BODHA
DHARMAJA ARYADHANA

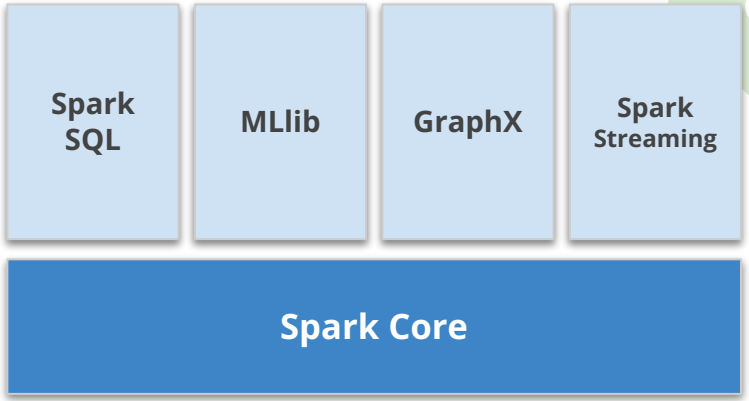
Apache Spark Data Processing

DATALEARNS24/7

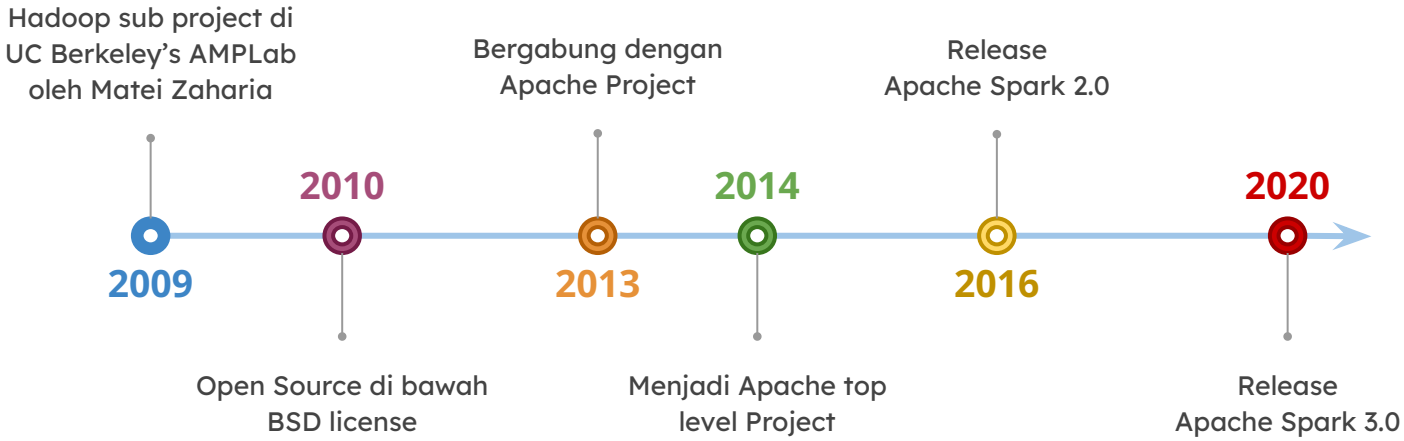
01 Overview

Apa itu Apache Spark ?

- Apache Spark™ adalah mesin multibahasa untuk data engineering, data science, dan machine learning, menggunakan node tunggal ataupun klaster.
- Fitur utama Spark adalah in-memory cluster computing.
- Mesin komputasi Spark disebut Spark Core, di atasnya dibangun berbagai library untuk SQL, Machine Learning, Streaming dan komputasi Graf.



Sejarah Apache Spark



Mengapa Apache Spark?

Fast Processing

Mencapai 100x lebih cepat dari Hadoop dengan in-memory computing, dan 10x lebih cepat dengan disk.

Fault Tolerant

Menggunakan **RDD (Resilient Distributed Dataset)** yang didesain untuk menangani kegagalan node dalam cluster.

