

NAMA : HERTIN NURHAYATI
KELAS :SIB 3E
NIM : 2241760025

Interaksi dengan Spark di Lingkungan Windows Menggunakan Docker

Dalam praktikum ini kita akan menjalankan Apache Spark di Windows menggunakan Docker dan mencoba membuat job sederhana dengan berbagai macam alternatif cara.

Prasyarat

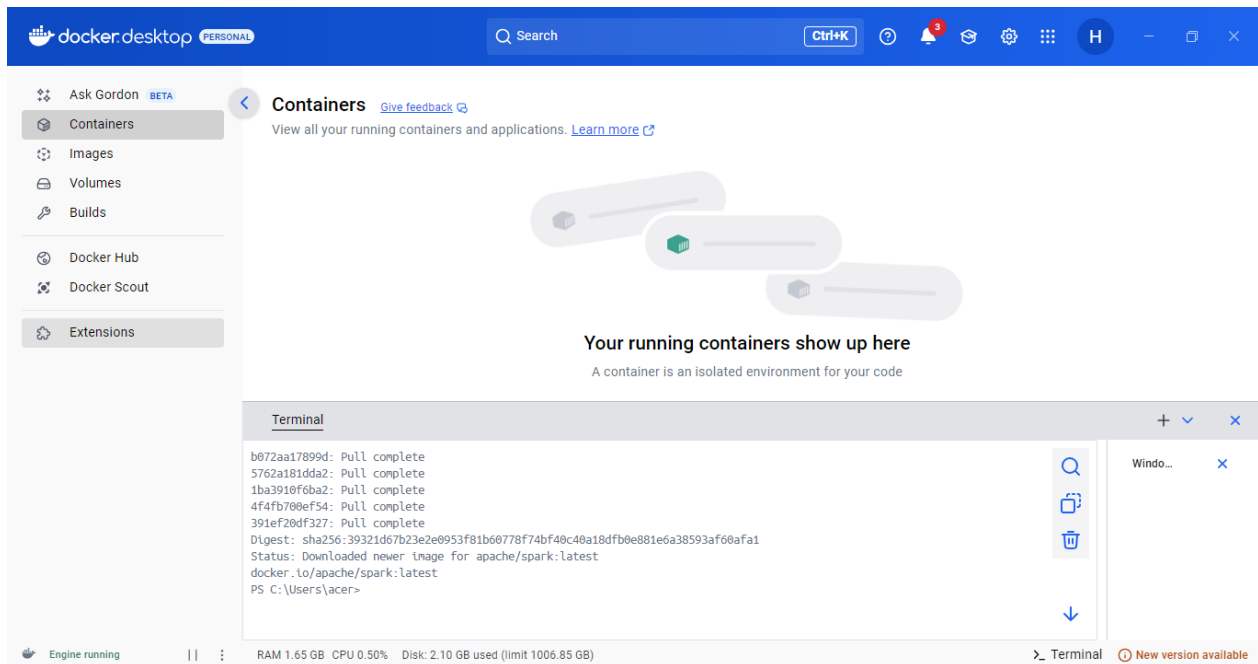
1. Windows 10/11 (64-bit) dengan versi Pro, Enterprise, atau Education
2. Docker Desktop untuk Windows diinstal dan berjalan
3. WSL 2 (Windows Subsystem for Linux versi 2) diaktifkan

Langkah-langkah 1. Pull Image Spark Resmi

docker pull apache/spark:latest

```
Terminal
f937e0a2086c: Pull complete
0f3083818c14: Pull complete
d3c7b6bd77aa: Pull complete
4d9bb71a5e54: Pull complete
b072aa17899d: Pull complete
5762a181dda2: Pull complete
1ba3910f6ba2: Pull complete
4f4fb700ef54: Pull complete
391ef20df327: Pull complete
Digest: sha256:39321d67b23e2e0953f81b60778f74bf40c40a18dfb0e881e6a38593af60afa1
Status: Downloaded newer image for apache/spark:latest
docker.io/apache/spark:latest
PS C:\Users\Acer> 
```

Hasil :



2. Menjalankan Spark Master

Sebelumnya buat docker network sebagai berikut

```
PS C:\Users\Acer> docker network create spark-net
00568fcd04abf0f4a9cf5d98aa568e7230e2dac924bee0bd70257c10ca3b3db
```

Hasil :

```
PS C:\Users\acer> docker network create spark-net
88c832cb4a110097733feb0d85cb2474c4b0b9d525f90f4acb1e893c287530ef
PS C:\Users\acer>
```

Kemudian jalankan spark-master dalam network tersebut

```
PS C:\Users\Acer> docker run -d -p 8080:8080 -p 7077:7077 --name spark-master --network spark-net -m 2g --cpus=2 apache/spark:latest /opt/spark/bin/spark-class org.apache.spark.deploy.master.Master
5bbd45e9dcf3af50dba8c63c111bb2b824f65a768fbb224a41bb795c51c28d70
```

Hasil:

