

SISTEM REKOMENDASI BERDASAR ANALISIS BIG DATA DENGAN ALGORITMA ALTERNATING LEAST SQUARE PARALLEL

2017730088 MUHAMMAD NAUFAL AKRAM



PENDAHULUAN

Latar belakang, rumusan masalah, dan tujuan

01

BIG DATA & TAHAPAN ANALISISNYA

Penjelasan mengenai *big data* beserta tahap-tahap analisisnya

02

SCALA

Penjelasan mengenai bahasa pemrograman *scala*

03

OUTLINE

04

SPARK

Penjelasan mengenai *framework spark*

05

ALS & KOMPUTASINYA

Penjelasan mengenai ALS & beserta komputasinya

06

EKSPLORASI

Pengumpulan studi kasus *big data* dan eksplorasi menggunakan *scala*, *spark*, dan ALS dengan library *MLlib*



01

PENDAHULUAN





PENDAHULUAN

- Pertumbuhan data yang cepat
- Data dapat dianalisis dan dimanfaatkan untuk system rekomendasi
- Algoritma *ALS* dapat dimanfaatkan untuk membuat system rekomendasi
- Algoritma *ALS* sudah diimplementasikan pada library *MLLib Spark*
- Penerapan *ALS* pada *spark* menggunakan *MLLib* merupakan salah satu solusi untuk membuat system rekomendasi





02

***BIG DATA & TAHAPAN
ANALISISNYA***



Big Data & Tahapan Analisisnya



VOLUME

- ◆ Amount of data generated
- ◆ Online & offline transactions
- ◆ In kilobytes or terabytes
- ◆ Saved in records, tables, files



VELOCITY

- ◆ Speed of generating data
- ◆ Generated in real-time
- ◆ Online and offline data
- ◆ In Streams, batch or bits



VARIETY

- ◆ Structured & unstructured
- ◆ Online images & videos
- ◆ Human generated - texts
- ◆ Machine generated - readings



Big Data & Tahapan Analisisnya (2)

Pengumpulan Data



Eksplorasi & Penyiapan Data



Analisis Data



Story Telling & Deploy



03

Scala



Scala

- Mirip dengan *java*
- Mengombinasikan *object oriented* & *functional programming*
- Dapat beroperasi dengan semua *library java*
- Cocok untuk pengolahan data
- Variabel : *mutable* & *immutable*
- Variabel dapat digunakan untuk membuat fungsi, butuh parameter & argument
- *Class* dapat memiliki *variable*, *method*, dan *parameter*
- *Object* : kelas yang hanya memiliki satu *instance*



Scala



Scala (2)

```
val s = "hello"    // immutable  
var i = 42         // mutable
```

```
val addOne = (x: Int) => x + 1  
println(addOne(1)) // 2
```

```
val x = 1 + 1
```



Scala



Scala (3)

```
class Greeter(prefix: String, suffix:  
String) {  
  def greet(name: String): Unit =  
    println(prefix + name + suffix)  
}  
val greeter = new Greeter("Hello, ", "!")  
greeter.greet("Naufal") // Hello, Naufal!
```



Scala



Scala (4)

```
object Circle{  
  def calcCircArea(r: Double): Unit =  
    println(Pi*r*r)  
}  
val call=Circle.calcCircArea(7.0)
```



