Nama : Muhammad Rifky Harto Biantoro

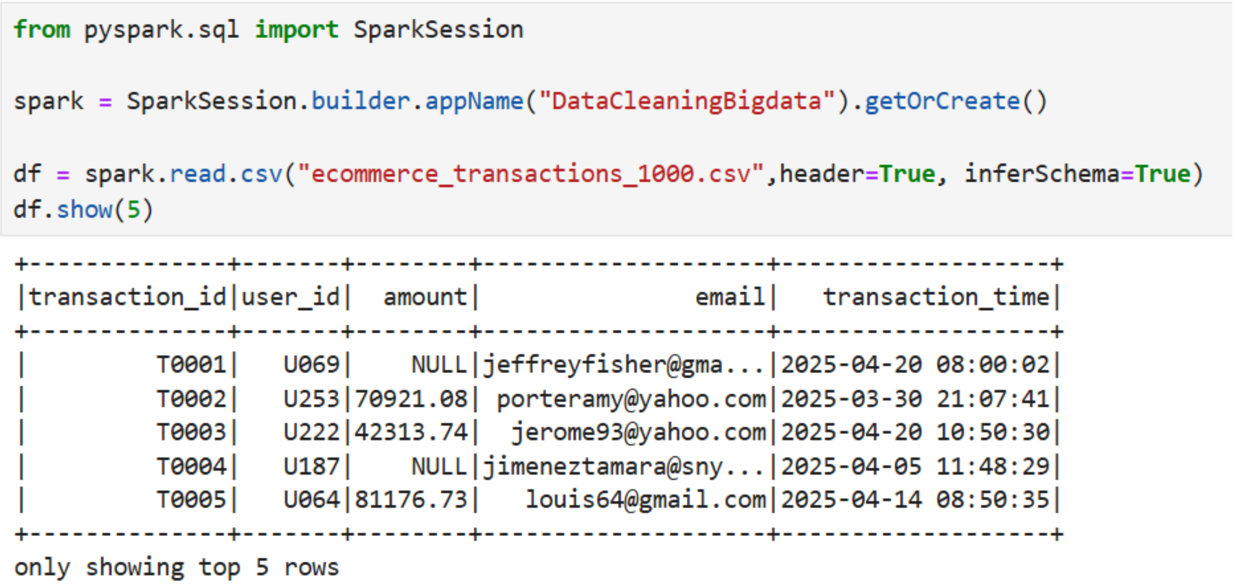
Kelas : TI-3D

NIM : 2241720176

Matkul : BigData

Praktikum 1

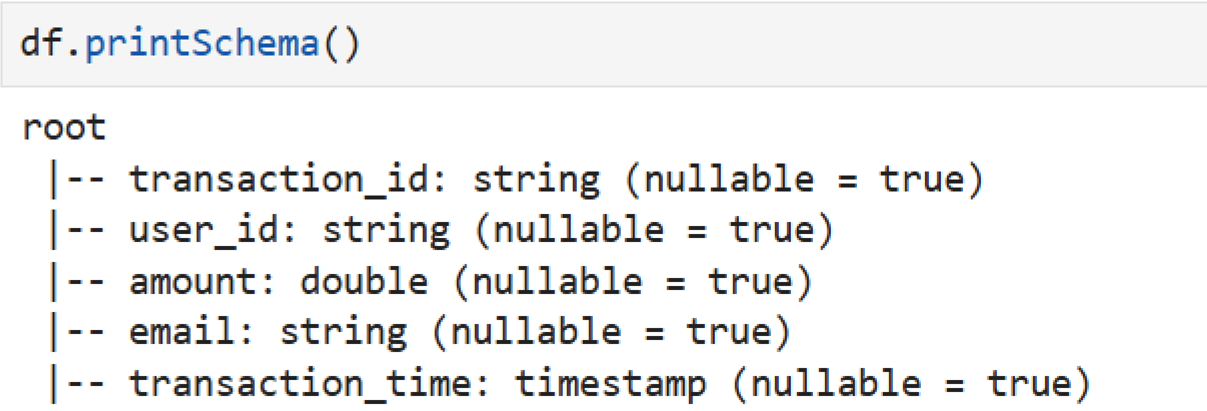
1. Load Dataset



```
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("DataCleaningBigdata").getOrCreate()

df = spark.read.csv("ecommerce_transactions_1000.csv",header=True, inferSchema=True)
df.show(5)
```

```
+--------------+-------+--------+--------------------+-------------------+
|transaction_id|user_id|  amount|               email|   transaction_time|
+--------------+-------+--------+--------------------+-------------------+
|         T0001|   U069|    NULL|jeffreyfisher@gma...|2025-04-20 08:00:02|
|         T0002|   U253|70921.08| porteramy@yahoo.com|2025-03-30 21:07:41|
|         T0003|   U222|42313.74|  jerome93@yahoo.com|2025-04-20 10:50:30|
|         T0004|   U187|    NULL|jimeneztamara@sny...|2025-04-05 11:48:29|
|         T0005|   U064|81176.73|   louis64@gmail.com|2025-04-14 08:50:35|
+--------------+-------+--------+--------------------+-------------------+
only showing top 5 rows
```