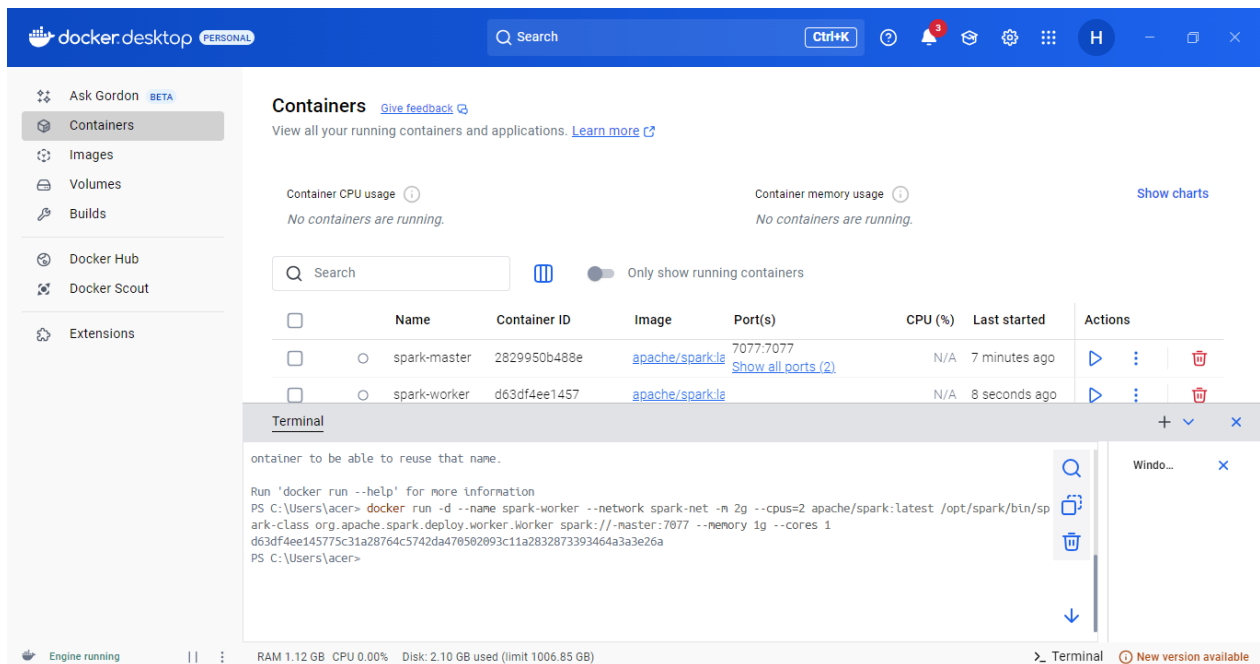
```
PS C:\Users\acer> docker run -d -p 8080:8080 -p 7077:7077 --name spark-master --network spark-net -m 2g --cpus=2 apache/spark:latest /opt/spark/bin/spark-class org.apache.spark.deploy.master.Master
2829950b488e710274778e4e59a0f488d873d43a9bbe89434644a3a48d39f64b
PS C:\Users\acer>
```

Kita alokasikan resource untuk memastikan tidak kekurangan resource dalam menjalankan job.

3. Menjalankan Spark Worker

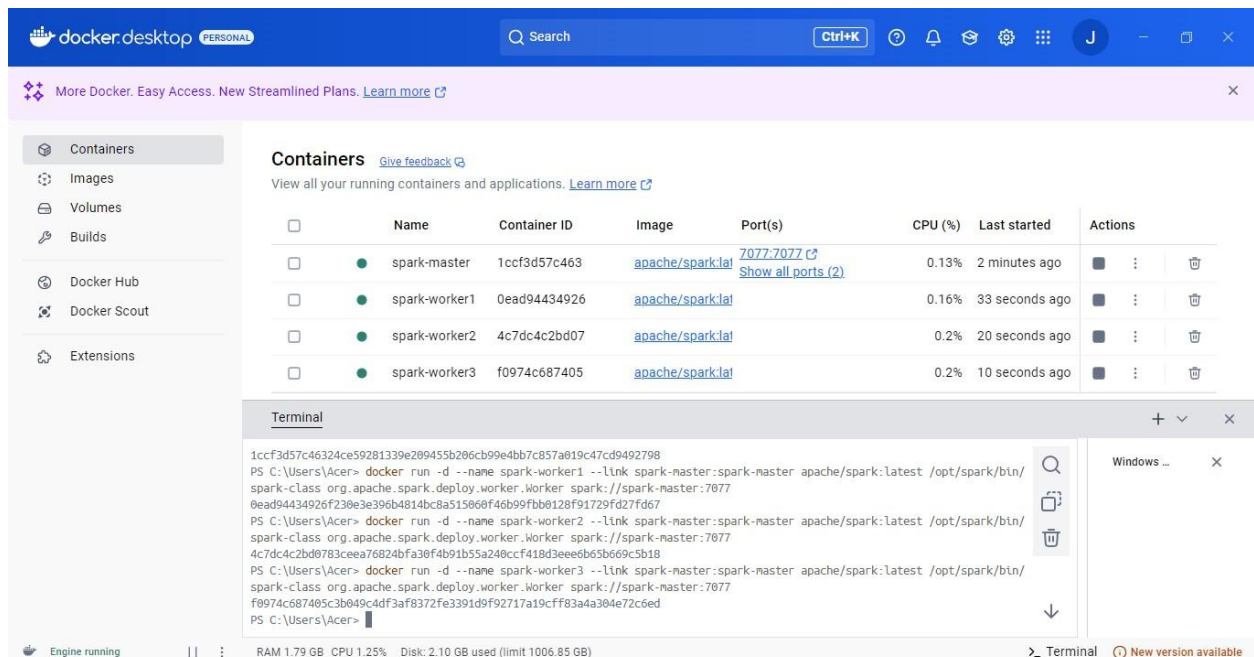
```
PS C:\Users\Acer> docker run -d --name spark-worker --network spark-net -m 2g --cpus=2 apache/spark:latest /opt/spark/bin/spark-class org.apache.spark.deploy.worker.Worker spark://spark-master:7077 --memory 1g --cores 1
58c18a90d6c65b112cc2e72ed7cad42c991bc8bb2710f459e2bce6b53168f9a8
```

