Laporan Responsi

Apache Spark dengan Jupyter Notebook

Author :

Riski - 225410072

# Pengenalan

Apache Spark merupakan sebuah platfom yang akan memproses data yang terdistribusi yang di gunakan untuk mengolah big data dengan cepat. Spark juga mendukung berbagai macam bahasa pemrograman, termasuk Python, R, Java, dan Scala. Kombinasi dari Apache Spark dengan Jupyter Notebook mempermudah eksplorasi data secara interaktif. Spark bekerja berdasarkan konsep RDD (Resilient Distributed Dataset) dan menyediakan fitur seperti Spark SQL, Spark Streaming, MLlib, dan GraphX.

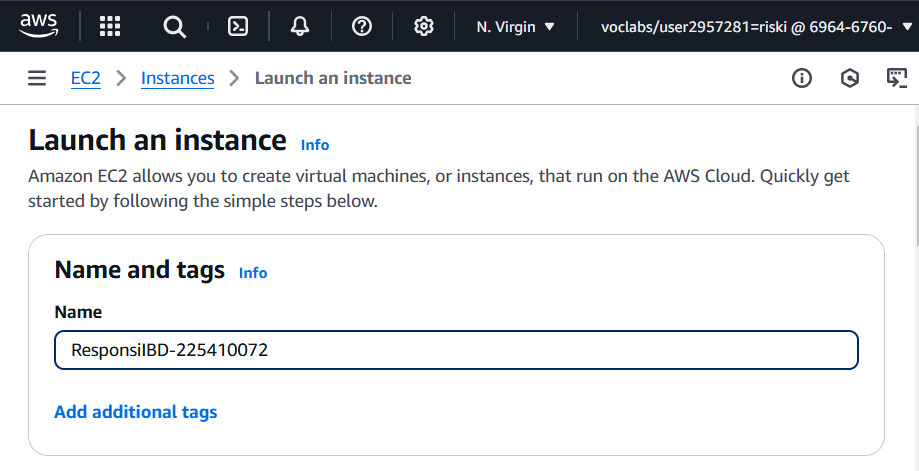
Cara kerja Spark:

* Data diproses dalam cluster yang terdistribusi.
* Spark membaca data dari berbagai sumber seperti HDFS, AWS S3, atau database lokal.
* Proses komputasi dilakukan di memori, membuatnya jauh lebih cepat dibandingkan MapReduce.

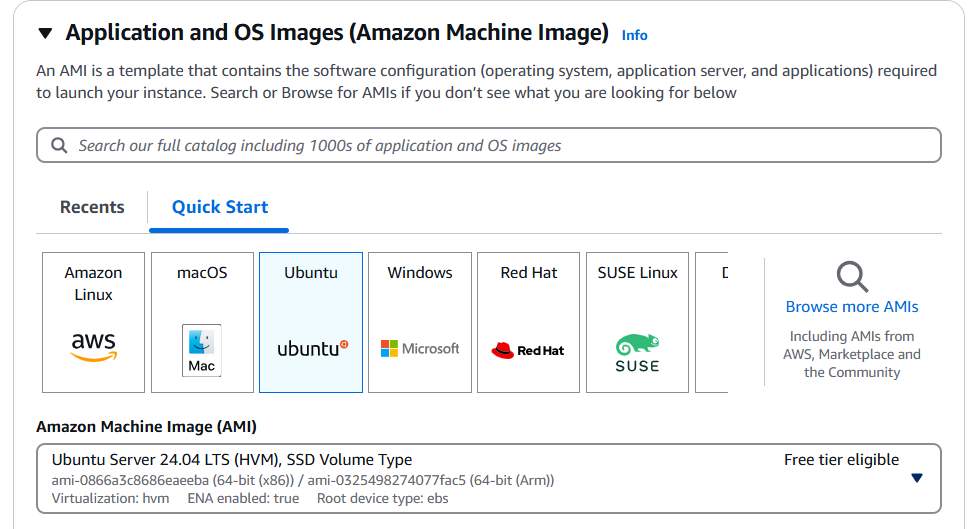
# Persiapan dan Instalasi

Siapkan Mesin Virtual

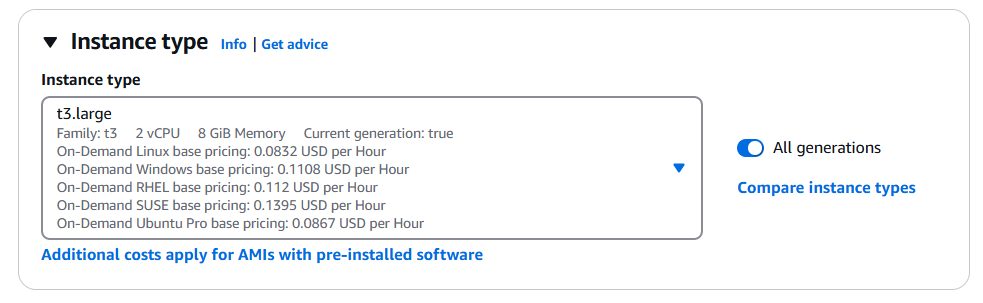
1. Langkah yang pertama ialah membuat nama sesuai ketentuan yang sudah di berikan



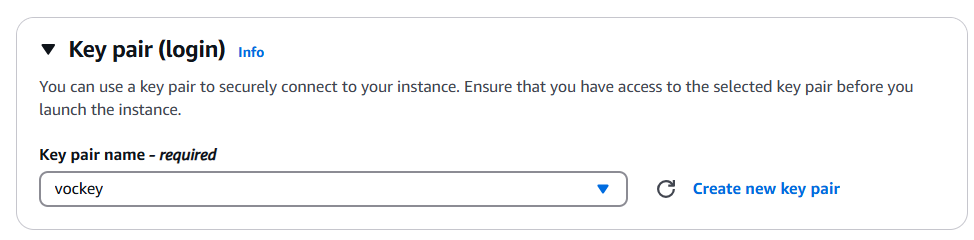
1. Langkah ke 2 membuat OS yang akan di gunakan :



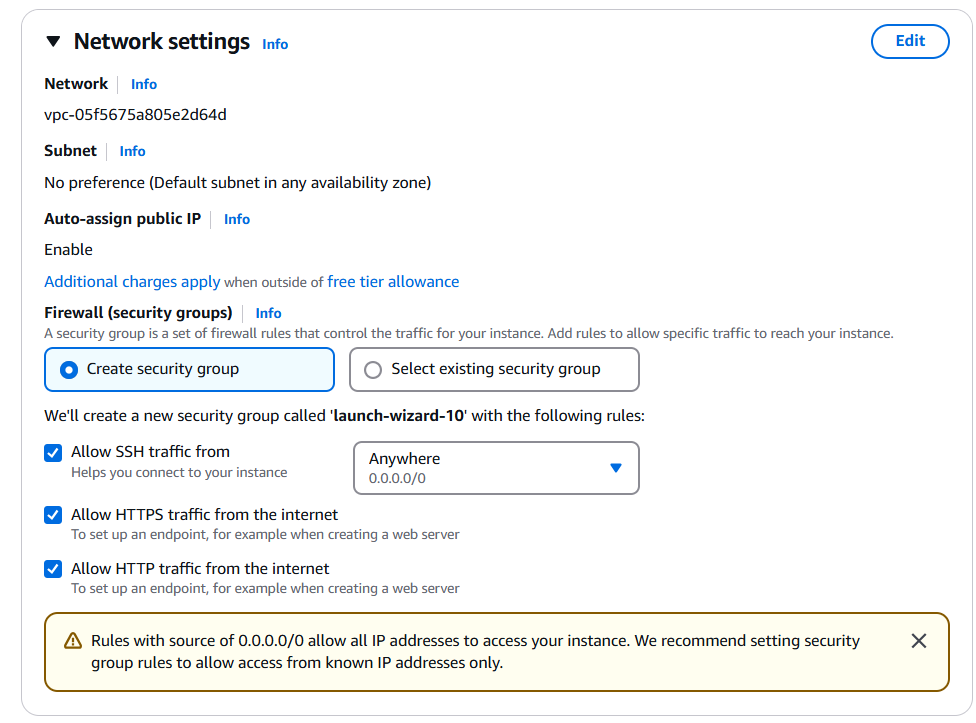
1. Langkah ke 3 ialah memilih instance type yang akan di gunakan :



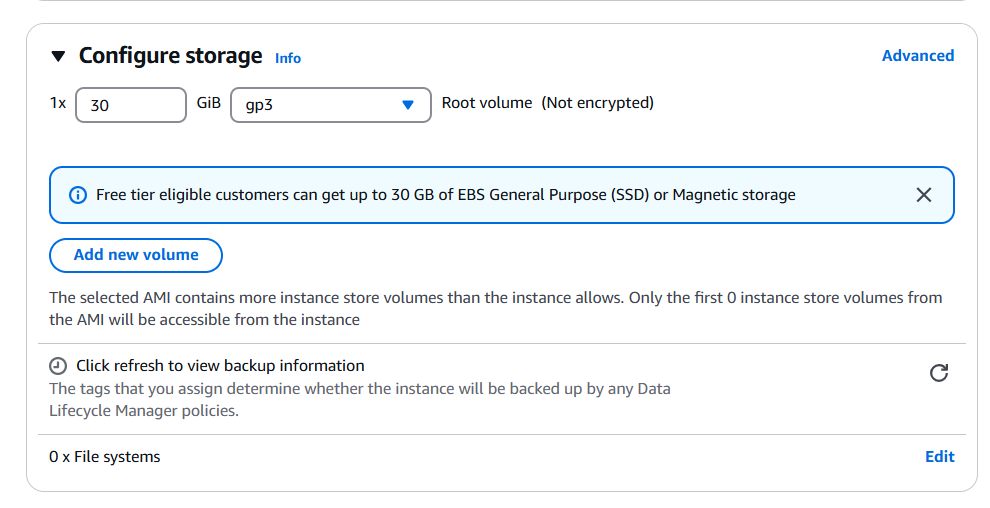
1. Langkah ke 4 ialah mengatur vockey :



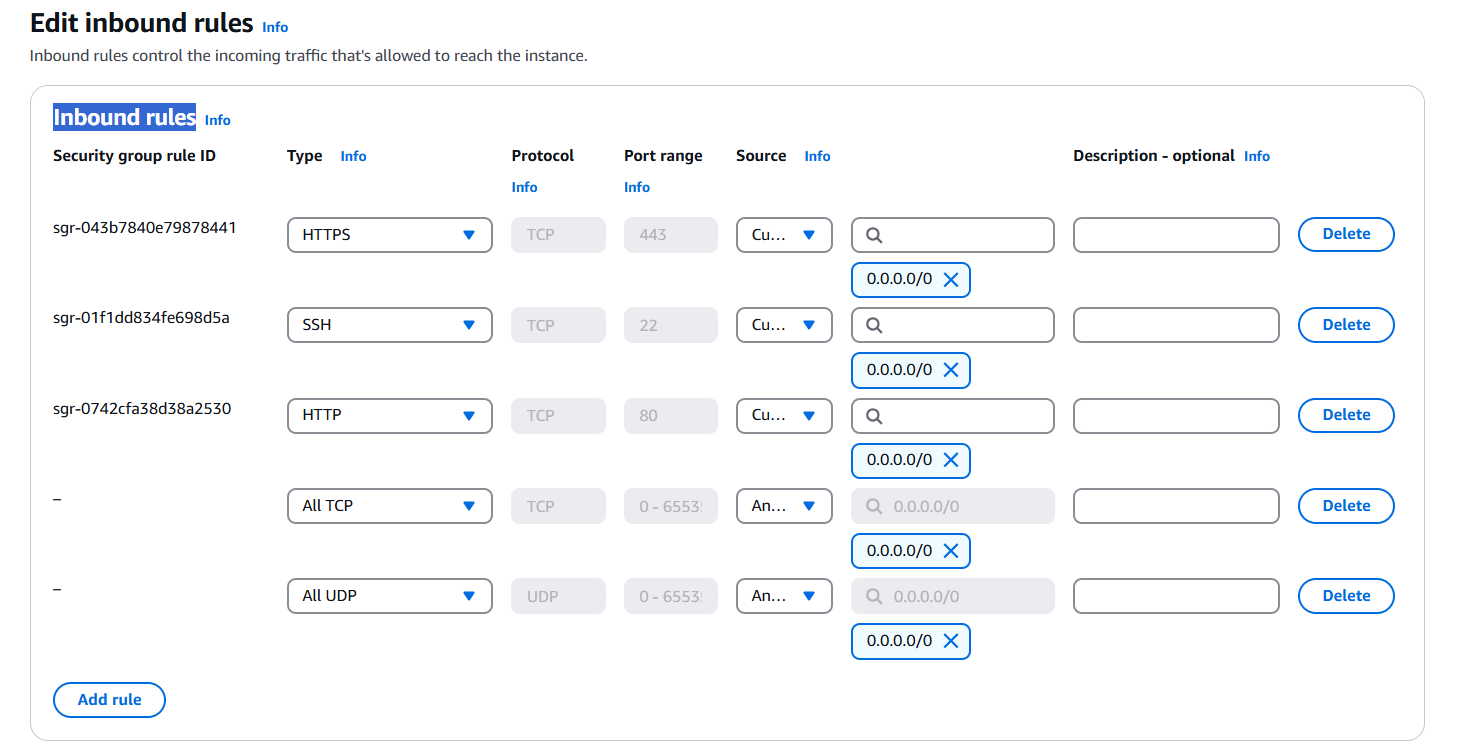
1. Langkah ke 5 ialah mengatur network sesuai kebutuhan :



