



Nama : Triyana Dewi Fatmawati
NIM : 2241720206
Kelas : TI – 3D
Nomor : 21
Mata Kuliah : Big Data

Tugas 9 – Spark SQL, DataSources, DataFrame, dan Dataset APIs

Pengerjaan:

Menyiapkan lingkungan Spark Cluster

Spark Master at spark://172.18.0.2:7077

URL: spark://172.18.0.2:7077

Active Workers: 2

Cores in use: 2 Total, 0 Used

Memory in use: 2.0 GiB Total, 0.0 B Used

Resources in use:

Applications: 0 Running, 0 Completed

Drivers: 0 Running, 0 Completed

Status: ALIVE

Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20250423080929-172.18.0.3-44499	172.18.0.3:44499	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20250423080944-172.18.0.4-36383	172.18.0.4:36383	ALIVE	1 (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
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

Docker Desktop PERSONAL

Containers [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Container CPU usage 0.72% / 2000% (20 CPUs available)

Container memory usage 770.69MB / 6.48GB

Show charts

Search

Only show running containers

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
spark-master	eb76fdea70ad	apache/spark:latest	7077:7077	0.25%	4 minutes ago	Show all ports (2)
spark-worker1	e9530508bae5	apache/spark:latest		0.18%	4 minutes ago	
spark-worker2	ee3d62fb6f29	apache/spark:latest		0.18%	3 minutes ago	
epic_kilby	cb5a78923af4	jupyter/all-spark-notebooks	4040:4040	0.01%	3 minutes ago	Show all ports (2)

Showing 5 items

Terminal

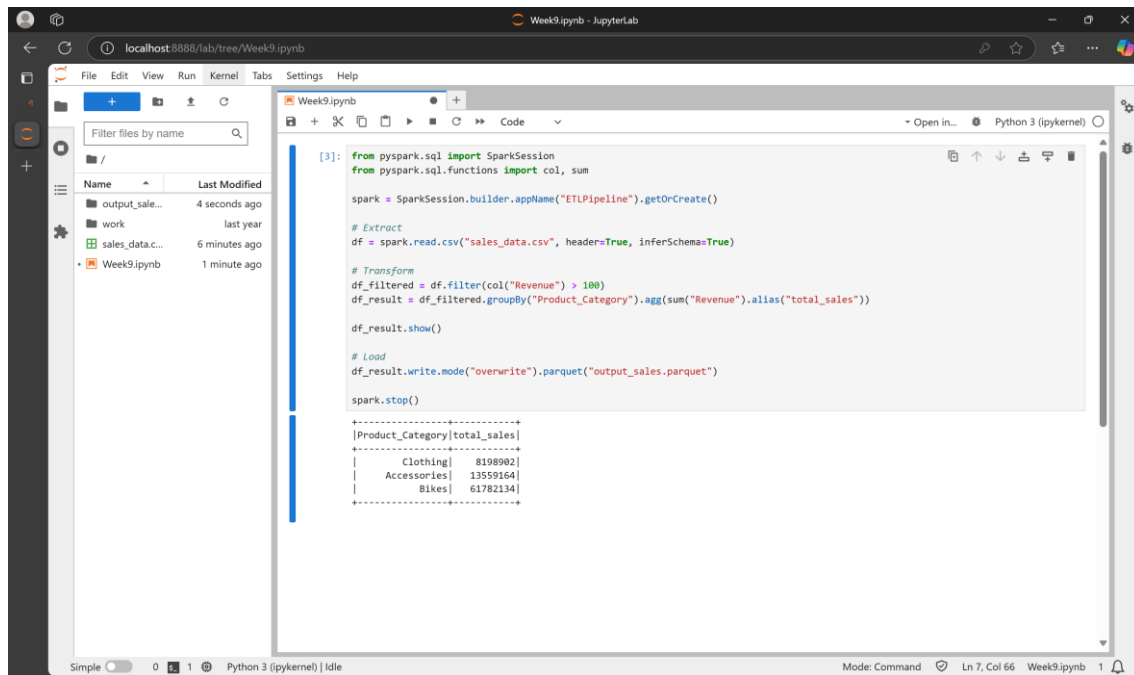
[I 2025-04-23 08:10:33.350 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-nodejs, javascript-typescript-languageserver, jedi-language-server, julia-language-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, unified-language-server, vscode-css-languagelanguage-server, vscode-html-languagelanguage-server, vscode-json-languageserver, vscode-languagelanguage-server

Engine running RAM 1.99 GB CPU 0.10% Disk 9.07 GB used (limit 1006.85 GB)

Terminal v4.40.0

Praktikum Membangun ETL Pipeline

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.



