**LAPORAN**

**RENCANA TUGAS MANDIRI (RTM) Ke-5**

**MATA KULIAH BIG DATA**

**“Automated Scoring System Menggunakan PySpark”**



**DISUSUN OLEH:**

Chelsea Ayu Adhigiadany ( 21083010028 )

**DOSEN PENGAMPU:**

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

**PROGRAM STUDI SAINS DATA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR**

**2023**

**Daftar Isi**

[**Import modul** 3](#_Toc134650188)

[**Membuat SparkSession** 3](#_Toc134650189)

[**Mendefinisikan skema dari dataset** 3](#_Toc134650190)

[**Load dataset** 5](#_Toc134650191)

[**Spliting data pertama** 6](#_Toc134650192)

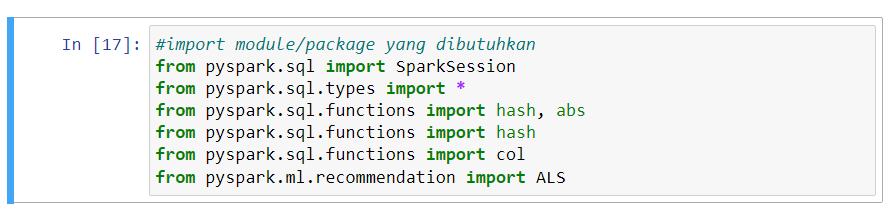
[**Hashing variabel** 6](#_Toc134650193)

[**Spliting data kedua** 7](#_Toc134650194)

[**Alternating Least Square** 7](#_Toc134650195)

[**Hasil akhir** 8](#_Toc134650196)

# **Import modul**

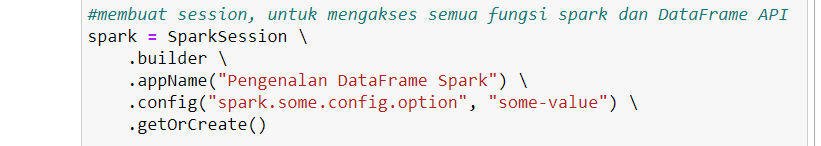


Langkah awal sebelum melakukan proses automated scoring tentunya yaitu menginport modul-modul yang akan digunakan selama proses, Adapun modul yang digunakan yaitu :

* SparkSession dari modul pyspark.sql digunakan untuk membuat sebuah SparkSession.
* StructType dan StructField dari modul pyspark.sql.types digunakan untuk mendefinisikan skema dari data yang akan dibaca.
* hash, abs, dan col dari modul pyspark.sql.functions digunakan untuk melakukan operasi pada DataFrame.
* ALS dari modul pyspark.ml.recommendation digunakan untuk mendefinisikan model ALS untuk collaborative filtering

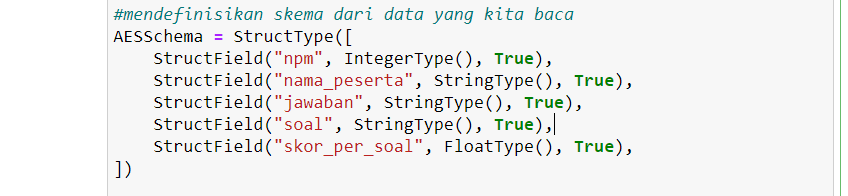
# **Membuat SparkSession**

Langkah selanjutnya yaitu membuat SparkSession untuk mengakses semua fungsi spark dan DataFrame API