1. Langkah ke 6 ialah mengatur storage sesuai kebutuhan :



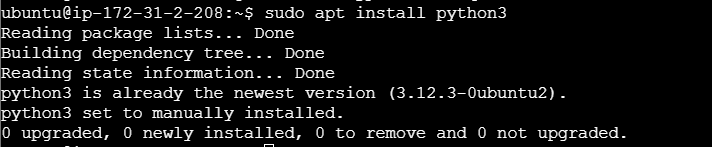
1. Selanjutnya aktifkan security group open all untuk TCP dan UDP



Instalasi Jupyter Notebook

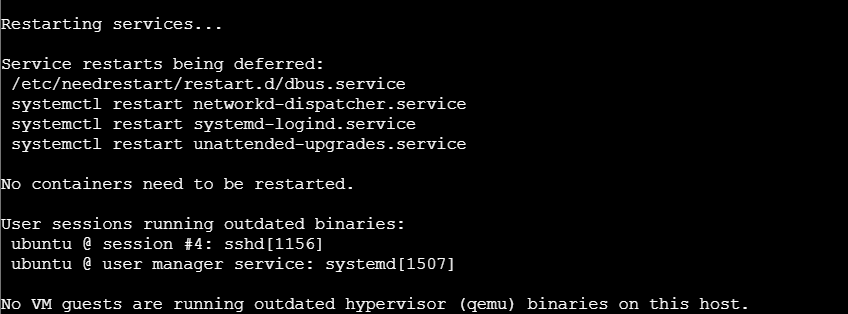
1. Sebelum menginstalasi Jupyter notebook user harus menginstall python3-pip dengan perintah:

**sudo apt install python3-pip**

****

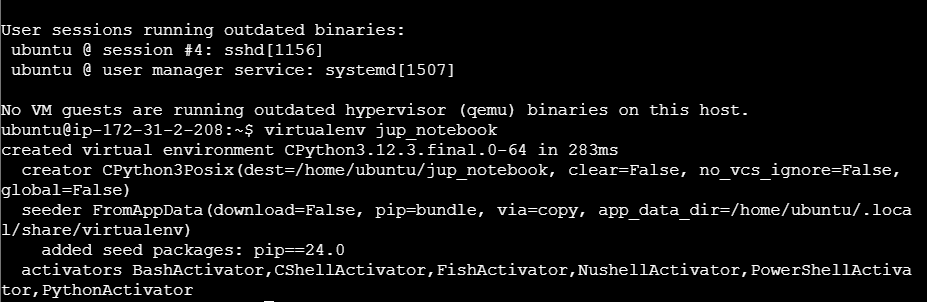
1. Selanjutnya Install **Virtual environment** untuk Python 3

**sudo apt install python3-virtualenv**

****

1. Selanjutnya Buat **virtual environment** baru dengan nama jup\_notebook

**virtualenv jup\_notebook**

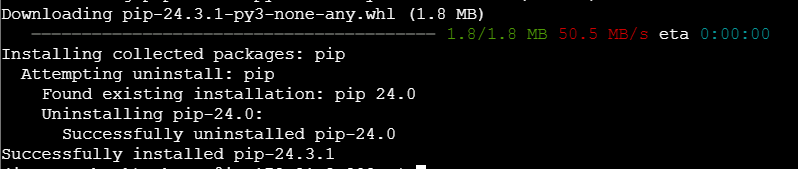
****

1. Selanjutnya Aktifkan virtual environment

**source jup\_notebook/bin/activate**

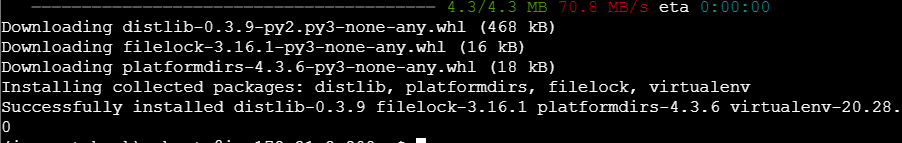
1. Perbarui pip (python package manager) ke versi terbaru

**pip3 install –upgrade pip**

****

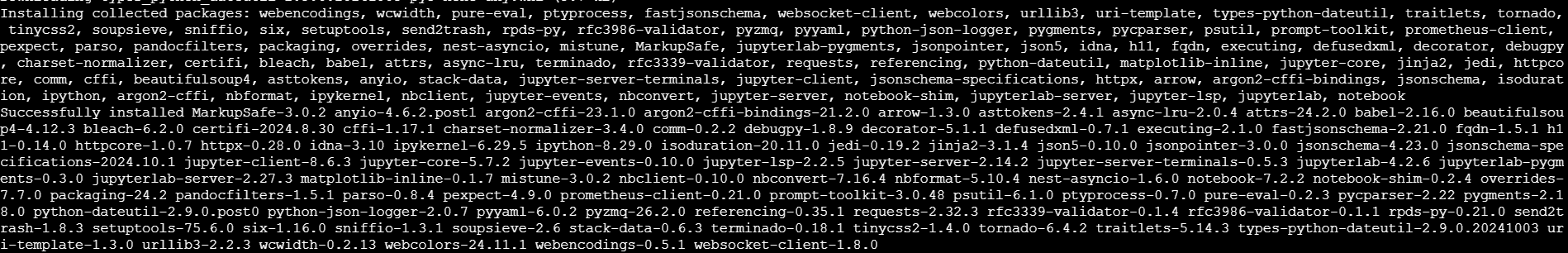
1. Install Virtualenv menggunakan pip3

**pip3 install virtualenv**

****

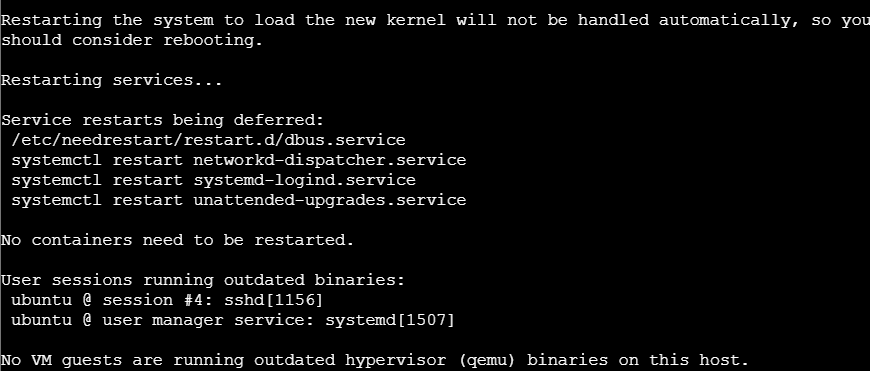
1. instal Jupyter Notebook menggunakan pip3

**pip3 install notebook**

****

1. Instal Jupyter Core dan Jupyter Notebook melalui paket sistem berbasis Ubuntu.

**sudo apt install jupyter-core jupyter-notebook**

****

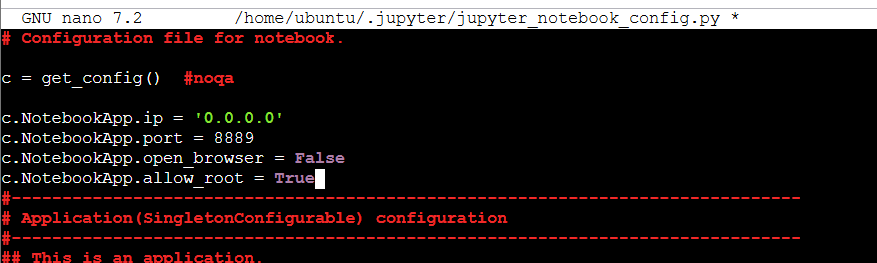
1. Selanjutnya Setelah selesai instalasi, kita perlu melakukan beberapa tahapan konfigurasi. Untuk membuat file konfigurasi dari jupyter notebook, gunakan perintah:

**jupyter-notebook --generate-config**

****

1. Buka file konfigurasi Jupyter Notebook menggunakan text editor nano

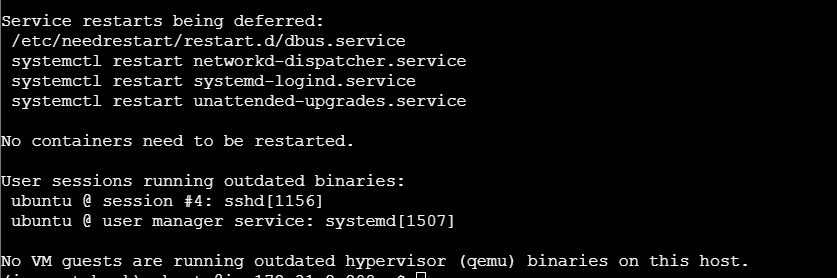
**nano /home/ubuntu/.jupyter/jupyter\_notebook\_config.py**

****

## Instalasi Apache Spark

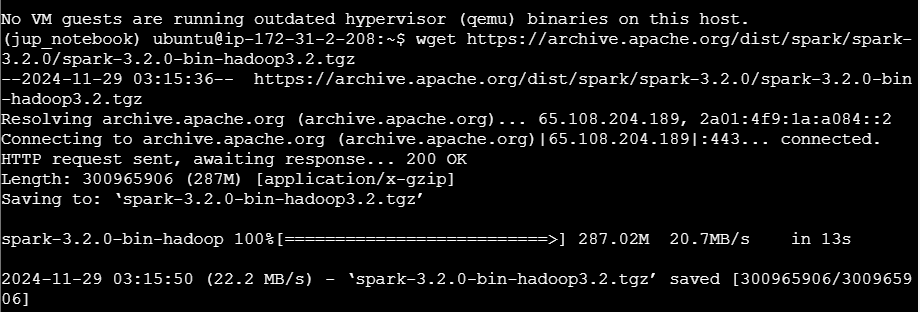
1. Menginstal JDK versi default, Menginstal bahasa pemrograman Scala da Menginstal sistem kontrol versi Git.

**sudo apt install default-jdk scala git -y**



1. Mengunduh file Apache Spark versi 3.2.0 yang sudah terkompresi dalam format .tgz dari situs arsip Apache.

**wget https://archive.apache.org/dist/spark/spark-3.2.0/spark-3.2.0-bin-hadoop3.2.tgz**

****

1. Mengekstrak file arsip .tgz

**tar xf spark-\***

1. Selanjutnya Memindahkan direktori Apache Spark dari lokasi unduhan ke lokasi /opt/spark.

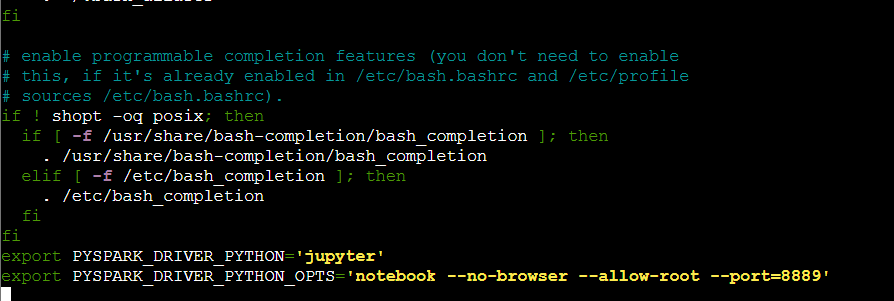
**sudo mv spark-3.2.0-bin-hadoop3.2 /opt/spark**

1. Selanjutnya Buka dan mengedit file .bashrc menggunakan editor teks nano

**nano .bashrc**

1. Tambahkan konfigurasi berikut :

**export SPARK\_HOME=/opt/spark  
export PATH=$PATH:$SPARK\_HOME/bin:$SPARK\_HOME/sbin  
export PYSPARK\_PYTHON=/usr/bin/python3**