Scala



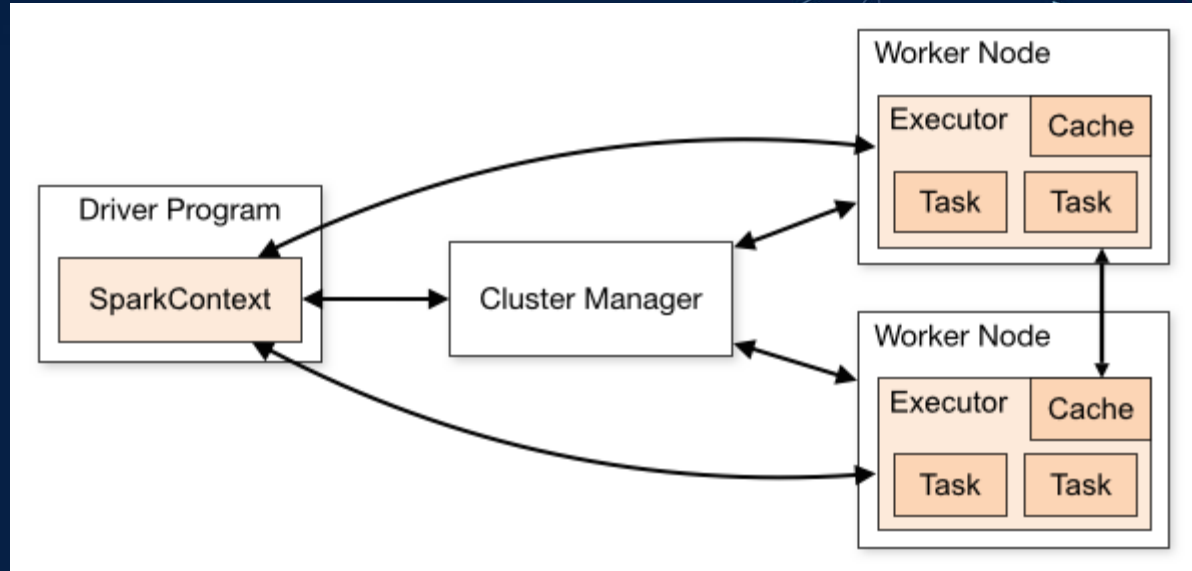
04

Spark



Spark

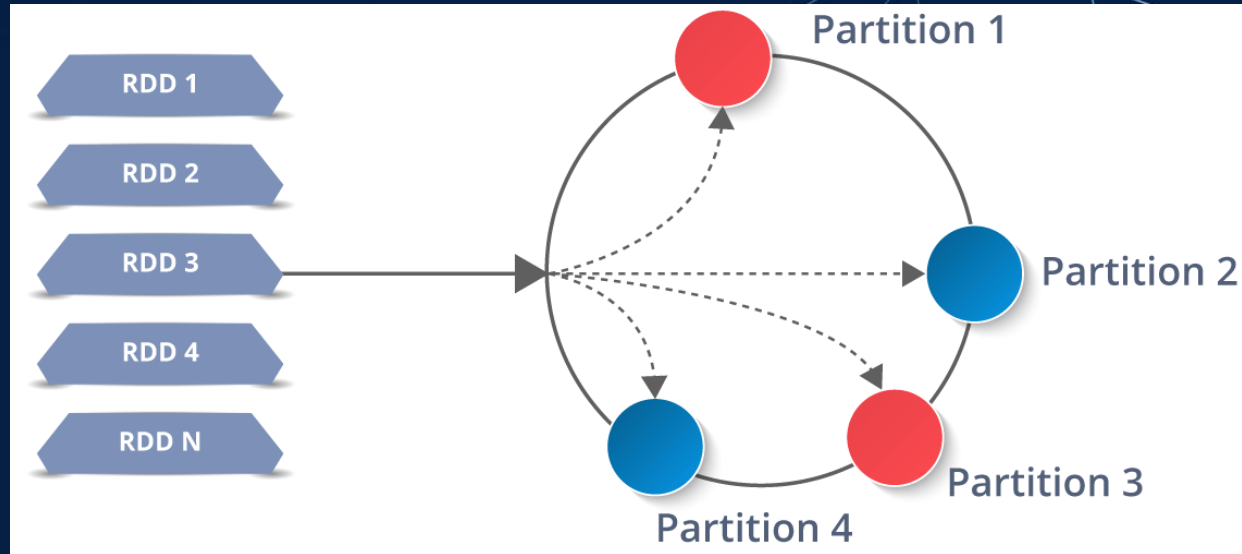
- *Open source framework*, untuk pemrosesan data dalam skala besar
- *Spark* menggunakan arsitektur *master/slave* dengan satu koordinator utama yang disebut *driver* dan beberapa *worker*
- *Driver* berkomunikasi dengan *worker* yang terdistribusi yang disebut *executor*
- *Driver* dan *Executor*-nya disebut sebagai spark application
- *Spark application* dijalankan di atas *cluster manager*



