

**LAPORAN UJIAN TENGAH SEMESTER  
MATA KULIAH BIG DATA  
Jobsheet 9 – Spark SQL**



**Dosen Pengampu:  
M. Hasyim Ratsanjani, S.Kom., M.Kom.**

**Disusun Oleh:**

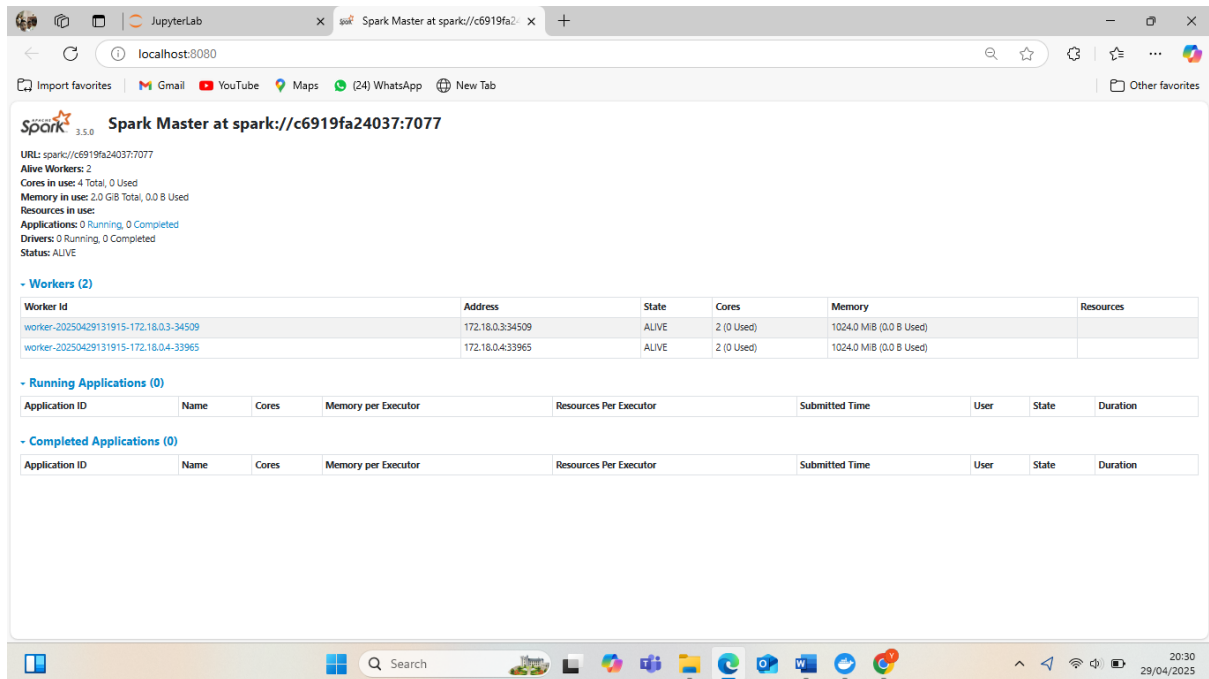
**Yuma Rakha Samodra Sikayo**

**NIM.2241720194**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2025**

# Spark SQL, DataSources, DataFrame, dan Dataset APIs

## Menyiapkan lingkungan Spark Cluster



The screenshot shows the Spark Master web interface in a browser. The URL is `localhost:8080`. The interface displays the following information:

- Spark Master at spark://c6919fa24037:7077**
- URL: `spark://c6919fa24037:7077`
- Alive Workers: 2
- Cores in use: 4 Total, 0 Used
- Memory in use: 2.0 GB Total, 0.0 B Used
- Resources in use:
- Applications: 0 Running, 0 Completed
- Drivers: 0 Running, 0 Completed
- Status: ALIVE