Flexible

Mendukung berbagai bahasa pemrograman populer : Java, SQL, Python, dan R.

Unified Engine

Mendukung pemrosesan data batch, graph, streaming, dan query interaktif dalam satu mesin.

Extensible

Mendukung berbagai jenis data store, termasuk HDFS, Cassandra, HBase, MongoDB, Hive, RDBMS, dll.

Wide Support

Lebih dari 2000 kontributor. Digunakan oleh banyak perusahaan, termasuk 80% perusahaan Fortune 500.

02 Bagaimana Spark Bekerja ?

Spark Execution Mode

Spark menggunakan arsitektur master/slave. Setiap Spark program memiliki satu koordinator pusat disebut **driver** yang berkomunikasi dengan banyak worker yang terdistribusi (**executor**).

Spark memiliki 3 mode eksekusi

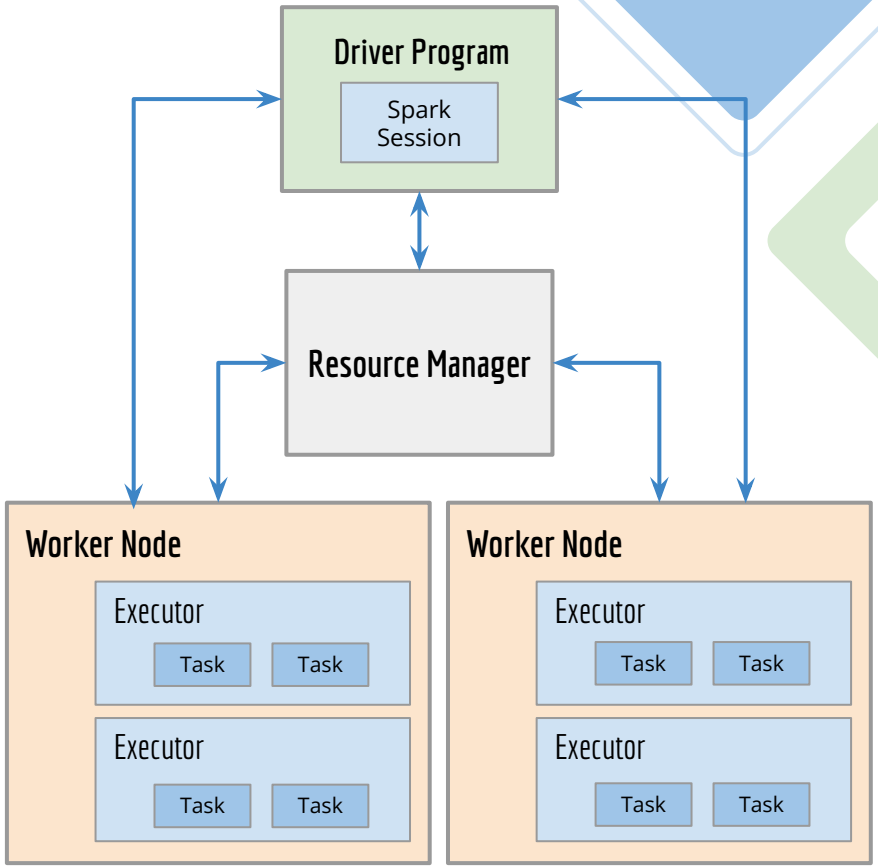
- Local
  Eksekusi program secara lokal, di notebook atau PC. Tidak menggunakan cluster yang terdistribusi

- Client mode
  Spark driver berada pada client di luar cluster, executor berada pada worker node

