

**LAPORAN**  
**RENCANA TUGAS MANDIRI (RTM) Ke-5**  
**MATA KULIAH BIG DATA C**  
**“AUTOMATIC ESSAY SCORING WITH PYSPARK”**



**DISUSUN OLEH:**

Mohamad Ibnu Fajar Maulana (21083010106)

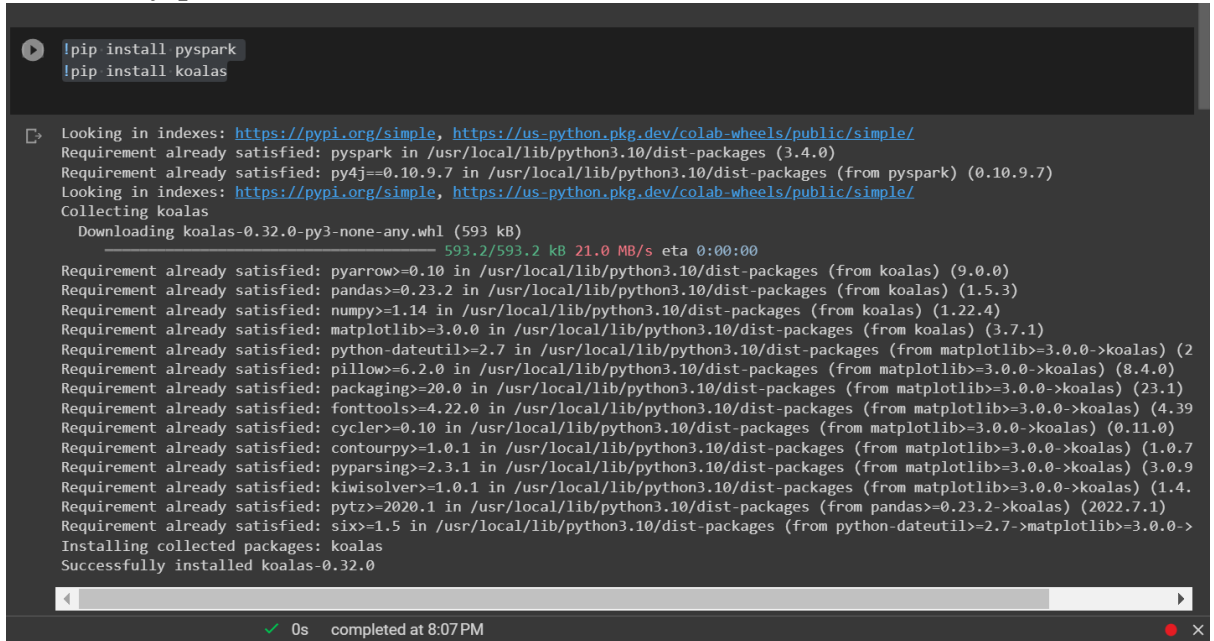
**DOSEN PENGAMPU:**

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

**PROGRAM STUDI SAINS DATA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA**  
**TIMUR**

**2022**

## 1. Instalasi Pyspark



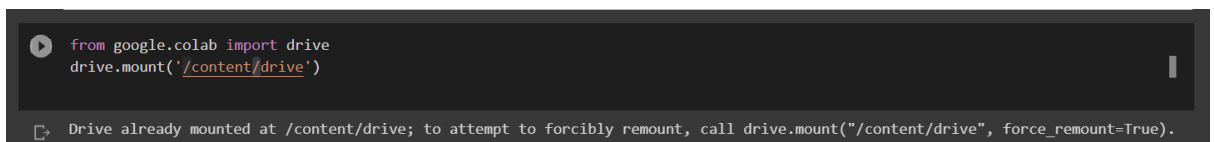
```
!pip install pyspark
!pip install koalas

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pyspark in /usr/local/lib/python3.10/dist-packages (3.4.0)
Requirement already satisfied: py4j==0.10.9.7 in /usr/local/lib/python3.10/dist-packages (from pyspark) (0.10.9.7)
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting koalas
  Downloading koalas-0.32.0-py3-none-any.whl (593 kB)
    593.2/593.2 kB 21.0 MB/s eta 0:00:00
Requirement already satisfied: pyarrow>=0.10 in /usr/local/lib/python3.10/dist-packages (from koalas) (9.0.0)
Requirement already satisfied: pandas>=0.23.2 in /usr/local/lib/python3.10/dist-packages (from koalas) (1.5.3)
Requirement already satisfied: numpy>=1.14 in /usr/local/lib/python3.10/dist-packages (from koalas) (1.22.4)
Requirement already satisfied: matplotlib>=3.0.0 in /usr/local/lib/python3.10/dist-packages (from koalas) (3.7.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (8.4.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (23.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (4.39)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (0.11.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (1.0.7)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (3.0.9)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.0.0->koalas) (1.4)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=0.23.2->koalas) (2022.7.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib>=3.0.0->koalas) (1.16.0)
Installing collected packages: koalas
Successfully installed koalas-0.32.0

0s completed at 8:07 PM
```

