Charlie's Angels Team

Learning Progress Review

WEEK 8

DigitalSkola

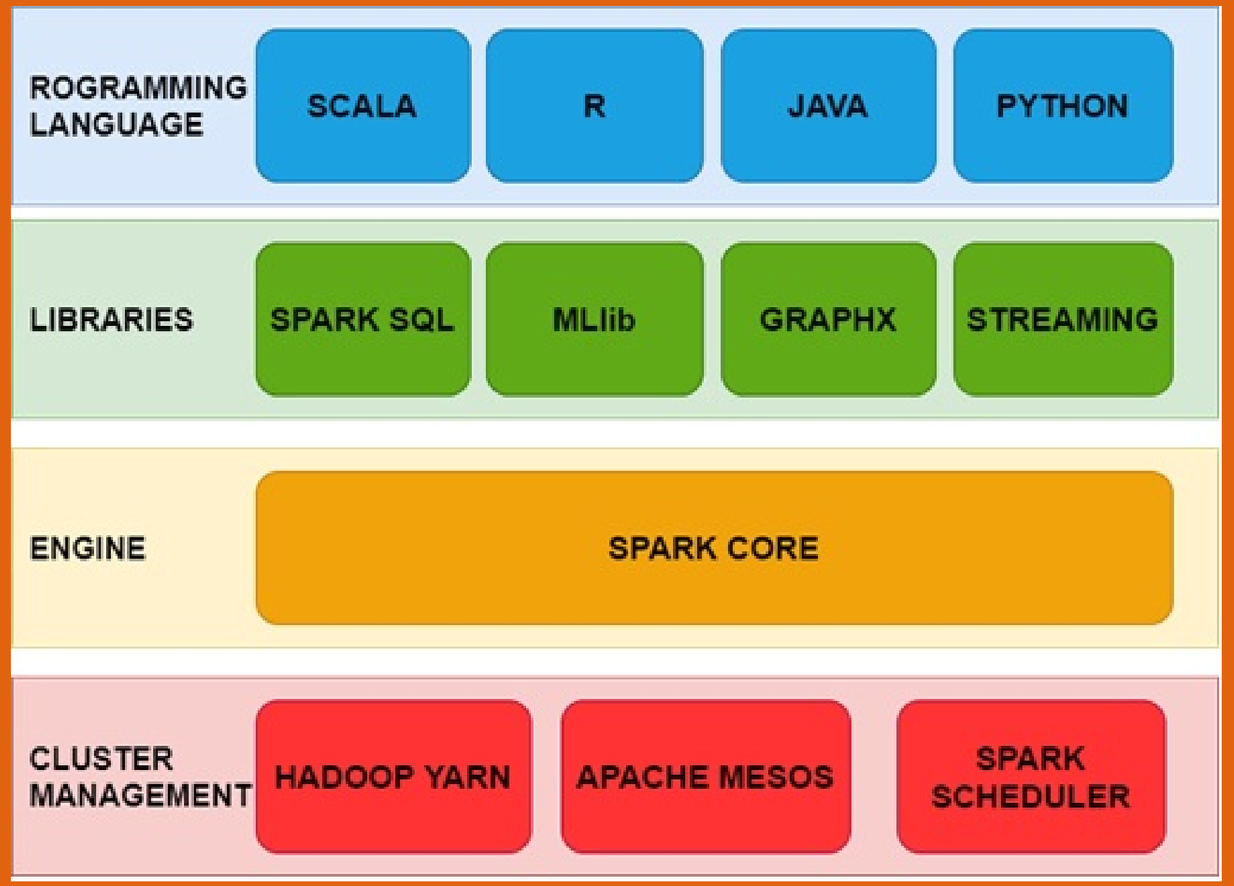
APACHE SPARK

Apache Spark adalah framework yang digunakan untuk memproses, menanyakan, dan menganalisis Big Data. Apache Spark melakukan pemrosesan data melalui in-memory, sehingga waktu pemrosesan lebih cepat daripada framework sejenis seperti MapReduce dan lainnya.