- Cluster mode
  Spark driver dan work berada pada node yang ada dalam cluster

Arsitektur Spark

- Setiap aplikasi Spark memiliki satu koordinator pusat disebut **Driver**
- Driver membuat object **Spark Session** untuk berkomunikasi dengan **Resource Manager**
- Resource Manager mengalokasikan **Executor**, yaitu proses yang menjalankan komputasi dan menyimpan data untuk aplikasi
- Spark Session mengirimkan task untuk dieksekusi oleh executor
- Spark dapat digunakan dengan berbagai Resource Manager, diantaranya **Spark**, **YARN, Mesos,** dan **Kubernetes**,



03 Spark API

Spark APIs

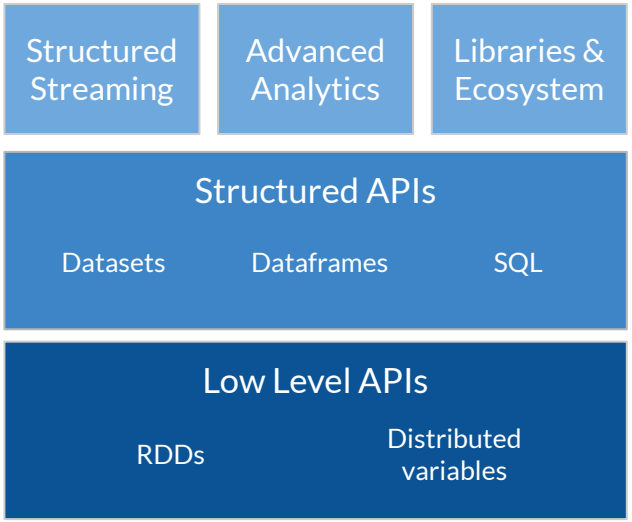
Spark memiliki dua set API dasar:

- API level rendah yang “tidak terstruktur”, yaitu :

  RDD dan **Distributed variables**

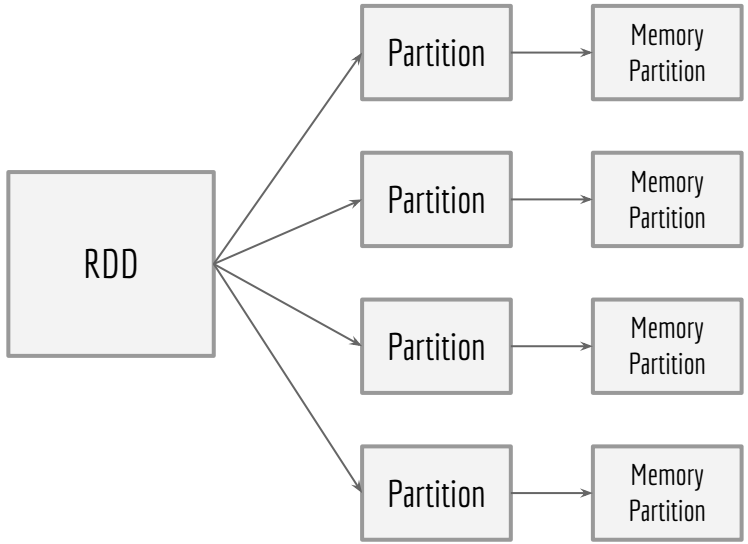
- API level lebih tinggi yang lebih terstruktur, yaitu :

  Dataset, **DataFrame** dan **SQL APIs**



Spark RDD

- Spark distributed data abstraction
- Immutable & resilient
- Lazy evaluation



Immutability

- Immutable berarti sekali dibentuk, RDD tidak bisa diubah
- Ketika RDD diubah, Spark membuat RDD baru dan menyimpan RDD asli beserta proses transformasinya
- Bisa direkonstruksi kapanpun diperlukan, termasuk jika terjadi kegagalan proses → **Resilient**
- Apakah hal ini tidak menyebabkan pemborosan resource?

Spark Transformations dan Actions

Spark Operations =

Fungsi yang menerima RDD sebagai input dan menghasilkan satu atau beberapa RDD baru dengan menerapkan operasi yang diwakilinya

TRANSFORMATIONS

filter(), map(), join(), distinct(), groupByKey(), etc.