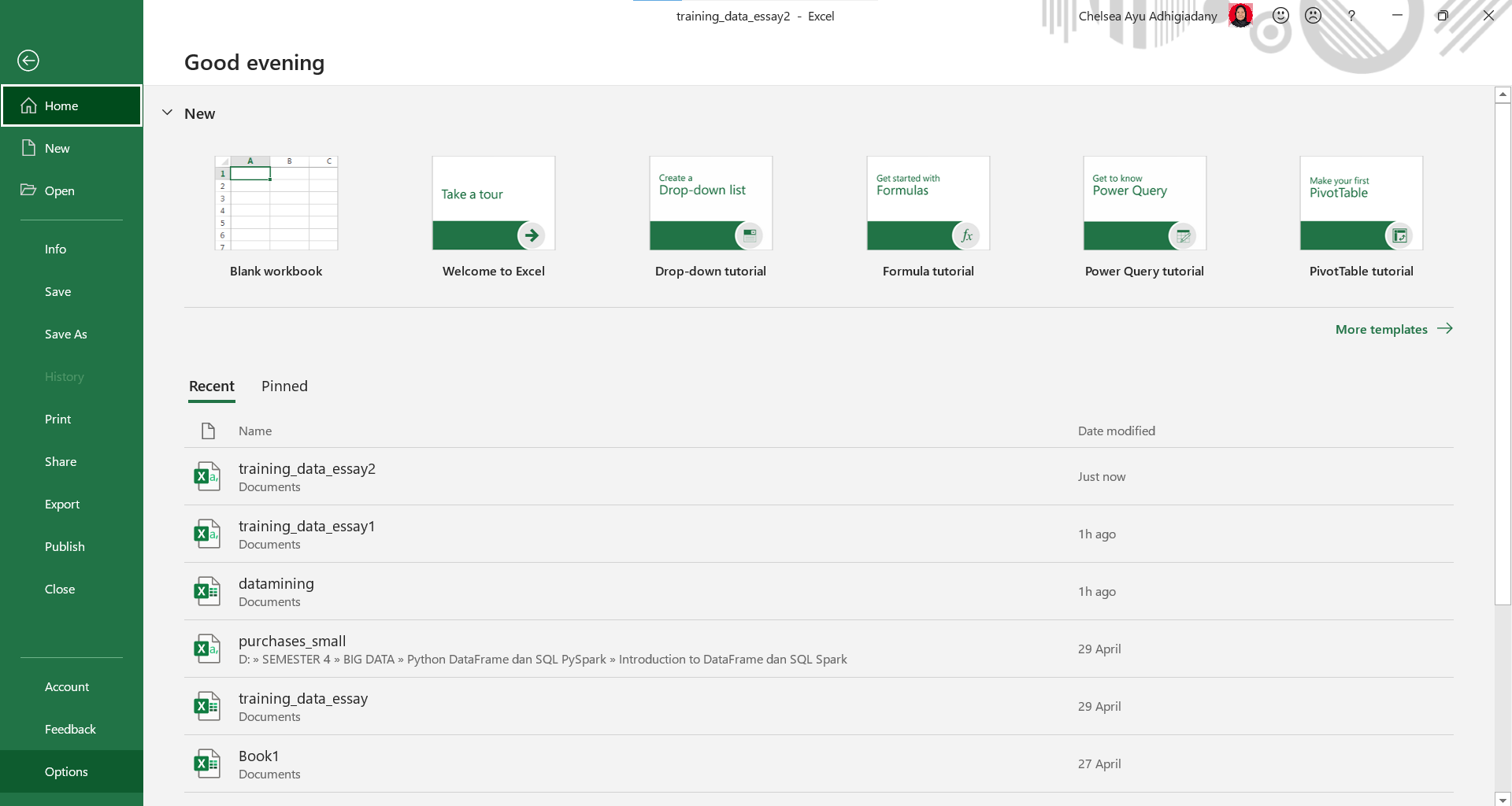


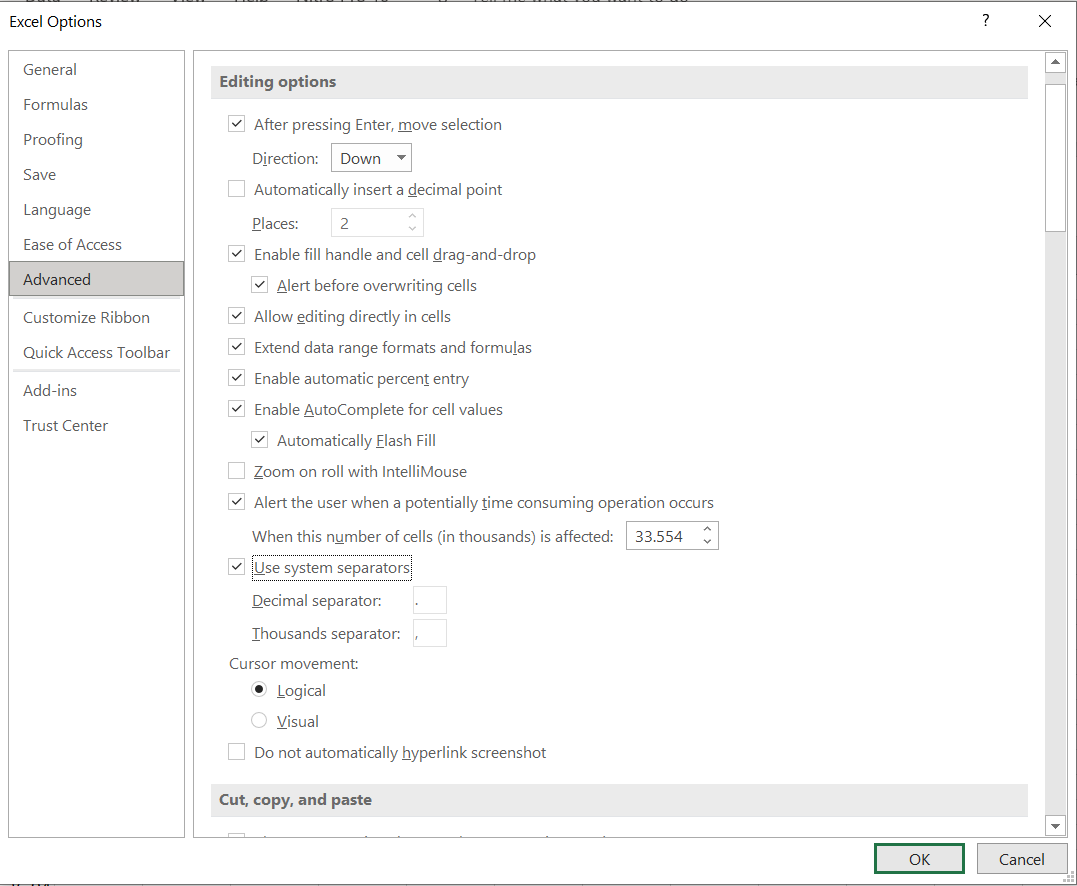
**builder** adalah API untuk membuat session. Dalam kasus ini, kita memberi nama aplikasi "Pengenalan DataFrame Spark" dan mengatur konfigurasi opsional **spark.some.config.option** dengan nilai "some-value". Jika sesi sudah ada, maka kita bisa menggunakan session yang sudah ada tersebut dengan menggunakan **.getOrCreate()**

# **Mendefinisikan skema dari dataset**



Script tersebut mendefinisikan skema atau struktur dari DataFrame yang akan dibaca oleh Spark. Skema tersebut terdiri dari lima kolom, yaitu **npm** dengan tipe data Integer, **nama\_peserta** dengan tipe data String, **jawaban** dengan tipe data String, **soal** dengan tipe data String, dan **skor\_per\_soal** dengan tipe data Float. Struktur skema ini akan digunakan untuk membaca file CSV yang berisi data yang ingin diproses dengan Spark. Pada tahap ini, pastikan juga bahwa format penulisan decimal pada dataset harus berbentuk titik(.) agar bisa terbaca oleh Spark, jika format penulisan decimal masih menggunakan pemisah koma(,) maka lakukan setiing agar berubah menjadi titik dengan cara berikut

