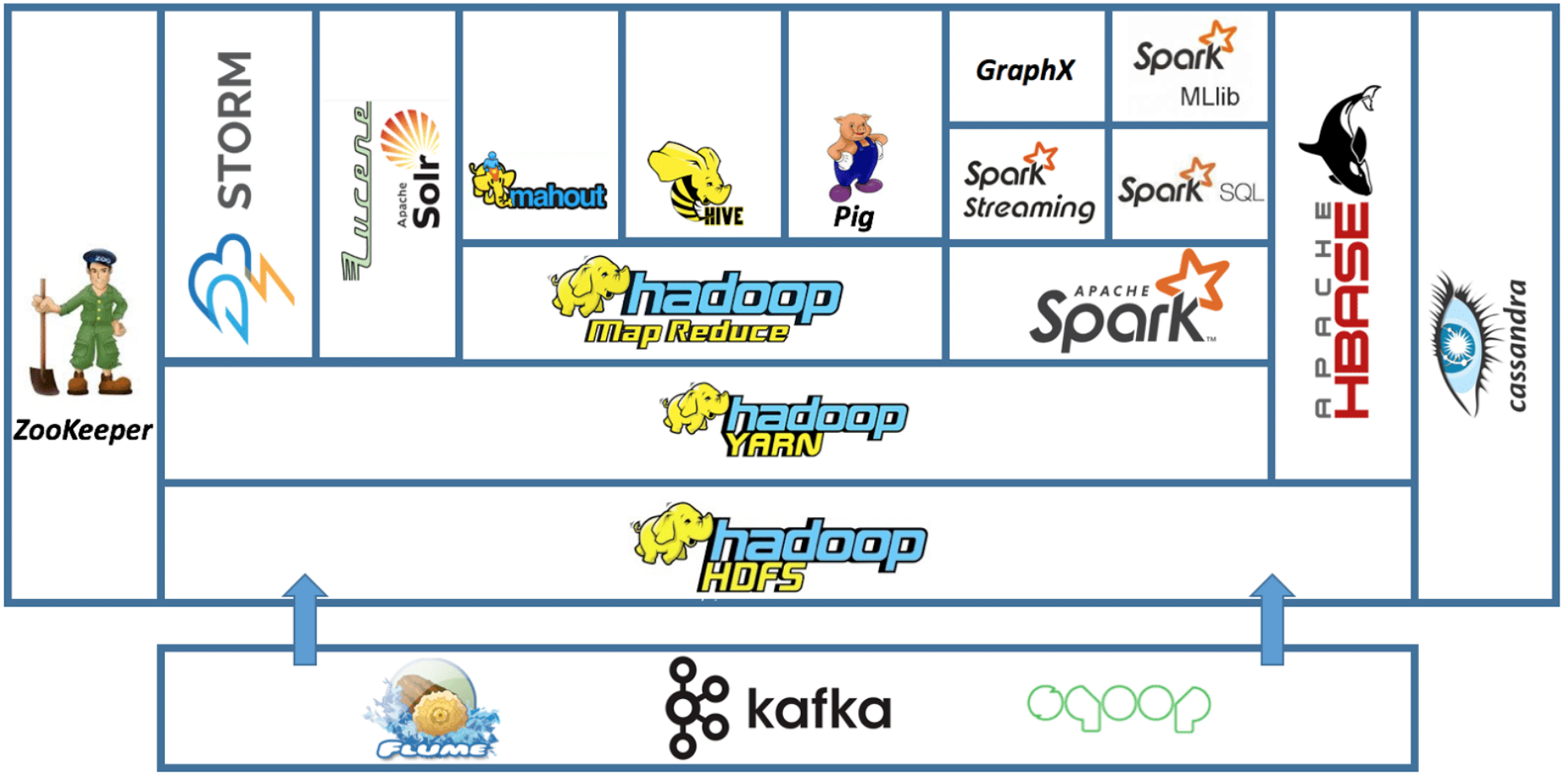
Apache Spark

Ahmad Rio Adriansyah S.Si. M.Si

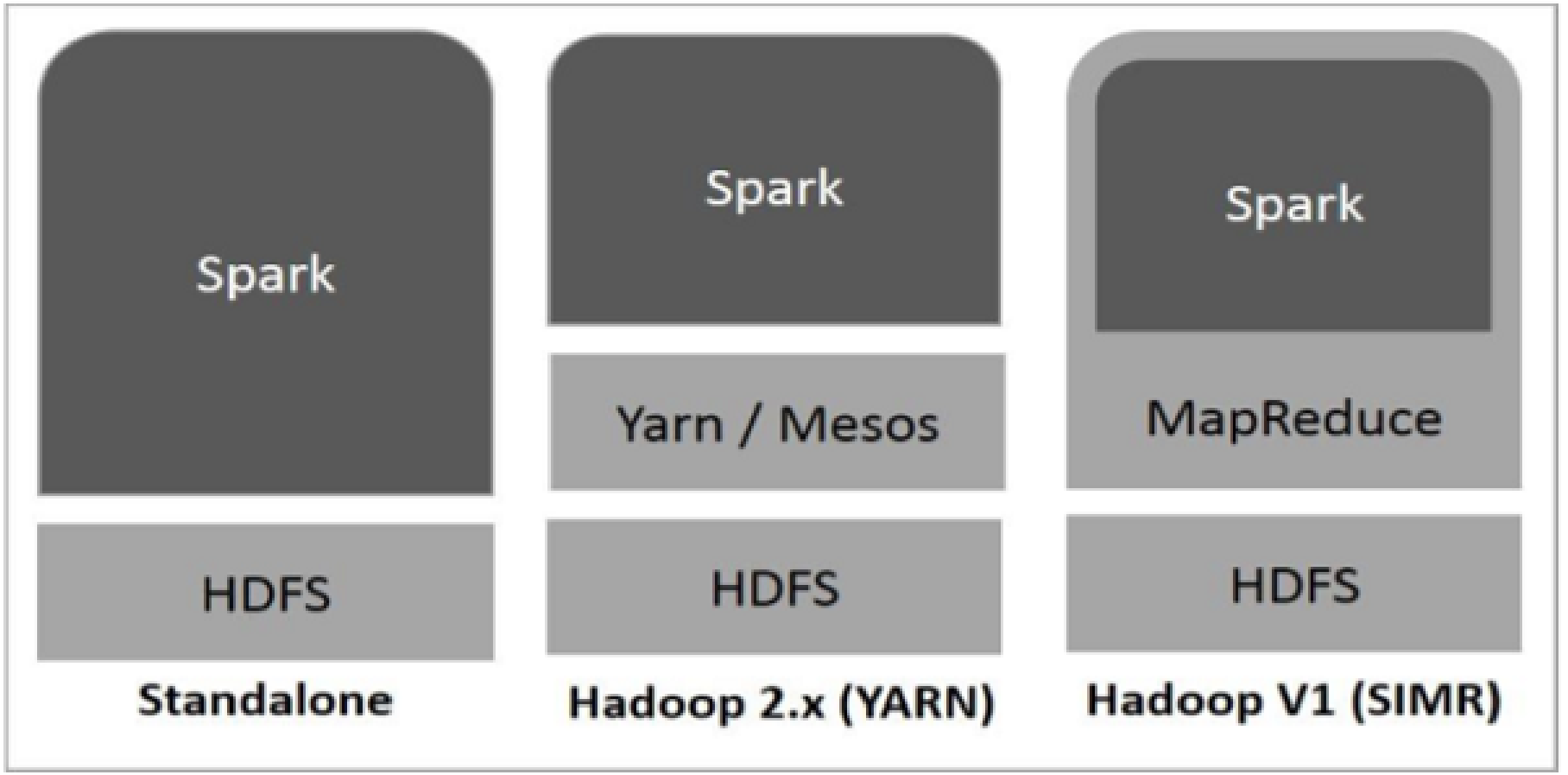
Ekosistem Hadoop



Spark

- Apache Spark didesain untuk komputasi cepat
- Berbasis Hadoop MapReduce
- Mendukung in-memory computing
- Fitur :
  - Proses lebih cepat dengan mengurangi IO ke disk
  - Memiliki API untuk Java, Scala, R, atau Python
  - Support SQL query, streaming data, machine learning, dan graph algorithm