+

ACTIONS

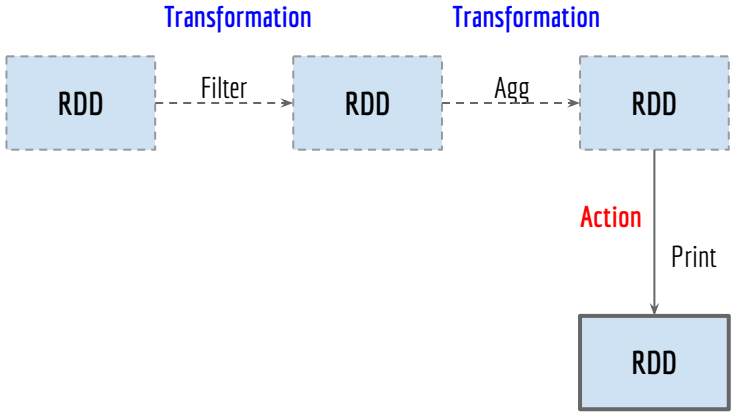
show(), collect(), count(), first(), etc.

Fungsi yang mengakses data di RDD.
Action mentrigger eksekusi semua transformasi terkait untuk mendapatkan data yang diperlukan.

https://training.databricks.com/visualapi.pdf

Lazy Evaluation

- Spark menunda eksekusi proses sampai benar-benar dibutuhkan
- Setiap operasi **Transformasi** terhadap RDD tidak langsung dieksekusi sampai menemukan operasi **Action**
- Dengan metode ini Spark dapat melakukan optimasi terhadap rangkaian proses yang akan dieksekusi



Spark Structured APIs

- DataFrames

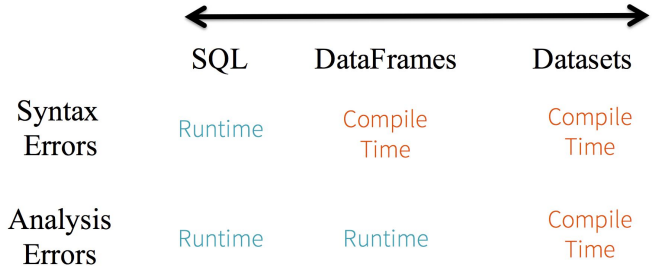
  Secara sederhana, DataFrame menggambarkan tabel dengan baris dan kolom. Konsepnya mirip dengan DataFrame R dan Python (Pandas). Konversi Spark DataFrame ke R atau Python dapat dilakukan dengan mudah.

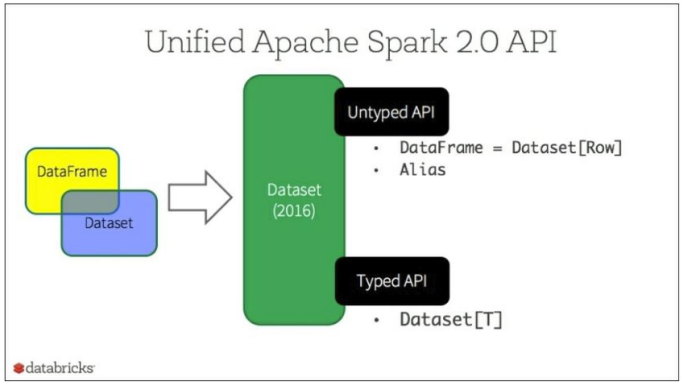
- Datasets

  Sama dengan DataFrame, DataSet memiliki struktur kolom dan baris. Bedanya, elemen Dataset merupakan objek yang *strongly-typed*.

- SQL tables and views

  Note : Sejak Spark 2.0 Dataset dan DataFrame disatukan dalam satu API. DataFrame dianggap sebagai Dataset yang bertipe Row (untyped). Untuk implementasi di pyspark hanya ada DataFrame saja.