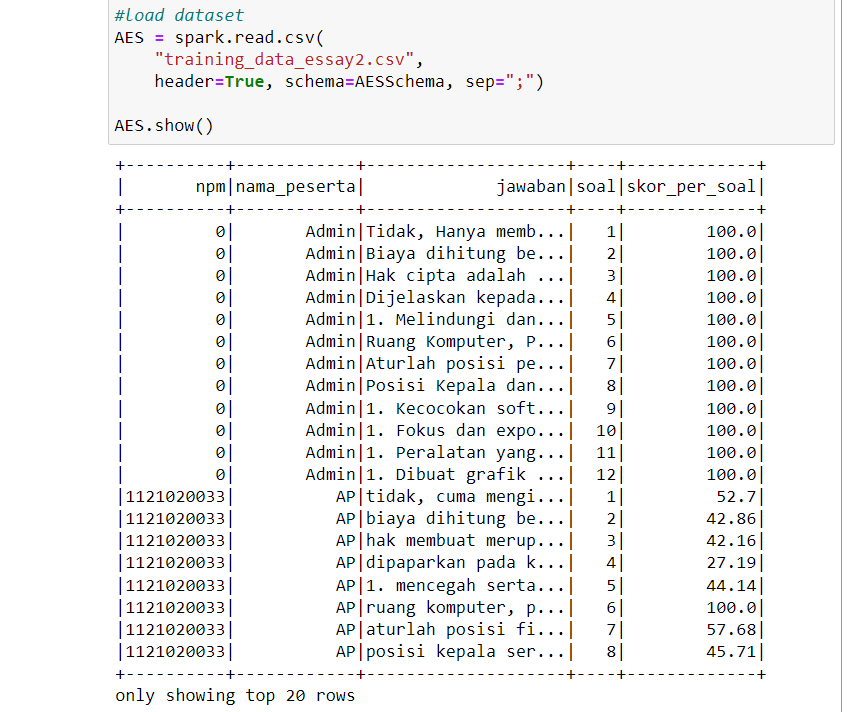




Pastikan sebelum mengaktifkan use system separator, decimal separatornya adalah titik, bukan koma.

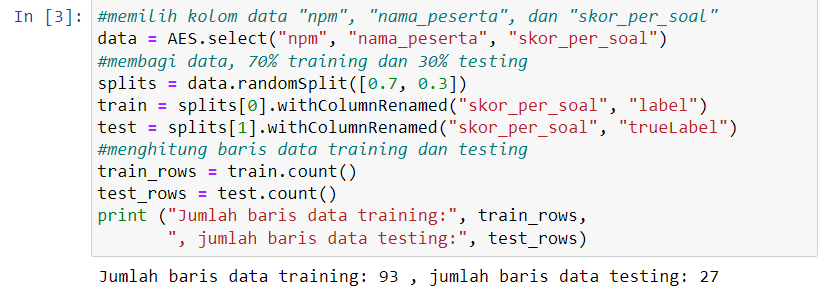
# **Load dataset**

Ketika dataset sudah siap dan semua skema sudah dibuat, maka kita bisa melalukan loading dataset.