The screenshot shows a JupyterLab window titled "Week9.ipynb" with a Python 3 (ipykernel) environment. The code in the cell [3] implements an ETL pipeline:

```
[3]: 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()
```

The output of the code is a table showing the total sales for each product category:

Product_Category	total_sales
Clothing	8198902
Accessories	13559164
Bikes	61782134

Analisis Data Retail

Dataset

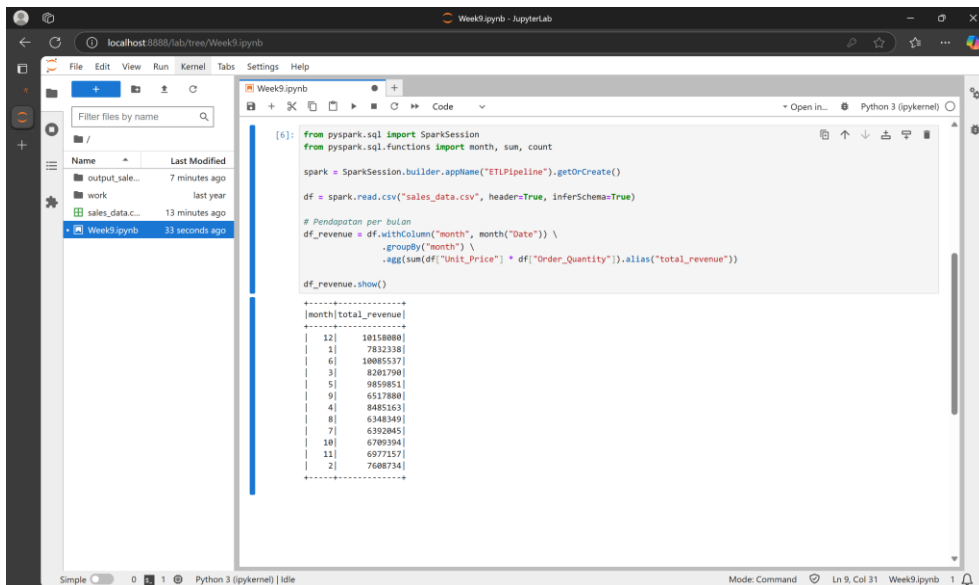
- **Format:** CSV (sales_data.csv)

Tugas

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

Pengerjaan

1. Pendapatan perbulan



```
[6]: from pyspark.sql import SparkSession
from pyspark.sql.functions import month, sum, count

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

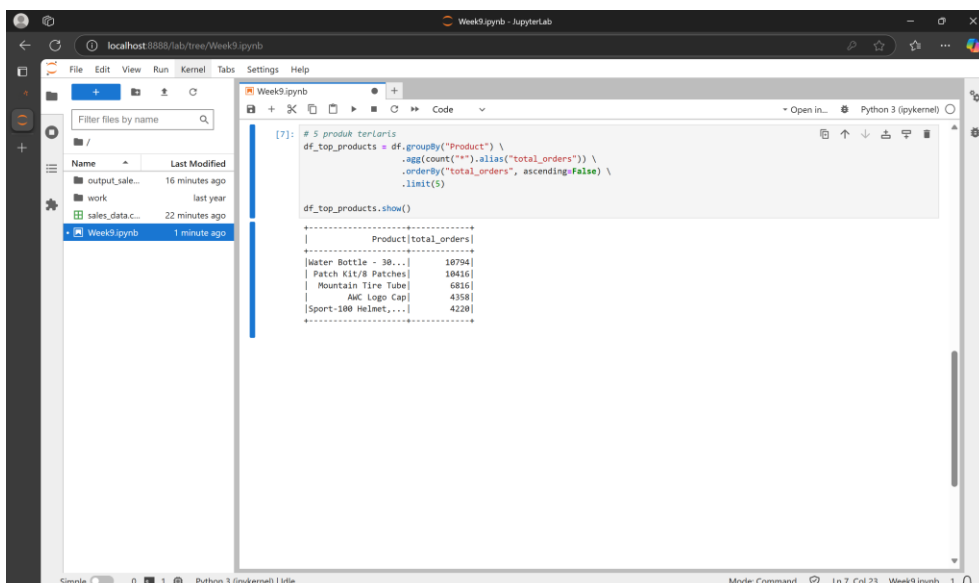
df = spark.read.csv("sales_data.csv", headers=True, inferSchema=True)

# 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| 10158000|
| 1| 7832338|
| 6| 10085537|
| 3| 8201790|
| 5| 9859851|
| 9| 6517880|
| 4| 8485163|
| 8| 6348349|
| 7| 6392945|
|10| 6709394|
|11| 6977157|
| 2| 7686754|
+-----+
```