FITUR DAN KOMPONEN PADA APACHE SPARK

FITUR

1. Performa lebih cepat dibandingkan framework pemrosesan data tradisional.
2. Mudah digunakan, aplikasi pengolahan data yang dibangun dengan Spark dapat dituliskan dalam bahasa pemrograman Python, R, Java, dan Scala.
3. Dilengkapi dengan SQL Library, Streaming, dan Graph Analysis yang memudahkan proses pengolahan dan analisis data.



Gambar 1.Komponen *Apache Spark*.

DigitalSkola

SPARK CLUSTER MANAGERS

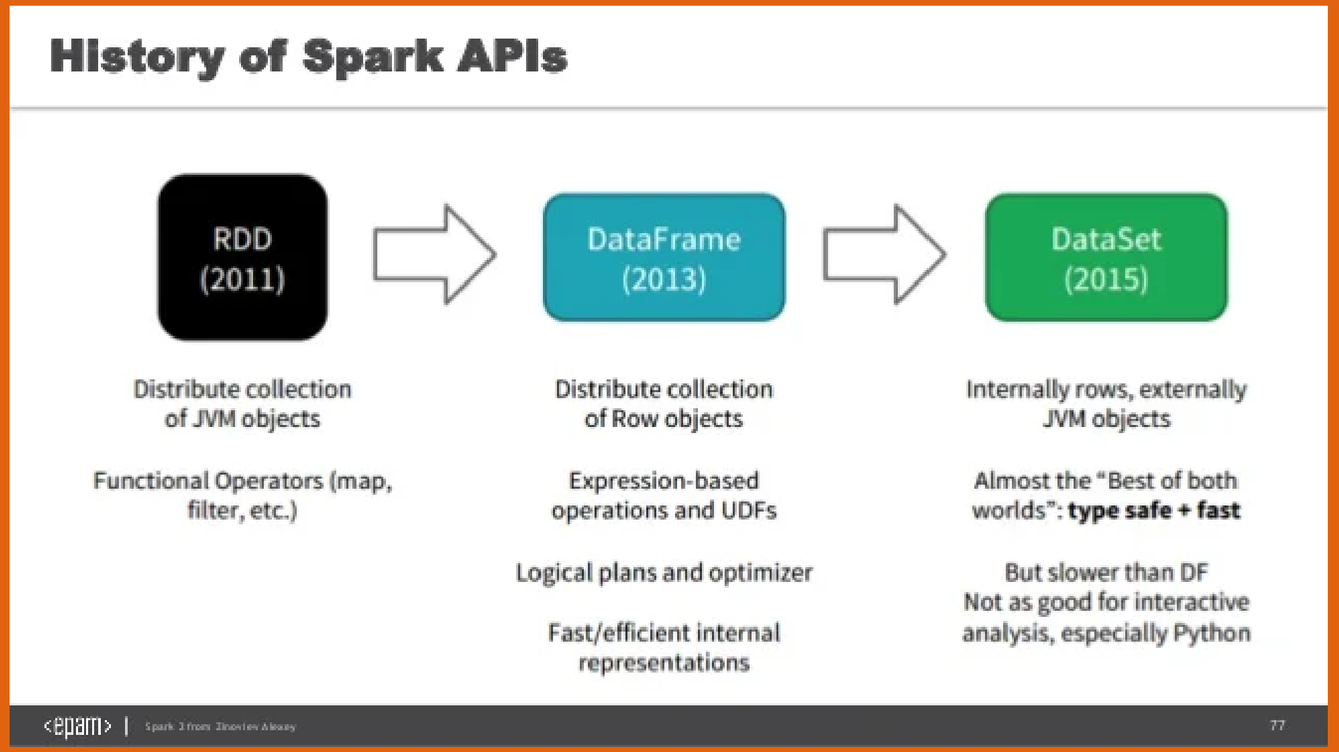
Spark Cluster Managers merupakan komponen Spark yang mendukung pengelolaan sumber daya/cluster berikut:

1. Spark Standalone – Cluster Manager sederhana yang disertakan dengan Spark untuk memudahkan pengaturan cluster.
2. Apache Mesos – Cluster Manager yang lebih umum yang dapat menjalankan environment lainnya seperti Hadoop
3. Apache Hadoop YARN – Resource manager untuk mengelola environment Hadoop
4. Kubernetes – sistem open source yang memanajemen berbagai macam aplikasi dalam bentuk kontainer.