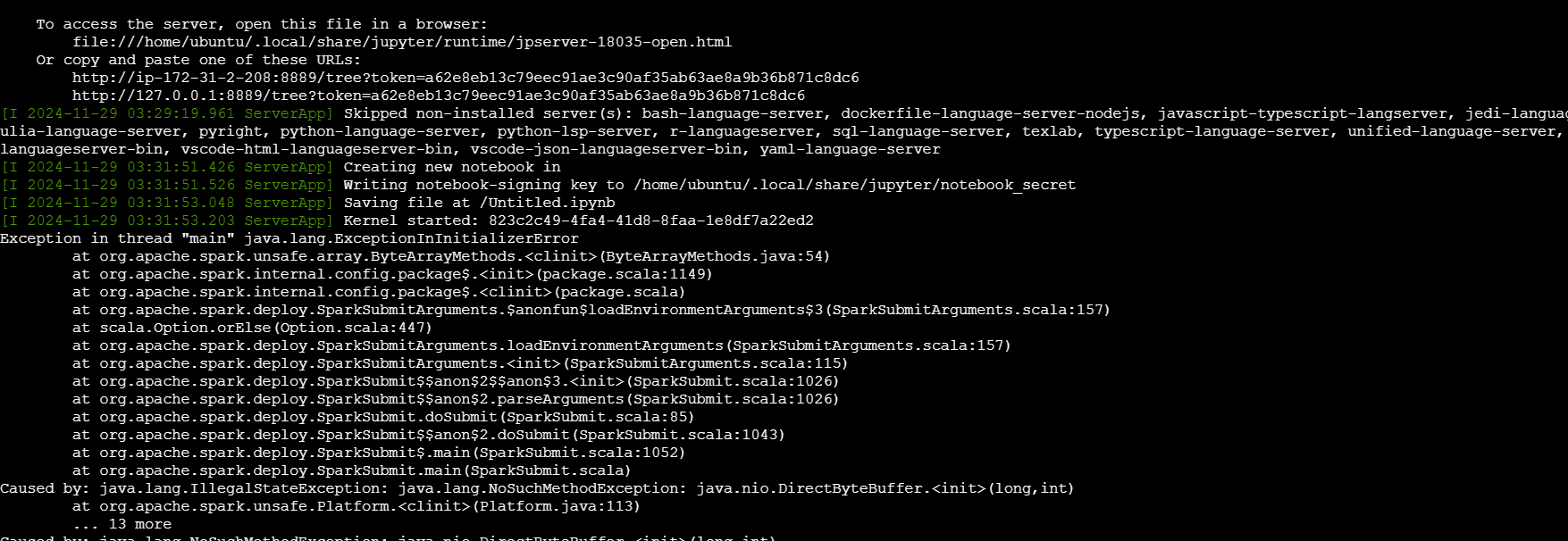


1. Terapkan perubahan pada file .bashrc

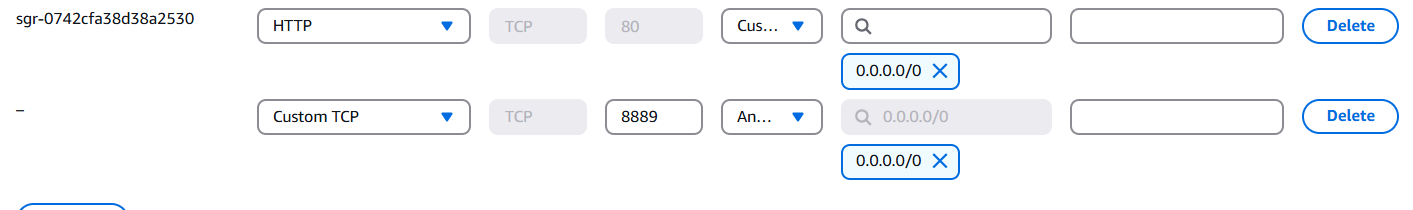
**source .bashrc**

1. Jalankan jupyter sekaligus menjalankan spark:

**/opt/spark/bin/pyspark**

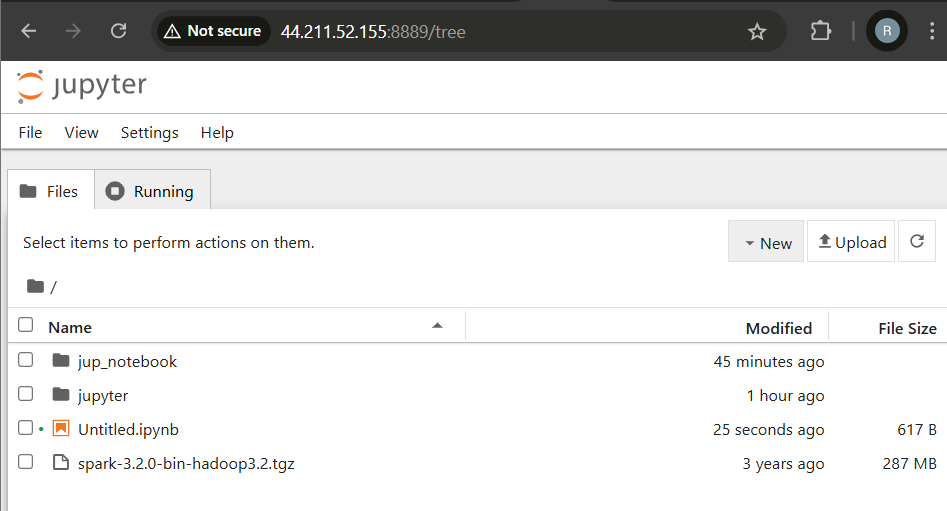


1. Menambahkan port 8889

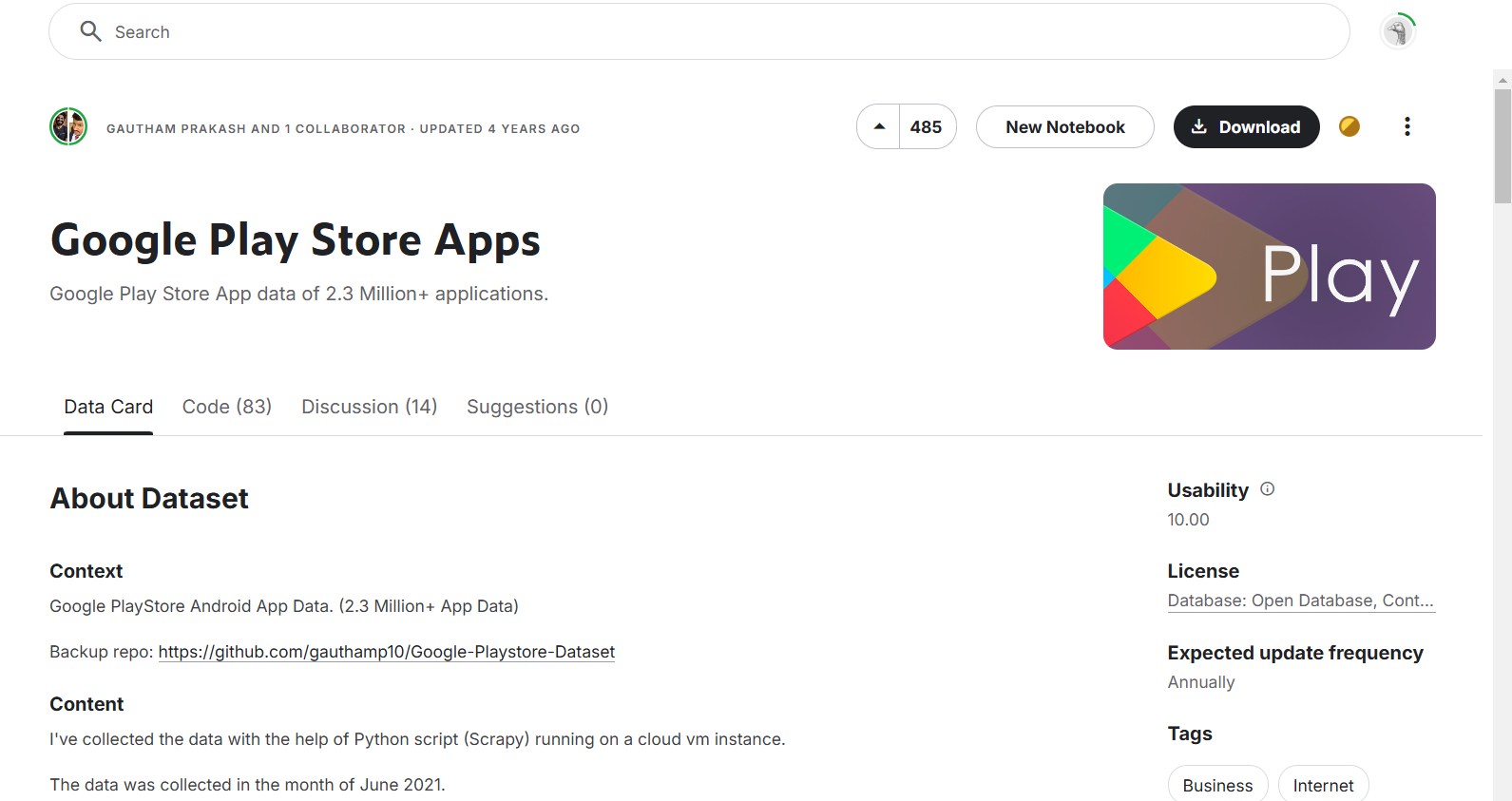


# Pengujian

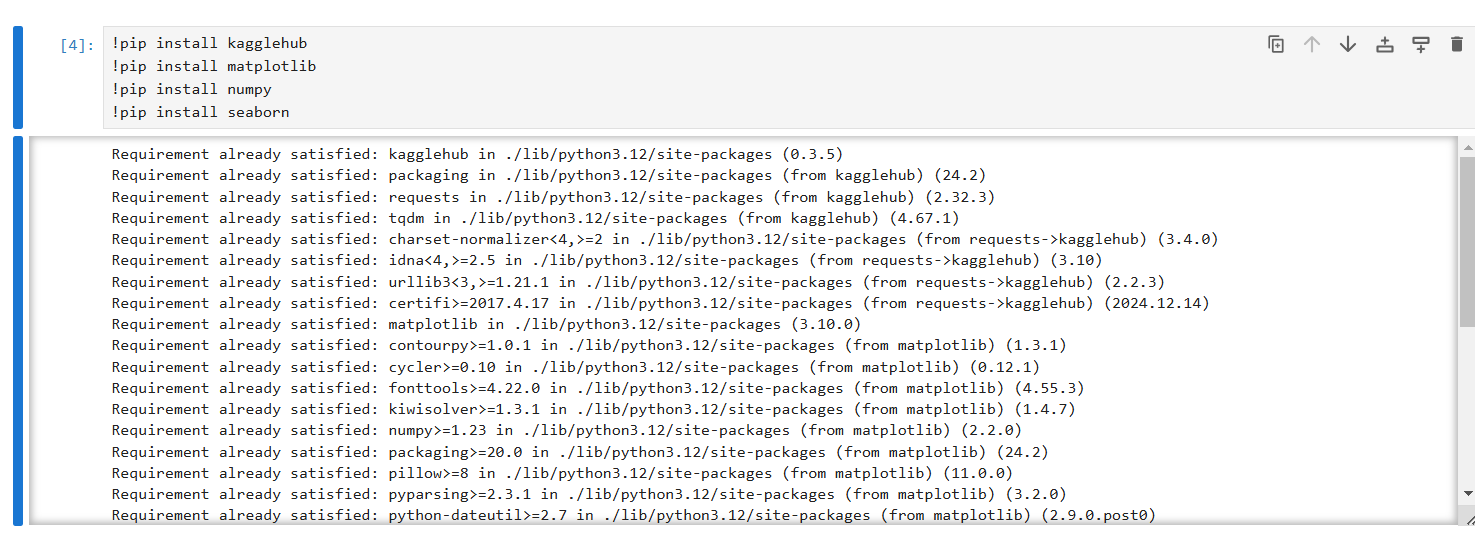
1. Akses Jupyter dengan menggunakan url yang ada pada terminal setelah itu membuat python 3 dengan menglik New pada pojok kanan maka tampilan akanseperti di bawah.



