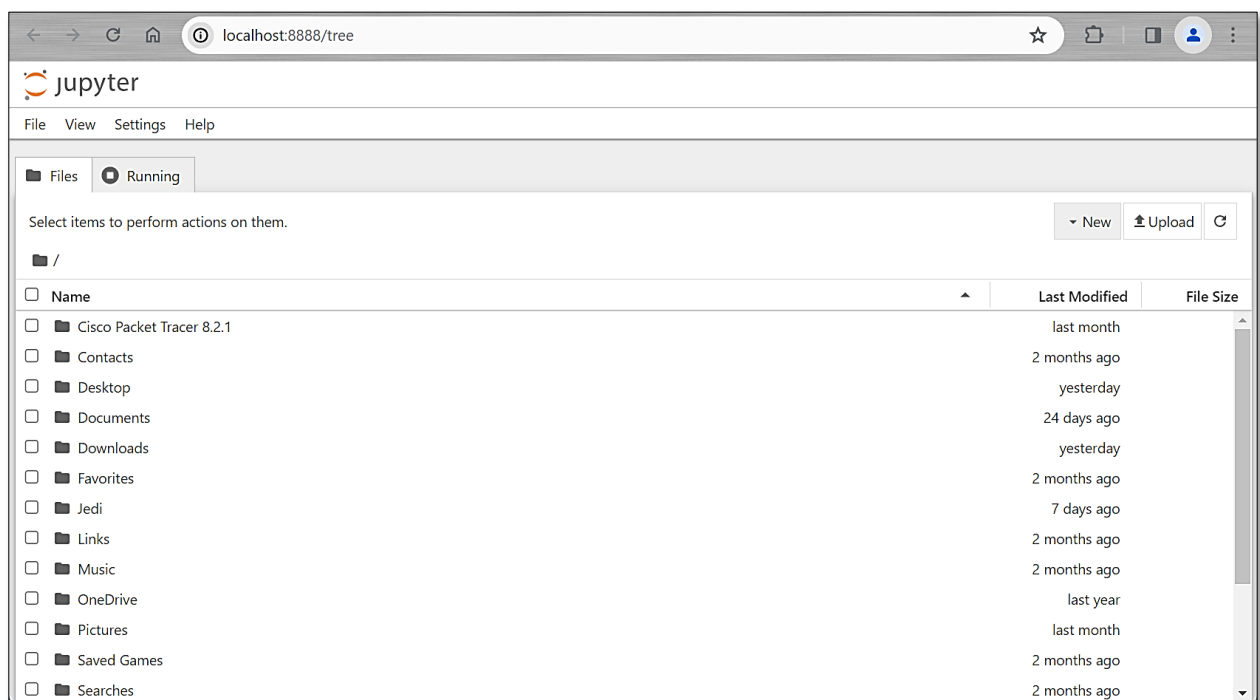


Jupyter Notebook

 Kata "Jupyter" berasal dari gabungan singkatan 3 nama bahasa pemrograman yaitu Julia, Python, dan R. Jupyter notebook merupakan aplikasi berbasis web yang banyak digunakan oleh para ilmuwan data (*data scientist*) untuk membuat dan membagikan dokumen yang di dalamnya mengandung kode, hasil hitungan, visualisasi, dan teks. Google juga menyediakan aplikasi semacam itu, namanya *Google Colaboratory*.

Menjalankan Jupyter Notebook

Buka Command Prompt, lalu berikan perintah **jupyter notebook**.



Menutup Jupyter Notebook

Pada Command Prompt, ketik shortcut **ctrl+c**.

Silahkan tutup kembali jendela Command Prompt dan web browser.

Manfaat Jupyter Notebook bagi Ilmuwan

Jupyter Notebook merupakan standar terbaik dalam mengorganisir data karena kecepatannya. Manfaat utama Jupyter Notebook bagi seorang ilmuwan data (*data scientist*) adalah merekam penelitian yang dilakukan dalam bentuk dokumen, membagikannya dengan cepat, dan mengeksplorasi data.