Hasil :



Kita perlu alokasikan resource misalnya 2G memori dan 2 core CPU

jalankan perintah di atas beberapa kali dengan nama yang berbeda untuk membuat beberapa worker. Misalnya spark-worker1, spark-worker2, dan seterusnya Contoh menggunakan 3 worker.



4. Mengakses Spark Web UI

<http://localhost:8080>

Spark Master at spark://172.17.0.2:7077

URL: spark://172.17.0.2:7077
Alive Workers: 3
Cores in use: 24 Total, 0 Used
Memory in use: 8.2 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory	Resources
worker-20250326042206-172.17.0.3-35705	172.17.0.3:35705	ALIVE	8 (0 Used)	2.7 GiB (0.0 B Used)	
worker-20250326042219-172.17.0.4-35299	172.17.0.4:35299	ALIVE	8 (0 Used)	2.7 GiB (0.0 B Used)	
worker-20250326042229-172.17.0.5-36505	172.17.0.5:36505	ALIVE	8 (0 Used)	2.7 GiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Hasil:

Spark Master at spark://71b1109d3b0f:7077

URL: spark://71b1109d3b0f:7077
Alive Workers: 3
Cores in use: 6 Total, 0 Used
Memory in use: 3.0 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (3)

Worker Id	Address	State	Cores	Memory	Resources
worker-20250611022301-172.18.0.3-39953	172.18.0.3:39953	ALIVE	2 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20250611022301-172.18.0.4-40465	172.18.0.4:40465	ALIVE	2 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20250611022523-172.18.0.6-35659	172.18.0.6:35659	ALIVE	2 (0 Used)	1024.0 MiB (0.0 B Used)	

Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

5. Menjalankan Spark Shell

```
PS C:\Users\Acer> docker run -it --rm --name spark-shell --network spark-net --link spark-master:spark-master apache/spark:latest /opt/spark/bin/spark-shell --master spark://spark-master:7077
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/03/26 04:52:31 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Spark context Web UI available at http://03f37d151387:4040
Spark context available as 'sc' (master = spark://spark-master:7077, app id = app-20250326045233-0000).
Spark session available as 'spark'.
Welcome to
```



Hasil:

```
PS C:\Users\acer> docker run -it --rm --name spark-shell --network spark-net --link spark-master:spark-master apache/spark:latest /opt/spark/bin/spark-shell --master spark://spark-master:7077
>>
```

Setting default log level to "WARN".

To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).

25/06/11 02:29:31 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java class

```

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | || |___| \
| |  | || |___| \
| |  | || |___| \
|_|  |_| \____/
version 3.5.5

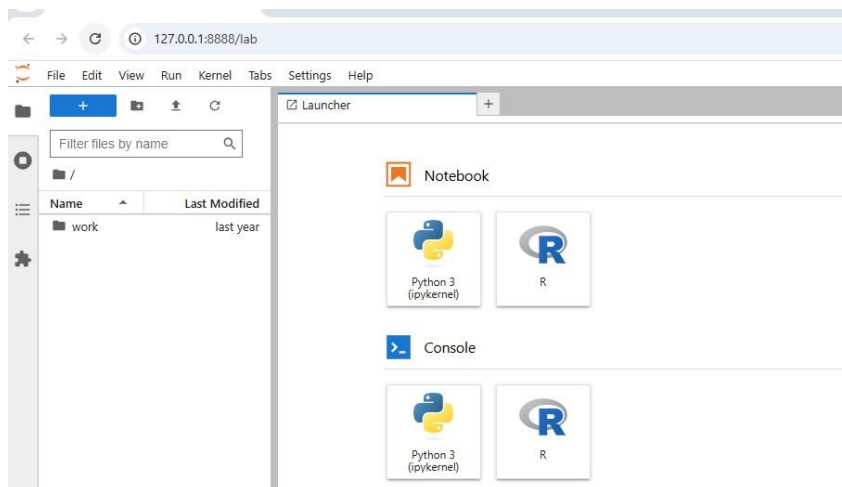
```