1. Langkah selanjutnya ialah copy code yang ada pada data yang sudah di pilih di kaggle



1. Instalasi Library yang di perlukan di dalam jupyter notebook disini saya menginstal library kagglehub, matplotlib, numpy, dan seaborn



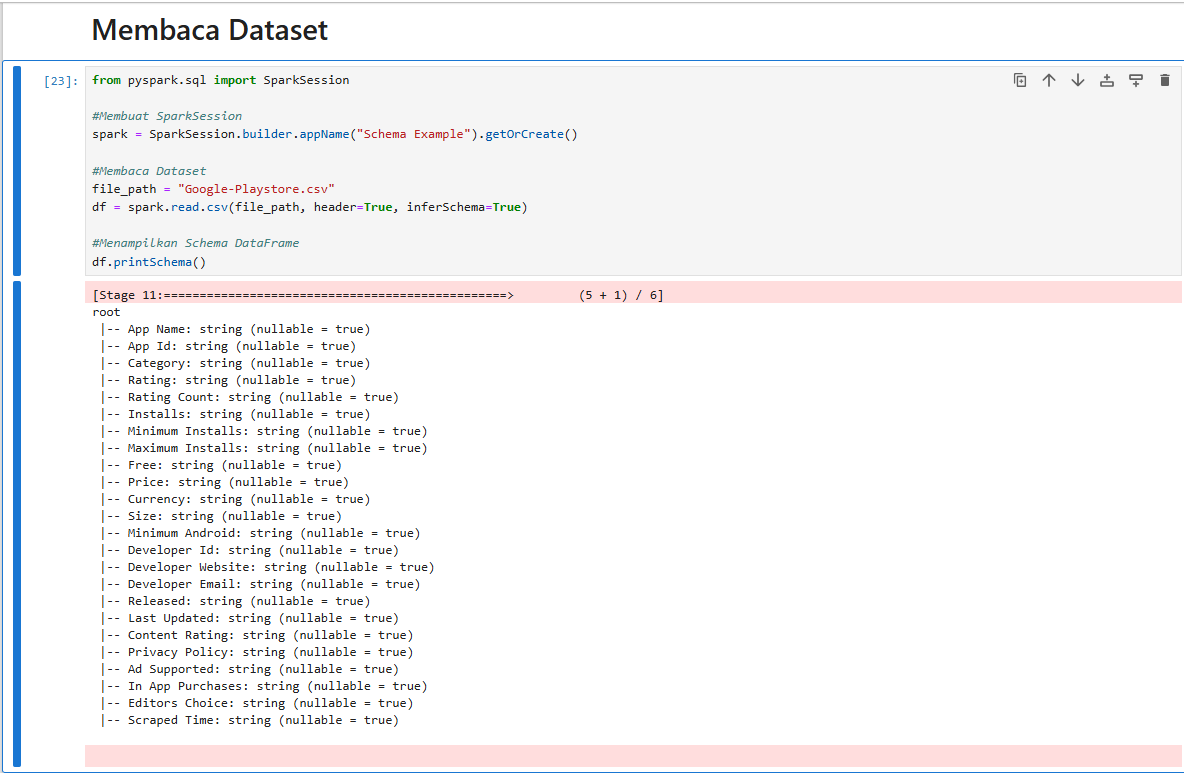
1. Disini saya download dataset yang berada di kaggle



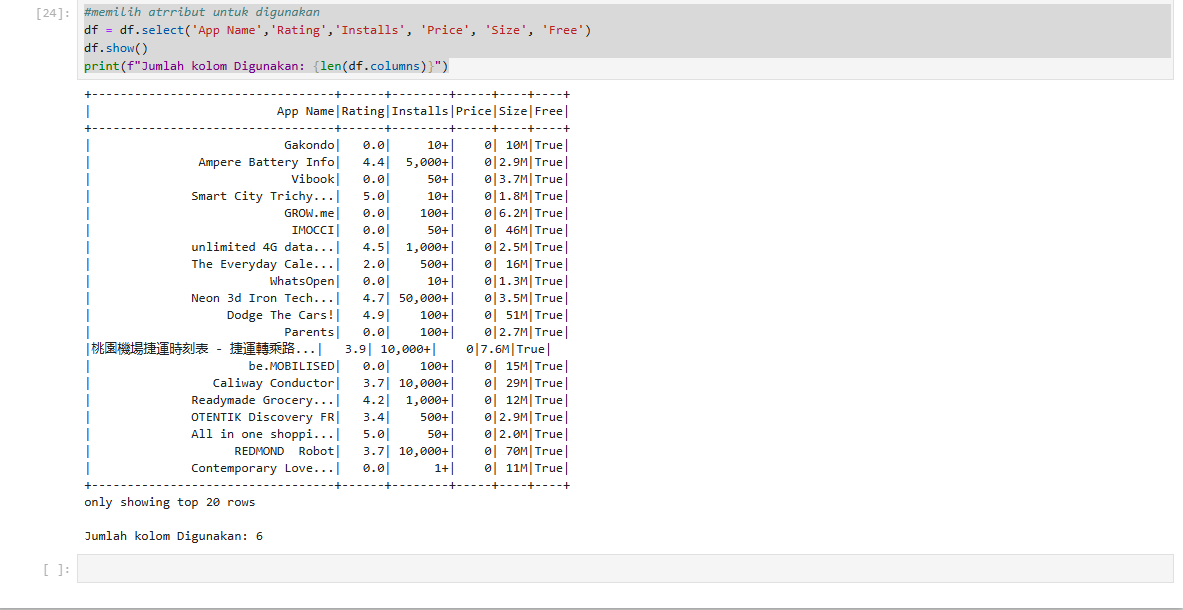
1. Memindahkan dataset ke direktori pekerjaan yang akan saya kerjakan



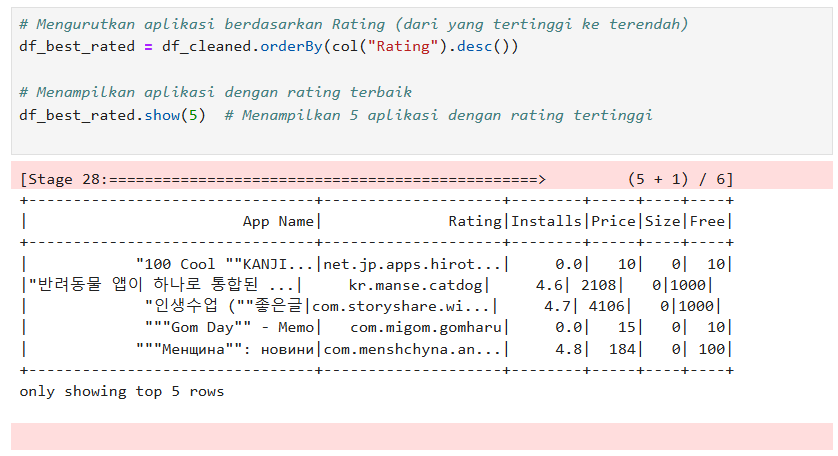
1. Disini saya membaca dataset yang telah saya pilih dari kaggle dan data mana yang saya ingin tampilkan



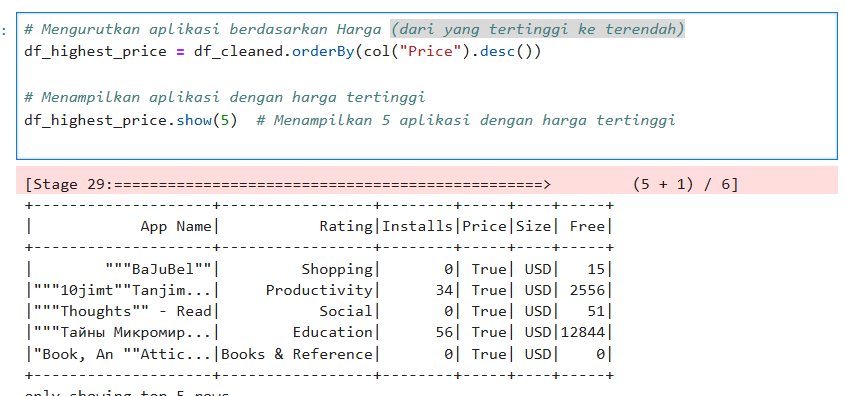
1. Disni saya memunculkan data yang saya ingin tampilkan, data yang saya tampilkan



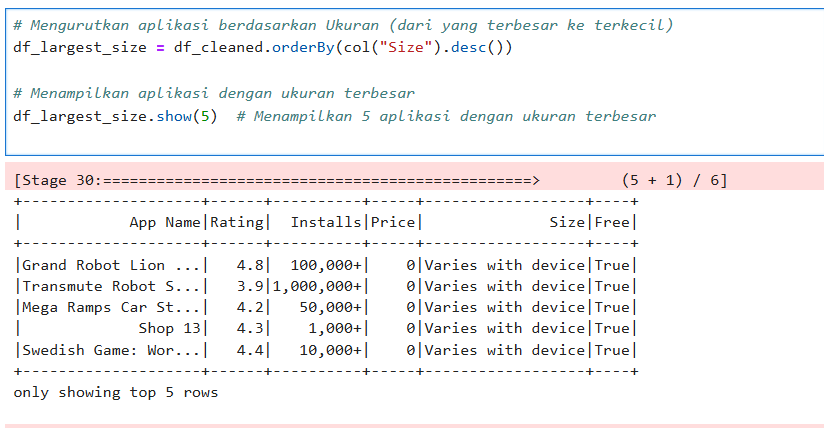
1. Lalu selanjutnya disini saya mengurutkan aplikasi berdasarkan ratting dan yang muncul seperti berikut



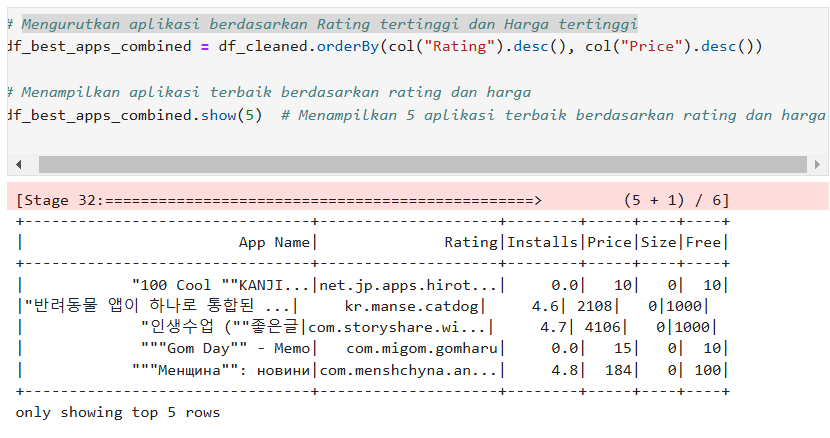
1. Lalu selanjutkan Saya mengurutkan aplikasi berdasarhakan harga (dari yang tertinggi ke terendah)



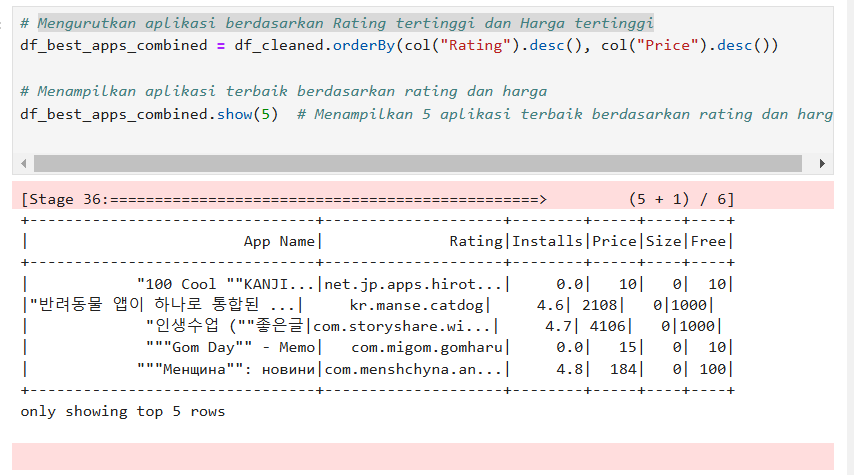
1. Lalu disini saya Mengurutkan aplikasi berdasarkan Ukuran (dari yang terbesar ke terkecil)



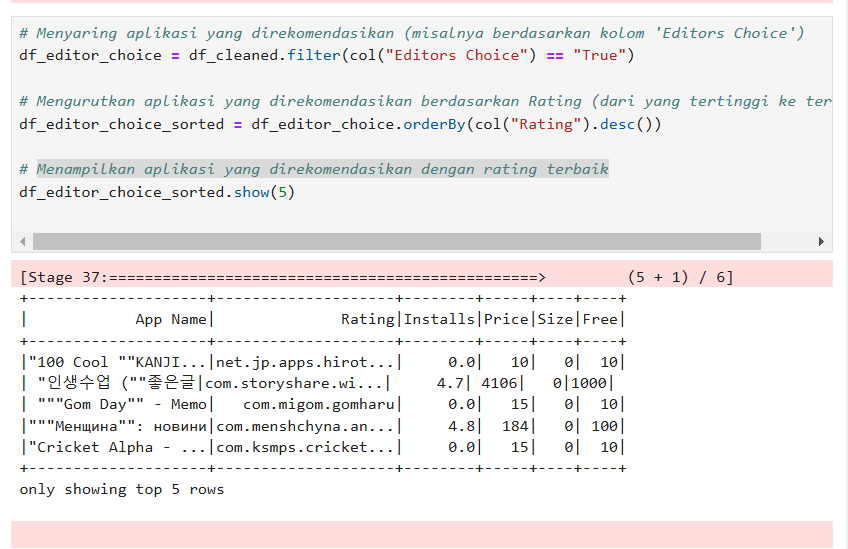
1. Lalu disini saya Mengurutkan aplikasi berdasarkan Rating tertinggi dan Harga tertinggi



1. Lalu disini saya Mengurutkan aplikasi berdasarkan Rating tertinggi dan Harga tertinggi



1. Kemudian disini saya Menampilkan aplikasi yang direkomendasikan dengan rating terbaik, Berikut adalah hasil akhirnya :