Spark (2)

RDD

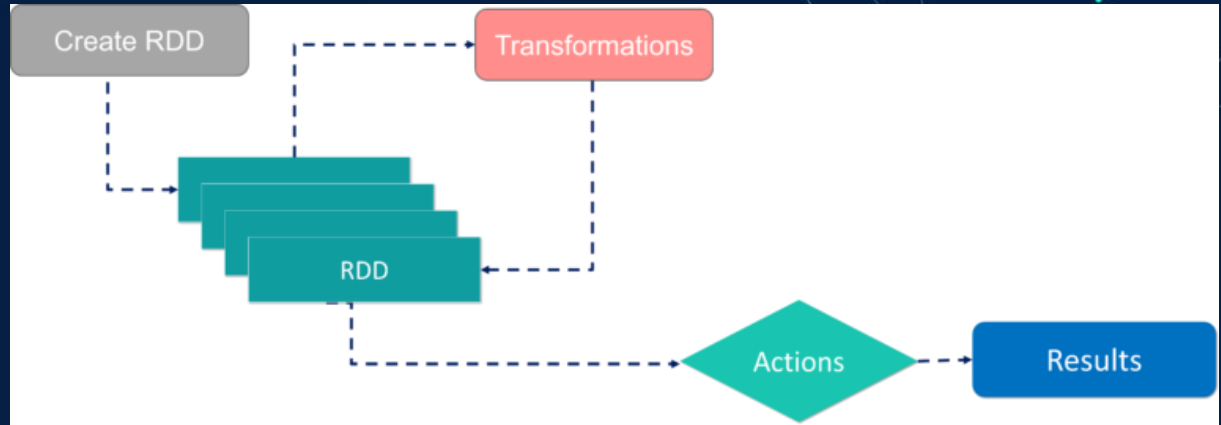
- Resilient Distributed dataset
- RDD abstraksi yang merepresentasikan kumpulan objek yang didistribusi
- RDD: immutable, partitioned, fault tolerant, In memory



Spark (2)

RDD Transformasi & Aksi

- Transformasi : Mengubah struktur nilai dan mengembalikan RDD baru, misal : `filter()`, `map()`, dll
- Aksi : Menerima masukan berupa *RDD* dan mengembalikan sebuah nilai, misal: `count()`, `collect()`, dll
- *Lazy execution*





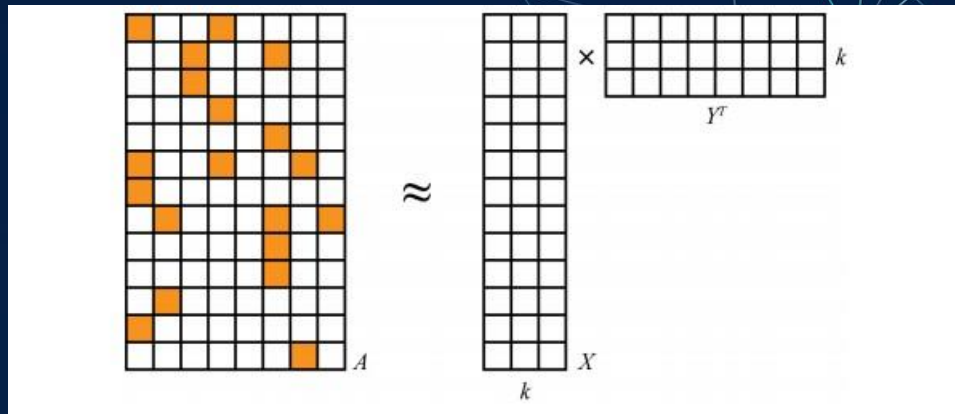
05

ALS &
Komputasinya



ALS & Komputasinya

- Pertama kali diperkenalkan pada *Netflix Price* tahun 2009
- Menggunakan konsep *matrix factorization*, $A=XY^T$
- Tujuannya adalah, mencari selisih kuadrat antara perkalian X dan Y^T dengan matriks A sekecil mungkin atau mendekati 0



$$A_i Y (Y^T Y)^{-1} = X_i$$
$$A_i X (X^T X)^{-1} = Y_i$$



ALS & Komputasinya (2)