Using Scala version 2.12.18 (OpenJDK 64-Bit Server VM, Java 11.0.26)

6. Menggunakan Jupyter Notebook dengan Spark

```
PS C:\Users\Acer> docker run -it -p 8888:8888 -p 4040:4040 --network spark-net jupyter/all-spark-notebook
Entered start.sh with args: jupyter lab
Running hooks in: /usr/local/bin/start-notebook.d as uid: 1000 gid: 100
Done running hooks in: /usr/local/bin/start-notebook.d
Running hooks in: /usr/local/bin/before-notebook.d as uid: 1000 gid: 100
Sourcing shell script: /usr/local/bin/before-notebook.d/spark-config.sh
Done running hooks in: /usr/local/bin/before-notebook.d
```

Setelah itu, akses Jupyter Notebook di: <http://localhost:8888>



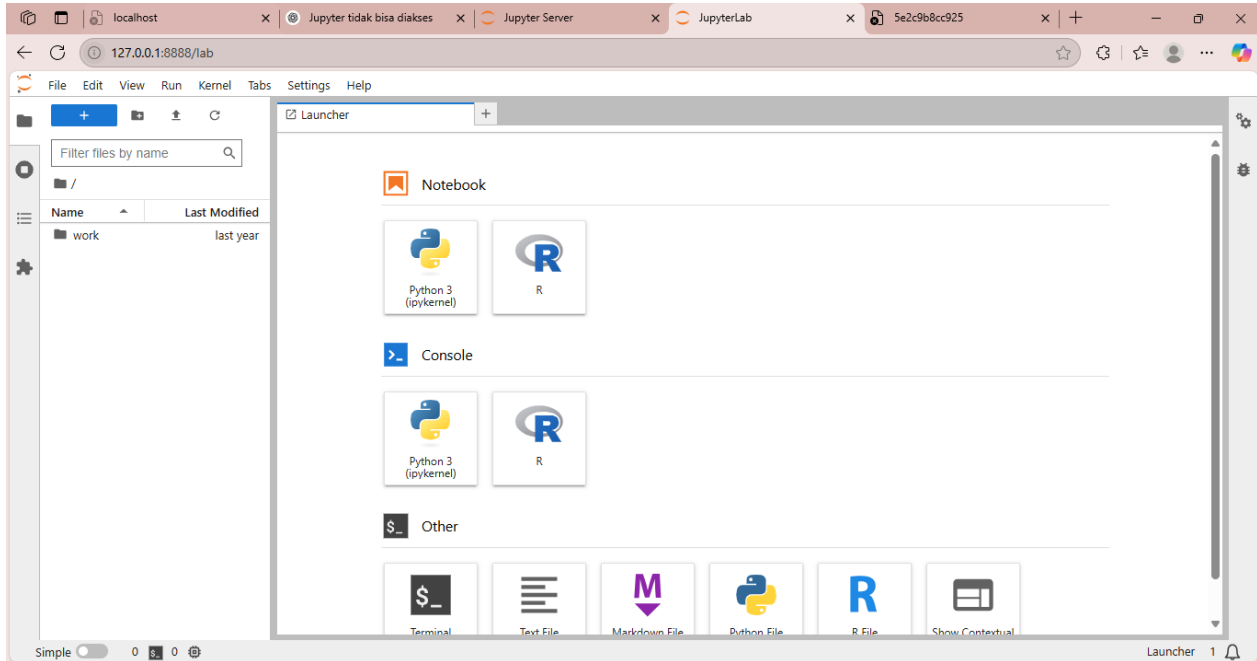
Hasil:

Jalankan jupyter

```
PS C:\Users\acer> docker logs bold_lalande
>>
Entered start.sh with args: jupyter lab
Running hooks in: /usr/local/bin/start-notebook.d as uid: 0 gid: 0
Done running hooks in: /usr/local/bin/start-notebook.d
Granting jovyan passwordless sudo rights!
Running hooks in: /usr/local/bin/before-notebook.d as uid: 0 gid: 0
Sourcing shell script: /usr/local/bin/before-notebook.d/spark-config.sh
Done running hooks in: /usr/local/bin/before-notebook.d
Running as jovyan: jupyter lab
[I 2025-05-27 06:50:47.157 ServerApp] Package jupyterlab took 0.0000s to import
```

Akses port 8888, lalu masukkan token kemudian log in.

```
To access the server, open this file in a browser:
file:///home/jovyan/.local/share/jupyter/runtime/jpserver-21-open.html
Or copy and paste one of these URLs:
http://5e2c9b8cc925:8888/lab?token=280c1fb9c88b9747bb05aff11fae079b2d0cd005aa2130b7
http://127.0.0.1:8888/lab?token=280c1fb9c88b9747bb05aff11fae079b2d0cd005aa2130b7
[I 2025-06-11 10:53:48.727 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nod
ejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright, python-language-server, python
-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, unified-language-server, vscode-css
-language-server-bi[jupyter] lab --ip=0.0.0.0 --port=8888 --allow-root --no-browser --NotebookApp.token='er
>> C:\Users\acer>
```