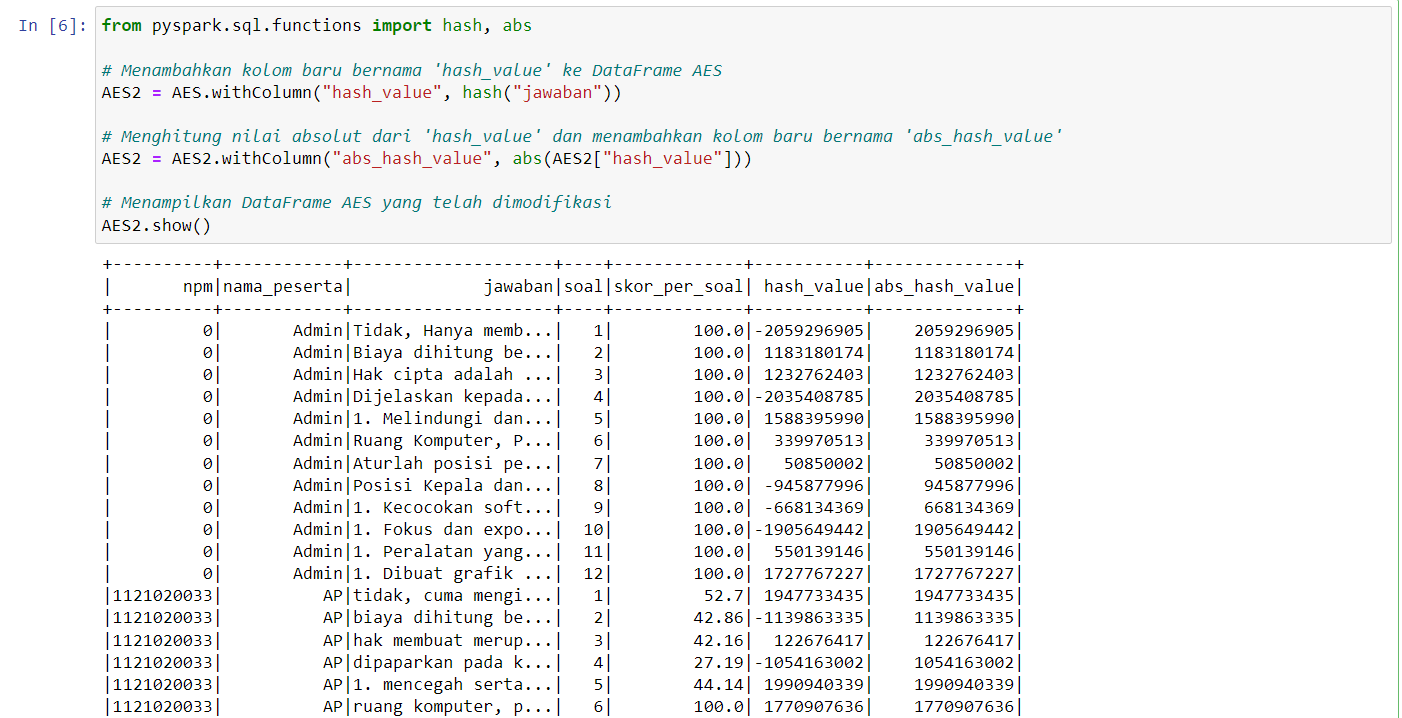
Script tersebut digunakan untuk membaca data dari file CSV "training\_data\_essay2.csv" dengan menggunakan SparkSession dan memuat data ke dalam sebuah DataFrame bernama "AES". Pada saat membaca file CSV, opsi "header=True" menandakan bahwa baris pertama dari file CSV akan digunakan sebagai header kolom. Opsi "schema=AESSchema" menginformasikan skema dari DataFrame yang ingin dibuat. Skema tersebut didefinisikan sebelumnya dalam variabel AESSchema. Opsi "sep=';'" menunjukkan bahwa separator dalam file CSV adalah tanda semicolon (;). Setelah berhasil memuat data ke dalam DataFrame AES, kemudian dilakukan pemanggilan method show() pada DataFrame AES untuk menampilkan beberapa baris pertama data dari DataFrame tersebut.