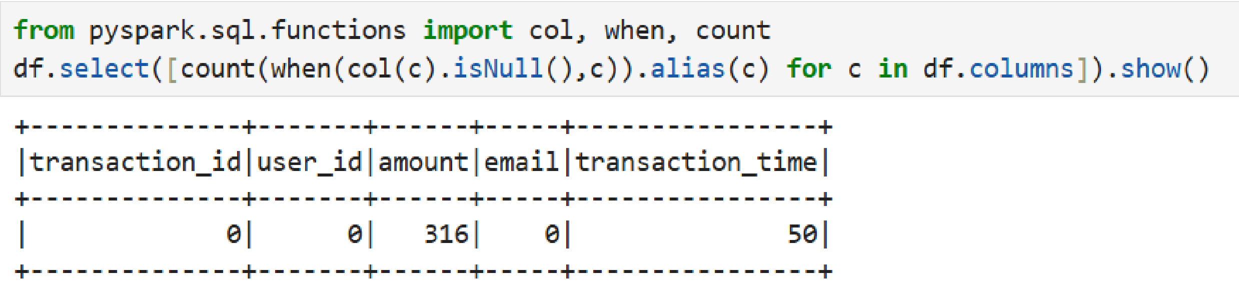
2. Inspeksi Data
   a. Lihat struktur schema



```
df.printSchema()
```

```
root
 |-- transaction_id: string (nullable = true)
 |-- user_id: string (nullable = true)
 |-- amount: double (nullable = true)
 |-- email: string (nullable = true)
 |-- transaction_time: timestamp (nullable = true)
```

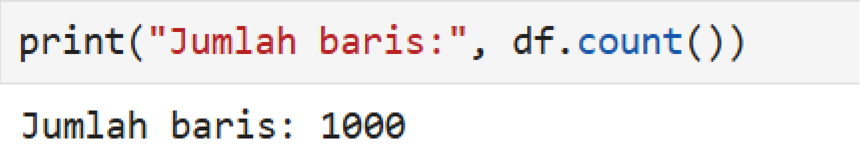
   b. Hitung missing values setiap kolom:



```
from pyspark.sql.functions import col, when, count
df.select([count(when(col(c).isNull(),c)).alias(c) for c in df.columns]).show()
```

```
+--------------+-------+------+-----+----------------+
|transaction_id|user_id|amount|email|transaction_time|
+--------------+-------+------+-----+----------------+
|             0|      0|   316|    0|              50|
+--------------+-------+------+-----+----------------+
```

c. Hitung jumlah total data



```
print("Jumlah baris:", df.count())
```

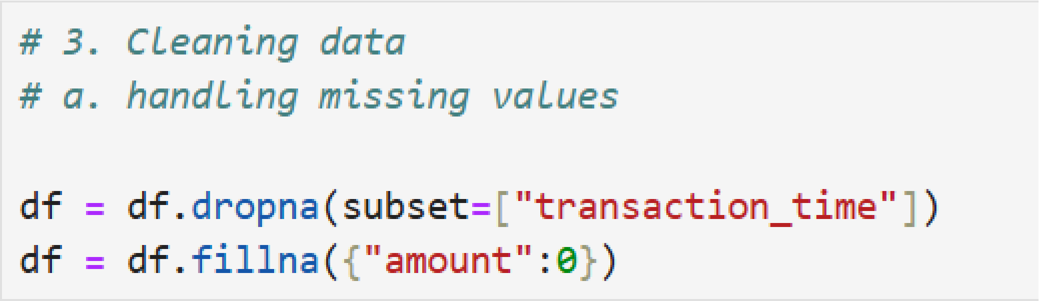
```
Jumlah baris: 1000
```

3. Cleaning Data

   a. Handling Missing values

   Drop transaksi yang tidak memiliki transaction_time

   Isi nilai kosong pada amount dengan 0

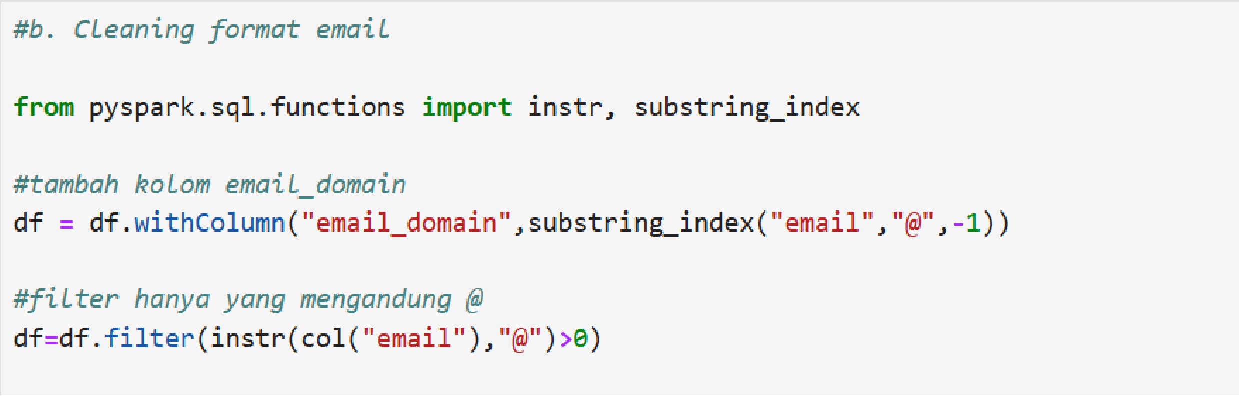


```
# 3. Cleaning data
# a. handling missing values

df = df.dropna(subset=["transaction_time"])
df = df.fillna({"amount":0})
```

   b. Cleaning format email

   Buat kolom baru email_domain yang berisi domain email

   Hapus transaksi yang emailnya tidak valid (tidak mengandung '@')

