LAPORAN

RENCANA TUGAS MAHASISWA (RTM) Ke-5

MATA KULIAH BIG DATA (B)

"sistem penskoran secara otomatis pada soal essay menggunakan PySpark"

DISUSUN OLEH:

Nabilah Selayanti (NPM. 22083010013)

DOSEN PENGAMPU:

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

PROGRAM STUDI SAINS DATA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN" JAWA TIMUR

2024

1. **Import Modul**

```
In [2]: #mengimport modul yang dibutuhkan
        from pyspark.sql import SparkSession
        from pyspark.sql.types import *
        from pyspark.sql.functions import *
        from pyspark.ml.recommendation import ALS
        from pyspark.ml.evaluation import RegressionEvaluator
        from pyspark.sql.functions import hash, abs
```

Import beberapa modul diatas digunakan untuk :

- **`SparkSession`** : Digunakan untuk membuat dan mengelola sesi Spark.
- **`pyspark.sql.types`** : Mengimpor semua tipe data yang didefinisikan dalam modul `types` di Spark SQL. Ini termasuk tipe data seperti `StringType`, `IntegerType`, dll.
- **`pyspark.sql.functions`** : Mengimpor semua fungsi yang didefinisikan dalam modul `functions` di Spark SQL.
- **`ALS`** : Digunakan untuk membangun model Collaborative Filtering menggunakan metode Alternating Least Squares (ALS) di Spark ML.
- **`RegressionEvaluator`** : Digunakan untuk mengevaluasi model regresi, termasuk model ALS, dengan metrik seperti RMSE (Root Mean Squared Error).
- **`hash, abs`** : Fungsi `hash` digunakan untuk menghasilkan nilai hash dari ekspresi atau kolom, sedangkan fungsi `abs` digunakan untuk menghitung nilai absolut dari suatu ekspresi atau kolom.

2. **Membuat Session**

```
In [3]: #membuat session
        appName = "Sistem Penskoran Otomatis pada Soal Essay"
        spark = SparkSession \
            .builder \
            .appName(appName) \
            .config("spark.some.config.option", "some-value") \
            .getOrCreate()
```

Code diatas digunakan untuk membuat session PySpark, berikut penjelasan code tersebut :

- **`appName = "Sistem Penskoran Otomatis pada Soal Essay"`** : Digunakan untuk menyimpan nama aplikasi yang akan ditetapkan untuk sesi Spark.
- **`SparkSession`** : Digunakan untuk berinteraksi dengan data dan menjalankan operasi di Spark.
- **`builder`** : Digunakan untuk memulai proses pembangunan sesi Spark.
- **`appName(appName)`** : Metode `.appName()` digunakan untuk menetapkan nama aplikasi untuk sesi Spark yang sedang dibangun.

- **`.config("spark.some.config.option", "some-value") \`:** Metode `.config()` digunakan untuk mengatur konfigurasi Spark.
- **`getOrCreate()`:** Metode `.getOrCreate()` digunakan untuk mendapatkan atau membuat sesi Spark.

3. Import Data

```
In [4]: df = spark.read.csv('/content/training_data_essay.csv', inferSchema=True, header=True, sep=';')
        df.show()
```