Untuk menghentikan container:

docker stop spark-master spark-worker

docker rm spark-master spark-worker

```
PS C:\Users\Acer> docker stop spark-master spark-worker1 spark-worker2
spark-master
spark-worker1
spark-worker2
PS C:\Users\Acer> docker rm spark-master spark-worker1 spark-worker2
spark-master
spark-worker1
spark-worker2
PS C:\Users\Acer> □
```

Contoh Program Word Count dengan Spark di Docker

Berikut adalah contoh program Word Count (menghitung kemunculan kata) menggunakan Apache Spark yang bisa dijalankan di lingkungan Docker:

Cara 1: Menggunakan Spark Shell

1. Jalankan Spark Shell di Docker seperti contoh di atas
2. Ketikkan kode berikut di Spark Shell:

```
// Buat RDD dari koleksi teks
val textData = List("Hello Spark", "Hello Docker", "Spark is awesome", "Docker makes Spark easy")
val rdd = sc.parallelize(textData)

// Lakukan word count
val wordCounts = rdd.flatMap(line => line.split(" ")) // Split menjadi kata-kata
                      .map(word => (word, 1))           // Map setiap kata ke tuple (word, 1)
                      .reduceByKey(_ + _)              // Reduce dengan menjumlahkan

// Tampilkan hasil
wordCounts.collect().foreach(println)
```

```
Terminal

scala> res1: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[3] at reduceByKey at <console>:24

scala> wordCounts.collect().foreach(println)
Hello
Spark
Hello
Docker
Spark
is
awesome
Docker
makes
Spark
```

Untuk keluar dari spark-shell gunakan:

```
System.exit(0)
```

Hasil:

Untuk menjalankan Spark Shell:



```
Terminal
res9: org.apache.spark.rdd.RDD[(String, Int)] = ShuffledRDD[8] at reduceByKey at <console>:24

scala>

scala> // Tampilkan hasil

scala> wordCounts.collect().foreach(println)
[Stage 2:>
Hello Docker
Hello Spark
Spark is awesome
Docker makes Spark easy

scala>
```

1. Jalankan PySpark Shell di Docker:

```
PS C:\Users\Acer> docker run -it --rm --name pyspark-shell --network spark-net --link spark-master:spark-master apache/spark:latest /opt/spark/bin/pyspark --master spark://spark-master:7077
Python 3.8.10 (default, Feb  4 2025, 15:02:54)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/03/26 05:09:36 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Welcome to
```

Hasil:


```
Terminal

PS C:\Users\acer> docker run -it --rm --name pyspark-shell --network spark-net --link spark-master:spark-master apache/spark:latest /opt/spark/bin/pyspark --master spark://spark-master:7077
>>
Python 3.8.10 (default, Feb  4 2025, 15:02:54)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/06/11 11:56:24 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java class
es where applicable
Welcome to

      _ _ _ _ _
     / _ _ _ _ _ \
    / _ _ _ _ _ \
   / _ _ _ _ _ \
  / _ _ _ _ _ \
 / _ _ _ _ _ \
/_ _ _ _ _ \

version 3.5.5

Using Python version 3.8.10 (default, Feb  4 2025 15:02:54)
Spark context Web UI available at http://9757cba22ec3:4040
Spark context available as 'sc' (master = spark://spark-master:7077, app id = app-20250611115636-0000).
SparkSession available as 'spark'.
>>> █
```

2. Ketikkan kode Python berikut:

```
from pyspark.sql import SparkSession

# Buat SparkSession
spark = SparkSession.builder.appName("WordCount").getOrCreate()

# Data contoh
data = ["Hello Spark", "Hello Docker", "Spark is awesome", "Docker makes Spark
easy"] rdd = spark.sparkContext.parallelize(data)

# Proses Word Count
word_counts = rdd.flatMap(lambda line: line.split(" ")) \
    .map(lambda word: (word, 1)) \
    .reduceByKey(lambda a, b: a + b)

# Tampilkan hasil
word_counts.collect()
```

```

Spark context available as 'sc' (master = spark://spark-master:7077, app id = app-20250618053323-0000).
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> spark = SparkSession.builder.appName("WordCount").getOrCreate()
25/03/26 05:11:39 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>> data = ["Hello Spark", "Hello Docker", "Spark is awesome", "Docker makes Spark easy"]
>>> rdd = spark.sparkContext.parallelize(data)
>>> word_counts = rdd.flatMap(lambda line: line.split(" ")) \
...                       .map(lambda word: (word, 1)) \
...                       .reduceByKey(lambda a, b: a + b)
>>> word_counts.collect()
[('Hello', 2), ('Spark', 3), ('is', 1), ('awesome', 1), ('Docker', 2), ('makes', 1), ('easy', 1)]
>>>