2. Identifikasi 5 produk terlaris

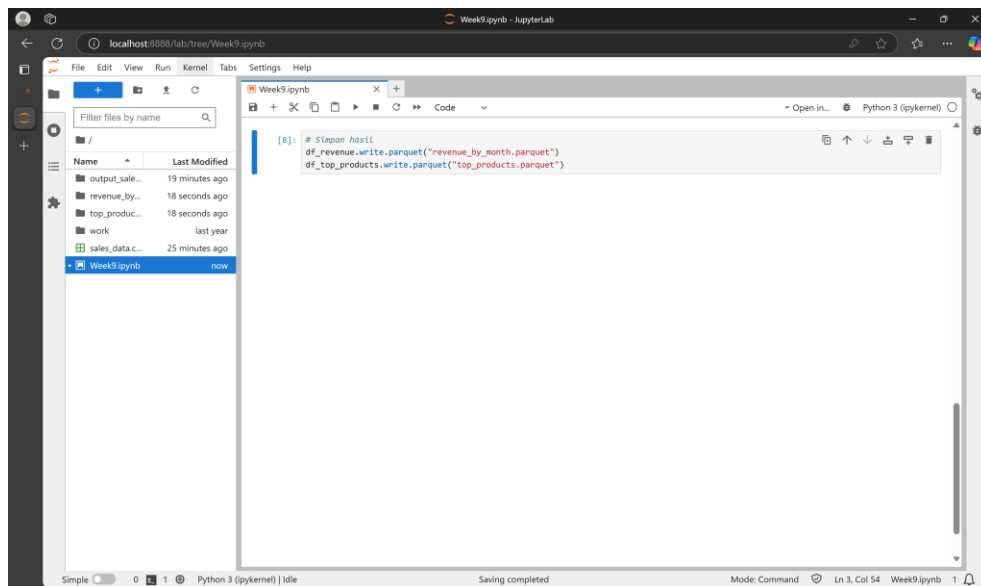


```
[7]: # 5 produk terlaris
df_top_products = df.groupBy("Product") \
    .agg(count("*").alias("total_orders")) \
    .orderBy("total_orders", ascending=False) \
    .limit(5)

df_top_products.show()

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

3. Simpan dalam format parquet



Evaluasi

Soal Latihan

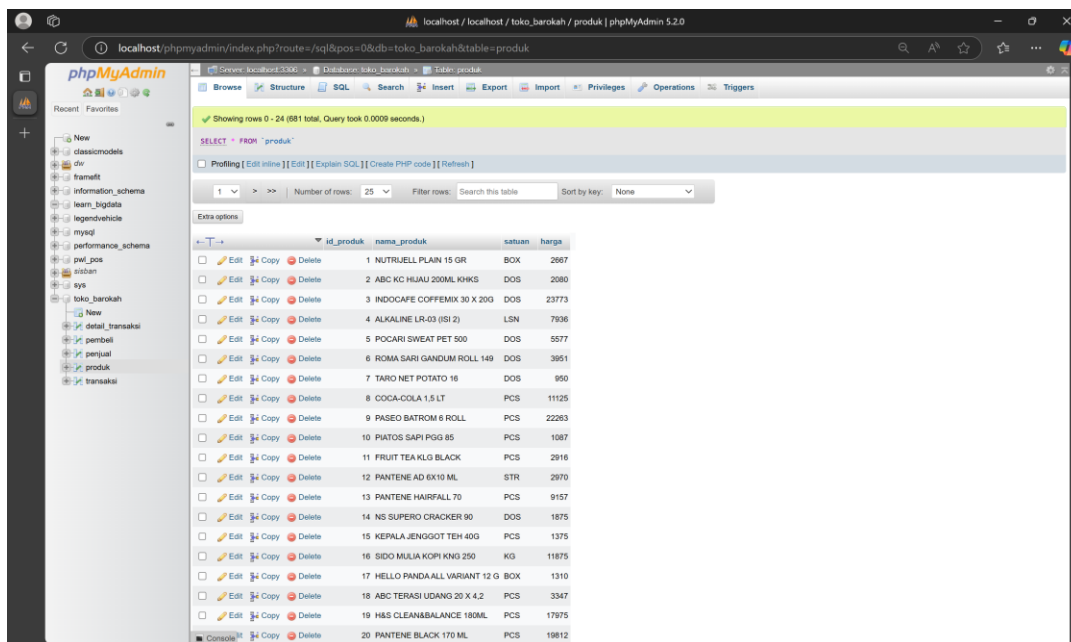
1. Baca data dari tabel di database MySQL anda menggunakan Spark, dengan cara berikut