Code ini untuk membaca data dari file CSV tersebut. Dalam membaca file, skema data akan diinfer dari file tersebut agar Spark dapat mengenali tipe data yang tepat **(`inferSchema=True`).** Code ini juga mengasumsikan bahwa baris pertama dalam file CSV adalah header yang berisi nama kolom **(`header=True`)**. Pemisah antar kolom dalam file CSV adalah titik koma **(`sep=';'`)**. Setelah membaca file, data yang berhasil dibaca sebagai berikut.

```
+----------+------------+--------------------+----+-------------+
|       npm|nama_peserta|             jawaban|soal|skor_per_soal|
+----------+------------+--------------------+----+-------------+
|         0|       Admin|Tidak, Hanya memb...|   1|          100|
|         0|       Admin|Biaya dihitung be...|   2|          100|
|         0|       Admin|Hak cipta adalah ...|   3|          100|
|         0|       Admin|Dijelaskan kepada...|   4|          100|
|         0|       Admin|1. Melindungi dan...|   5|          100|
|         0|       Admin|Ruang Komputer, P...|   6|          100|
|         0|       Admin|Aturlah posisi pe...|   7|          100|
|         0|       Admin|Posisi Kepala dan...|   8|          100|
|         0|       Admin|1. Kecocokan soft...|   9|          100|
|         0|       Admin|1. Fokus dan expo...|  10|          100|
|         0|       Admin|1. Peralatan yang...|  11|          100|
|         0|       Admin|1. Dibuat grafik ...|  12|          100|
|1121020033|          AP|tidak, cuma mengi...|   1|         52,7|
|1121020033|          AP|biaya dihitung be...|   2|        42,86|
|1121020033|          AP|hak membuat merup...|   3|        42,16|
|1121020033|          AP|dipaparkan pada k...|   4|        27,19|
|1121020033|          AP|1. mencegah serta...|   5|        44,14|
|1121020033|          AP|ruang komputer, p...|   6|          100|
|1121020033|          AP|aturlah posisi fi...|   7|        57,68|
|1121020033|          AP|posisi kepala ser...|   8|        45,71|
+----------+------------+--------------------+----+-------------+
only showing top 20 rows
```

4. Pre-Processing Data

```
In [5]: # Ganti koma (",") dengan titik (".") pada kolom "skor_per_soal".
        df1 = df.withColumn("skor_per_soal", regexp_replace(col("skor_per_soal"), ",", "."))
        df1.show()
```

Untuk melakukan mengganti tanda koma **(",")** menjadi titik **(".")** pada kolom `"skor_per_soal"` dalam DataFrame `df`, menggunakan fungsi `withColumn()` untuk membuat kolom baru dengan nama `"skor_per_soal"`. Pada kolom baru ini, menggunakan fungsi `regexp_replace()` untuk mengganti setiap tanda koma dengan titik. Setelah itu, hasilnya ditampilkan sebagai berikut.

```
+----------+------------+--------------------+----+-------------+
|       npm|nama_peserta|             jawaban|soal|skor_per_soal|
+----------+------------+--------------------+----+-------------+
|         0|       Admin|Tidak, Hanya memb...|   1|          100|
|         0|       Admin|Biaya dihitung be...|   2|          100|
|         0|       Admin|Hak cipta adalah ...|   3|          100|
|         0|       Admin|Dijelaskan kepada...|   4|          100|
|         0|       Admin|1. Melindungi dan...|   5|          100|
|         0|       Admin|Ruang Komputer, P...|   6|          100|
|         0|       Admin|Aturlah posisi pe...|   7|          100|
|         0|       Admin|Posisi Kepala dan...|   8|          100|
|         0|       Admin|1. Kecocokan soft...|   9|          100|
|         0|       Admin|1. Fokus dan expo...|  10|          100|
|         0|       Admin|1. Peralatan yang...|  11|          100|
|         0|       Admin|1. Dibuat grafik ...|  12|          100|
|1121020033|          AP|tidak, cuma mengi...|   1|         52.7|
|1121020033|          AP|biaya dihitung be...|   2|        42.86|
|1121020033|          AP|hak membuat merup...|   3|        42.16|
|1121020033|          AP|dipaparkan pada k...|   4|        27.19|
|1121020033|          AP|1. mencegah serta...|   5|        44.14|
|1121020033|          AP|ruang komputer, p...|   6|          100|
|1121020033|          AP|aturlah posisi fi...|   7|        57.68|
|1121020033|          AP|posisi kepala ser...|   8|        45.71|
+----------+------------+--------------------+----+-------------+
only showing top 20 rows
```