User/Item	a	b	c
1	4	5	2
2	3	-	2
3	3	4	2

$$X =$$

User/Genre	G1	G2
1	-	-
2	-	-
3	-	-

$$Y^T =$$

Genre/Item	a	b	c
G1	-	-	-
G2	-	-	-



ALS & Komputasinya (3)

$$A_i Y (Y^T Y)^{-1} = X_i$$

$$Y^T = \begin{bmatrix} 0,7 & 0,7 & 0,5 \\ 0,7 & 0,6 & 0,3 \end{bmatrix}$$

$$A = \begin{bmatrix} 4 & 5 & 2 \\ 3 & - & 2 \\ 3 & 4 & 2 \end{bmatrix}$$

$$\begin{array}{l} A_1 = \begin{bmatrix} 4 & 5 & 2 \end{bmatrix} \\ A_1 Y = \begin{bmatrix} 7,3 & 6,4 \end{bmatrix} \end{array} \quad Y = \begin{bmatrix} 0,7 & 0,7 \\ 0,7 & 0,6 \\ 0,5 & 0,3 \end{bmatrix}$$

$$Y^T Y = \begin{bmatrix} 1,23 & 1,06 \\ 1,06 & 0,94 \end{bmatrix}$$



ALS & Komputasinya (4)

$$A_i Y (Y^T Y)^{-1} = X_i$$

$$A_1 = \begin{bmatrix} 4 & 5 & 2 \end{bmatrix}$$

$$A_1 Y = \begin{bmatrix} 7,3 & 6,4 \end{bmatrix}$$

$$Y^T Y = \begin{bmatrix} 1,23 & 1,06 \\ 1,06 & 0,94 \end{bmatrix}$$

$$A = \begin{bmatrix} 4 & 5 & 2 \\ 3 & - & 2 \\ 3 & 4 & 2 \end{bmatrix}$$

$$(Y^T Y)^{-1} = 1/0,0326 \begin{bmatrix} 0,94 & -1,06 \\ -1,06 & 1,23 \end{bmatrix}$$

$$(Y^T Y)^{-1} = \begin{bmatrix} 28,83435583 & -32,51533742 \\ -32,51533742 & 37,73006135 \end{bmatrix}$$

$$X_1 = A_1 Y (Y^T Y)^{-1} = \begin{bmatrix} 2,392638037 & 4,110429448 \end{bmatrix}$$



ALS & Komputasinya (5)

$$(Y^T Y)^{-1} = 1/0,0326 \begin{bmatrix} 0,94 & -1,06 \\ -1,06 & 1,23 \end{bmatrix}$$

$$(Y^T Y)^{-1} = \begin{bmatrix} 28,83435583 & -32,51533742 \\ -32,51533742 & 37,73006135 \end{bmatrix}$$

$$X_1 = A_1 Y (Y^T Y)^{-1} = \begin{bmatrix} 2,392638037 & 4,110429448 \end{bmatrix}$$

$$X_1 = A_1 Y (Y^T Y)^{-1} = \begin{bmatrix} 0,37 & 0,63 \end{bmatrix}$$



ALS & Komputasinya (6)

$$X = \begin{bmatrix} 0,37 & 0,63 \\ 0,6 & 0,4 \\ 0,88 & 0,12 \end{bmatrix}$$



ALS & Komputasinya (7)

$$A_i X (X^T X)^{-1} = Y_i$$

$$X^T = \begin{bmatrix} 0,37 & 0,6 & 0,88 \\ 0,63 & 0,4 & 0,12 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} 4 & 5 & 2 \end{bmatrix} \quad X = \begin{bmatrix} 0,37 & 0,63 \\ 0,6 & 0,4 \\ 0,88 & 0,12 \end{bmatrix}$$
$$A_1 X = \begin{bmatrix} 6,24 & 4,76 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 1,27 & 0,75 \\ 0,75 & 0,86 \end{bmatrix}$$