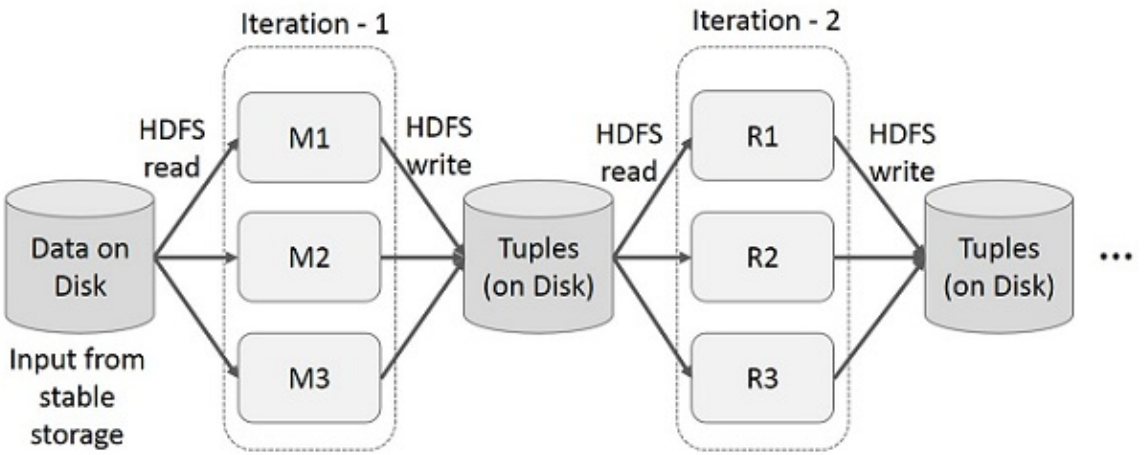
Spark dalam Hadoop



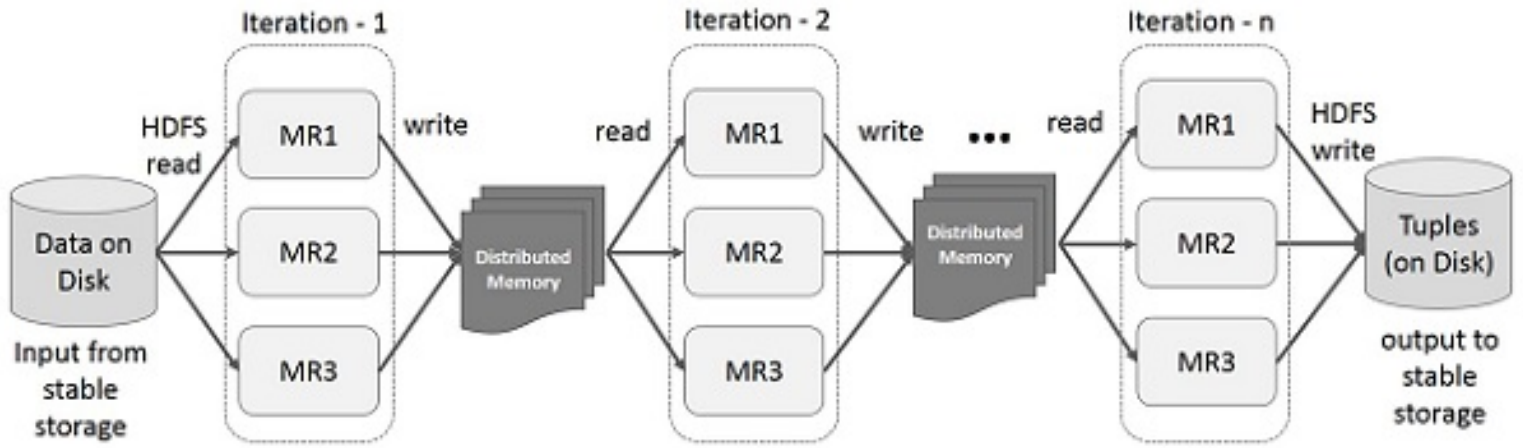
RDD

- Resilient Distributed Database
- Data struktur fundamental di Spark
- Read only, partitioned collection of records.
- Create RDD :
  - Memparalelisasi collection pada driver programnya
  - Mereferensi ke dataset yang berada di distributed storage system seperti HDFS