```

Untuk keluar dari pyspark-shel menggunakan: exit()

Hasil:

```

Using Python version 3.8.10 (default, Feb  4 2025 15:02:54)
Spark context Web UI available at http://c9529144f0cc:4040
Spark context available as 'sc' (master = spark://spark-master:7077, app id = app-20250618053323-0000).
SparkSession available as 'spark'.
>>> from pyspark.sql import SparkSession
>>> spark = SparkSession.builder.appName("WordCount").getOrCreate()
25/06/18 05:34:51 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
>>> data = ["Hello Spark", "Hello Docker", "Spark is awesome"]
>>> rdd = spark.sparkContext.parallelize(data)
>>> word_counts = rdd.flatMap(lambda line: line.split(" ")).map(lambda word: (word, 1)).reduceByKey(lambda a, b: a + b)
>>> word_counts.collect()

```

Cara 3: Menggunakan Jupyter Notebook

Jika Anda menggunakan Jupyter Notebook (seperti di container jupyter/all-spark-notebook):

```

from pyspark.sql import SparkSession

# Inisialisasi Spark

```

```

spark = SparkSession.builder \
    .appName("WordCount") \
    .getOrCreate()

# Baca file teks (jika ingin membaca dari file)
# text_file =
spark.sparkContext.textFile("hdfs://.../input.txt")

# Untuk contoh, kita gunakan data dalam memori
data = ["Apache Spark is a unified analytics
engine",
        "Spark can run on Hadoop, Apache Mesos,
Kubernetes",
        "Spark is awesome for big data
processing"] rdd =
spark.sparkContext.parallelize(data)

#   Word   Count   counts   =
rdd.flatMap(lambda x: x.split(' ')) \
    .map(lambda x: (x, 1)) \
    .reduceByKey(lambda a, b: a + b) \
    .sortBy(lambda x: x[1], ascending=False)

# Tampilkan 10 kata paling sering muncul
counts.take(10)

```

The screenshot shows a JupyterLab window with a single notebook titled 'Untitled.ipynb'. The notebook contains the following code:

```
# Untuk contoh, kita gunakan data dalam memori
data = ["Apache Spark is a unified analytics engine",
        "Spark can run on Hadoop, Apache Mesos, Kubernetes",
        "Spark is awesome for big data processing"]
rdd = spark.sparkContext.parallelize(data)

# Word Count
counts = rdd.flatMap(lambda x: x.split(' ')) \
            .map(lambda x: (x, 1)) \
            .reduceByKey(lambda a, b: a + b) \
            .sortBy(lambda x: x[1], ascending=False)

# Tampilkan 10 kata paling sering muncul
counts.take(10)
```

The output of the code is displayed below the code cell:

```
[2]: [('Spark', 3),
      ('Apache', 2),
      ('is', 2),
      ('a', 1),
      ('unified', 1),
      ('engine', 1),
      ('can', 1),
      ('processing', 1),
      ('big', 1),
      ('Hadoop', 1)]
```

The JupyterLab interface includes a file browser on the left, a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help), and a status bar at the bottom indicating 'Python 3 (ipykernel) | Idle'.

Hasil:

The screenshot shows a JupyterLab window with a single notebook titled 'Untitled.ipynb'. The notebook contains the following code:

```
[1]: from pyspark.sql import SparkSession

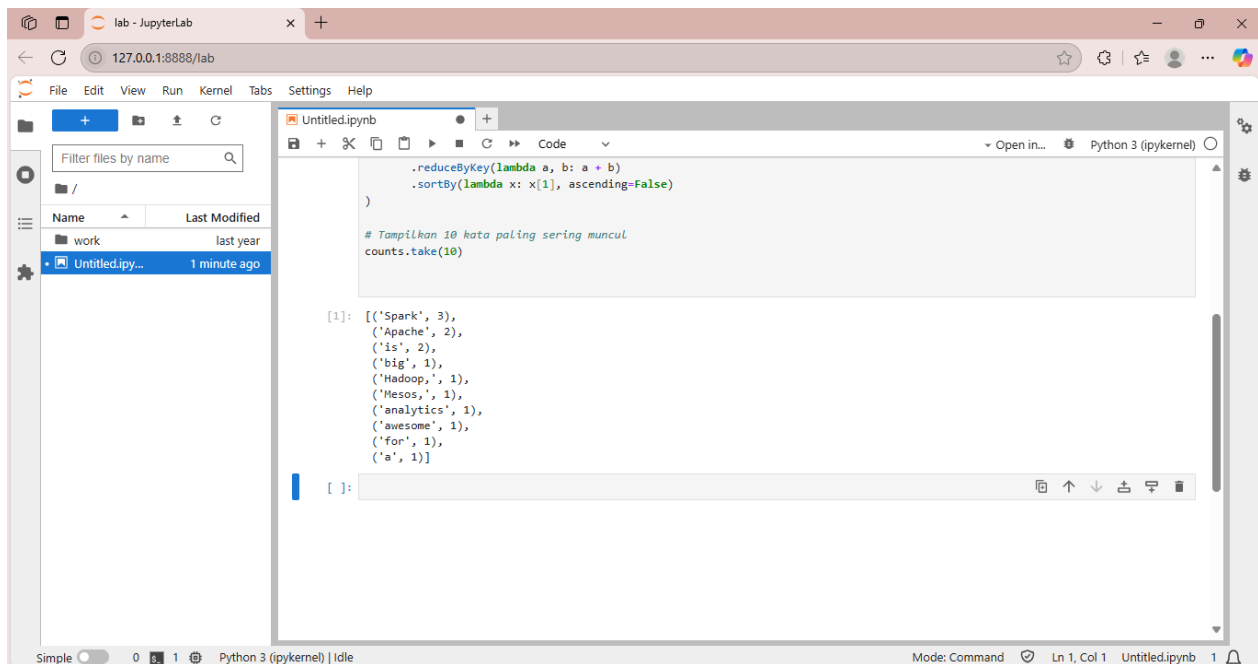
# Inisialisasi Spark
spark = (
    SparkSession.builder
        .appName("WordCount")
        .getOrCreate()
)

# Untuk contoh, kita gunakan data dalam memori
data = [
    "Apache Spark is a unified analytics engine",
    "Spark can run on Hadoop, Apache Mesos, Kubernetes",
    "Spark is awesome for big data processing"
]
rdd = spark.sparkContext.parallelize(data)

# Word Count
counts = (
    rdd.flatMap(lambda x: x.split(' '))
        .map(lambda x: (x, 1))
        .reduceByKey(lambda a, b: a + b)
        .sortBy(lambda x: x[1], ascending=False)
)

# Tampilkan 10 kata paling sering muncul
counts.take(10)
```