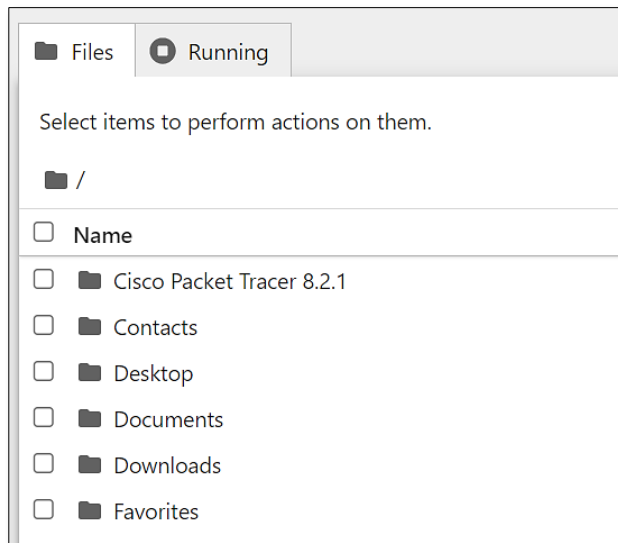
Penggunaan Jupyter Notebook dalam eksplorasi data memberikan narasi komputasi. Jadi, seorang *data scientist* bisa menambahkan analisis, hipotesis, dan keputusan pada dokumen tersebut.

Penggunaan Jupyter Notebook memberikan keuntungan, terutama untuk pekerjaan penelitian dan data scientist, dimana fokusnya adalah pada analisis dan manipulasi data, bukan produksi kode.

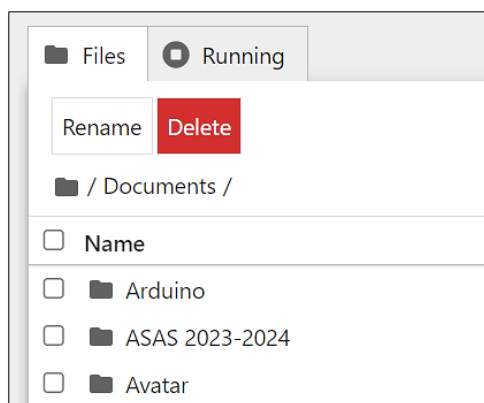
Penggunaan Jupyter Notebook

Root Directory

Root direktory adalah bagian paling luar dari sistem komputer. Tidak disarankan menyimpan file pada root direktory. Silahkan gunakan sub-direktory "Documents" untuk menyimpan file kerja.



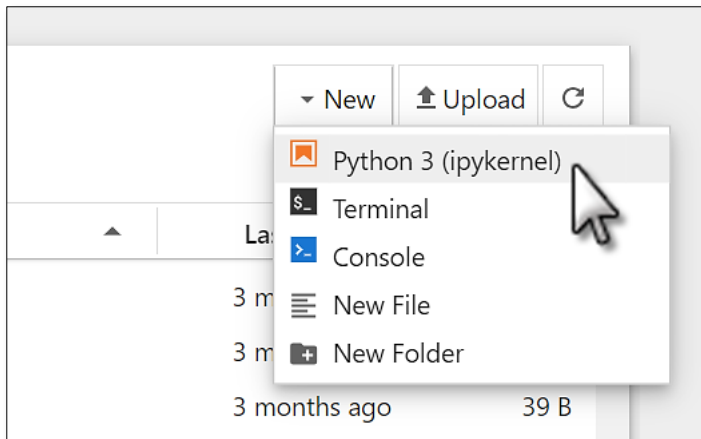
Untuk beralih ke sub-direktory "Documents" silahkan dobel-klik folder yang bersangkutan. Sekarang kita sudah berada di dalam folder "Documents".