Mapreduce vs Spark RDD



Mapreduce



Spark RDD

Instalasi Spark

- Download
- Extract
  $ tar -xvzf spark-2.4.4-bin-hadoop2.7.tgz
- Tambahkan Spark ke sistem
  $ sudo gedit .bashrc

```
export PATH=$PATH:/home/ubuntu/BigData/spark-2.4.4-bin-hadoop2.7/bin
```

Menjalankan Spark

- Untuk menjalankan spark dengan bahasa scala gunakan perintah

  $ spark-shell

  atau dengan bahasa python

  $ pyspark

RDD : Word Count

```
>>> lines = sc.textFile("hdfs://...")
>>> words = lines.flatMap(lambda line : line.split(" "))
>>> pairs = words.map(lambda s: (s,1))
>>> counts = pairs.reduceByKey(lambda a,b : a+b)
>>> counts.saveAsTextFile("hdfs://...")
```

RDD : Sorted Word Count

```
>>> sorted = counts.map(lambda (a,b) : (b,a))
                .sortByKey(0,1)
                .map(lambda (a,b) : (b,a))
```

PySpark

- Gunakan pyspark shell untuk menjalankan dan menguji kode-kodenya
- Jika ingin mengeksekusi filenya, dapat menyimpannya dalam sebuah script python
  dengan menambahkan kode berikut di atasnya

```
from pyspark import SparkContext
sc = SparkContext.getOrCreate()
```

lalu menjalankannya dengan perintah

```
$ spark-submit script.py
```

PySpark

- Load File :

  >>> text = spark.read.text("hdfs://...")

- Show first row :

  >>> text.first()

- Counting lines :

  >>> text.count()

- Filtering :

  >>> text.filter(text.value.contains("Holmes"))
  .take(10)

SQL : Counting Max Words

```
>>> from pyspark.sql.functions import *
>>> text.select(size(split(text.value, "\s+"))
        .name("nWords"))
        .agg(max(col("nWords")))
        .collect()
```

Note : \s+ adalah regex yang mewakili whitespace (termasuk \n, \t, dll)

SQL : Word Counts

```
>>> text.select(explode(split(text.value, "\s+"))
        .alias("word"))
        .groupBy("word")
        .count()
        .collect()
```

SQL : Sorted Word Counts

```
>>> text.select(explode(split(text.value, "\s+"))
        .alias("word"))
        .groupBy("word")
        .count()
        .sort('count', ascending=[0,1])
        .collect()
```

Spark SQL

- Spark SQL dapat digunakan untuk
  - menjalankan SQL Queries
  - Membaca data dari instalasi Hive
- Saat dipanggil, akan menghasilkan objek berupa Dataset / DataFrame
  - Dataset = koleksi atas data yang terdistribusi
  - DataFrame = Dataset yang terorganisir dalam kolom-kolom yang bernama (mirip seperti tabel pada RDBMS)

Spark SQL : Load and Save

- Spark dapat mengambil input dari file maupun query SQL
- JSON atau CSV

```
>>> spark.read.load(namafile, format="json")
>>> spark.read.load(namafile, format="csv")

>>> df.select("name","age").write
        .save(outfile, format="json")
```

- SQL Query

```
>>> spark.sql("SELECT * FROM ...")
```

Kasus : Game Sales

- Ada dataset penjualan game dalam folder Dataset/vgsales.csv
- Load data tersebut dalam spark
- Hitung genre apa yang paling tinggi penjualan globalnya di tahun 2016

Kasus : Game Sales

```
>>> dfgame = spark.read.load("vgsales.csv",
                     header=true)

>>> sales = dfgame.select("Genre",
                                    "Year",
                                    "Global_Sales")

>>> sales = sales.filter(sales.Year==2016)

>>> sales = sales.withColumn("Global_Sales",
                     sales.Global_Sales.cast("double"))
```

Kasus : Game Sales

```
>>> sumsales = sales.groupBy("Genre")
                    .agg(sum("Global_Sales")
                    .alias("Total"))

>>> sorted = sumsales.sort("Total",
                  ascending=0)

>>> sorted.collect()
```

Latihan

- Tahun berapa penjualan game di Jepang paling banyak?
- Tampilkan total penjualan game (seluruh genre) di dunia dari tahun ke tahun.