SPARK STRUCTURED API

Structured API adalah tools untuk memanipulasi semua jenis data, dari file log tidak terstruktur hingga file CSV semi-terstruktur dan file Parket yang sangat terstruktur. API ini merujuk pada tiga tipe inti dari API kumpulan terdistribusi:

- Datasets
- DataFrames
- SQL tables and views



Gambar 2. Sejarah *Spark APIs*.

DigitalSkola

LAZY EVALUATION

Di map reduce banyak waktu pemborosan pengembang dalam meminimalkan jumlah map reduce. Itu terjadi dengan menggabungkan operasi bersama. Sementara di Spark kami tidak membuat grafik eksekusi tunggal, melainkan kami menggabungkan banyak operasi sederhana.
Dengan demikian itu menciptakan perbedaan antara Hadoop MapReduce vs Apache Spark.
Di Spark, program driver memuat kode ke cluster. Ketika kode dijalankan setelah setiap operasi, tugas akan memakan waktu dan memori. Karena setiap kali data masuk ke cluster untuk evaluasi.

KEUNTUNGAN

- Meningkatkan Pengelolaan : pengguna dapat mengatur program Aphace menjadi operasi yang kecil
- Menghemat Komputasi dan meningkatkan Kecepatan : park Lazy Evaluation memainkan peran kunci dalam menghemat overhead kalkulasi. Karena hanya nilai yang diperlukan yang mendapatkan komputasi. Ini menghemat perjalanan antara driver dan cluster, sehingga
- Mengurangi Kompleksitas : Dua kompleksitas utama dari setiap operasi adalah kompleksitas waktu dan ruang
- Optimasi : Ini memberikan pengoptimalan dengan mengurangi jumlah queri

SPARK ACTIONS

- Action pada Spark merupakan trigger untuk melakukan dan meninstruksikan Spark untuk melakukan komputasi dari sebuah proses transformasi
- Spark Actions:
  - count() : mengitung jumlah elemen pada RDD/DataFrame
  - collect() : return semua elemen data sebagai array ke driver program. Biasanya berguna setelah filtering data
  - ttake(n) : return array sejumlah n pertama dari DataFrame
  - top(n) : extract top n elements dari dataframe
  - countbyvalue(): meanmpilkan berapa kali data pada sebuah dataframe muncul
  - reduce() : menggabungkan 2 atau lebih elemen dataframe dengan menggunakan fungsi tambahan dan return hasilnya dengan tipe data yang sama seperti tipe data inputnya.
  - foreach() : untuk mengaplikasikan function/operation pada setiap element dataframe tanpa melakukan return value ke driver.

STRUCTURED API

- Tool yang digunakan untuk memanipulasi berbagai macam data. Dari data yang bersifat unstructured seperti log, semi structured seperti CSV file, sampai data structured tingkat tinggi seperti Parquet file
- API tersebut mengacu pada 3 tipe distributed collection berikut:
  - Datasets
  - DataFrames
  - SQL tables and views

SCHEMA

- Merupakan structure dari DataFrame atau Dataset
- Define dengan menggunakan StructType Class yang merupakan collection dari StructField
- StructField akan mendefinisikan column name (String), column type (DataType), nullable column (Boolean), dan metadata (MetaData)