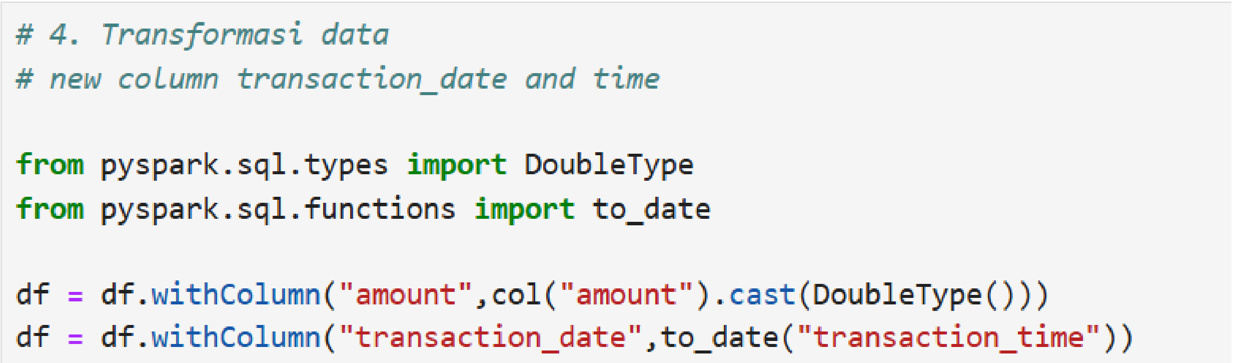


```
#b. Cleaning format email

from pyspark.sql.functions import instr, substring_index

#tambah kolom email_domain
df = df.withColumn("email_domain",substring_index("email","@",-1))

#filter hanya yang mengandung @
df=df.filter(instr(col("email"),"@")>0)
```

4. Transformasi Data

   a. Ubah kolom amount menjadi DoubleType.

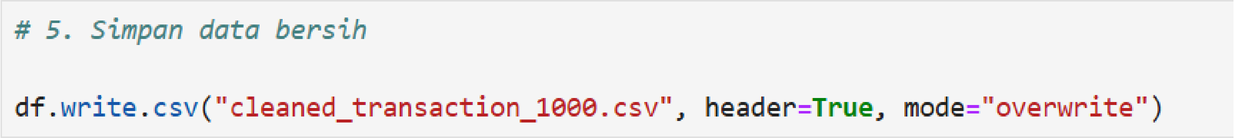
b. Tambahkan kolom baru transaction_date dari transaction_time



```
# 4. Transformasi data
# new column transaction_date and time

from pyspark.sql.types import DoubleType
from pyspark.sql.functions import to_date

df = df.withColumn("amount",col("amount").cast(DoubleType()))
df = df.withColumn("transaction_date",to_date("transaction_time"))
```

5. Simpan Data Bersih

a. Simpan dataframe hasil cleaning ke file baru

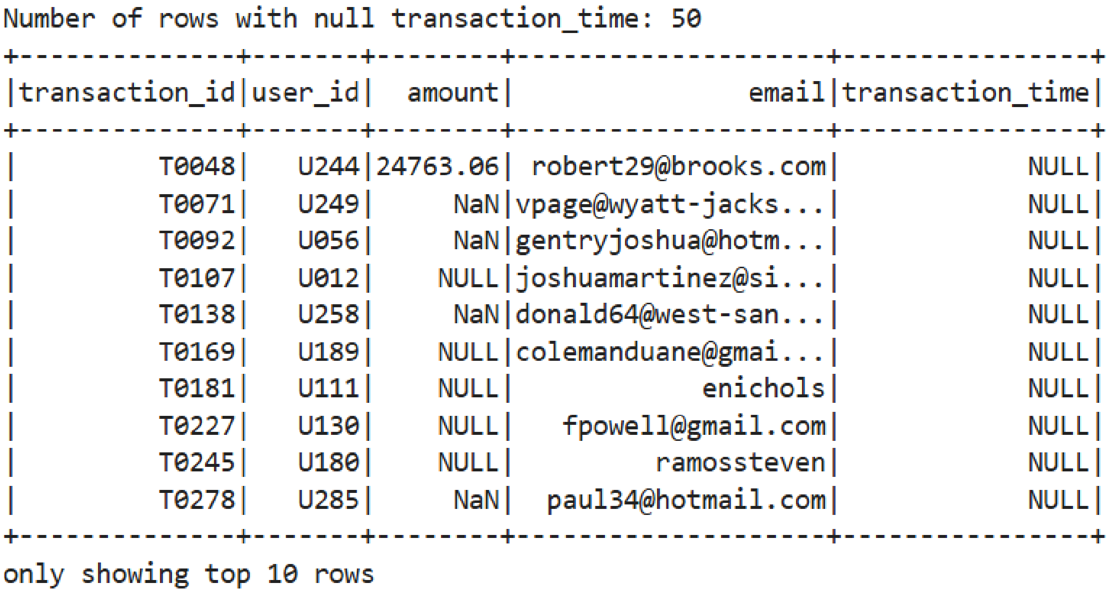


```
# 5. Simpan data bersih

df.write.csv("cleaned_transaction_1000.csv", header=True, mode="overwrite")
```