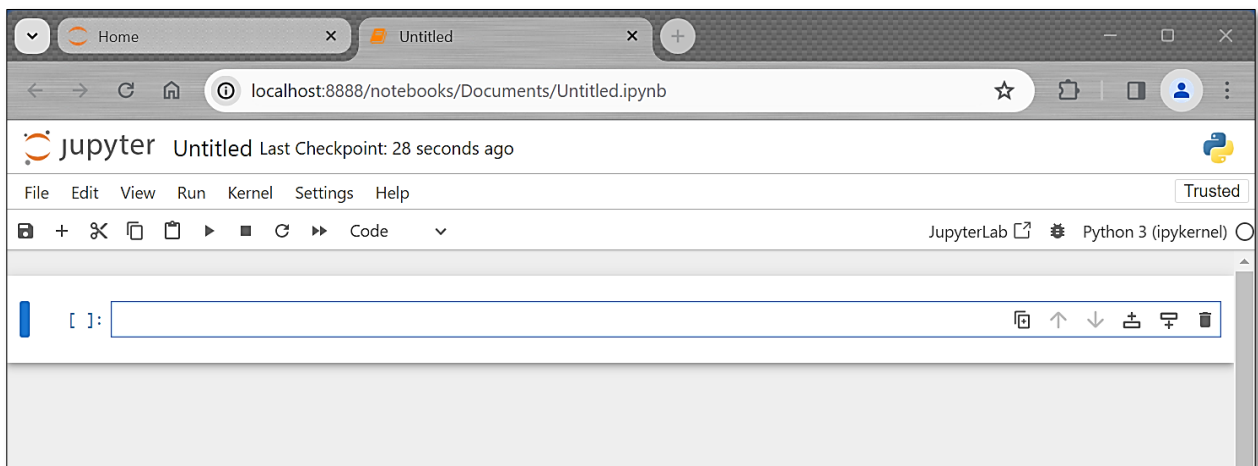
Manfaatkan folder tersebut untuk menyimpan file hasil kerja.

Membuat Nootebook Baru

Buka **New**, lalu pilih **Python 3 (ipkernel)**.

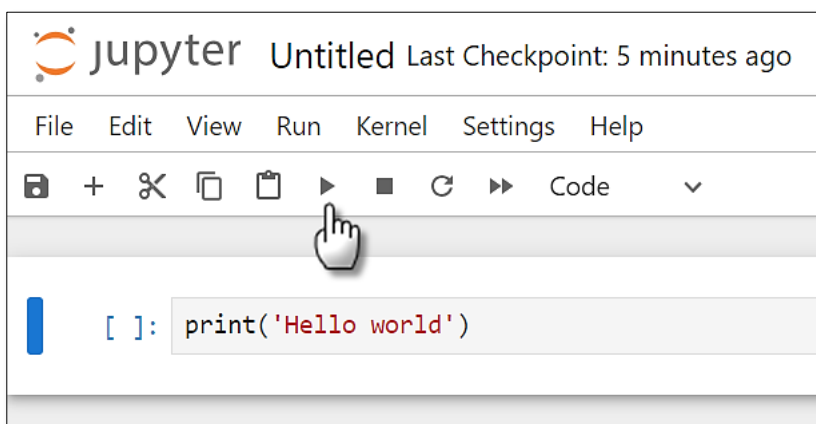


Notebook siap untuk digunakan.
Untuk sementara notebook baru diberi nama bawaan "Untitled.ipynb".
Tersedia sebuah sel code untuk menulis kode program di dalamnya.



Menulis dan Menjalankan Kode

Tulis baris kode `print('Hello world')` di dalam sel code.
Klik ikon **Run** untuk menjalankan kode.



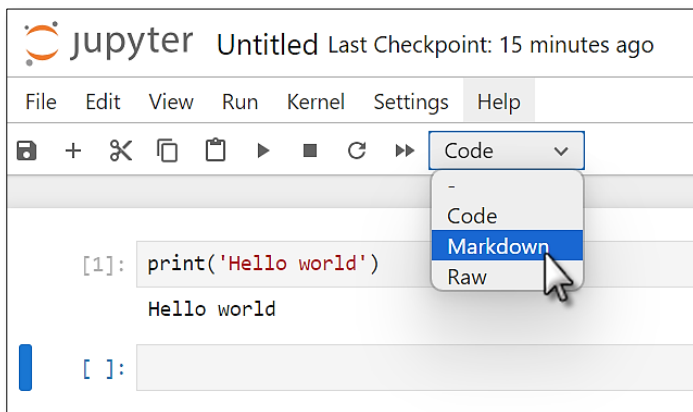
Hasil eksekusi kode diperlihatkan tepat di bawah sel code.

```
[1]: print('Hello world')
      Hello world
```

```
[ ]:
```

Menambahkan Sel Markdown

Pilih sebuah sel, lalu pilih **Markdown** dari daftar sel.