# **Spliting data pertama**



Script tersebut digunakan untuk memilih kolom tertentu dari DataFrame AES dan mendeklarasikannya dengan nama ‘data’, yang berisi kolom "npm", "nama\_peserta", dan "skor\_per\_soal". Kemudian data dibagi menjadi 70% untuk data pelatihan dan 30% untuk data pengujian menggunakan fungsi randomSplit(). Selanjutnya, dilakukan renaming pada kolom "skor\_per\_soal" di data pelatihan menjadi "label" dan di data pengujian menjadi "trueLabel". Terakhir, jumlah baris data pelatihan dan pengujian dihitung dan dicetak ke layar menggunakan fungsi count() dan print(). Dari hasil tersebut didapatkan bahwa banyaknya baris untuk data training adalah sebanyak 93 baris sedangkan data testing adalah 27 baris.

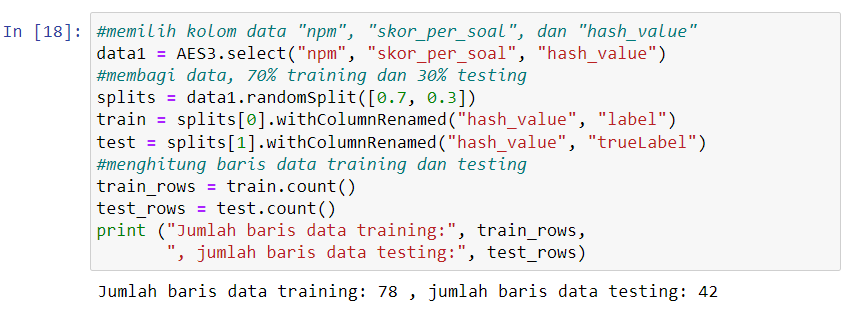
# **Hashing variabel**



Script di atas digunakan untuk menambahkan kolom baru ke dalam DataFrame AES. Kolom baru tersebut dinamakan 'hash\_value' dan nilainya dihitung dengan menggunakan fungsi hash() pada kolom 'jawaban'. Selain itu, script juga menambahkan kolom baru bernama 'abs\_hash\_value', yang berisi nilai absolut dari 'hash\_value'. Terakhir, DataFrame AES2 ditampilkan dengan menggunakan metode show() untuk memeriksa kolom baru yang telah ditambahkan.