Pertanyaan:

1. Berapa banyak data yang dibuang karena transaction_time kosong?

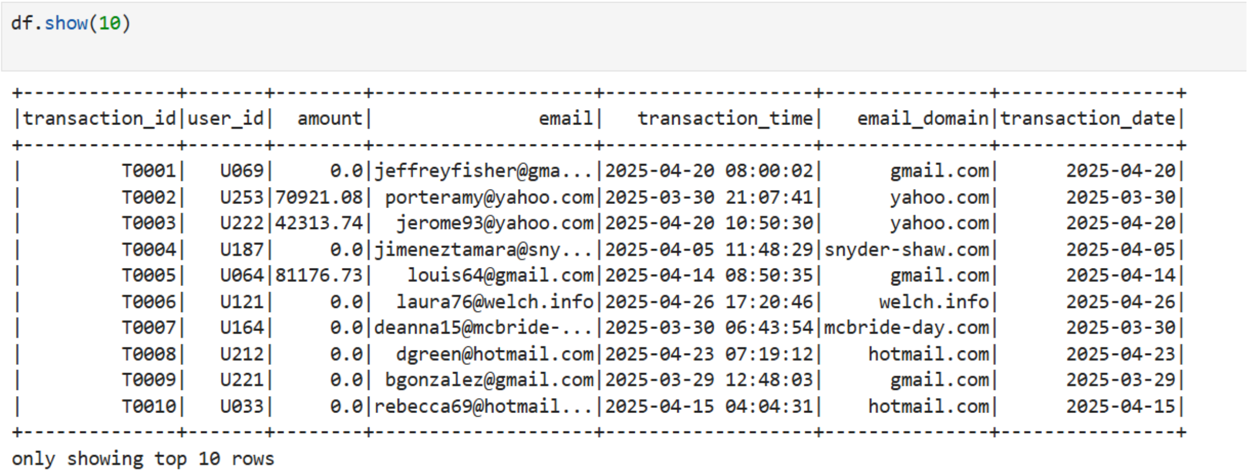
Jawab: Berikut merupakan baris data yang pada kolom transaction_time kosong



```
Number of rows with null transaction_time: 50
+--------------+-------+--------+--------------------+----------------+
|transaction_id|user_id|  amount|               email|transaction_time|
+--------------+-------+--------+--------------------+----------------+
|         T0048|   U244|24763.06| robert29@brooks.com|            NULL|
|         T0071|   U249|     NaN|vpage@wyatt-jacks...|            NULL|
|         T0092|   U056|     NaN|gentryjoshua@hotm...|            NULL|
|         T0107|   U012|    NULL|joshuamartinez@si...|            NULL|
|         T0138|   U258|     NaN|donald64@west-san...|            NULL|
|         T0169|   U189|    NULL|colemanduane@gmai...|            NULL|
|         T0181|   U111|    NULL|            enichols|            NULL|
|         T0227|   U130|    NULL|   fpowell@gmail.com|            NULL|
|         T0245|   U180|    NULL|         ramossteven|            NULL|
|         T0278|   U285|     NaN|  paul34@hotmail.com|            NULL|
+--------------+-------+--------+--------------------+----------------+
only showing top 10 rows
```

2. Apakah semua data amount sudah bertipe numerik setelah cleaning?

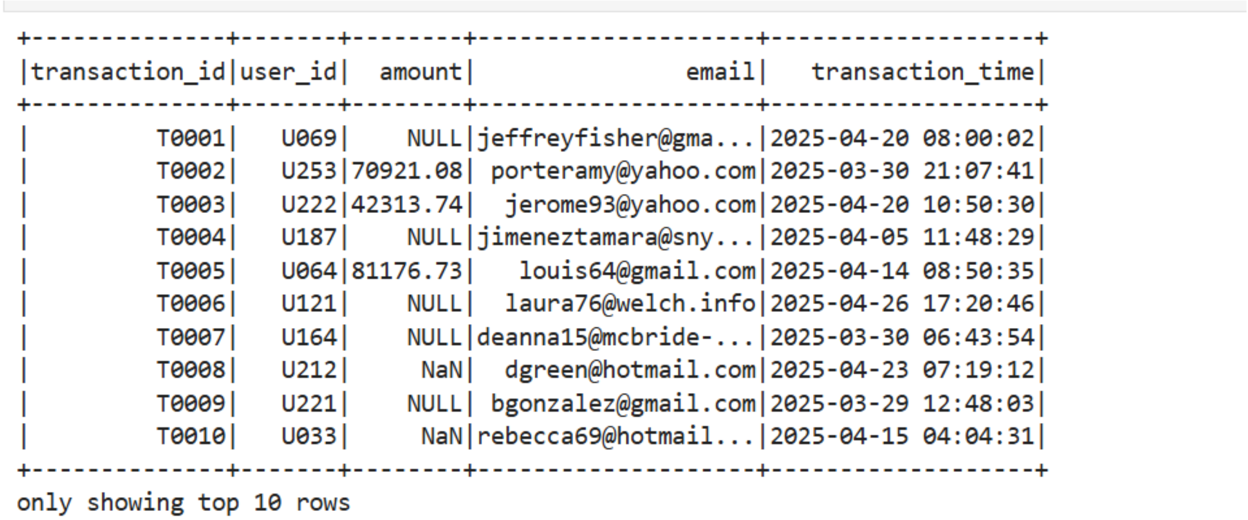
Jawab: Ya, sudah, berikut buktinya



```
df.show(10)
```

```
+--------------+-------+--------+--------------------+-------------------+---------------+----------------+
|transaction_id|user_id|  amount|               email|   transaction_time|   email_domain|transaction_date|
+--------------+-------+--------+--------------------+-------------------+---------------+----------------+
|         T0001|   U069|     0.0|jeffreyfisher@gma...|2025-04-20 08:00:02|      gmail.com|      2025-04-20|
|         T0002|   U253|70921.08| porteramy@yahoo.com|2025-03-30 21:07:41|      yahoo.com|      2025-03-30|
|         T0003|   U222|42313.74|  jerome93@yahoo.com|2025-04-20 10:50:30|      yahoo.com|      2025-04-20|
|         T0004|   U187|     0.0|jimeneztamara@sny...|2025-04-05 11:48:29|snyder-shaw.com|      2025-04-05|
|         T0005|   U064|81176.73|   louis64@gmail.com|2025-04-14 08:50:35|      gmail.com|      2025-04-14|
|         T0006|   U121|     0.0|  laura76@welch.info|2025-04-26 17:20:46|     welch.info|      2025-04-26|
|         T0007|   U164|     0.0|deanna15@mcbride-...|2025-03-30 06:43:54|mcbride-day.com|      2025-03-30|
|         T0008|   U212|     0.0|  dgreen@hotmail.com|2025-04-23 07:19:12|    hotmail.com|      2025-04-23|
|         T0009|   U221|     0.0| bgonzalez@gmail.com|2025-03-29 12:48:03|      gmail.com|      2025-03-29|
|         T0010|   U033|     0.0|rebecca69@hotmail...|2025-04-15 04:04:31|    hotmail.com|      2025-04-15|
+--------------+-------+--------+--------------------+-------------------+---------------+----------------+
only showing top 10 rows
```