Sel markdown hanya untuk diisi dengan teks, bukan kode.

```
Ucapkan salam
```

```
[1]: print('Hello world')
      Hello world
```

```
[ ]:
```

Tentang Markdown

Markdown adalah bahasa markup ringan untuk menciptakan teks terformat menggunakan teks biasa (plain text).

Markdown	Output
# Heading 1	Heading 1
## Heading 2	Heading 2
### Heading 3	Heading 3
#### Heading 4	Heading 4
##### Heading 5	Heading 5
##### Heading 6	Heading 6
Miring	<i>Miring</i>

Markdown	Output
Tebal	Tebal
1. Item 1 2. Item 2	1. Item 1 2. Item 2
- Item 1 - Item 2	• Item 1 • Item 2

Teks "Ucapkan salam" pada contoh di atas adalah *plain text*. Untuk mengubah formatnya menjadi tajuk level pertama, cukup tambahkan simbol #.

```
# Ucapkan salam

[2]: print('Hello world')
      Hello world
```

Untuk menerapkan markdown, pilih sel berisi markdown, lalu klik ikon **Run**.

```
▼ Ucapkan salam

[2]: print('Hello world')
      Hello world
```

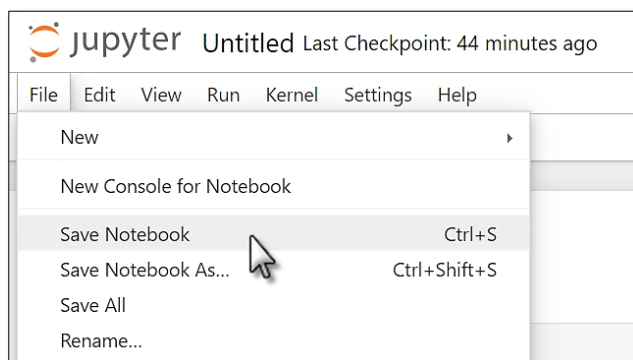
Pengatur Sel

Setiap sel dilengkapi dengan ikon-ikon untuk membuat duplikat sel, memindah posisi sel, menambahkan sel baru, dan menghapus sel.

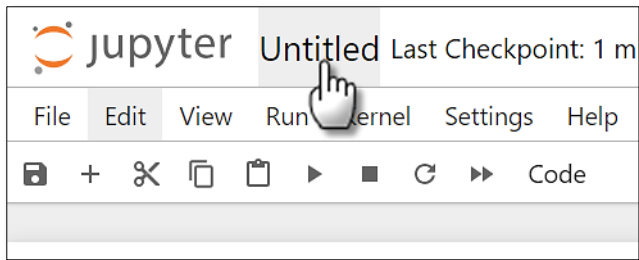


Menyimpan Notebook

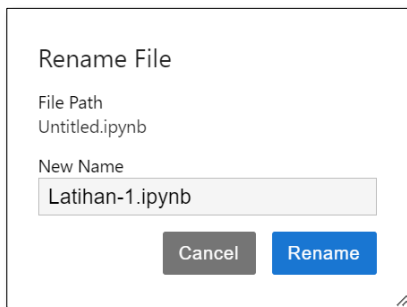
Gunakan **Save Notebook** dari menu **File**.



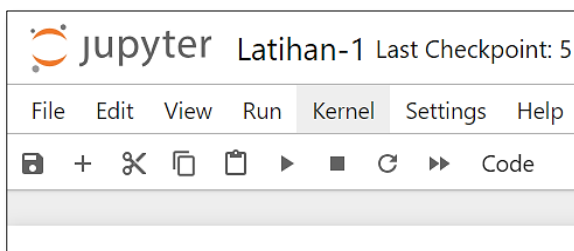
Setiap kali sebuah notebook baru diciptakan, nama "Untitled" akan diberikan. Untuk mengubah nama tersebut, klik pada nama file tersebut.



Berikan nama baru, lalu klik **Rename**.



Sekarang, notebook sudah berganti nama.