```
df = spark.read.format("jdbc") \
    .option("url", "jdbc:mysql://localhost:3306/db") \
    .option("dbtable", "table_name") \
    .option("user", "user") \
    .option("password", "password") \
    .load()
```

Baca tabel apa saja.

2. Buat query Spark SQL untuk menghitung Jumlah row dalam tabel tersebut

Pengerjaan :



id_produk	nama_produk	satuan	harga
1	NUTRIJELL PLAIN 15 GR	BOX	2667
2	ABC KC HIJAU 200ML KHKGS	DOS	2080
3	INDOCAFE COFFEMIX 30 X 200	DOS	23773
4	ALKALINE LR-03 (ISI 2)	LSN	7936
5	POCARI SWEAT PET 500	DOS	5577
6	ROMA SARI GANDUM ROLL 149	DOS	3951
7	TARO NET POTATO 16	DOS	850
8	COCA-COLA 1.5 LT	PCS	11125
9	PASEO BATROM 6 ROLL	PCS	22263
10	PIATOS SAPI PIGG 85	PCS	1087
11	FRUIT TEA KLG BLACK	PCS	2916
12	PANTENE AD 6X10 ML	STR	2970
13	PANTENE HAIRFALL 70	PCS	9157
14	NS SUPERO CRACKER 90	DOS	1875
15	KEPALA JENGOTOT TEH 40G	PCS	1375
16	SIDO MULIA KOPRI KNG 250	KG	11875
17	HELLO PANDA ALL VARIANT 12 G	BOX	1310
18	ABC TERASI UDANG 20 X 4.2	PCS	3347
19	H&S CLEANBALANCE 180ML	PCS	17975
20	PANTENE BLACK 170 ML	PCS	19812

Pada pengerjaan soal latihan ini saya menggunakan database “toko_barokah” tabel “produk”.

1. Baca tabel

Saya membaca tabel “produk” dan menampilkan 5 produk teratas.

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

spark = SparkSession.builder \
    .appName("MySQLConnection") \
    .config("spark.jars", "mysql-connector-j-9.3.0.jar") \
    .getOrCreate()

# Membaca tabel dari MySQL
df = spark.read.format("jdbc") \
    .option("url", "jdbc:mysql://host.docker.internal:3306/toko_barokah") \
    .option("dbtable", "produk") \
    .option("user", "root") \
    .option("password", "") \
    .option("driver", "com.mysql.cj.jdbc.Driver") \
    .load()

df.show(5)
```

id_produk	nama_produk	satuan	harga
1	NUTRIJELL PLAIN 15 GR	BOX	2667.0
2	ABC KC HIJAU 200ML KHKGS	DOS	2080.0
3	INDOCAFE COFFEMIX 30 X 200	DOS	23773.0
4	ALKALINE LR-03 (ISI 2)	LSN	7936.0
5	POCARI SWEAT PET 500	DOS	5577.0

only showing top 5 rows

2. Query menghitung jumlah row dalam tabel

```
[5]: # Daftarkan DataFrame jadi table sementara
df.createOrReplaceTempView("produk")

# Query Spark SQL untuk menghitung jumlah baris
hasil = spark.sql("SELECT COUNT(*) AS jumlah_row FROM produk")

# Tampilkan hasil
hasil.show()
```

jumlah_row
681

Kesimpulan

- Spark SQL menyediakan antarmuka terstruktur untuk pemrosesan data besar.
- DataFrame & Dataset APIs memungkinkan manipulasi data dengan sintaks mirip SQL.
- DataSources API mendukung integrasi dengan berbagai format penyimpanan.