Sebelum cleaning:



```
+--------------+-------+--------+--------------------+-------------------+
|transaction_id|user_id|  amount|               email|   transaction_time|
+--------------+-------+--------+--------------------+-------------------+
|         T0001|   U069|    NULL|jeffreyfisher@gma...|2025-04-20 08:00:02|
|         T0002|   U253|70921.08| porteramy@yahoo.com|2025-03-30 21:07:41|
|         T0003|   U222|42313.74|  jerome93@yahoo.com|2025-04-20 10:50:30|
|         T0004|   U187|    NULL|jimeneztamara@sny...|2025-04-05 11:48:29|
|         T0005|   U064|81176.73|   louis64@gmail.com|2025-04-14 08:50:35|
|         T0006|   U121|    NULL|  laura76@welch.info|2025-04-26 17:20:46|
|         T0007|   U164|    NULL|deanna15@mcbride-...|2025-03-30 06:43:54|
|         T0008|   U212|     NaN|  dgreen@hotmail.com|2025-04-23 07:19:12|
|         T0009|   U221|    NULL| bgonzalez@gmail.com|2025-03-29 12:48:03|
|         T0010|   U033|     NaN|rebecca69@hotmail...|2025-04-15 04:04:31|
+--------------+-------+--------+--------------------+-------------------+
only showing top 10 rows
```

3. Kenapa lebih baik memperbnaiki email invalid sebelum menganalisis data?
   Jawab : Menurut saya hal ini meminimalisir data bias, serta email yang valid dapat menjadi jaminan kualitas data, secara langsung kita akan lebih percaya terhadap email yang valid dibanding invalid.

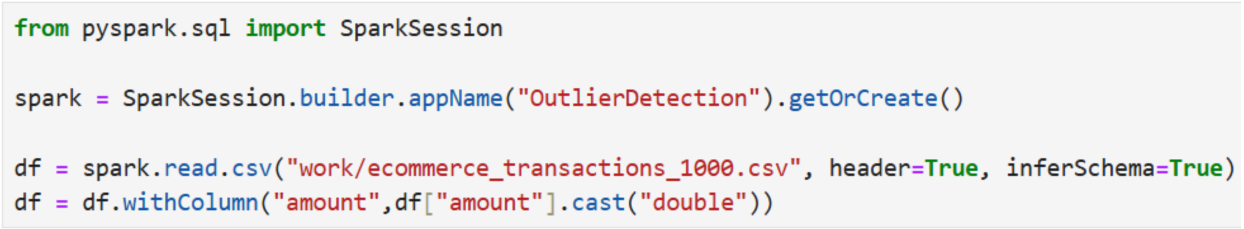
Praktikum 2: Deteksi Outlier Sederhana di Spark