# Penutup

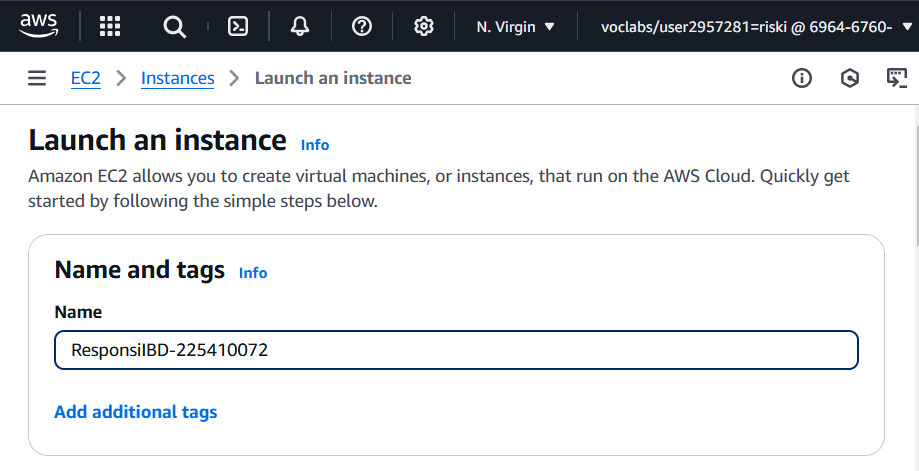
Penggunaan Apache Spark bersama dengan Jupyter Notebook dalam mengelola data aplikasi Google Play Store yang diambil dari Kaggle telah terbukti efektif dan efisien. Apache Spark menyediakan kemampuan komputasi terdistribusi yang memungkinkan pemrosesan data dalam skala besar dengan kecepatan tinggi, sementara Jupyter Notebook menawarkan lingkungan interaktif yang memudahkan analisis dan visualisasi data secara real-time. Dengan memanfaatkan kombinasi ini, analisis terhadap data Google Play Store apps dapat dilakukan secara mendalam, mulai dari pembersihan data, transformasi, hingga penerapan algoritma machine learning untuk mendapatkan wawasan yang berharga. Selain itu, integrasi ini memungkinkan kolaborasi yang lebih baik antar tim data scientist dan pengembang, serta mempercepat proses pengambilan keputusan berbasis data. Secara keseluruhan, integrasi Apache Spark dan Jupyter Notebook merupakan solusi yang kuat untuk mengelola dan menganalisis dataset besar seperti Google Play Store apps, mendukung pengembangan aplikasi yang lebih baik dan strategi bisnis yang lebih tepat sasaran.

# Log Activity

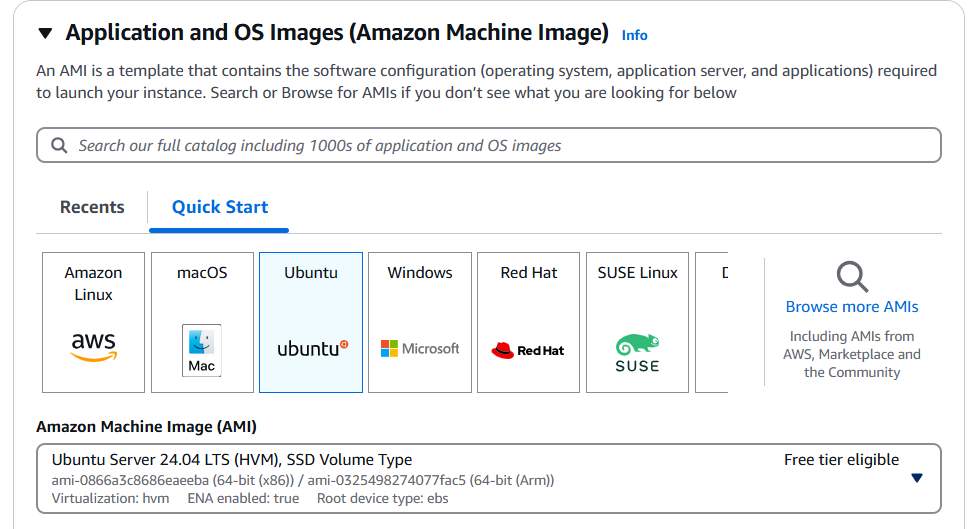
Pertemuan 9

Pada Pertemuan ini saya membuat mesin dengan nama sesuai yang sudah di tentukan oleh dosen, dan saya ingin mengerjakan topik yang telah saya ambil yaitu Apache Spark dengan Jupyter Noteebok. langkah pertama yang saya lakukan ialah menginstalasi jupyter notebook menggunakan modul yang ada di e-learning.

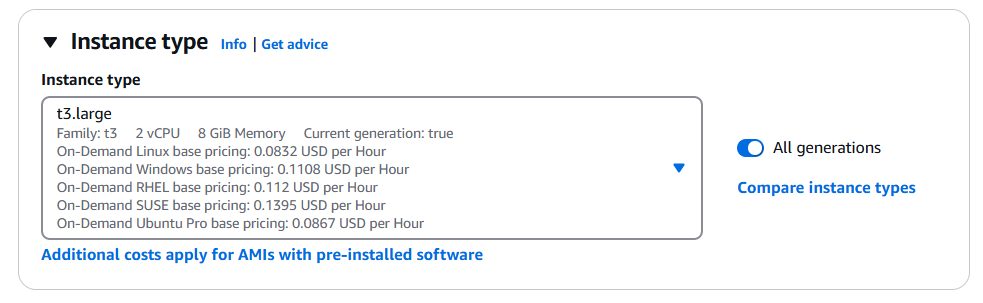
1. Langkah yang pertama ialah membuat nama sesuai ketentuan yang sudah di berikan



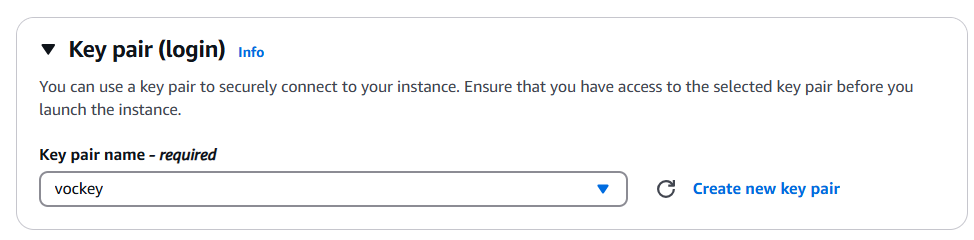
1. Langkah ke 2 membuat OS yang akan di gunakan :



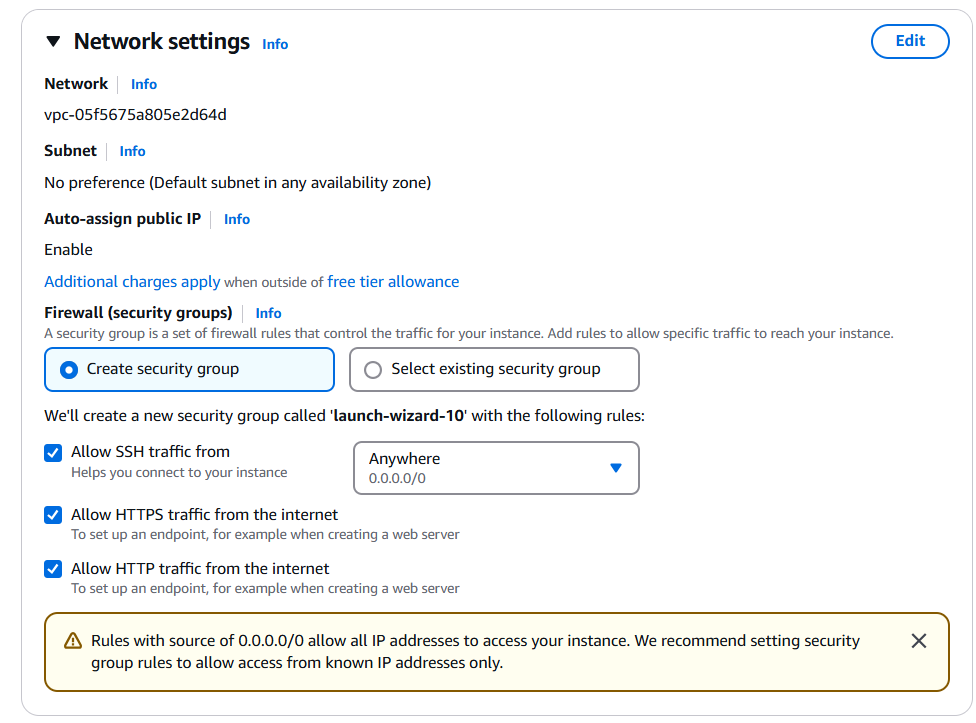
1. Langkah ke 3 ialah memilih instance type yang akan di gunakan :



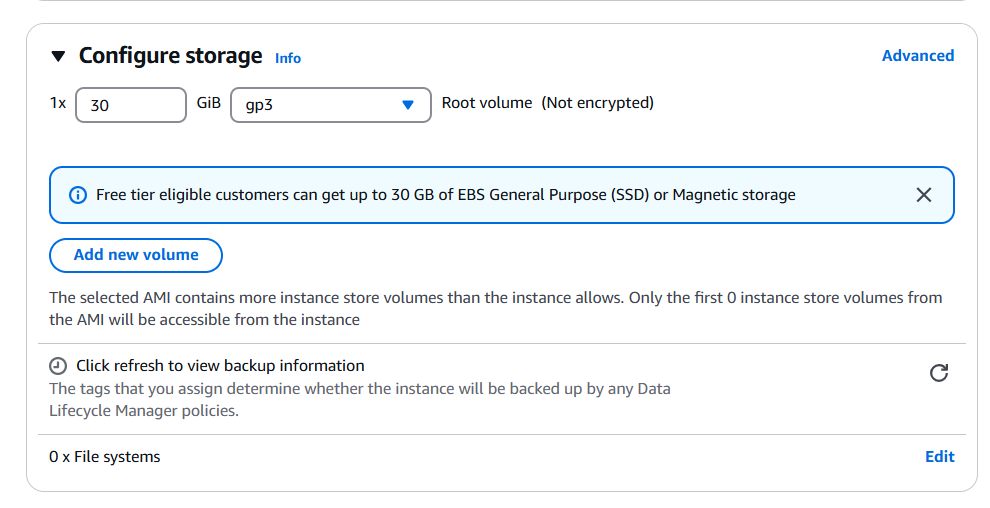
1. Langkah ke 4 ialah mengatur vockey :



1. Langkah ke 5 ialah mengatur network sesuai kebutuhan :



1. Langkah ke 6 ialah mengatur storage sesuai kebutuhan :

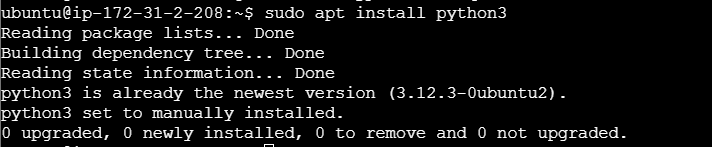


Pertemuan 10

Disini saya menginstalasi jupyter notebook dengan menggunakan refrensi dari modul dan refrensi yang berada di google. Jupyter notebook penting untuk di instalasi, di karenakan jika tidak menginstalasi maka tidak akan bisa di buka kecuali sudah di instalasi pada virtual machine.