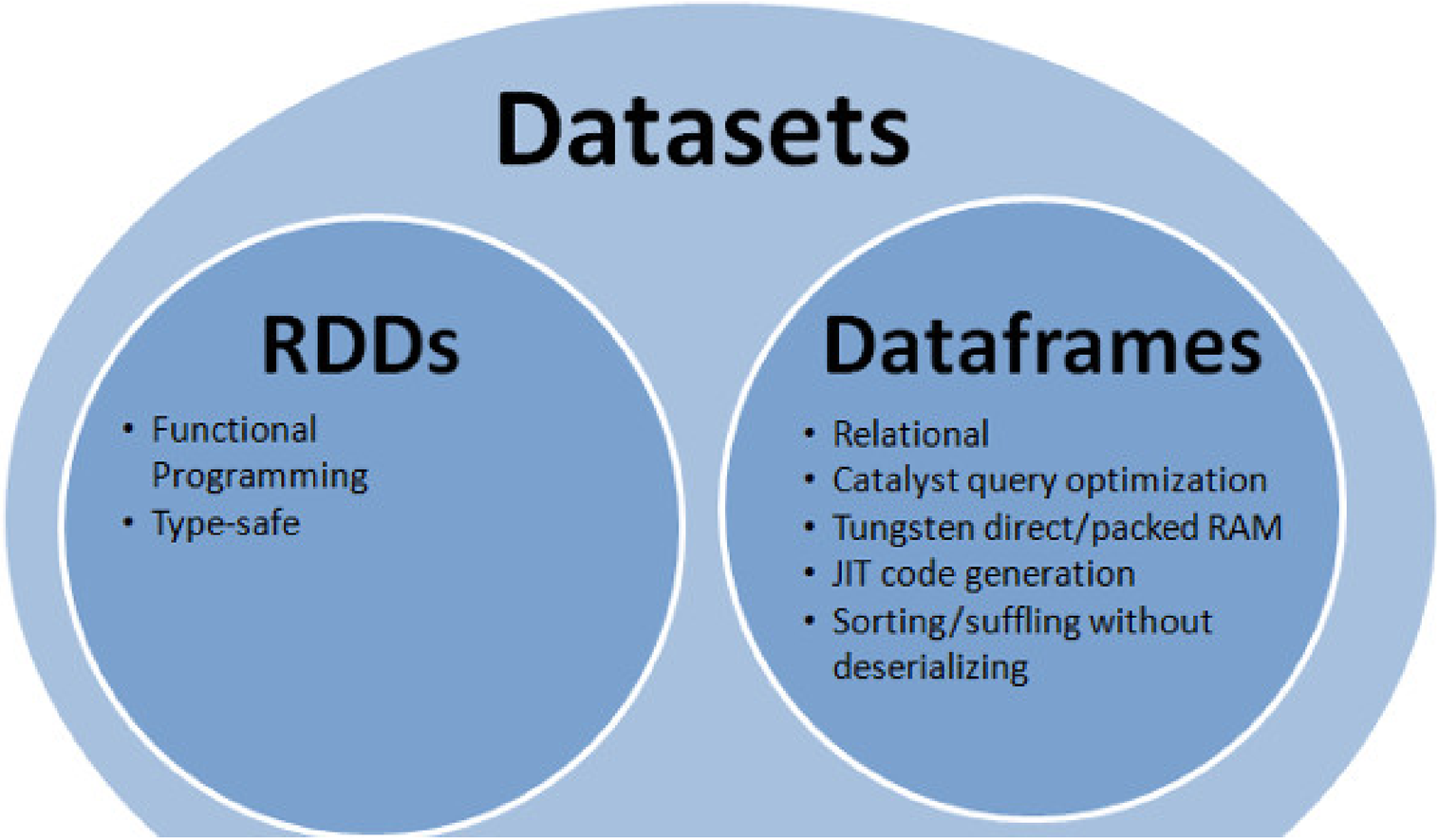
STRUCTURED SPARK TYPES

- Spark menggunakan engine yang disebut “Catalyst”
- Catalyst akan melakukan pengelolaan tipe informasinya sendiri
- Spark memiliki lookup table untuk setiap Bahasa pemrograman (Scala, Java, Python, SQL, R)
- Manipulasi data dilakukan pada Spark types, bukan Python types

DATAFRAMES VS DATASETS

- DataFrames dan Datasets berbentuk seperti table dengan data pada column yang terdefine dengan baik
- DataFrame dan Dataset berbentuk Row
- Row merupakan tipe data representasi internal Spark dengan format yang telah dioptimasi untuk komputasi
- Pada Spark Python dan Scala hanya memiliki DataFrame

DATAFRAMES VS DATASETS



Sources: www.linkedin.com/pulse/apache-spark-rdd-vs-dataframe-dataset-chandan-prakash/