Langkah pre-processing selanjutnya adalah memeriksa tipe data `skor_per_soal`.

```
In [6]: # Menampilkan skema DataFrame
        df1.printSchema()

        root
         |-- npm: integer (nullable = true)
         |-- nama_peserta: string (nullable = true)
         |-- jawaban: string (nullable = true)
         |-- soal: integer (nullable = true)
         |-- skor_per_soal: string (nullable = true)
```

Pada output diatas, variabel `skor_per_soal` bertipe string'. Sehingga, perlu dilakukan konversi tipe data menjadi 'integer'. Hal ini diperlukan karena pada pemodelan ALS, data yang digunakan harus memiliki tipe data 'integer'. Berikut Codenya.

```
In [7]: # Ubah tipe data kolom "skor_per_soal" menjadi integer
        data_essay = df1.withColumn("skor_per_soal", col("skor_per_soal").cast("float"))

        data_essay.printSchema()
```

Pada code tersebut, dengan menggunakan metode **`cast("float")`** untuk perubahan tipe data. Sehingga, struktur schema dari DataFrame `data_essay` dapat dilihat sebagai berikut.

```
root
 |-- npm: integer (nullable = true)
 |-- nama_peserta: string (nullable = true)
 |-- jawaban: string (nullable = true)
 |-- soal: integer (nullable = true)
 |-- skor_per_soal: float (nullable = true)
```

5. Menyiapkan Data

```
In [8]: data = data_essay.select("soal","jawaban",'skor_per_soal')
        data.show()
```

Code tersebut memilih kolom **"soal", "jawaban", dan "skor_per_soal"** dari DataFrame `data_essay` menggunakan metode **`select()`**, yang digunakan untuk pemodelan. Kemudian, data yang dipilih ditampilkan menggunakan metode **`show()`**, outputnya sebagai berikut.

```
+----+--------------------+-------------+
|soal|             jawaban|skor_per_soal|
+----+--------------------+-------------+
|   1|Tidak, Hanya memb...|        100.0|
|   2|Biaya dihitung be...|        100.0|
|   3|Hak cipta adalah ...|        100.0|
|   4|Dijelaskan kepada...|        100.0|
|   5|1. Melindungi dan...|        100.0|
|   6|Ruang Komputer, P...|        100.0|
|   7|Aturlah posisi pe...|        100.0|
|   8|Posisi Kepala dan...|        100.0|
|   9|1. Kecocokan soft...|        100.0|
|  10|1. Fokus dan expo...|        100.0|
|  11|1. Peralatan yang...|        100.0|
|  12|1. Dibuat grafik ...|        100.0|
|   1|tidak, cuma mengi...|         52.7|
|   2|biaya dihitung be...|        42.86|
|   3|hak membuat merup...|        42.16|
|   4|dipaparkan pada k...|        27.19|
|   5|1. mencegah serta...|        44.14|
|   6|ruang komputer, p...|        100.0|
|   7|aturlah posisi fi...|        57.68|
|   8|posisi kepala ser...|        45.71|
+----+--------------------+-------------+
only showing top 20 rows
```

6. Mengonversi kolom 'jawaban' menjadi representasi numerik

```
In [9]: hash_data = data.withColumn("hashed_answer", hash("jawaban"))

        hash_data.select("soal", "jawaban","skor_per_soal", "hashed_answer")
        hash_data.show()
```