Kasus

Kita mau cek outlier pada kolom amount di data transaksi

Langkah praktikum

1. Load Data

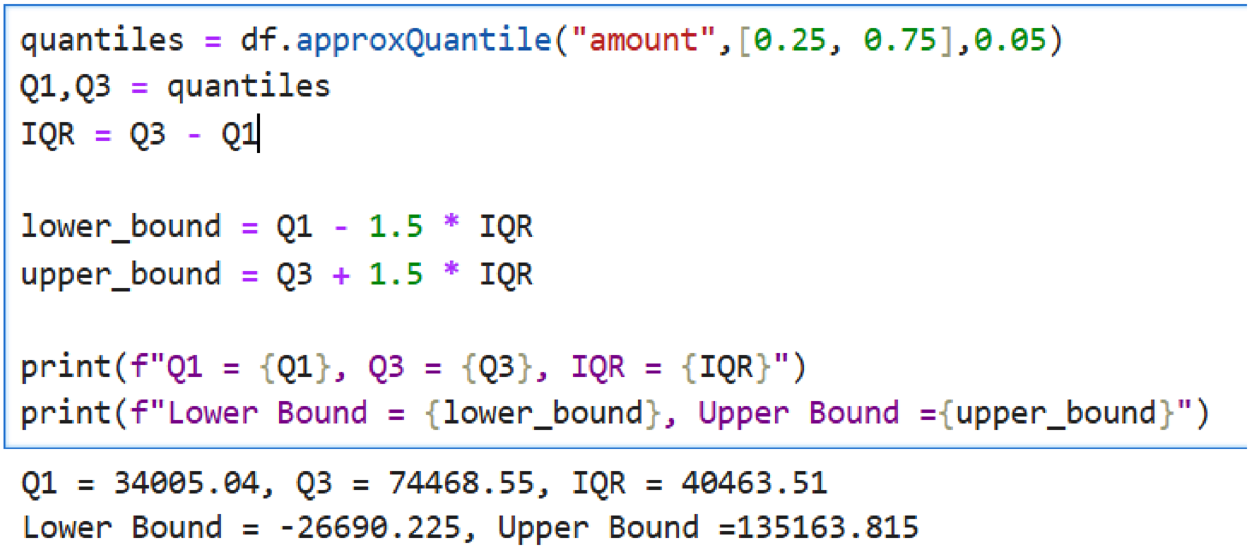


```python
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("OutlierDetection").getOrCreate()

df = spark.read.csv("work/ecommerce_transactions_1000.csv", header=True, inferSchema=True)
df = df.withColumn("amount",df["amount"].cast("double"))
```

2. Hitung statistic dasar

   Kita butuh:

   - Q1 (25th percentile)
   - Q3 (75th percentile)
   - IQR (Interquartile Range)

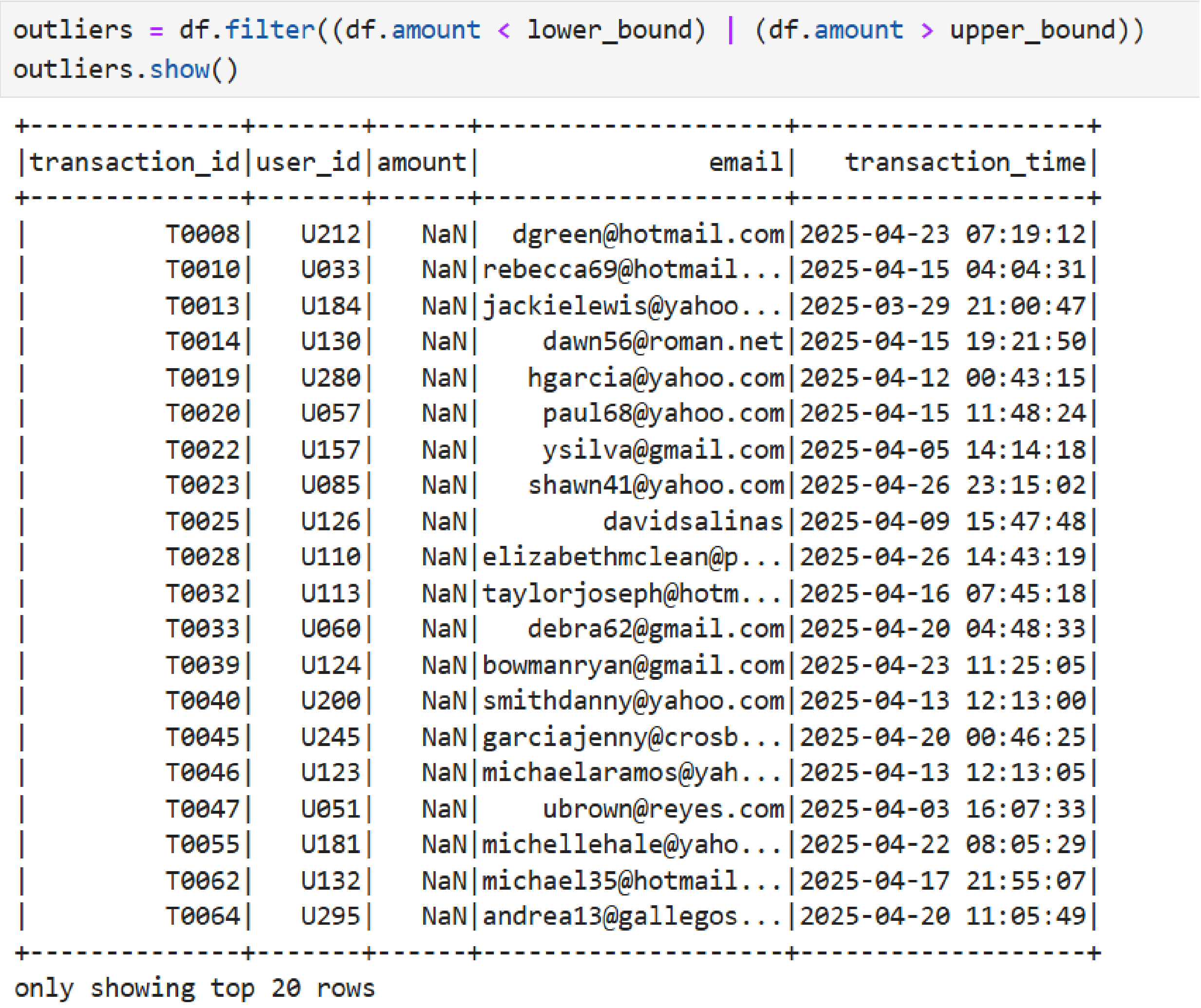


```python
quantiles = df.approxQuantile("amount",[0.25, 0.75],0.05)
Q1,Q3 = quantiles
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

print(f"Q1 = {Q1}, Q3 = {Q3}, IQR = {IQR}")
print(f"Lower Bound = {lower_bound}, Upper Bound ={upper_bound}")
```

```
Q1 = 34005.04, Q3 = 74468.55, IQR = 40463.51
Lower Bound = -26690.225, Upper Bound =135163.815
```