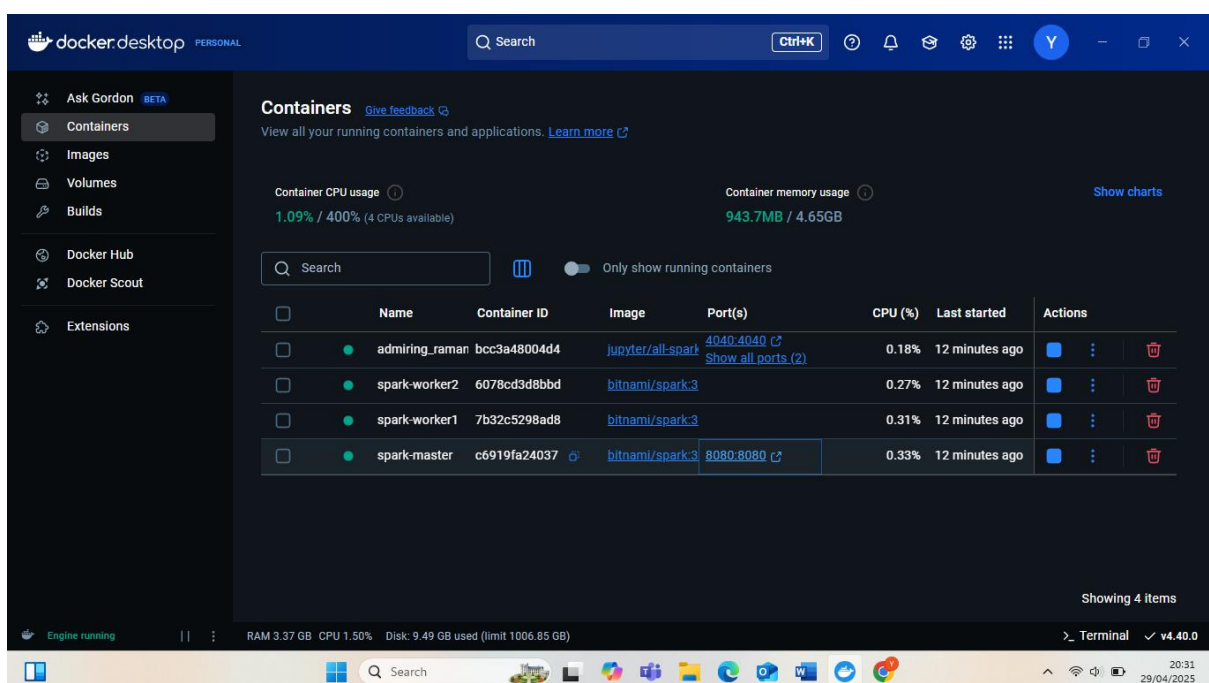
Below this information, there are three sections:

- Workers (2)**: A table showing the details of the two workers.
- Running Applications (0)**: A table showing no running applications.
- Completed Applications (0)**: A table showing no completed applications.

Worker Id	Address	State	Cores	Memory	Resources
worker-20250429131915-172.18.0.3-34509	172.18.0.3:34509	ALIVE	2 (0 Used)	1024.0 MB (0.0 B Used)	
worker-20250429131915-172.18.0.4-33965	172.18.0.4:33965	ALIVE	2 (0 Used)	1024.0 MB (0.0 B Used)	

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

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



The screenshot shows the Docker Desktop interface. The left sidebar contains the following menu items:

- Ask Gordon **BETA**
- Containers
- Images
- Volumes
- Builds
- Docker Hub
- Docker Scout
- Extensions

The main area displays the **Containers** section. It shows the following summary:

- Container CPU usage: 1.09% / 400% (4 CPUs available)
- Container memory usage: 943.7MB / 4.65GB

Below the summary, there is a table of running containers:

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	admiring_raman	bcc3a48004d4	jupyter/all-spark	4040:4040	0.18%	12 minutes ago	
<input type="checkbox"/>	spark-worker2	6078cd3d8bbd	bitnami/spark:3		0.27%	12 minutes ago	
<input type="checkbox"/>	spark-worker1	7b32c5298ad8	bitnami/spark:3		0.31%	12 minutes ago	
<input type="checkbox"/>	spark-master	c6919fa24037	bitnami/spark:3	8080:8080	0.33%	12 minutes ago	

At the bottom, it says "Showing 4 items".