- **pyspark** adalah pustaka yang memungkinkan pengguna untuk mengakses Apache Spark menggunakan bahasa pemrograman Python. Apache Spark sendiri adalah kerangka kerja yang dirancang untuk memproses data secara terdistribusi, dengan dukungan untuk berbagai bahasa pemrograman dan jenis data.
- **koalas** adalah pustaka yang memungkinkan pengguna untuk menggunakan antarmuka Pandas untuk bekerja dengan data terdistribusi di Apache Spark. Dengan Koalas, pengguna dapat menulis kode Pandas biasa dan menggunakannya untuk memproses data yang jauh lebih besar daripada yang dapat ditangani oleh satu mesin.

## 2. Pre-Processing



```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

google.colab adalah modul Python yang menyediakan alat-alat dan utilitas untuk menjalankan kode di Colab. Dalam kode di atas, kita mengimpor fungsi drive dari modul google.colab, yang digunakan untuk mengakses Google Drive dari Colab.

Selanjutnya, kita memanggil fungsi mount dan memberikan argumen /content/drive kepadanya. Fungsi ini akan meminta izin akses ke Google Drive dan akan menghasilkan tautan yang harus diikuti pengguna untuk memberikan izin tersebut. Setelah izin diberikan, Colab akan dapat mengakses Google Drive melalui direktori /content/drive.

```
[ ] from pyspark.sql import SparkSession
    from pyspark.ml.feature import Tokenizer, CountVectorizer, IDF
    from pyspark.ml.regression import LinearRegression
    from pyspark.ml.evaluation import RegressionEvaluator
```

- pyspark.sql adalah modul yang menyediakan API untuk memproses data dalam format tabel menggunakan Spark SQL.
- SparkSession adalah kelas yang menyediakan antarmuka untuk mengakses Spark dalam pengolahan data dengan PySpark.
- pyspark.ml.feature adalah modul yang menyediakan API untuk fitur pemrosesan dan transformasi data.
- Tokenizer adalah salah satu fitur Transformer yang digunakan untuk memecah teks menjadi token.
- CountVectorizer adalah fitur Transformer yang digunakan untuk menghitung frekuensi kemunculan kata dalam dokumen.
- IDF adalah Transformer yang digunakan untuk menghitung bobot kata dalam dokumen berdasarkan frekuensi kemunculannya.
- pyspark.ml.regression adalah modul yang menyediakan API untuk model regresi.
- LinearRegression adalah salah satu algoritma regresi yang umum digunakan.
- pyspark.ml.evaluation adalah modul yang menyediakan API untuk evaluasi model.
- RegressionEvaluator adalah salah satu evaluator yang digunakan untuk menghitung metrik evaluasi dalam model regresi.

Dengan mengimpor modul ini, pengguna dapat melakukan pemrosesan data, transformasi fitur, dan membangun model regresi dalam Apache Spark menggunakan bahasa pemrograman Python.

```
[ ] spark = SparkSession.builder.appName("Automatic Essay Scoring").getOrCreate()
```

- SparkSession adalah kelas yang menyediakan antarmuka untuk mengakses Spark dalam pengolahan data dengan PySpark. Dalam kode di atas, kita menggunakan metode builder untuk membuat objek SparkSession.
- Pada objek SparkSession, kita memberikan nama aplikasi dengan metode appName("Automatic Essay Scoring"). Nama aplikasi ini akan muncul pada antarmuka pengguna Spark.

Jika sudah ada sesi Spark yang berjalan, maka sesi tersebut akan digunakan. Jika tidak ada, maka akan dibuat sesi baru dengan nama aplikasi yang diberikan. Hal ini dilakukan dengan memanggil metode getOrCreate(). Dalam praktiknya, objek spark ini akan digunakan untuk membaca dan memproses data dalam format tertentu, membangun model machine learning, dan melakukan evaluasi kinerja pada model.

- Kemudian kita akan menampilkan DataFrame