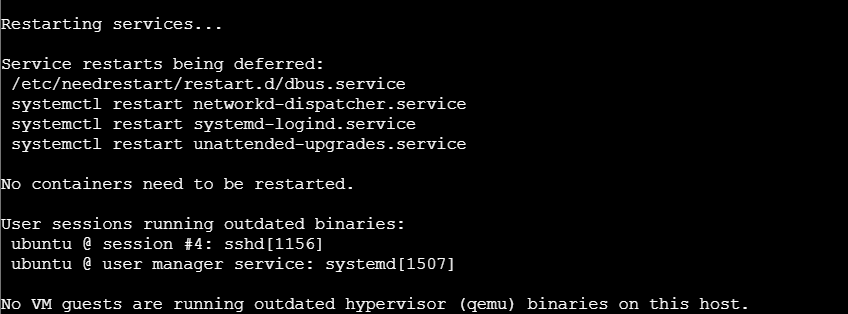
1. Sebelum menginstalasi Jupyter notebook user harus menginstall python3-pip dengan perintah:

**sudo apt install python3-pip**

****

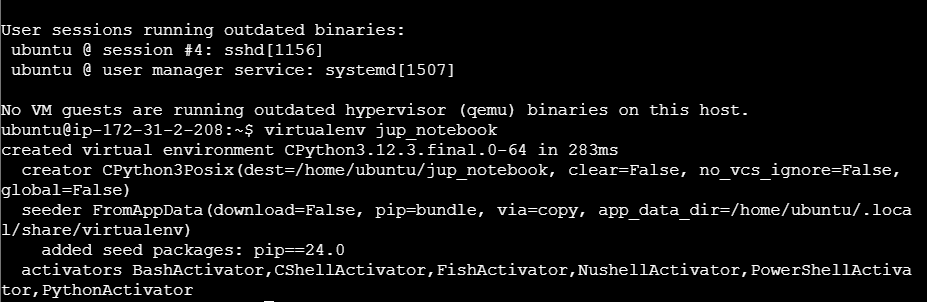
1. Selanjutnya Install **Virtual environment** untuk Python 3

**sudo apt install python3-virtualenv**

****

1. Selanjutnya Buat **virtual environment** baru dengan nama jup\_notebook

**virtualenv jup\_notebook**

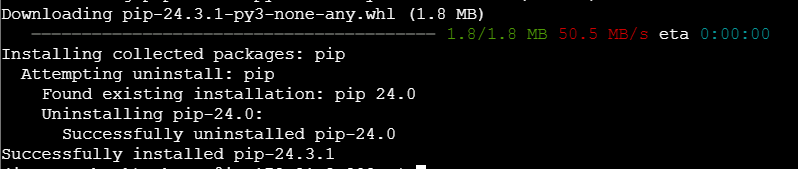
****

1. Selanjutnya Aktifkan virtual environment

**source jup\_notebook/bin/activate**

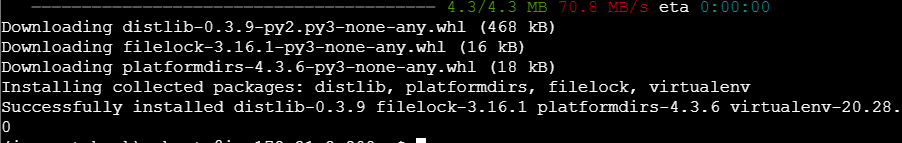
1. Perbarui pip (python package manager) ke versi terbaru

**pip3 install –upgrade pip**

****

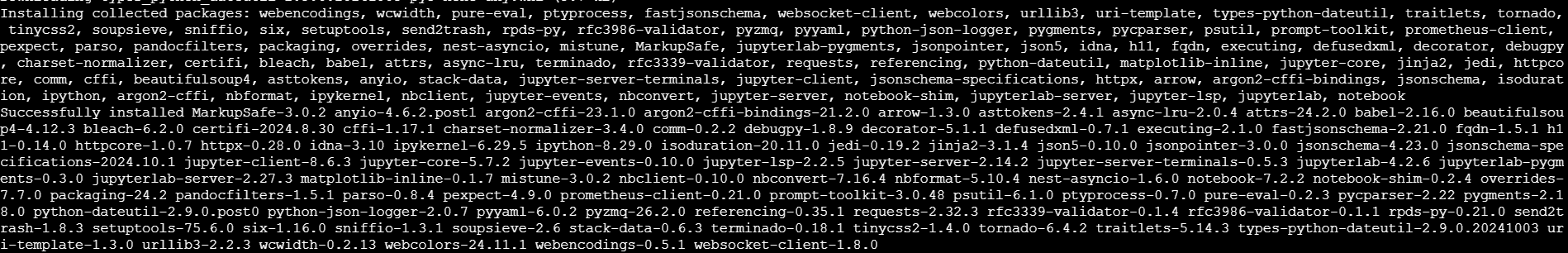
1. Install Virtualenv menggunakan pip3

**pip3 install virtualenv**

****

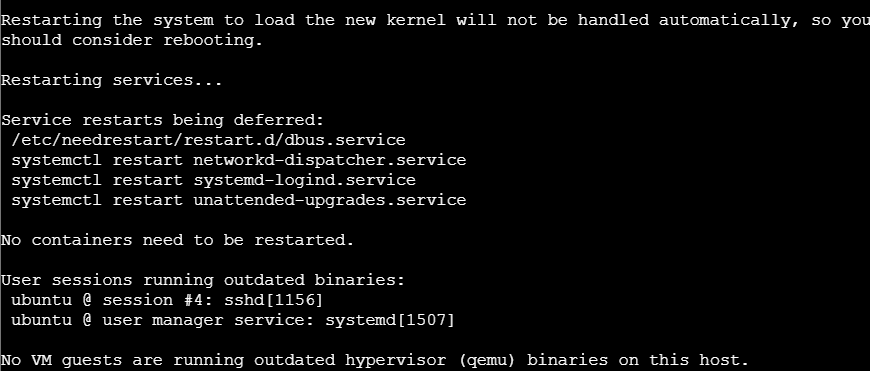
1. instal Jupyter Notebook menggunakan pip3

**pip3 install notebook**

****

1. Instal Jupyter Core dan Jupyter Notebook melalui paket sistem berbasis Ubuntu.

**sudo apt install jupyter-core jupyter-notebook**

****

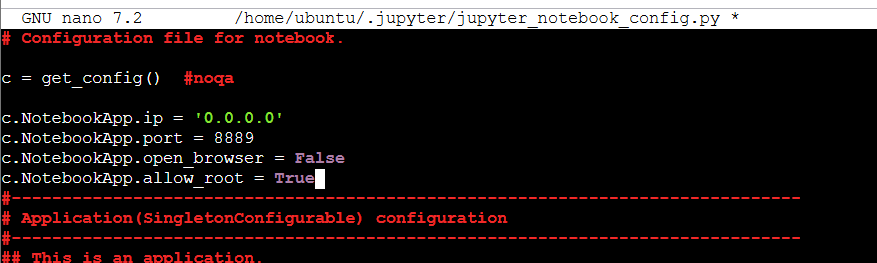
1. Selanjutnya Setelah selesai instalasi, kita perlu melakukan beberapa tahapan konfigurasi. Untuk membuat file konfigurasi dari jupyter notebook, gunakan perintah:

**jupyter-notebook --generate-config**

****

1. Buka file konfigurasi Jupyter Notebook menggunakan text editor nano

**nano /home/ubuntu/.jupyter/jupyter\_notebook\_config.py**

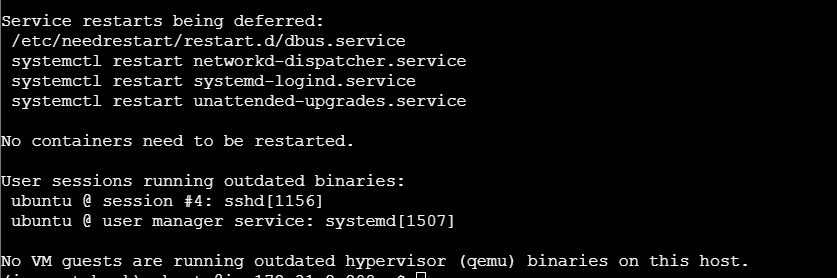
****

## Pertemuan 11

Pada bagian pertemuan 11, disini saya melakukan instalasi pada Apache Spark dan disini saya melakukannya menggunakan referensi dari modul dan pada google. Dan berikut adalah command yang saya gunakan untuk menginstalasi pada machine virtual saya :

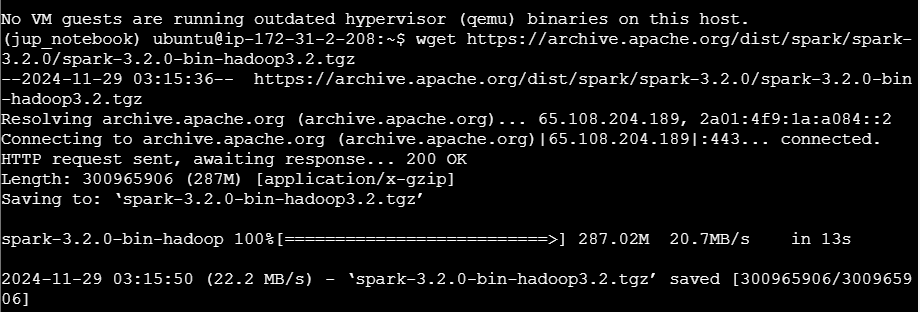
1. Menginstal JDK versi default, Menginstal bahasa pemrograman Scala da Menginstal sistem kontrol versi Git.

**sudo apt install default-jdk scala git -y**



1. Mengunduh file Apache Spark versi 3.2.0 yang sudah terkompresi dalam format .tgz dari situs arsip Apache.

**wget https://archive.apache.org/dist/spark/spark-3.2.0/spark-3.2.0-bin-hadoop3.2.tgz**

****

1. Mengekstrak file arsip .tgz

**tar xf spark-\***

1. Selanjutnya Memindahkan direktori Apache Spark dari lokasi unduhan ke lokasi /opt/spark.

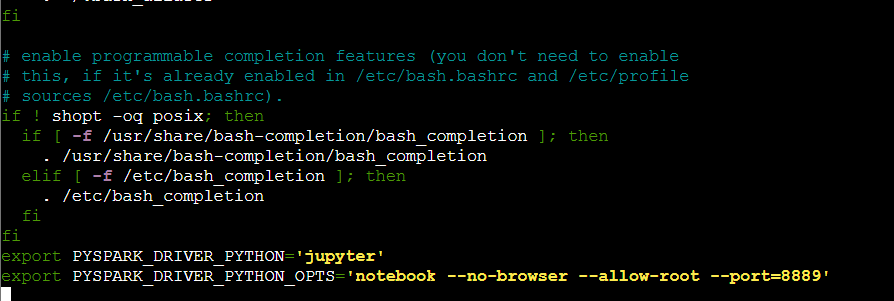
**sudo mv spark-3.2.0-bin-hadoop3.2 /opt/spark**

1. Selanjutnya Buka dan mengedit file .bashrc menggunakan editor teks nano

**nano .bashrc**

1. Tambahkan konfigurasi berikut :

**export SPARK\_HOME=/opt/spark  
export PATH=$PATH:$SPARK\_HOME/bin:$SPARK\_HOME/sbin  
export PYSPARK\_PYTHON=/usr/bin/python3**