The JupyterLab interface includes a file browser on the left, a menu bar (File, Edit, View, Run, Kernel, Tabs, Settings, Help), and a status bar at the bottom indicating 'Python 3 (ipykernel) | Idle'.



Menjalankan Program sebagai Script

1. Buat file wordcount.py dengan isi berikut:

```
from pyspark.sql import SparkSession

if __name__ == "__main__":
    spark = SparkSession.builder.appName("WordCount").getOrCreate()

    # Untuk versi membaca file
    # lines = spark.read.text("input.txt").rdd.map(lambda r: r[0])

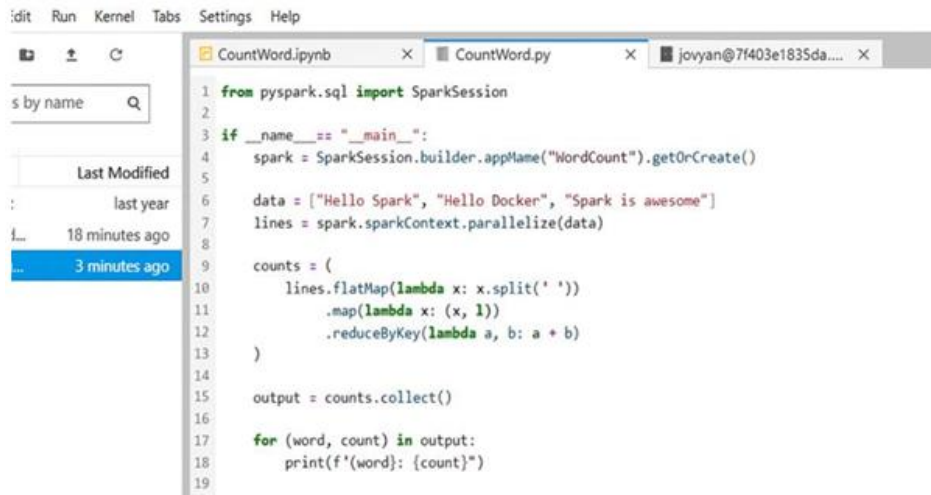
    # Untuk versi data contoh
    data = ["Hello Spark", "Hello Docker", "Spark is
awesome"]    lines = spark.sparkContext.parallelize(data)

    counts = lines.flatMap(lambda x: x.split(' ')) \
        .map(lambda x: (x, 1)) \
        .reduceByKey(lambda a, b: a + b)

    output = counts.collect()
    for (word, count) in output:
        print("%s: %i" % (word, count))

    spark.stop()
```

Hasil:



The screenshot shows a Jupyter Notebook interface with a file explorer on the left and a code editor on the right. The file explorer lists files by name and last modified time. The code editor shows a Python script named 'CountWord.py' that uses PySpark to count words in a dataset.

```
1 from pyspark.sql import SparkSession
2
3 if __name__ == "__main__":
4     spark = SparkSession.builder.appName("WordCount").getOrCreate()
5
6     data = ["Hello Spark", "Hello Docker", "Spark is awesome"]
7     lines = spark.sparkContext.parallelize(data)
8
9     counts = (
10         lines.flatMap(lambda x: x.split(' '))
11         .map(lambda x: (x, 1))
12         .reduceByKey(lambda a, b: a + b)
13     )
14
15     output = counts.collect()
16
17     for (word, count) in output:
18         print(f'{word}: {count}')
```