```
[ ] df = spark.read.csv('/content/drive/My Drive/training_data_essay.csv', header=True, inferSchema=True)

df.printSchema()
df.show()
```

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

| npm         | nama_peserta | jawaban               | soal | skor_per_soal |
|-------------|--------------|-----------------------|------|---------------|
| 0           | Admin        | Tidak, Hanya memb...  | 1    | 100.0         |
| 0           | Admin        | biaya dihitung be...  | 2    | 100.0         |
| 0           | Admin        | law cipra adalah ...  | 3    | 100.0         |
| 0           | Admin        | dijelaskan kepada...  | 4    | 100.0         |
| 0           | Admin        | 1. Melindungi dan...  | 5    | 100.0         |
| 0           | Admin        | uang komputer, p...   | 6    | 100.0         |
| 0           | Admin        | Atur lah posisi pe... | 7    | 100.0         |
| 0           | Admin        | Posisi kepala dan...  | 8    | 100.0         |
| 0           | Admin        | 1. Kecelakaan soft... | 9    | 100.0         |
| 0           | Admin        | 1. Fokus dan expo...  | 10   | 100.0         |
| 0           | Admin        | 1. Peralatan yang...  | 11   | 100.0         |
| 0           | Admin        | 1. dibuat grafik ...  | 12   | 100.0         |
| 11218208033 | AP           | tidak, cuma mengi...  | 1    | 52.7          |
| 11218208033 | AP           | biaya dihitung be...  | 2    | 42.86         |
| 11218208033 | AP           | hak membuat merup...  | 3    | 42.16         |
| 11218208033 | AP           | dipaparkan pada k...  | 4    | 22.19         |
| 11218208033 | AP           | 1. mencegah serta...  | 5    | 44.14         |
| 11218208033 | AP           | uang komputer, p...   | 6    | 100.0         |
| 11218208033 | AP           | atur lah posisi fi... | 7    | 57.68         |
| 11218208033 | AP           | posisi kepala ser...  | 8    | 45.71         |

only showing top 20 rows

- Gambar diatas menampilkan DataFrame dengan Pyspark
- Selanjutnya

```
from pyspark.ml.feature import StringIndexer
indexer = StringIndexer(inputCol='soal', outputCol='skor_per_soal')
df = indexer.fit(df).transform(df)
```

- Kemudian melakukan training data jawaban(essay)

```
[ ] from pyspark.ml.feature import Tokenizer, StopWordsRemover
from pyspark.ml.feature import HashingTF, IDF

# Tokenize the essay text
tokenizer = Tokenizer(inputCol='jawaban', outputCol='words')
df = tokenizer.transform(df)

# Remove stop words
stop_words = StopWordsRemover(inputCol='words', outputCol='filtered_words')
df = stop_words.transform(df)

# Apply TF-IDF
hashingTF = HashingTF(inputCol='filtered_words', outputCol='raw_features', numFeatures=10000)
featurized_data = hashingTF.transform(df)
idf = IDF(inputCol='raw_features', outputCol='features')
idf_model = idf.fit(featurized_data)
df = idf_model.transform(featurized_data)
```

- Dimana nantinya akan dibagi data training 80% dan data testing 20%]

```
[ ] (trainingData, testData) = df.randomSplit([0.8, 0.2], seed=42)
```

- Selanjutnya

```
[ ] from pyspark.ml.classification import RandomForestClassifier

rf = RandomForestClassifier(labelCol='skor_per_soal', featuresCol='features', numTrees=10)
model = rf.fit(trainingData)
```

- Gambar diatas melakukan automatic system scoring dengan pyspark clasification dengan merandom value
- Selanjutnya

```
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

predictions = model.transform(testData)
evaluator = MulticlassClassificationEvaluator(labelCol='skor_per_soal', predictionCol='prediction', metricNames='accuracy')
accuracy = evaluator.evaluate(predictions)
print('Accuracy:', accuracy)
```

Accuracy: 0.9545454545454546

- Menghitung ketepatan skor\_per\_soal dengan automaatc essay scoring yang mana mendapatkan Accuracy: 0.9545454545454546