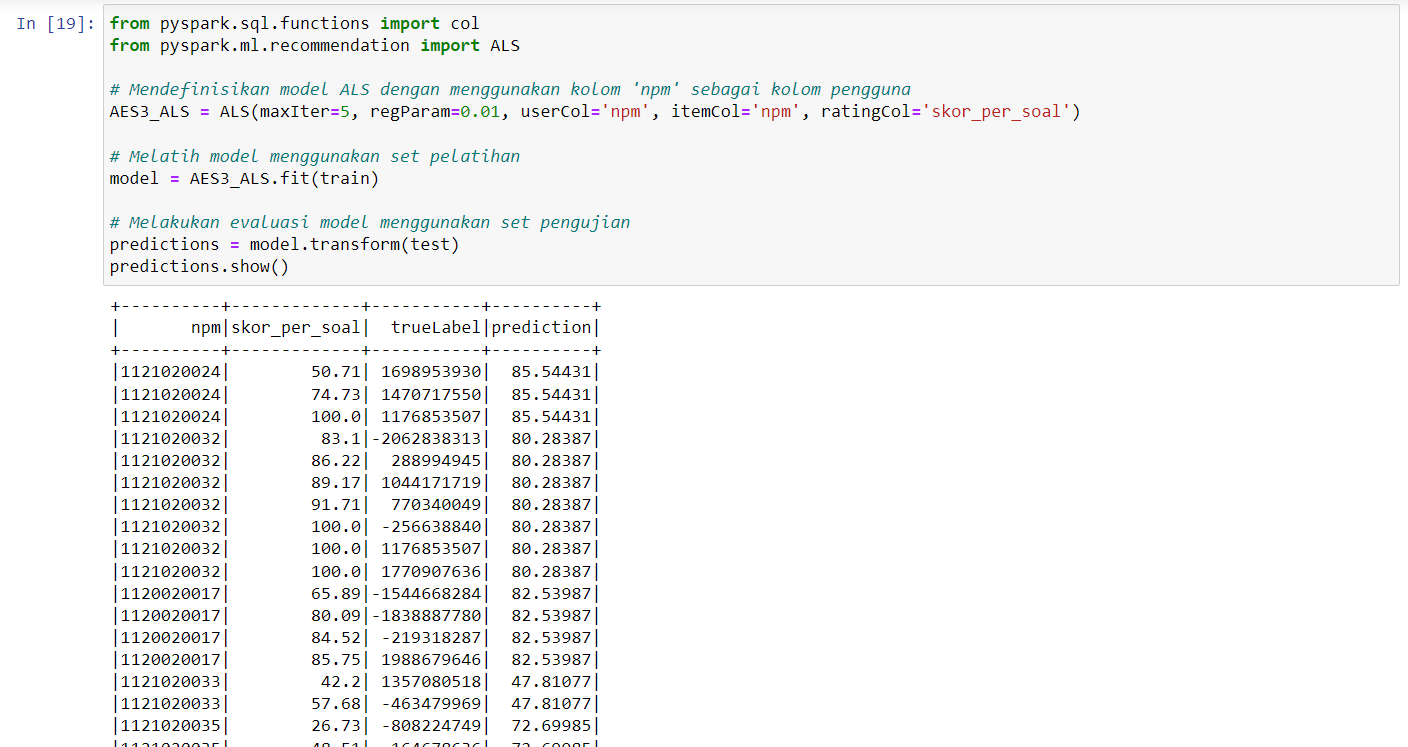
# **Spliting data kedua**

Karena adanya penambahan kolom baru yang merupakan hasil dari hashing kolom jawaban, maka dilakukan splitting data lagi dengan menjadikan kolom “hash\_value” sebagai data training dan data testing.



Script tersebut digunakan untuk memilih kolom tertentu dari DataFrame AES dan mendeklarasikannya dengan nama ‘dat1a’, yang berisi kolom "npm", "skor\_per\_soal", dan "hash\_value". Kemudian data dibagi menjadi 70% untuk data pelatihan dan 30% untuk data pengujian menggunakan fungsi randomSplit(). Selanjutnya, dilakukan renaming pada kolom "hash\_value" di data pelatihan menjadi "label" dan di data pengujian menjadi "trueLabel". Terakhir, jumlah baris data pelatihan dan pengujian dihitung dan dicetak ke layar menggunakan fungsi count() dan print(). Dari hasil tersebut didapatkan bahwa banyaknya baris untuk data training adalah sebanyak 78 baris sedangkan data testing adalah 42 baris.