## Praktikum : Membangun ETL Pipeline

### Tugas

1. **Extract**: Baca data dari file CSV (`sales_data.csv`).

## 2. Transform:

- Filter transaksi dengan Revenue > \$100.
- Hitung total penjualan per kategori.

## 3. Load: Simpan hasil ke Parquet.

### Solusi

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, sum

spark = SparkSession.builder.appName("ETLPipeline").getOrCreate()

# Extract
df = spark.read.csv("sales_data.csv", header=True, inferSchema=True)

# Transform
df_filtered = df.filter(col("Revenue") > 100)
df_result = df_filtered.groupBy("Product_Category").agg(sum("Revenue").alias("total_sales"))

df_result.show()

# Load
df_result.write.mode("overwrite").parquet("output_sales.parquet")

spark.stop()
```

### Hasil

```
+-----+
|Product_Category|total_sales|
+-----+
|      Clothing|      8198902|
|   Accessories|   13559164|
|         Bikes|   61782134|
+-----+
```

### Analisis Data Retail

#### Dataset

- **Format:** CSV (sales\_data.csv)

#### Tugas

1. Hitung total pendapatan per bulan.
2. Identifikasi 5 produk terlaris.
3. Simpan hasil dalam format Parquet.

### Solusi

1. Pendapatan perbulan

```

from pyspark.sql import SparkSession
from pyspark.sql.functions import month, sum, count

spark = SparkSession.builder.appName("ETLPipeline").getOrCreate()

df = spark.read.csv("sales_data.csv", header=True, inferSchema=True)

#pendapatan per bulan
df_revenue = df.withColumn("month", month("Date")) \
                .groupBy("month") \
                .agg(sum(df["unit_price"]*df["Order_Quantity"]).alias("total_revenue"))
df_revenue.show()

```

```

+-----+-----+
|month|total_revenue|
+-----+-----+
| 12|    10158080|
|  1|     7832338|
|  6|    10085537|
|  3|     8201790|
|  5|     9859851|
|  9|     6517880|
|  4|     8485163|
|  8|     6348349|
|  7|     6392045|
| 10|     6709394|
| 11|     6977157|
|  2|     7608734|
+-----+-----+

```

## 2. Identikasi 5 Produk terlaris

```

from pyspark.sql.functions import count

df_top_products = df.groupBy("Product") \
                    .agg(count("*").alias("total_orders")) \
                    .orderBy("total_orders", ascending=False) \
                    .limit(5)

df_top_products.show()