$$A = \begin{bmatrix} 4 & 5 & 2 \\ 3 & - & 2 \\ 3 & 4 & 2 \end{bmatrix}$$



ALS & Komputasinya (8)

$$A_i Y (Y^T Y)^{-1} = X_i$$

$$A_1 = \begin{bmatrix} 4 & 5 & 2 \\ 6,24 & 4,76 \end{bmatrix}$$

$$X^T X = \begin{bmatrix} 1,27 & 0,75 \\ 0,75 & 0,86 \end{bmatrix}$$

$$A = \begin{bmatrix} 4 & 5 & 2 \\ 3 & - & 2 \\ 3 & 4 & 2 \end{bmatrix}$$

$$(X^T X)^{-1} = 1/0,53 \begin{bmatrix} 0,86 & -0,75 \\ -0,75 & 1,27 \end{bmatrix}$$

$$(X^T X)^{-1} = \begin{bmatrix} 1,647865782 & -1,443929539 \\ -1,443929539 & 2,420767009 \end{bmatrix}$$

$$Y_1 = A_1 X (X^T X)^{-1} = \begin{bmatrix} 3,409577878 & 2,512730639 \end{bmatrix}$$

$$Y_1 = A_1 X (X^T X)^{-1} = \begin{bmatrix} 0,58 & 0,42 \end{bmatrix}$$



ALS & Komputasinya (9)

$$Y = \begin{bmatrix} 0,58 & 0,42 \\ 0,62 & 0,38 \\ 0,7 & 0,3 \end{bmatrix}$$



ALS & Komputasinya (10)

$$X \times Y^T = \begin{bmatrix} 0,4767 & 0,4688 & 0,448 \\ 0,516 & 0,524 & 0,54 \\ 0,5608 & 0,5912 & 0,652 \end{bmatrix}$$

$$2,5 \times (X \times Y^T) = \begin{bmatrix} 1,19175 & 1,172 & 1,12 \\ 1,29 & 1,31 & 1,35 \\ 1,402 & 1,478 & 1,63 \end{bmatrix}$$

ALS & Komputasinya (11)

$$(2,5(X \times Y^T)) - A = -13,05625$$

$$((2,5(X \times Y^T)) - A)^2 = 170,4656641$$



ALS & Komputasinya (12)

$$X \times Y^T = \begin{bmatrix} -0,1134 & -0,2644 & 0,5184 \\ 0,4942 & 0,0883 & 0,5025 \\ 0,501 & -0,088 & 0,513 \end{bmatrix}$$

$$2,5 \times (X \times Y^T) = \begin{bmatrix} -0,2835 & -0,661 & 1,296 \\ 1,2355 & 0,22075 & 1,25625 \\ 1,2525 & -0,22 & 1,2825 \end{bmatrix}$$



ALS & Komputasinya (13)

$$(2,5(X \times Y^T)) - A = -19,621$$

$$((2,5(X \times Y^T)) - A)^2 = 384,983641$$



ALS & Komputasinya (14)

$$X \times Y^T = \begin{bmatrix} 0,2304 & 1,1 & 0,4 \\ 0,556 & 0,26 & 0,554 \\ 0,248 & 1,58 & 0,312 \end{bmatrix}$$

$$2,5 \times (X \times Y^T) = \begin{bmatrix} 0,576 & 2,75 & 1 \\ 1,39 & 0,65 & 1,385 \\ 0,62 & 3,95 & 0,78 \end{bmatrix}$$



ALS & Komputasinya (15)

$$(2,5(X \times Y^T)) - A = -11,899$$

$$((2,5(X \times Y^T)) - A)^2 = 141,586201$$



HASIL TIAP ITERASI

	ITERASI 1	ITERASI 2	ITERASI 3	ITERASI 4
SELISIH	-13,05	-19,61	-12,35	-11,89
SELISIH KUADRAT	170,46	384,98	152,72	141,58
PERBEDAAN DENGAN ITERASI SEBELUMNYA	0	+214,52	-132,26	-11,14