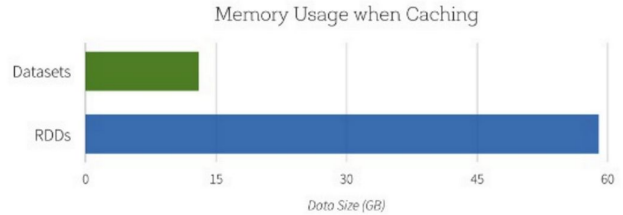


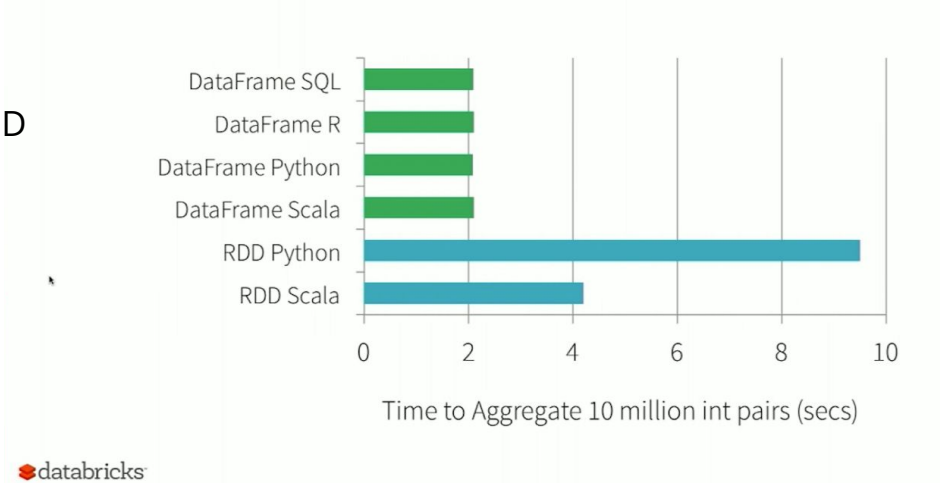


Mengapa Structured API?

- Mudah digunakan → tell Spark **what to do** not **how to do it**
- API yang konsisten untuk semua high level API : Machine learning, Deep learning, Graph, Streaming, dll
- Optimized storage and operations

  Berbeda dengan operasi dan data pada RDD yang bersifat transparan, Structured API memungkinkan Spark mengenali jenis operasi dan data yang diproses, sehingga dapat melakukan optimasi lebih lanjut, misalnya predicate pushdown, dll.





Kapan Menggunakan RDD?

Meskipun kita disarankan untuk menggunakan high level structured API, akan tetapi ada kasus di mana kita perlu menggunakan RDD atau low level API

- Jika perlu menggunakan fungsi yang tidak ada di *higher level API*
- Jika masih menggunakan legacy code yg menggunakan RDD
- Jika ingin melakukan custom partitioning (meskipun kita bisa melakukan ini dengan DataFrame sampai batas tertentu)
- Jika ingin melakukan manipulasi shared custom variable yang tidak ada di structured API

Karena pada dasarnya DataFrame akan dieksekusi oleh Spark sebagai RDD, memahami RDD akan sangat membantu dalam melakukan design, monitoring, maupun troubleshooting.

Hands-On

Eksplorasi DataFrame

Deskripsi

Pada praktek ini, kita akan mempelajari :

- Beberapa cara untuk membentuk DataFrame dari beberapa sumber data:
  - Membentuk DataFrame dari list
  - Membentuk DataFrame dari Pandas DataFrame
  - Membaca file csv dan simple JSON file
- Operasi umum pada DataFrame:
  - Filtering
  - Sorting
  - Aggregation
  - Join