3. Deteksi outliers

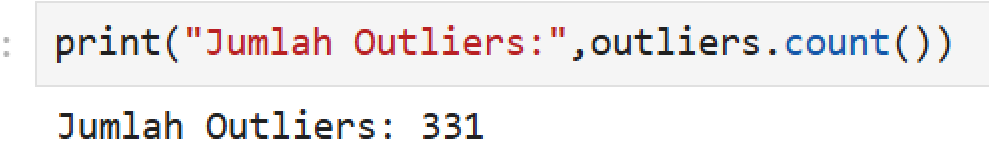
   Cari data amount yang lebih kecil dari lower bound atau lebih besar dari upper bound



```
outliers = df.filter((df.amount < lower_bound) | (df.amount > upper_bound))
outliers.show()
```

```
+--------------+-------+------+--------------------+-------------------+
|transaction_id|user_id|amount|               email|   transaction_time|
+--------------+-------+------+--------------------+-------------------+
|         T0008|   U212|   NaN|  dgreen@hotmail.com|2025-04-23 07:19:12|
|         T0010|   U033|   NaN|rebecca69@hotmail...|2025-04-15 04:04:31|
|         T0013|   U184|   NaN|jackielewis@yahoo...|2025-03-29 21:00:47|
|         T0014|   U130|   NaN|    dawn56@roman.net|2025-04-15 19:21:50|
|         T0019|   U280|   NaN|   hgarcia@yahoo.com|2025-04-12 00:43:15|
|         T0020|   U057|   NaN|    paul68@yahoo.com|2025-04-15 11:48:24|
|         T0022|   U157|   NaN|    ysilva@gmail.com|2025-04-05 14:14:18|
|         T0023|   U085|   NaN|   shawn41@yahoo.com|2025-04-26 23:15:02|
|         T0025|   U126|   NaN|        davidsalinas|2025-04-09 15:47:48|
|         T0028|   U110|   NaN|elizabethmclean@p...|2025-04-26 14:43:19|
|         T0032|   U113|   NaN|taylorjoseph@hotm...|2025-04-16 07:45:18|
|         T0033|   U060|   NaN|   debra62@gmail.com|2025-04-20 04:48:33|
|         T0039|   U124|   NaN|bowmanryan@gmail.com|2025-04-23 11:25:05|
|         T0040|   U200|   NaN|smithdanny@yahoo.com|2025-04-13 12:13:00|
|         T0045|   U245|   NaN|garciajenny@crosb...|2025-04-20 00:46:25|
|         T0046|   U123|   NaN|michaelaramos@yah...|2025-04-13 12:13:05|
|         T0047|   U051|   NaN|    ubrown@reyes.com|2025-04-03 16:07:33|
|         T0055|   U181|   NaN|michellehale@yaho...|2025-04-22 08:05:29|
|         T0062|   U132|   NaN|michael35@hotmail...|2025-04-17 21:55:07|
|         T0064|   U295|   NaN|andrea13@gallegos...|2025-04-20 11:05:49|
+--------------+-------+------+--------------------+-------------------+
only showing top 20 rows
```

4. Hitung banyak outliers

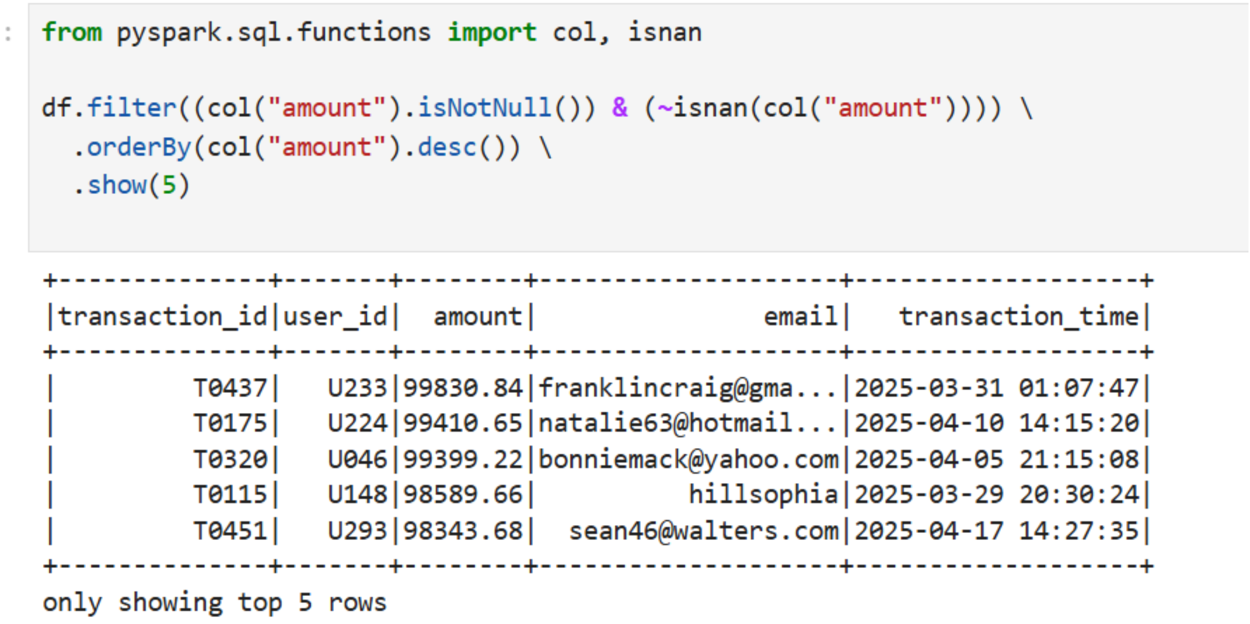


```
print("Jumlah Outliers:",outliers.count())
```

```
Jumlah Outliers: 331
```