Contoh Tugas Proyek

Seorang guru ingin mengetahui korelasi antara lamanya siswa menghabiskan waktu belajar dan nilai tes yang mereka raih. Ia mengambil sampel 24 siswa. Data mentah disimpannya pada url: <https://raw.githubusercontent.com/patrahost/waone/master/grades.csv>
Bantu guru tersebut melakukan analisis data dengan memanfaatkan Jupyter Notebook.

Penyelesaian

Siapkan dulu library yang dibutuhkan, dalam hal ini adalah pandas karena kita akan melakukan pekerjaan menganalisis data dengan Python.

Silahkan mengimpor library tersebut ke dalam program yang akan dikerjakan. Gunakan `pd` sebagai nama lain untuk library tersebut. Hal ini mempersingkat penulisan nama library tersebut pada saat menggunakan fungsi/method yang ada di dalamnya.



Lakukan pembacaan terhadap file csv yang berisi data mentah.

```
[ ]: url = 'https://raw.githubusercontent.com/patrahost/waone/master/grades.csv'

## Baca data mentah

[ ]: baca = pd.read_csv(url)

## Cetak data mentah

[ ]: print(baca)
```

Terlihat bahwa data mentah terdiri dari kolom "NamaSiswa", "JamBelajar" dan "Nilai". Sekarang kita kelompokkan data untuk masing-masing kolom.

```
## Kelompokkan data per kolom

[ ]: siswa = baca['NamaSiswa']
    jam = baca['JamBelajar']
    nilai = baca['Nilai']
```

Ciptakan tabulasi data, dan cobalah untuk menayangkannya.

```
## Ciptakan tabulasi data

[ ]: df = pd.DataFrame({'Nama Siswa':siswa,'Jam Belajar':jam,'Nilai Tes':nilai})

## Cetak tabulasi data

print(df)
```

Periksa tabulasi data yang diperlihatkan guna menemukan nilai yang tidak lengkap (*NaN* atau *Not a Number*).

```
## Data cleansing

### Temukan nilai tak lengkap dalam data

[ ]: print(df.isna().sum())
```

Lakukan pembersihan data (*data cleansing*) dengan membuang seluruh data yang nilainya tidak lengkap.

```
### Buang data dengan nilai tak lengkap  
[ ]: df = df.dropna()
```

Lakukan manipulasi data sebagai berikut:

1. Tetapkan kriteria kelulusan, misalkan 60

```
## Manipulasi data  
  
### Kriteria kelulusan  
[71]: kriteria = df['Nilai Tes'] >= 60
```

2. Siapkan kolom baru untuk menampung nilai berdasarkan kriteria tersebut.

```
### Siapkan kolom baru  
[ ]: kolom = pd.Series(kriteria)
```

3. Sabungkan kolom baru ke tabulasi data.

```
### Sabungkan kolom baru ke tabulasi data  
[ ]: df = pd.concat([df, kolom.rename('Lulus')], axis=1)
```

4. Lihat tabulasi data dengan tambahan kolom baru.

```
### Lihat tabulasi data dengan tambahan kolom baru  
[ ]: print(df)
```

5. Hitung banyaknya data kelulusan.

```
### Hitung banyaknya data kelulusan  
[ ]: print(df.groupby(df['Lulus']).count())
```

Lakukan analisis korelasi


```
▼ ## Analisis korelasi

[ ]: kolom = ['Jam Belajar', 'Nilai Tes']
      korelaai = df[kolom].corr()
      print(korelasi)
```

Berapa nilai korelasi yang diperoleh, dan apa artinya itu?

Visualisasikan korelasi tersebut dengan grafik scatter.

```
## Vsualisasi dengan grafik scatter

### Impor library yang diperlukan

[77]: from matplotlib import pyplot as plt

### Data untuk grafik

[80]: sumbuX = df['Jam Belajar']
      sumbuY = df['Nilai Tes']

### Kelengkapan grafik

[ ]: plt.title('KORELASI JAM BELAJAR DAN NILAI TES')
      plt.xlabel('Lamanya Belajar')
      plt.ylabel('Nilai Tes')

### Cetak grafik

[ ]: plt.scatter(sumbuX, sumbuY)
```