



NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA - 09 Spark SQL

Siapkan lingkungan Spark Cluster

Spark Master at spark://172.18.0.2:7077

URL: spark://172.18.0.2:7077
Alive 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-20250429181327-172.18.0.3-40039	172.18.0.3:40039	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	
worker-20250429181339-172.18.0.4-40951	172.18.0.4:40951	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
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

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. (Hasil Praktikum di bawah ini)

praktikum.ipynb

MEMBANGUN ETL PIPELINE

1. Extract: Baca data dari file CSV (sales_data.csv).
2. Transform:
 - o Filter transaksi dengan Revenue > \$100 .
 - o Hitung total penjualan per kategori.
3. Load: Simpan hasil ke Parquet.

```
[2]: # 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()
```

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



NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA - 09 Spark SQL

Analisis Data Retail

Dataset

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

sales_data.csv yesterday

Tugas

1. Hitung total pendapatan per bulan.

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.

```
[3]: # 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. Identifikasi 5 produk terlaris.

```
[4]: # Solusi 2 : 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 Patches	10416
Mountain Tire Tube	6816
AWC Logo Cap	4358
Sport-100 Helmet,...	4220



NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA - 09 Spark SQL

3. Simpan hasil dalam format Parquet.

```
[5]: # Solusi 3 : Simpan hasil
df_revenue.write.parquet("revenue_by_month.parquet")
df_top_products.write.parquet("top_products.parquet")
```

revenue_by_month.parquet	yesterday
top_products.parquet	yesterday

Evaluasi

Soal Latihan

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

DB : framefit

Tabel : kacamata

	kacamata_id	model	jenis	gender	bentuk	deskripsi	foto
<input type="checkbox"/>	1	Aviator(2)	Aviator	female	Aviator	Bingkai tipe Aviator	static/Frame/female/Aviator/Aviator(2).png
<input type="checkbox"/>	2	Aviator(3)	Aviator	female	Aviator	Bingkai tipe Aviator	static/Frame/female/Aviator/Aviator(3).png
<input type="checkbox"/>	3	Butterfly(1)	Butterfly	female	Butterfly	Bingkai tipe Butterfly	static/Frame/female/Butterfly/Butterfly(1).png
<input type="checkbox"/>	4	Butterfly(2)	Butterfly	female	Butterfly	Bingkai tipe Butterfly	static/Frame/female/Butterfly/Butterfly(2).png
<input type="checkbox"/>	5	Butterfly(3)	Butterfly	female	Butterfly	Bingkai tipe Butterfly	static/Frame/female/Butterfly/Butterfly(3).png

- Konfigurasi JDBC, download file mysql-connector-j-9.3.0.jar. Kemudian file mysql-connector-j-9.3.0.jar di copy ke spark-notebook dengan perintah docker cp mysql-connector-j-9.3.0.jar beautiful_bassi:/usr/share/java/.
- Dilanjutkan dengan menjalankan kode program seperti dibawah

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

spark = SparkSession.builder \
    .appName("Read MySQL Table") \
    .config("spark.jars", "/usr/share/java/mysql-connector-j-9.3.0.jar") \
    .getOrCreate()

df = spark.read.format("jdbc") \
    .option("url", "jdbc:mysql://host.docker.internal:3306/framefit") \
    .option("dbtable", "kacamata") \
    .option("user", "root") \
    .option("password", "") \
    .option("driver", "com.mysql.cj.jdbc.Driver") \
    .load()

df.show(5)
```

kacamata_id	model	jenis	gender	bentuk	deskripsi	foto
1	Aviator(2)	Aviator	female	Aviator	Bingkai tipe Aviator	static/Frame/female/Aviator/Aviator(2).png
2	Aviator(3)	Aviator	female	Aviator	Bingkai tipe Aviator	static/Frame/female/Aviator/Aviator(3).png
3	Butterfly(1)	Butterfly	female	Butterfly	Bingkai tipe Butterfly	static/Frame/female/Butterfly/Butterfly(1).png
4	Butterfly(2)	Butterfly	female	Butterfly	Bingkai tipe Butterfly	static/Frame/female/Butterfly/Butterfly(2).png
5	Butterfly(3)	Butterfly	female	Butterfly	Bingkai tipe Butterfly	static/Frame/female/Butterfly/Butterfly(3).png

only showing top 5 rows



NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA - 09 Spark SQL

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

```
df.createOrReplaceTempView("kacamata")  
  
jumlah_row = spark.sql("SELECT COUNT(*) AS jumlah_row FROM kacamata").show()
```

```
+-----+  
|jumlah_row|  
+-----+  
|         62|  
+-----+
```

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.