06

Eksplorasi



Eksplorasi

 artist_alias.txt	06/10/2019 4:40	Text Document	9 KB
 artist_data.txt	06/10/2019 4:40	Text Document	712 KB
 user_artist_data.txt	06/10/2019 4:40	Text Document	878 KB

```
val base = "D:\\Data  
Opank\\Kuliah\\Skripsi\\Music-Recommender-System-  
using-ALS-Algorithm-with-Apache-Spark-and-Python-  
master\\data_raw\\";
```

```
val rawUserArtistData = spark.read.textFile(base  
+ "user_artist_data.txt")
```

```
val rawArtistData = spark.read.textFile(base +  
"artist_data.txt")
```

```
val rawArtistAlias = spark.read.textFile(base +  
"artist_alias.txt")
```



Eksplorasi (2)

```
val bArtistAlias =  
  spark.sparkContext.broadcast(buildArtistAlias(raw  
    ArtistAlias))
```



Eksplorasi (3)

```
def buildArtistAlias(rawArtistAlias:
Dataset[String]): Map[Int, Int] = {

  rawArtistAlias.flatMap { line =>

    val Array(artist, alias) =
line.split('\t')

    if (artist.isEmpty) {

      None

    } else {

      Some((artist.toInt,
alias.toInt))

    }

  }.collect().toMap

}
```



Eksplorasi (4)

```
val allData = buildCounts(rawUserArtistData,  
bArtistAlias).cache()
```



Eksplorasi (5)

```
def buildCounts(  
    rawUserArtistData:  
    Dataset[String],  
    bArtistAlias: Broadcast[Map[Int,  
Int]]) : DataFrame = {  
    rawUserArtistData.map { line =>  
        val Array(userID, artistID, count) =  
        line.split(' ').map(_.toInt)  
        val finalArtistID =  
        bArtistAlias.value.getOrElse(artistID, artistID)  
        (userID, finalArtistID, count)  
    }.toDF("user", "artist", "count")  
}
```



Eksplorasi (6)

```
val bArtistAlias =  
  spark.sparkContext.broadcast(buildArtistAlias(rawArtistAlias))  
  
val allData =  
  buildCounts(rawUserArtistData,  
    bArtistAlias).cache()
```



Eksplorasi (7)

```
val model = new ALS().  
    setSeed(Random.nextLong()).  
    setImplicitPrefs(true). setRank(10).setRegParam(1.0).setAlpha(40.0).setMaxIter(20).  
    setUserCol("user").setItemCol("artist").  
    setRatingCol("count").setPredictionCol("prediction").  
    fit(allData)  
allData.unpersist()  
  
model.write.overwrite().save("D:\\Data  
Opank\\Kuliah\\Skripsi\\ALS\\model")
```



Eksplorasi (8)

```
val userID = 2093760

val topRecommendations =
  makeRecommendations(model, userID, 5)

def makeRecommendations(model: ALSModel, userID:
  Int, howMany: Int): DataFrame = {

  val toRecommend = model.itemFactors.

    select($"id".as("artist")).

    withColumn("user", lit(userID))

  model.transform(toRecommend).

    select("artist", "prediction").

    orderBy($"prediction".desc).

    limit(howMany)

}
```



Eksplorasi (9)

```
val recommendedArtistIDs =  
topRecommendations.select("artist").as[  
Int].collect()
```


```
val artistByID =  
buildArtistByID(rawArtistData)
```

```
artistByID.join(spark.createDataset(rec  
ommendedArtistIDs).toDF("id", "id").
```

```
select("name").show()
```

```
+-----+  
|      name|  
+-----+  
|Peace Orchestra|  
|      DJ Krush|  
|      Tortoise|  
| Sophie Zelmani|  
|      Björk|  
+-----+
```





**Terima
kasih**