Code tersebut menambahkan kolom baru dengan nama "hashed_answer" ke DataFrame `data` yang berisi nilai hash dari kolom "jawaban" menggunakan fungsi **`hash()`**. Kemudian, DataFrame yang sudah dimodifikasi ditampilkan dengan kolom **"soal", "jawaban", "skor_per_soal", dan "hashed_answer"** menggunakan metode **`select()`** dan **`show()`**. Sehingga, output dapat dilihat sebagai berikut.

```
+----+--------------------+-------------+-------------+
|soal|             jawaban|skor_per_soal|hashed_answer|
+----+--------------------+-------------+-------------+
|   1|Tidak, Hanya memb...|        100.0|  -2059296905|
|   2|Biaya dihitung be...|        100.0|   1183180174|
|   3|Hak cipta adalah ...|        100.0|   1232762403|
|   4|Dijelaskan kepada...|        100.0|  -2035408785|
|   5|1. Melindungi dan...|        100.0|   1588395990|
|   6|Ruang Komputer, P...|        100.0|    339970513|
|   7|Aturlah posisi pe...|        100.0|     50850002|
|   8|Posisi Kepala dan...|        100.0|   -945877996|
|   9|1. Kecocokan soft...|        100.0|   1576366224|
|  10|1. Fokus dan expo...|        100.0|  -1905649442|
|  11|1. Peralatan yang...|        100.0|    550139146|
|  12|1. Dibuat grafik ...|        100.0|   1727767227|
|   1|tidak, cuma mengi...|         52.7|   1947733435|
|   2|biaya dihitung be...|        42.86|  -1139863335|
|   3|hak membuat merup...|        42.16|    122676417|
|   4|dipaparkan pada k...|        27.19|  -1054163002|
|   5|1. mencegah serta...|        44.14|   1990940339|
|   6|ruang komputer, p...|        100.0|   1770907636|
|   7|aturlah posisi fi...|        57.68|   -463479969|
|   8|posisi kepala ser...|        45.71|   -412537011|
+----+--------------------+-------------+-------------+
only showing top 20 rows
```

7. Spliting Data

```
In [45]: #membagi data, 70% training dan 30% testing
         splits = hash_data.randomSplit([0.7, 0.3])
         train = splits[0].withColumnRenamed("skor_per_soal", "label")
         test = splits[1].withColumnRenamed("skor_per_soal", "trueLabel")

         #menghitung baris data training dan testing
         print("Jumlah baris data training :", train.count())
         print("jumlah baris data testing :", test.count())
```

Code tersebut membagi DataFrame `hash_data` menjadi dua bagian: data training (70%) dan data testing (30%) menggunakan metode **`randomSplit()`**. Setelah itu, kolom "skor_per_soal" diubah namanya menjadi "label" untuk data training dan "trueLabel" untuk data testing menggunakan metode **`withColumnRenamed()`**. Kemudian, jumlah baris data training dan testing dicetak untuk memeriksa pembagian data.

```
Jumlah baris data training : 86
jumlah baris data testing : 34
```

8. Mendefinisikan Model ALS dan Mentrainingnya

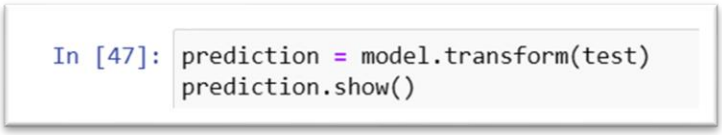
```
In [46]: #mendefinisikan algoritma ALS untuk sistem Penskoran Otomatis pada Soal Essay
         als = ALS(maxIter=19, regParam=0.01, userCol="hashed_answer",
                   itemCol="soal", ratingCol="label")

         #mentraining model dengan fungsi ".fit()"
         model = als.fit(train)
         print("Model telah selesai ditraining!")

         Model telah selesai ditraining!
```

Code tersebut mendefinisikan model ALS (Alternating Least Squares) untuk sistem penskoran otomatis pada soal esai. Model ini memiliki 19 iterasi maksimum menggunakan parameter `**maxIter**`, lalu menggunakan parameter `**regParam**` sebesar 0.01, dengan kolom "hashed_answer" sebagai identitas pengguna `**userCol**`, kolom "soal" sebagai identitas item `**itemCol**`, dan kolom "label" sebagai rating dari pengguna terhadap item. Setelah model didefinisikan, dilakukan proses pelatihan menggunakan data pelatihan, dan setelah selesai, pesan "`Model telah selesai ditraining!`" dicetak.