Pertanyaan:

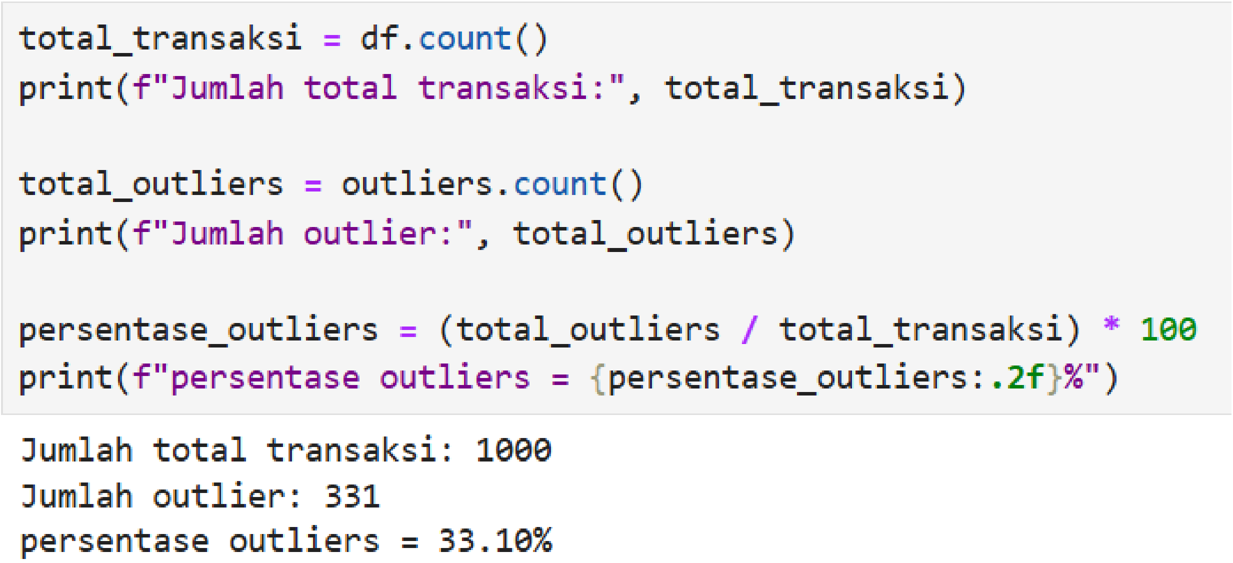
1. Tampilkan top 5 transaksi dengan amount terbesar?
   Disini saya tampilkan 5 transaksi dengan nilai amount terbesar, berikut code dan hasilnya:



```python
from pyspark.sql.functions import col, isnan

df.filter((col("amount").isNotNull()) & (~isnan(col("amount")))) \
  .orderBy(col("amount").desc()) \
  .show(5)
```

```
+--------------+-------+--------+--------------------+-------------------+
|transaction_id|user_id|  amount|               email|   transaction_time|
+--------------+-------+--------+--------------------+-------------------+
|         T0437|   U233|99830.84|franklincraig@gma...|2025-03-31 01:07:47|
|         T0175|   U224|99410.65|natalie63@hotmail...|2025-04-10 14:15:20|
|         T0320|   U046|99399.22|bonniemack@yahoo.com|2025-04-05 21:15:08|
|         T0115|   U148|98589.66|          hillsophia|2025-03-29 20:30:24|
|         T0451|   U293|98343.68|   sean46@walters.com|2025-04-17 14:27:35|
+--------------+-------+--------+--------------------+-------------------+
only showing top 5 rows
```

2. Hitung jumlah total transaksi?
3. Hitung jumlah outlier?
4. Hitung persentase outlier terhadap seluruh transaksi?



```python
total_transaksi = df.count()
print(f"Jumlah total transaksi:", total_transaksi)

total_outliers = outliers.count()
print(f"Jumlah outlier:", total_outliers)

persentase_outliers = (total_outliers / total_transaksi) * 100
print(f"persentase outliers = {persentase_outliers:.2f}%")
```

```
Jumlah total transaksi: 1000
Jumlah outlier: 331
persentase outliers = 33.10%
```