COLUMN

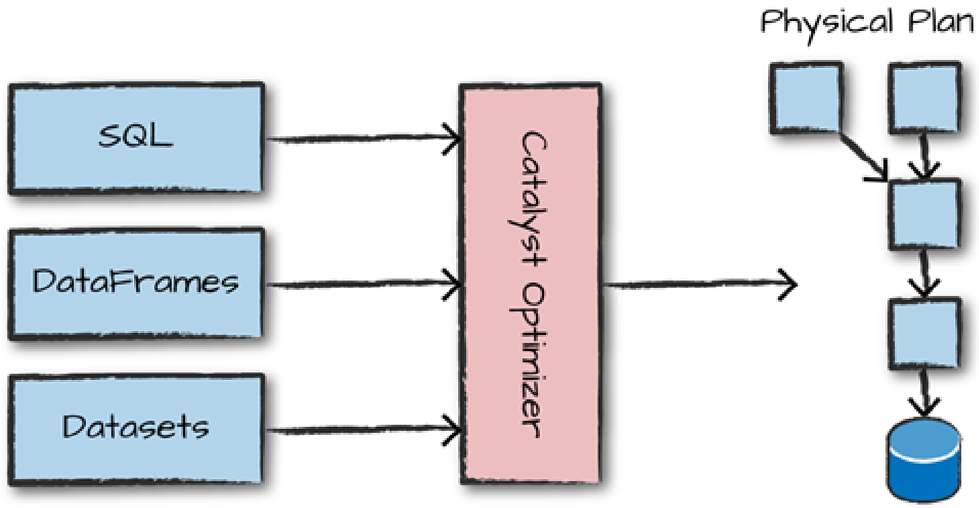
- Column pada Spark dapat dianalogikan seperti column pada table
- Merepresentasikan simple type (contoh: integer/string), complex type (array, map), atau nilai kosong (null)
- Merupakan generator value untuk setiap row pada dataset

ROW

- Fields collection yang berurutan dimulai dari index 0
- Merupakan record data, record pada DataFrame harus bertipe Row
- Row dapat dibuat secara manual dari SQL, RDD, data sources, dan lain-lain.

STRUCTURED API EXECUTION

- Menggambarkan bagaimana sebuah code dieksekusi pada cluster
  - Menulis code -> submit code ke Spark
  - Code yang disubmit akan diubah menjadi logical plan yang bersifat unresolved
  - Unresolved logical plan kemudian akan dianalisis oleh analyzer dan diubah menjadi resolved logical plan
  - Resolved logical plan akan diteruskan melalui Catalyst Optimizer (menentukan bagaimana code akan dieksekusi secara optimal)
  - Spark akan mengeksekusi pyshical plan pada cluster



Sources: www.oreilly.com/library/view/spark-the-definitive/9781491912201/assets/spdg_0401.png

STRUCTURED OPERATIONS

- Infer schema pada Spark dapat menentukan schema secara otomatis
- Untuk menghindari masalah precision data type pada ETL, Schema sebaiknya didefine secara manual

SPARKSQL DAN DATA FRAME

Spark SQL merupakan modul Spark yang dirancang untuk memproses data secara struktural yang merupakan bagian dari Apache Spark

Spark SQL memiliki struktur utama berupa data frame yang merupakan kumpulan RDD dari Row. RDD ini dirancang untuk mendukung penyimpanan data dalam memori serta didistribusikan di seluruh cluster dengan cara yang efisien

SPARK SQL VS HIVE

Pada mulanya, Spark SQL dibuat sebagai Apache Hive uang berguna untuk menjalankan Spark. Hive merupakan infrastruktur data warehouse berbasis Hadoop, yang mana konsep Hive sendiri hampir mirip dengan basis data relasional

Perbedaan antara Spark SQL dengan Hive adalah Spark SQL tidak mendukung beberapa perintah dalam SQL seperti SELECT TOP, ROWNUM, INSERT INTO, UPDATE, DELETE CONSTRAINTS, INDEX sedangkan Hive hampir mendukung semua perintah dalam SQL kecuali TOP dan ROWNUM

KEGUNAAN SPARK SQL

1. Sebagai sumber data frame API, kumpulan pustaka untuk bekerja dengan tabel data
2. Dataframe API membantu menemukan frame data yang berisi baris dan kolom
3. Catalyst Optimizer

FITUR SPARK SQL

- Terintegrasi dengan program Spark yang membuat user dapat meminta data terstruktur dari program-program Spark menggunakan SQL atau DataFrame API dan dapat digunakan untuk Java, Scala, Python, dan R.
- Mendukung dalam mengakses berbagai sumber data serta membantu menggabungkan data untuk mengakomodasi keperluan pengguna.