9. Melakukan prediksi menggunakan model yang telah ditraining terhadap data testing



```
In [47]: prediction = model.transform(test)
         prediction.show()
```

Code tersebut digunakan untuk melakukan prediksi menggunakan model yang telah ditraining terhadap data testing. Dengan menggunakan metode `**.transform(test)**`, model akan melakukan prediksi skor pada data testing berdasarkan item (soal) yang diberikan. Hasil prediksi ditampilkan sebagai berikut.

```
+----+--------------------+---------+-------------+----------+
|soal|             jawaban|trueLabel|hashed_answer|prediction|
+----+--------------------+---------+-------------+----------+
|   1|tidak, hanya memb...|    100.0|   -256638840|  99.99993|
|   1|tidak, hanya memb...|    100.0|   -256638840|  99.99993|
|   6|posisi tubuh, pos...|    90.37|  -1702735646|       NaN|
|   6|        posisi tubuh|    67.08|  -1240688658|       NaN|
|   6|ruang komputer, p...|    100.0|   1770907636|  99.99987|
|   6|ruang komputer, p...|    100.0|   1770907636|  99.99987|
|   6|ruang komputer, p...|    100.0|   1770907636|  99.99987|
|   3|hak cipta adalah ...|    83.43|  -1876419705|  83.42993|
|   3|emperbanyak cipta...|    47.67|    944450734|       NaN|
|   5|melindungi dan me...|    74.73|  -1932865057|  74.72994|
|   5|melindungi dan me...|    69.63|  -1210907486|       NaN|
|   5|memlihara sumber ...|     88.3|     40672765|       NaN|
|   4|Dijelaskan kepada...|    100.0|  -2035408785|       NaN|
|   4|dijelaskan kepada...|    72.06|  -1834227989| 72.059906|
|   8|posisi kepala dan...|    48.41|    899751389|       NaN|
|   7|aturlah posisi pe...|    86.22|  -1392782412|  86.21994|
|   2|biaya dihitung be...|    84.52|   -219318287|       NaN|
|   2|biaya dihitung be...|    84.52|   -219318287|       NaN|
|   2|biaya dihitung be...|    100.0|   1176853507|  99.99985|
|   2|biaya dihitung be...|    100.0|   1176853507|  99.99985|
+----+--------------------+---------+-------------+----------+
only showing top 20 rows
```

10. Evaluasi Model

```
In [48]: evaluator = RegressionEvaluator(
             labelCol="trueLabel", predictionCol="prediction", metricName="rmse")
         rmse = evaluator.evaluate(prediction)
         print ("Root Mean Square Error (RMSE):", rmse)

         Root Mean Square Error (RMSE): nan
```

Code tersebut digunakan untuk menghitung Root Mean Square Error (RMSE) antara skor aktual dan skor prediksi yang dihasilkan oleh model. Pertama, objek **`RegressionEvaluator`** dibuat dengan menspesifikasikan kolom label aktual (`trueLabel`) dan kolom prediksi (`prediction`) yang ingin dievaluasi, serta metric yang digunakan (`rmse`). Kemudian, dengan menggunakan metode **`.evaluate(prediction)`**, RMSE dihitung berdasarkan data prediksi yang dihasilkan sebelumnya. Nilai **`nan`** menunjukkan bahwa ada nilai yang hilang atau tidak valid dalam data prediksi atau label aktual, yang dapat memengaruhi hasil evaluasi.

```
In [49]: prediction.count()
         a = prediction.count()
         print("jumlah baris sebelum di hapus data kosong: ", a)
         cleanPred = prediction.dropna(how="any", subset=["prediction"])
         b = cleanPred.count()
         print("jumlah baris setelah di hapus data kosong: ", b)
         print("jumlah baris data kosong: ", a-b)
```