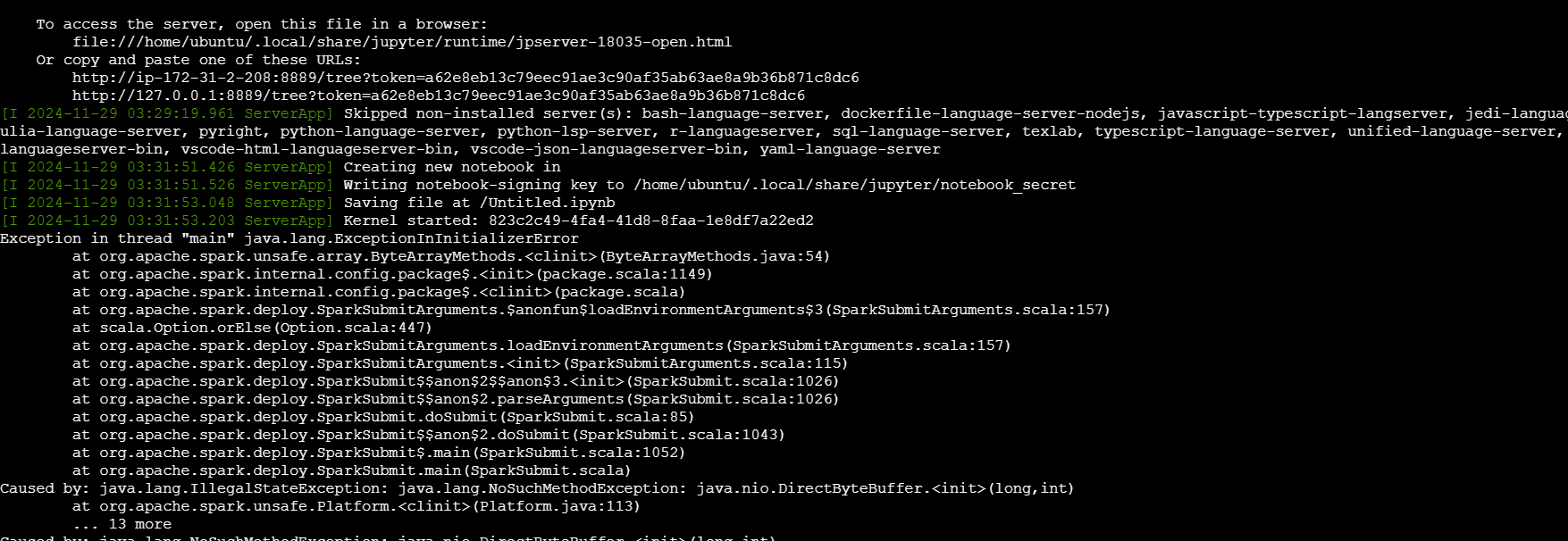


1. Terapkan perubahan pada file .bashrc

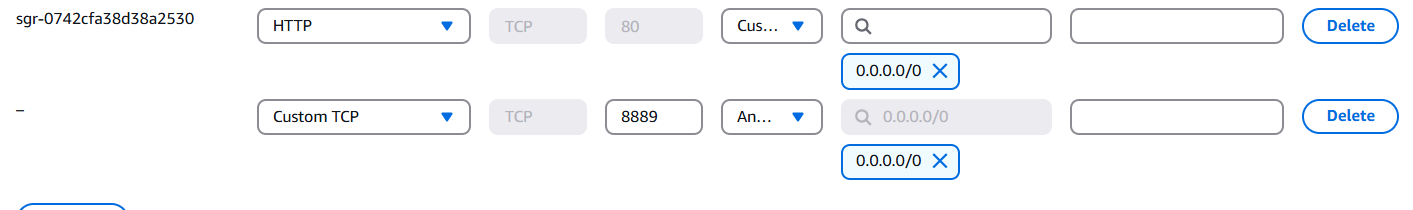
**source .bashrc**

1. Jalankan jupyter sekaligus menjalankan spark:

**/opt/spark/bin/pyspark**

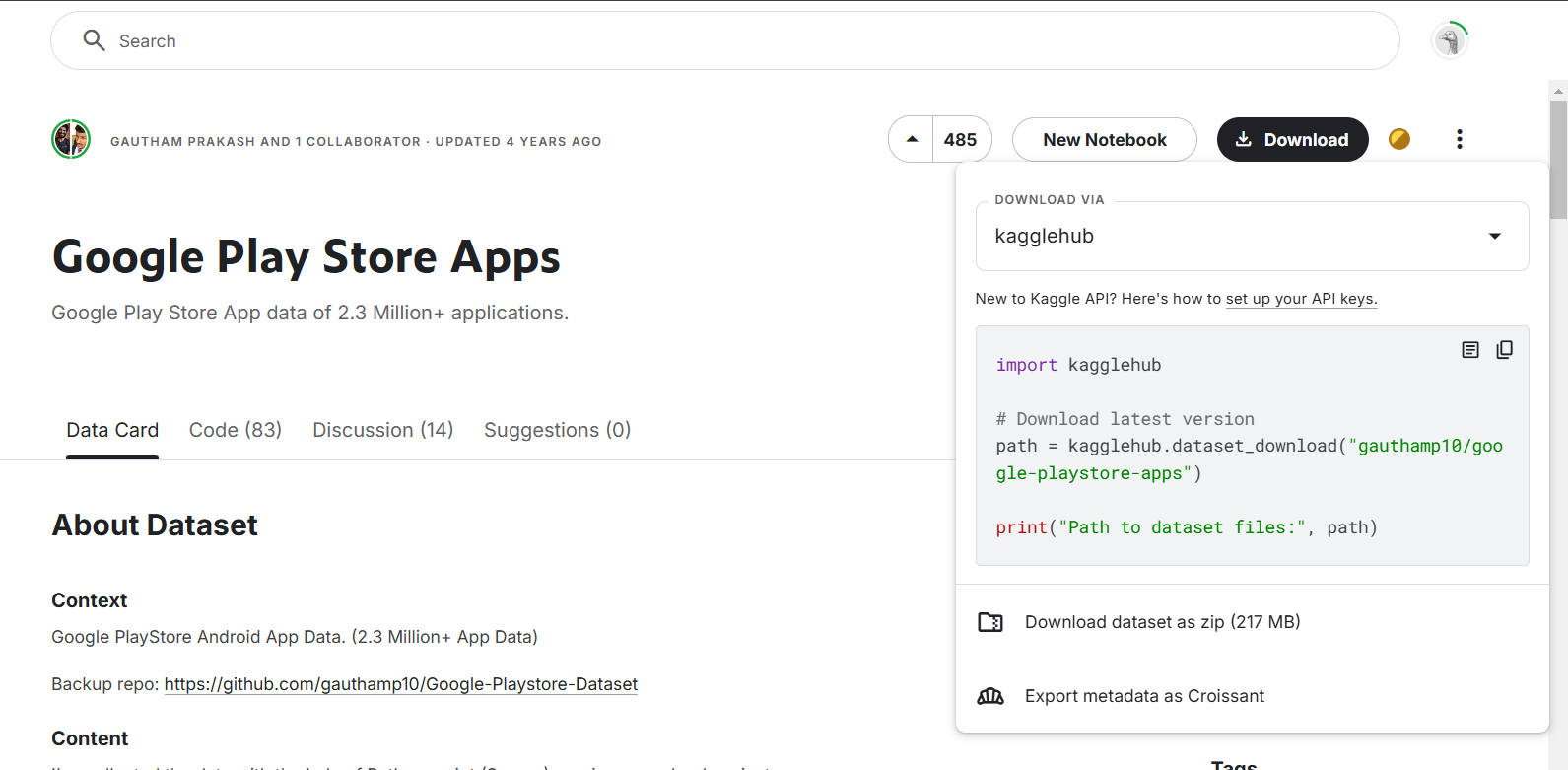


1. Menambahkan port 8889



## Pertemuan 12

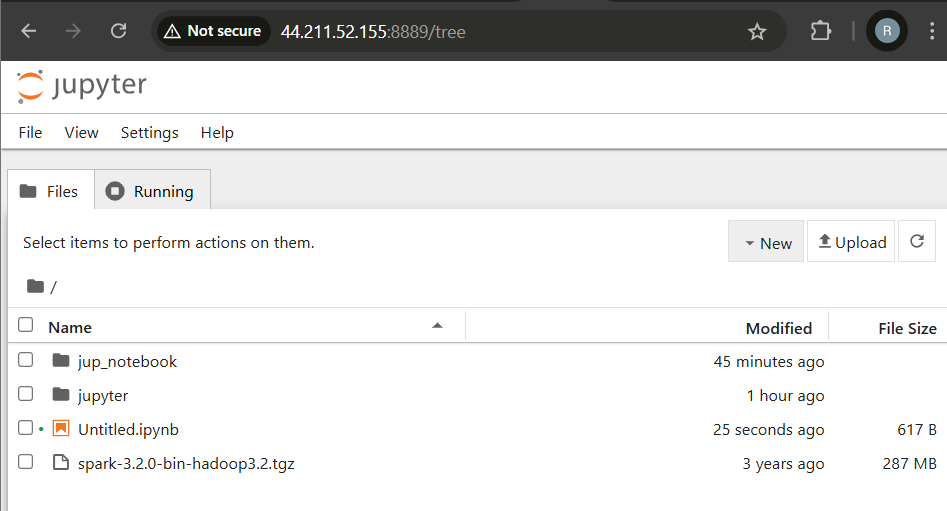
Di pertemuan ke 12 ini saya mencari data dengan menggunakan kaggle. Pada saat pencarian data saya menggunakan filter agar memudahkan dalam mencari data yang mau di cari dan disini saya mencari data dengan nama Google Play Store Apps. Dan datanya harus sesuai ketentuan yang telah di berikan yaitu minimal 200Mb :



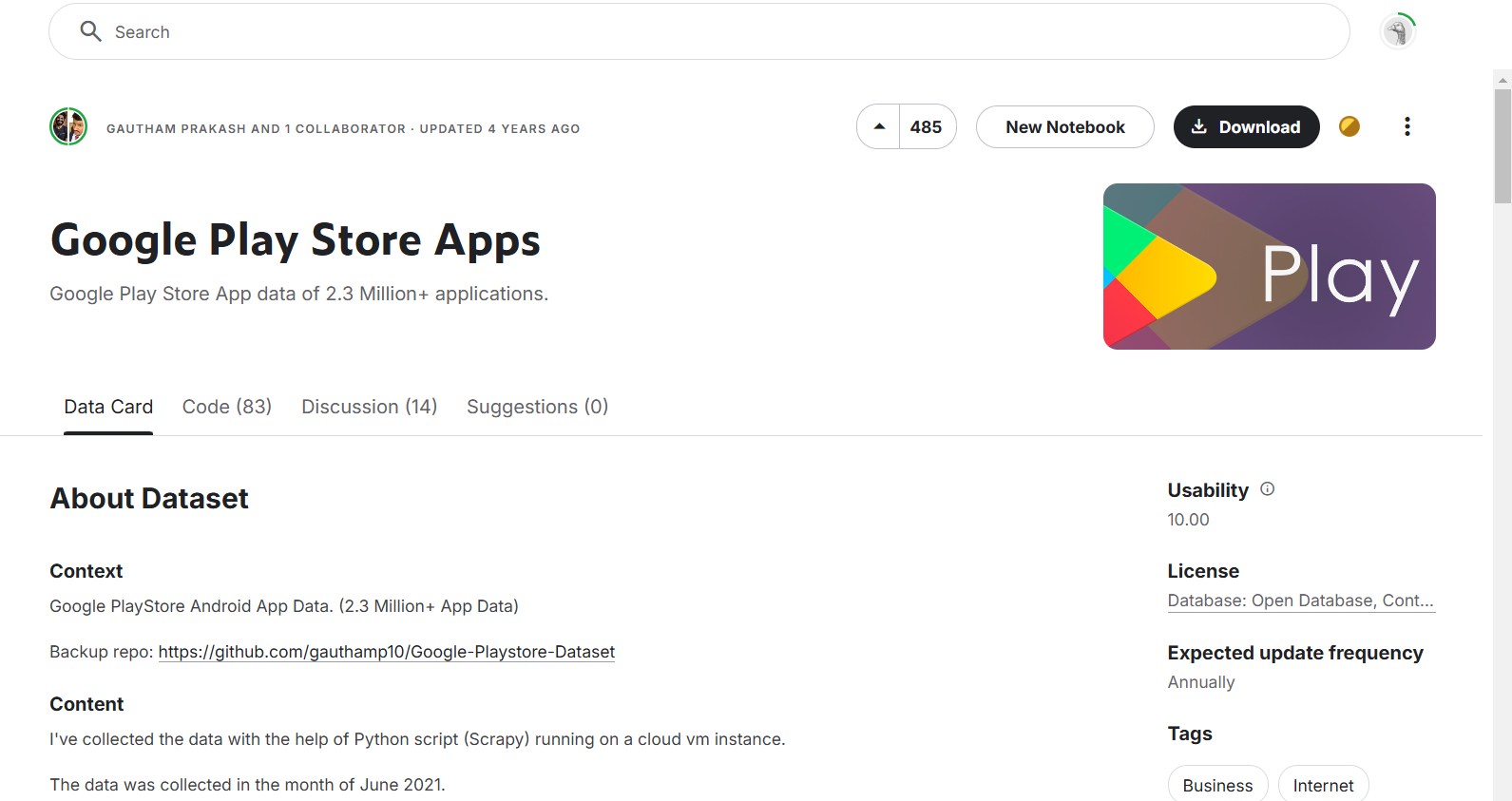
## Pertemuan 13

Pada pertemuan ini saya mengelola data yang saya ambil dari kaggle dan saya mengelolanya menggunakan jupyter notebook, untuk perintah yang di gunakan untuk visualisasi data saya mencari dari refrensi-refrensi yang berada pada google. Dan saya pada praktikum ini mengelola data tentang Google Play Store Apps, data yang ada pada Google Play Store Apps banyak yang bisa di kelola, tetapi disini saya mengambil sedikit saja berikut proses pengelolaan datanya :

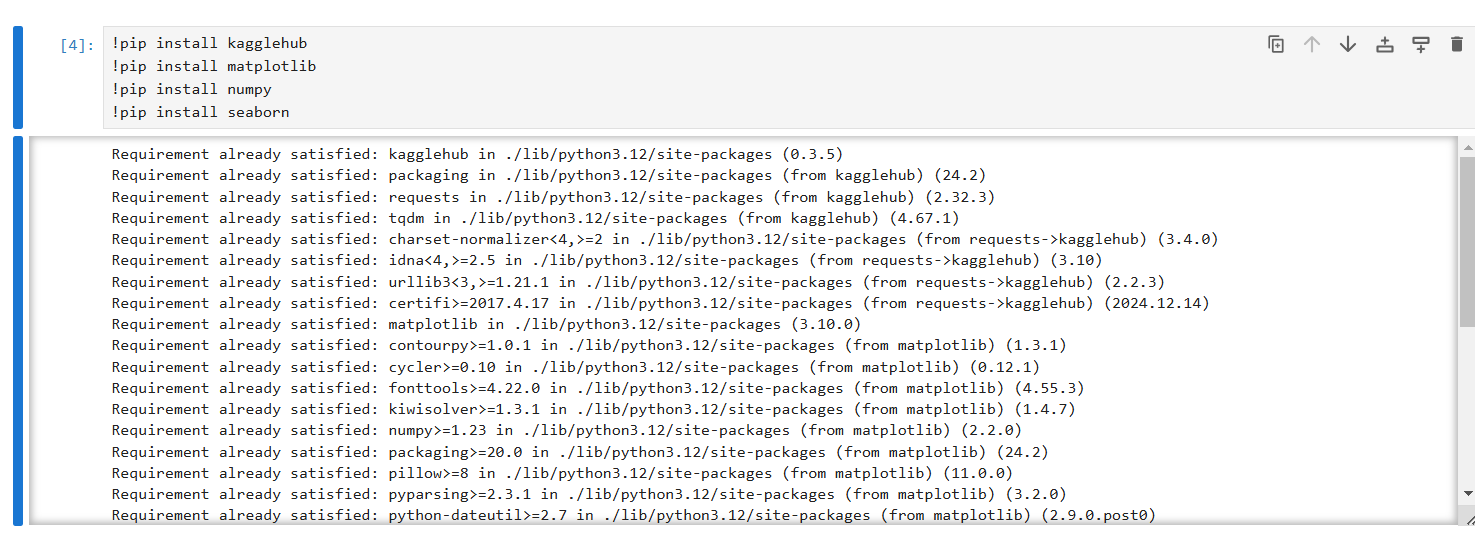
1. Akses Jupyter dengan menggunakan url yang ada pada terminal setelah itu membuat python 3 dengan menglik New pada pojok kanan maka tampilan akanseperti di bawah.