FITUR SPARK SQL

- Kompatibel dengan Hive dengan menulis ulang frontend dari Hive dan meta store.
- Mampu memberi koneksi melalui JDBC atau ODBC yaitu industri dengan konektivitas untuk alat bisnis intelijen.
- Memiliki fungsi UDF (User-Defined Functions) yang saling terintegrasi

EKSEKUSI COMMAND SQL PADA DATAFRAME

```
# install java
!apt-get install openjdk-8-jdk-headless -qq > /dev/null

# install spark (change the version number if needed)
!wget -q https://archive.apache.org/dist/spark/spark-3.0.0/spark-3.0.0-bin-hadoop3.2.tgz

# unzip the spark file to the current folder
!tar xf spark-3.0.0-bin-hadoop3.2.tgz

# set your spark folder to your system path environment.
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-8-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.0.0-bin-hadoop3.2"

# install findspark using pip
!pip install -q findspark
!pip install pyspark
!pip install findspark
```

DigitalSkola

EKSEKUSI COMMAND SQL PADA DATAFRAME

```
import findspark
findspark.init()
from pyspark.sql import SparkSession
from pyspark.sql import functions as func
from pyspark.sql.types import StructType, StructField, StringType, IntegerType, FloatType, TimestampType
```

```
spark = SparkSession.builder.appName("MinAmountCustomer").getOrCreate()
```

DigitalSkola

EKSEKUSI COMMAND SQL PADA DATAFRAME

```
schema = StructType([ \
                     StructField("InvoiceNo", IntegerType(), True), \
                     StructField("StockCode", StringType(), True), \
                     StructField("Description", StringType(), True), \
                     StructField("Quantity", IntegerType(), True), \
                     StructField("InvoiceDate", TimestampType(), True), \
                     StructField("Amount", FloatType(), True), \
                     StructField("CustomerID", FloatType(), True), \
                     StructField("Country", StringType(), True)])
df = spark.read.schema(schema).csv("/content/sample_data/retail-data-full.csv", sep=";")
df.printSchema()
```