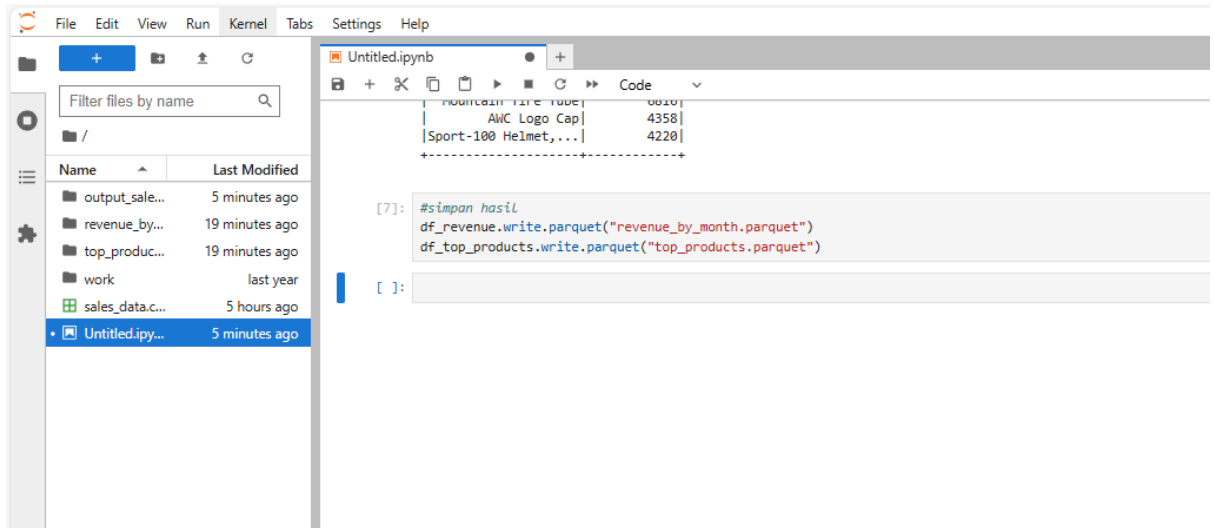
```

```

+-----+-----+
|          Product|total_orders|
+-----+-----+
|Water Bottle - 30...|      10794|
|Patch Kit/8 Patches|      10416|
|Mountain Tire Tube|       6816|
|      AWC Logo Cap|       4358|
|Sport-100 Helmet,...|       4220|
+-----+-----+

```

## 3. Simpan dalam format parquet



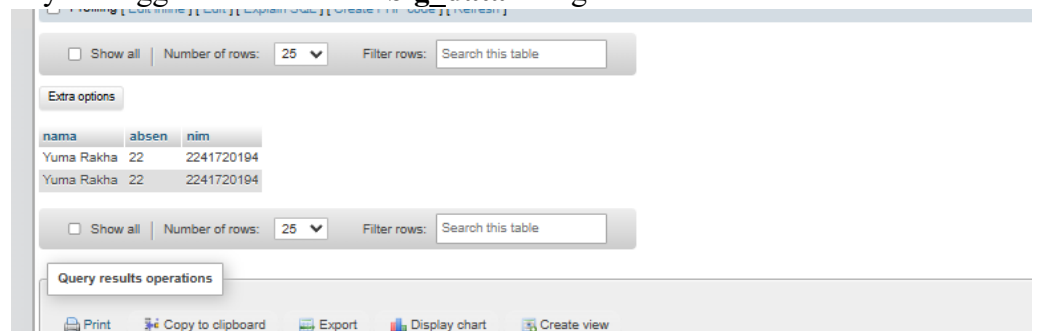
## Evaluasi

### Soal Latihan

1. Baca data dari table di database MySQL anda menggunakan Spark, dengan cara berikut

Jawab :

- a. Saya menggunakan database **big\_data** sebagai contoh



- b. Melakukan konfigurasi

```

from pyspark.sql import SparkSession

# Inisialisasi SparkSession
spark = SparkSession.builder \
    .appName("Read MySQL") \
    .config("spark.jars", "/home/jovyan/jars/mysql-connector-j-9.3.0.jar") \
    .getOrCreate()

```

- c. Output

```

+-----+-----+-----+
|      nama|absen|      nim|
+-----+-----+-----+
|Yuma Rakha |   22|2241720194|
|Yuma Rakha |   22|2241720194|
+-----+-----+-----+

```

2. Buat query Spark SQL untuk menghitung jumlah row dalam table tersebut

```
[3]: df.createOrReplaceTempView("mahasiswa")
```

```
[4]: # Query Spark SQL untuk menghitung jumlah row
jumlah_row = spark.sql("SELECT COUNT(*) AS total_mahasiswa FROM mahasiswa")
```

```
# Tampilkan hasilnya
jumlah_row.show()
```

```
+-----+
|total_mahasiswa|
+-----+
|                2|
+-----+
```