1. Langkah selanjutnya ialah copy code yang ada pada data yang sudah di pilih di kaggle



1. Instalasi Library yang di perlukan di dalam jupyter notebook disini saya menginstal library kagglehub, matplotlib, numpy, dan seaborn



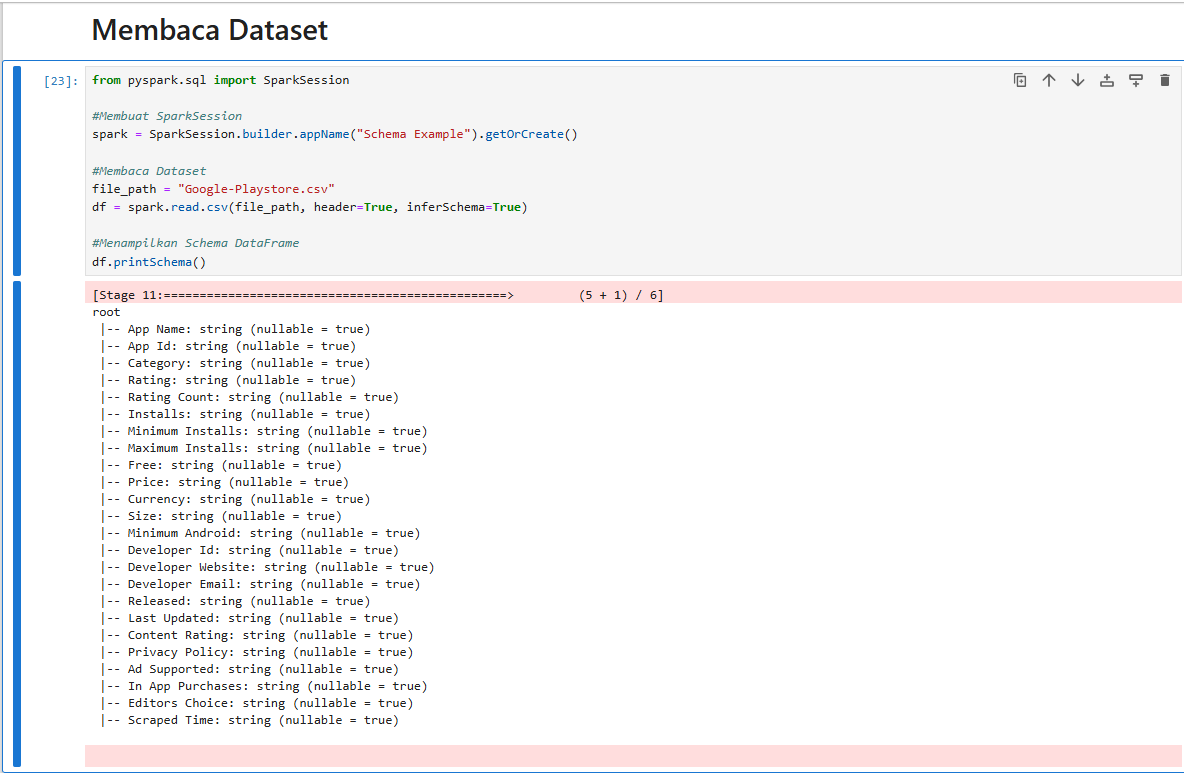
1. Disini saya download dataset yang berada di kaggle



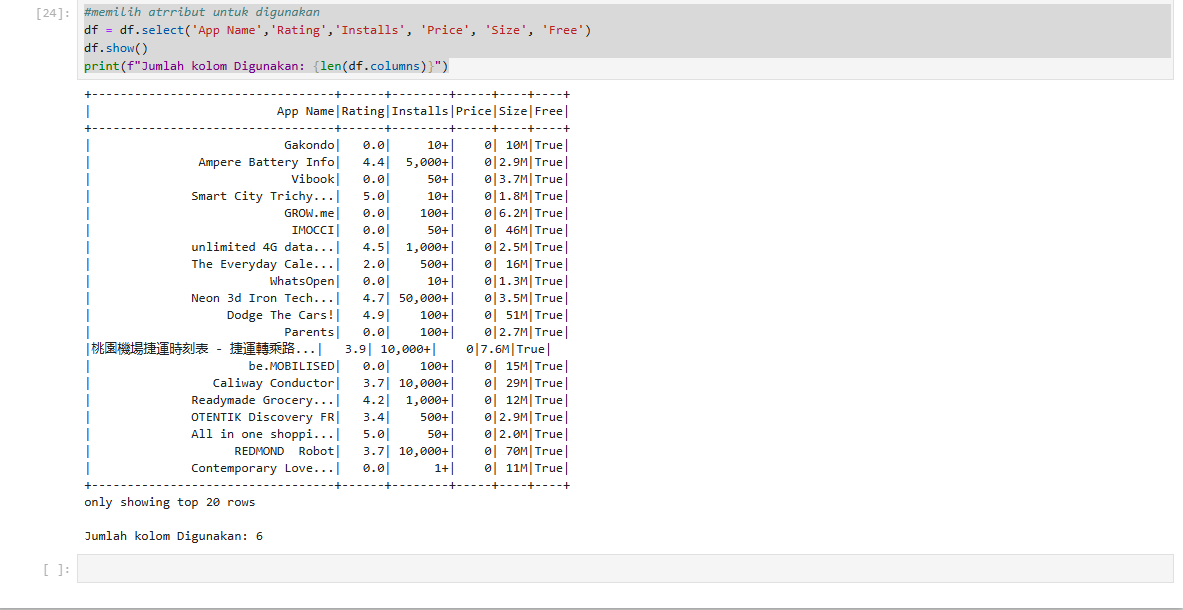
1. Memindahkan dataset ke direktori pekerjaan yang akan saya kerjakan



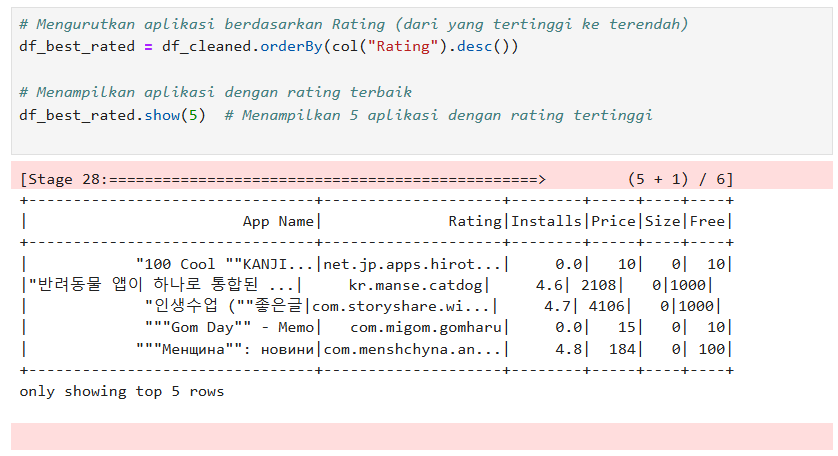
1. Disini saya membaca dataset yang telah saya pilih dari kaggle dan data mana yang saya ingin tampilkan



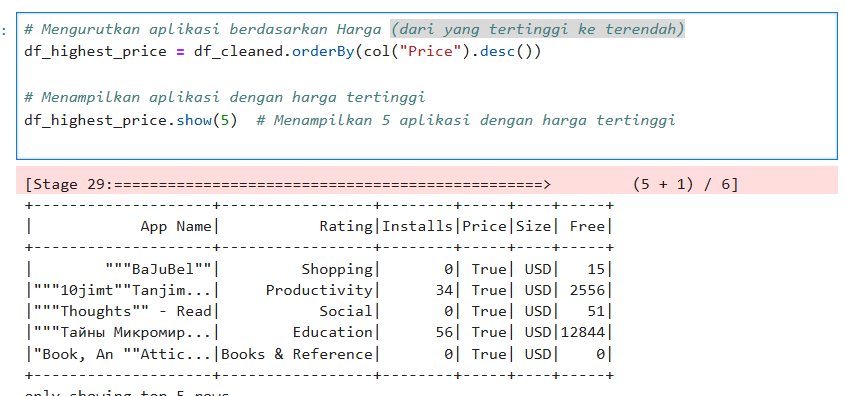
1. Disni saya memunculkan data yang saya ingin tampilkan, data yang saya tampilkan



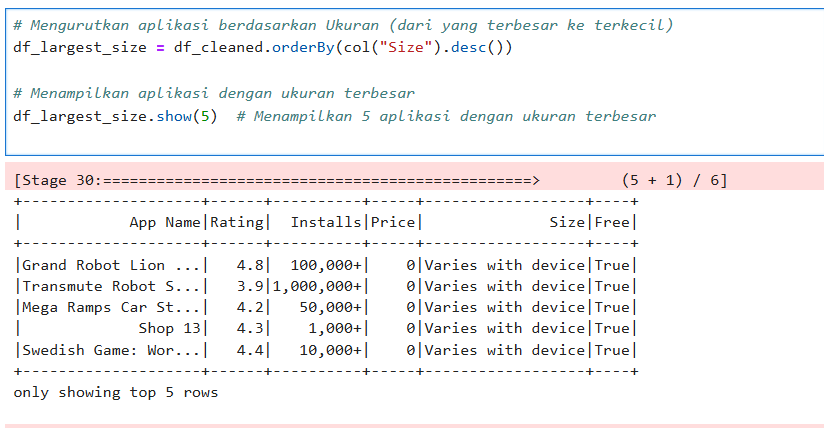
1. Lalu selanjutnya disini saya mengurutkan aplikasi berdasarkan ratting dan yang muncul seperti berikut



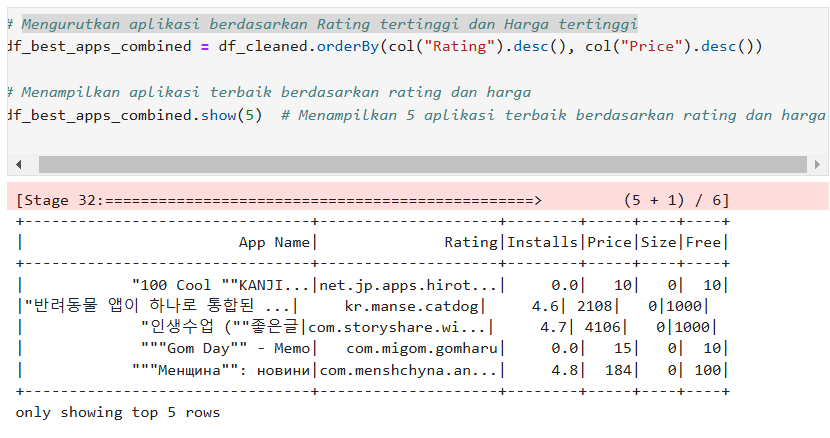
1. Lalu selanjutkan Saya mengurutkan aplikasi berdasarhakan harga (dari yang tertinggi ke terendah)



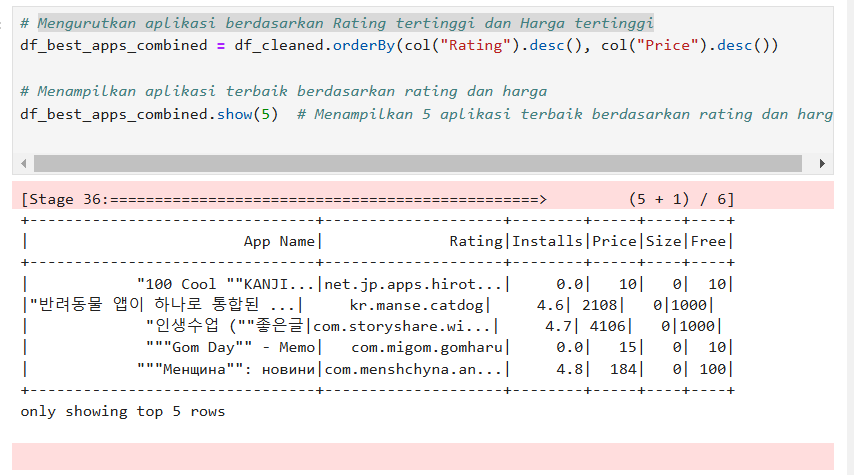
1. Lalu disini saya Mengurutkan aplikasi berdasarkan Ukuran (dari yang terbesar ke terkecil)



1. Lalu disini saya Mengurutkan aplikasi berdasarkan Rating tertinggi dan Harga tertinggi



1. Lalu disini saya Mengurutkan aplikasi berdasarkan Rating tertinggi dan Harga tertinggi



1. Kemudian disini saya Menampilkan aplikasi yang direkomendasikan dengan rating terbaik, Berikut adalah hasil akhirnya :