DigitalSkola

EKSEKUSI COMMAND SQL PADA DATAFRAME

```
df.show(truncate=False)
```

```
+---------+---------+-----------------------------------+--------+-------------------+------+----------+----------------+
|InvoiceNo|StockCode|Description                        |Quantity|InvoiceDate        |Amount|CustomerID|Country         |
+---------+---------+-----------------------------------+--------+-------------------+------+----------+----------------+
|536365   |85123A   |WHITE HANGING HEART T-LIGHT HOLDER |6       |2010-12-01 08:26:00|2.55  |17850.0   |United Kingdom,,|
|536365   |71053    |WHITE METAL LANTERN                |6       |2010-12-01 08:26:00|3.39  |17850.0   |United Kingdom,,|
|536365   |84406B   |CREAM CUPID HEARTS COAT HANGER     |8       |2010-12-01 08:26:00|2.75  |17850.0   |United Kingdom,,|
|536365   |84029G   |KNITTED UNION FLAG HOT WATER BOTTLE|6       |2010-12-01 08:26:00|3.39  |17850.0   |United Kingdom,,|
|536365   |84029E   |RED WOOLLY HOTTIE WHITE HEART.     |6       |2010-12-01 08:26:00|3.39  |17850.0   |United Kingdom,,|
|536365   |22752    |SET 7 BABUSHKA NESTING BOXES       |2       |2010-12-01 08:26:00|7.65  |17850.0   |United Kingdom,,|
|536365   |21730    |GLASS STAR FROSTED T-LIGHT HOLDER  |6       |2010-12-01 08:26:00|4.25  |17850.0   |United Kingdom,,|
|536366   |22633    |HAND WARMER UNION JACK             |6       |2010-12-01 08:28:00|1.85  |17850.0   |United Kingdom,,|
|536366   |22632    |HAND WARMER RED POLKA DOT          |6       |2010-12-01 08:28:00|1.85  |17850.0   |United Kingdom,,|
|536367   |84879    |ASSORTED COLOUR BIRD ORNAMENT      |32      |2010-12-01 08:34:00|1.69  |13047.0   |United Kingdom,,|
|536367   |22745    |POPPY'S PLAYHOUSE BEDROOM          |6       |2010-12-01 08:34:00|2.1   |13047.0   |United Kingdom,,|
|536367   |22748    |POPPY'S PLAYHOUSE KITCHEN          |6       |2010-12-01 08:34:00|2.1   |13047.0   |United Kingdom,,|
|536367   |22749    |FELTCRAFT PRINCESS CHARLOTTE DOLL  |8       |2010-12-01 08:34:00|3.75  |13047.0   |United Kingdom,,|
|536367   |22310    |IVORY KNITTED MUG COSY             |6       |2010-12-01 08:34:00|1.65  |13047.0   |United Kingdom,,|
|536367   |84969    |BOX OF 6 ASSORTED COLOUR TEASPOONS |6       |2010-12-01 08:34:00|4.25  |13047.0   |United Kingdom,,|
|536367   |22623    |BOX OF VINTAGE JIGSAW BLOCKS       |3       |2010-12-01 08:34:00|4.95  |13047.0   |United Kingdom,,|
|536367   |22622    |BOX OF VINTAGE ALPHABET BLOCKS     |2       |2010-12-01 08:34:00|9.95  |13047.0   |United Kingdom,,|
|536367   |21754    |HOME BUILDING BLOCK WORD           |3       |2010-12-01 08:34:00|5.95  |13047.0   |United Kingdom,,|
|536367   |21755    |LOVE BUILDING BLOCK WORD           |3       |2010-12-01 08:34:00|5.95  |13047.0   |United Kingdom,,|
|536367   |21777    |RECIPE BOX WITH METAL HEART        |4       |2010-12-01 08:34:00|7.95  |13047.0   |United Kingdom,,|
+---------+---------+-----------------------------------+--------+-------------------+------+----------+----------------+
only showing top 20 rows
```

DigitalSkola

EKSEKUSI COMMAND SQL PADA DATAFRAME

```
results = df.groupBy("CustomerID").min("Amount")
```

```
results.show()
```

```
+----------+-----------+
|CustomerID|min(Amount)|
+----------+-----------+
|   17951.0|        2.1|
|   17548.0|       0.29|
|   12748.0|       4.95|
|   16029.0|       1.25|
|   15862.0|       0.55|
|   15012.0|       0.42|
|   18144.0|       1.25|
|   14911.0|       0.65|
|   17850.0|       1.06|
|   17069.0|       0.85|
|   18074.0|       0.65|
|   12868.0|       0.65|
|   16218.0|       0.55|
|   18085.0|        2.1|
|   18011.0|       0.19|
|      null|        0.0|
|   17968.0|       0.29|
|   17897.0|       0.42|
|   15485.0|       0.42|
|   13093.0|       0.85|
+----------+-----------+
only showing top 20 rows
```

DigitalSkola

MENGGUNAKAN DATAFRAME DIBANDINGKAN RDD

Faktor yang mendukung penggunaan DataFrame dibanding RDD

- Meningkatnya trend penggunaan DataFrame dibanding RDD pada Spark
- DataFrame memungkinkan interoperabiitas yang lebih baik dibanding RDD
- DataFrame lebih mudah untuk dikembangkan (banyak operasi-operasi SQL yang dapat dieksekusi dalam single line code)

TERIMA KASIH!

Sampai jumpa di Learning Progress Review kami berikutnya!

- Hilda Meiranita Prastika Dewi
- Nur Indrasari
- Rezha Sulvian
- Thasha Dinya Ainsha

DigitalSkola