Code tersebut digunakan untuk menghitung jumlah baris sebelum dan setelah menghapus baris yang memiliki nilai kosong (null) pada kolom "prediction". Pertama, dilakukan perhitungan jumlah baris sebelum penghapusan data kosong dengan menggunakan metode **`.count()`** pada DataFrame prediction dan disimpan dalam variabel `'a'`. Kemudian, dilakukan penghapusan data kosong dengan menggunakan metode `.dropna()` dengan parameter **`how="any"`** yang berarti menghapus baris yang memiliki setidaknya satu nilai kosong, dan subset yang disebutkan adalah kolom "prediction". Jumlah baris setelah penghapusan data kosong dihitung kembali menggunakan metode **`.count()`** dan disimpan dalam variabel `'b'`. Selanjutnya, dihitung jumlah baris data kosong sebelum dan setelah penghapusan dengan mengurangkan nilai `'a'` dengan nilai `'b'`. Sehingga output sebagai berikut.

```
jumlah baris sebelum di hapus data kosong:  34
jumlah baris setelah di hapus data kosong:  13
jumlah baris data kosong:  21
```

Dari hasil output, terlihat bahwa sebanyak 21 baris data kosong berhasil dihapus dari total 34 baris sebelumnya, sehingga tersisa 13 baris data setelah penghapusan. Berikut tampilan 13 baris data tersebut.

```
In [50]: cleanPred.show()
```

```
+----+--------------------+---------+-------------+----------+
|soal|             jawaban|trueLabel|hashed_answer|prediction|
+----+--------------------+---------+-------------+----------+
|   1|tidak, hanya memb...|    100.0|   -256638840|  99.99993|
|   1|tidak, hanya memb...|    100.0|   -256638840|  99.99993|
|   6|ruang komputer, p...|    100.0|   1770907636|  99.99987|
|   6|ruang komputer, p...|    100.0|   1770907636|  99.99987|
|   6|ruang komputer, p...|    100.0|   1770907636|  99.99987|
|   3|hak cipta adalah ...|    83.43|  -1876419705|  83.42993|
|   5|melindungi dan me...|    74.73|  -1932865057|  74.72994|
|   9|kecocokan softwar...|    65.89|    447829639| 65.889946|
|   4|dijelaskan kepada...|    72.06|  -1834227989| 72.059906|
|   8|posisi kepala dan...|    100.0|   1782344444|  99.99991|
|   7|aturlah posisi pe...|    86.22|  -1392782412|  86.21994|
|   2|biaya dihitung be...|    100.0|   1176853507|  99.99985|
|   2|biaya dihitung be...|    100.0|   1176853507|  99.99985|
+----+--------------------+---------+-------------+----------+
```

Selanjutnya menghitung RMSE setelah menghapus baris yang memiliki nilai kosong pada kolom "prediction".

```
In [51]: rmse = evaluator.evaluate(cleanPred)
         print("Root Mean Square Error (RMSE):", rmse)
```

- **`rmse = evaluator.evaluate(cleanPred)`**: Ini menggunakan objek evaluator yang telah didefinisikan sebelumnya sebagai `RegressionEvaluator` untuk mengevaluasi model berdasarkan data `'cleanPred'`, yaitu data yang telah dibersihkan dari nilai kosong.
- **`evaluate(cleanPred)`** akan menghitung RMSE, yang merupakan metrik untuk mengukur seberapa baik prediksi model dibandingkan dengan nilai sebenarnya.

Sehingga hasil RMSE sebagai berikut.

```
Root Mean Square Error (RMSE): 0.00010312173339142772
```

Outputnya menunjukkan bahwa RMSE adalah $\mathbf{0.00010}$, yang berarti model memiliki tingkat kesalahan prediksi yang sangat kecil. RMSE yang sangat rendah menunjukkan bahwa prediksi model sangat dekat dengan nilai sebenarnya dalam dataset yang diuji.