Hands-On

SQL Query dengan DataFrame

Deskripsi

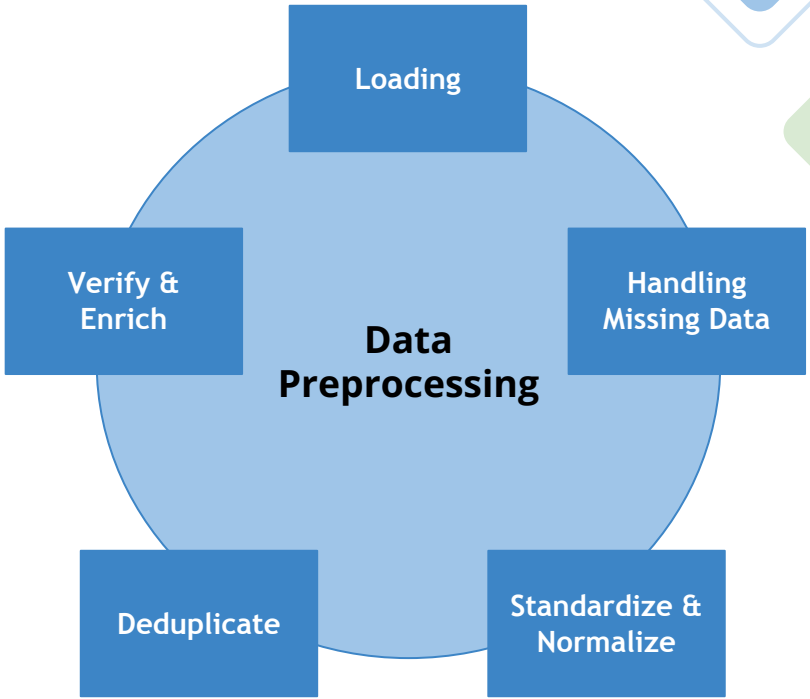
Pada praktek ini, kita akan mempelajari mengenai bagaimana mengolah data menggunakan SQL Query pada DataFrame

04 Data Cleaning and Enrichment

Data Preprocessing

Siapa yang perlu mempelajari data preprosesing

- Data Engineers
- Data Scientists/Data Analyst
- Machine Learning Engineers



Apa Yang Harus Diperhatikan?

- Invalid/Reject Records
- Null Values
- Invalid values
- Nonstandard values
- Duplicate records
- etc.

Null Values

- Perlu memahami data untuk menentukan penyebab/arti dari NULL : missing? unknown? broken records? dll
- Apa yang bisa dilakukan dengan nilai NULL?
  - Menghapusnya
  - Menggantikan dengan nilai mean/median/mode
  - Memberikan unique category
  - Memprediksi nilai yang hilang tersebut
  - dll

Data Standardization

Beberapa hal yang harus diperhatikan untuk melakukan standarisasi data

- Extra Spaces dan Blank Cells
- Data type format : number, string, date
- Data format : date, time, etc
- Text formatting : upper/lower/proper case
- Spell check
- dll

Data Enrichment : Menggabungkan DataFrames

- Ada 2 cara untuk menggabungkan data : merge and join
- Join biasanya digunakan pada proses enrichment
- Untuk melakukan merge DataFrame, gunakan fungsi transformasi **`union()`**
- **`union()`** tidak memeriksa duplikasi data
- Operasi Join pada PySpark memiliki potensi masalah kinerja bila tidak dirancang dengan hati-hati karena dapat melibatkan shuffling data secara masif

Hands-On

Data Cleansing and Enrichment

Deskripsi

Pada praktek ini, kita akan mempelajari:

- Penggunaan spark dataframe untuk membaca file csv
- Membersihkan data, yaitu
  - Menangani missing value
  - Menangani data invalid
  - Menangani data duplikat
- Enrichment dengan data referensi

THE REAL TRAINING BEGINS WHEN THE CLASS ENDS

Module development team

M. Urfah
Sigit Prasetyo

document version : 2.00.0203.23

DATALearns247 is educational program developed by Solusi247 focusing on building Indonesian data talents through curriculum based on the real world experience in big data and artificial intelligence implementation