# **Alternating Least Square**



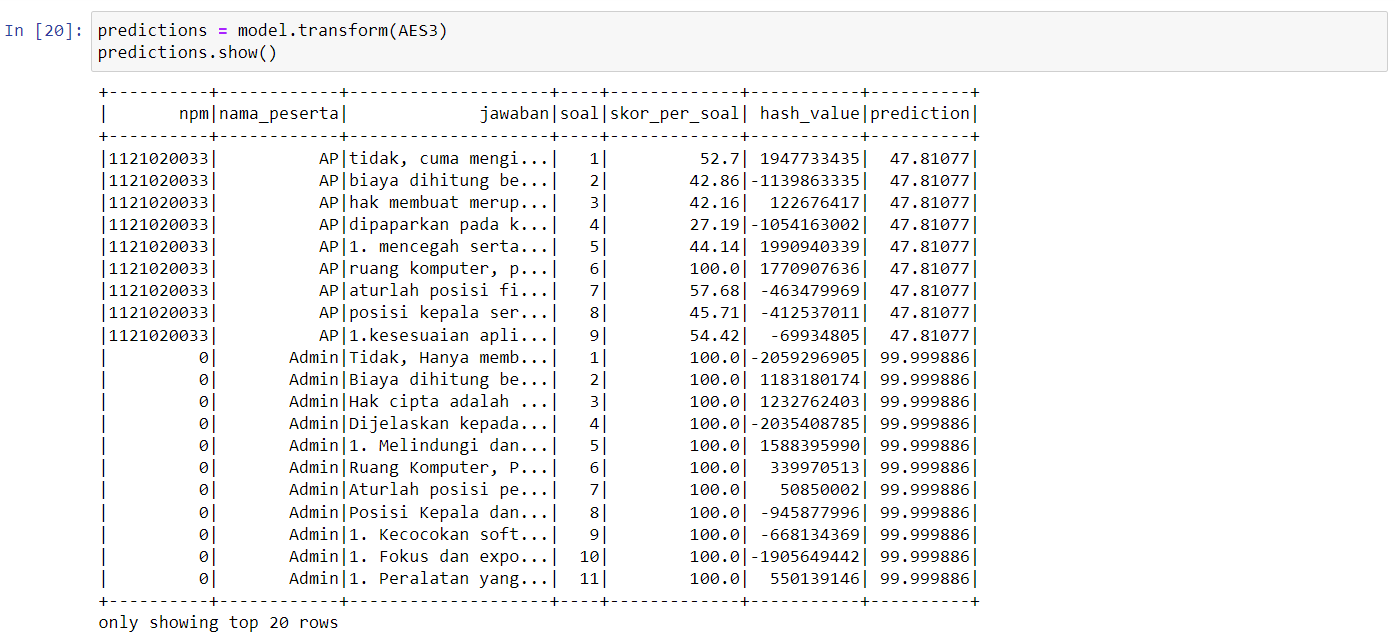
Script tersebut melakukan proses pemodelan dan evaluasi menggunakan ALS (Alternating Least Squares) pada dataset AES yang telah diproses sebelumnya. ALS merupakan algoritma yang umum digunakan dalam collaborative filtering untuk merekomendasikan produk atau item berdasarkan preferensi pengguna sebelumnya.

Pertama, model ALS didefinisikan dengan mengatur beberapa parameter, seperti **maxIter** untuk jumlah iterasi yang dilakukan, **regParam** untuk parameter regularisasi, **userCol** untuk menentukan kolom pengguna (dalam hal ini, kolom "npm" digunakan), **itemCol** untuk menentukan kolom item, dan **ratingCol** untuk menentukan kolom rating (dalam hal ini, kolom "skor\_per\_soal" digunakan).

Selanjutnya, model tersebut dilatih dengan menggunakan set pelatihan **train**. Setelah model dilatih, evaluasi dilakukan dengan menggunakan set pengujian **test** dan prediksi hasilnya disimpan pada variabel **predictions**.

Terakhir, hasil prediksi tersebut ditampilkan dengan menggunakan fungsi **show()** pada DataFrame **predictions**. Hasil prediksi ini akan berisi rating prediksi untuk setiap pasangan pengguna dan item pada set pengujian **test**, yang akan digunakan untuk mengevaluasi performa model ALS.

# **Hasil akhir**



Script ini, **model.transform(test)** digunakan untuk melakukan prediksi nilai skor dari kolom **hash\_value**. Cara kerja prediksi ini yaitu memberikan nilai dari setiap jawaban pada kolom **jawaban** yang telah dilakukan proses hashing menjadi **hash\_value**. Pada bagian jawaban oleh admin, didapatkan nilai prediksi sebesar 99% karena itu merupakan jawaban kunci yang menjadi acuan, sedangkan pada bagian jawaban oleh AP dan lain-lain mendapatkan nilai dibawah admin. Dapat dilihat bahwasanya hasil dari prediksi hampir sama dengan nilai dari kolom **skor\_per\_soal** yang mana prediksi tersebut sesuai dengan skor yang sudah tercantum di dataset tersebut.