2. Jalankan jika sudah memastikan skrip di atas benar: - Simpan file CountWord.py - Jalankan: spark-submit CountWord.py

by name

Last Modified

laid your

ord... 4 minutes ago

ords 5. abingers .pp

CountWordJpymb X CountWord.py joryam@77435x1835da X +

tor driver) (84133)
25/00/14 14:14:55 INFO Executor: Finished task 11.0 in stage 1.0 (TID 17) 2480 bytes result sent to driver
25/00/14 16:14:55 INFO PythonKrunner: Tzann: Setal t 208, boot = 709, 101, = 245), finish = 7
25/00/14 16:14:55 INFO TaskSetManager: Finished task 31.0 in stage 1.0 (TID 20) in 488 ms on 7748x:1855da (ame
tor driver) (40:81)
25/00/14 16:14:55 INFO PythonKrunner: Tzann: Setal t 200, boot = 308, 1661, = 589, finish = 1
25/00/14 16:14:55 INFO TaskSetManager: Finished task 3.0 in stage 1.0 (TID 10) in 488 ms on 77483x1835da (amee
tor driver) (0:88)
25/00/14 16:14:55 INFO PythonKrunner: Tzann: Setal t 209, boot = 700, 136, = 285), finish = 3
25/00/14 16:14:55 INFO TaskSetManager: Finished task 3.0 in stage 1.0 (TID 18) in 488 ms on 77483x1835da (amee
tor driver) (0:81)
25/00/14 14:14:55 INFO Executor: Finished task 1.0 in stage 1.0 (TID 17), 2480 bytes result sent to driver
25/00/14 16:14:55 INFO TaskSetManager: Finished task 3.0 in stage 1.0 (TID 12) in 488 ms on 77483x1835da (amee
tor driver) (0:82)
25/00/14 16:14:55 INFO Executor: Finished task 0.0 in stage 1.0 (TID 17), 255 bytes result sent to driver
25/00/14 16:14:55 INFO Executor: Finished task 0.0 in stage 1.0 (TID 17) 744 bytes result sent to driver
25/00/14 16:14:55 INFO TaskSetManager: Finished task 3.0 in stage 1.0 (TID 17) in 488 ms on 77483x1835da (amee
tor driver) (44:02)
25/00/14 16:14:55 INFO TaskSetManager: Finished task 3.0 in stage 1.0 (TID 17) in 588 ms on 77483x1835da (amee
tor driver) (44:02)
25/00/14 16:14:55 INFO TaskSetManager: Finished task 3.0 in stage 1.0 (TID 17) in 588 ms on 77483x1835da (amee
tor driver) (64:182)
25/00/14 14:14:55 INFO TablSchedulerImpl: Reared fastat t.0, where nasks tore all comitted, fren past
25/00/14 16:14:55 INFO DmScheduler: SppilStage = (Collect at .basSilerincCoordind.dp:13) Tine/663 is 0,508
25/00/14 16:14:55 INFO DmScheduler: Jo0.0 is finished. Cancelling potential speculative or speculative tasks for ti
in net
25/00/14 16:14:55 INFO TabSchedulerImpl: Killing off running tasks in stage 3! Stage finished
25/00/14 16:14:55 INFO DmScheduler: Jo0.0 finished collected faeserjoryam@77435x1835da, task 3.142400 =
Batu: 2
Sun X
Sancvee: 1
Gaskn: 4
Edon.. 1
25/00/14 16:14:55 INFO SparkContext: SparkContext is stopping with asitCode 0.
25/00/14 16:14:55 INFO Spouter: Storgen apopt.cad 0d at Gate..76148&65090a-8090
25/00/14 16:14:55 INFO SparkContext: Yindiness! asouu = basioipdflwKenthiterIndpoint stopped!
25/00/14 16:14:55 INFO SpoutYubber YatorMosenrtiopend
25/00/14 16:14:55 INFO OlannKoorper: Plebblanager stapnd
25/00/14 16:14:55 INFO OleatTookranddner: DloSbkoncbetrugter stopped
25/00/14 16:14:55 INFO OutputCommitCoordinatorOutputCommitCoordinator: OutputCommitCoordinator stoppe
25/00/14 16:14:55 INFO SparkContext: Successfully stopped SparkContext
25/00/14 16:14:55 INFO SparkContextManager: Deletetune:buee.153103
25/00/14 16:14:55 INFO SparkContextManager: Deleting directory /tmp/apark-579uBeX-18ee-43af-8o3d-adfb8872len5
25/00/14 16:14:55 INFO SparkContextManager: Deleting directory /tmp/apark-8ud6596x-d48f-dadf-033d-36477817lebo
turidor Priken.. 7705-0781-0008-7882f0a3Kob1
25/00/14 16:14:55 INFO SparkContextManager: Deleting directory /tmp/apark-8ud6e050-d08f-4aof-b34b-00477817bc5d
(0sd) 0d105097023bx40559:-0