LAPORAN UJIAN TENGAH SEMESTER
MATA KULIAH BIG DATA
Jobsheet 10

Dosen Pengampu:
M. Hasyim Ratsanjani, S.Kom., M.Kom.

Disusun Oleh:

Yuma Rakha Samodra Sikayo **NIM.2241720194**

PROGRAM STUDI D4 TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2025

Persiapan

```
(base) PS D:\polinema\tugas_kuliah\semester6\big_Data> docker cp ecommerce_transactions_1000.csv spark-ma
ster:/opt/spark_data/
Successfully copied 58.4kB to spark-master:/opt/spark_data/
(base) PS D:\polinema\tugas_kuliah\semester6\big_Data>
```

```
(base) PS C:\users\user> docker exec -u root -it spark-master bash
root@e9e019bcf75c:/opt/spark/work-dir# ls /opt/spark_data/
ecommerce_transactions_1000.csv
```

Praktikum 1

1. Load Data

```
from pyspark.sql import SparkSession

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

df = spark.read.csv("/opt/spark_data/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

```
[2]:
#Inspeksi Data
#lihat sturktur 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)
```

```
[3]:
#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|
+--------------+-------+------+-----+----------------+
```

```
#Hitung jumlah total data
print("Jumlah baris:", df.count())
```

```
Jumlah baris: 1000
```

3. Cleaning Data

```
#Cleaning data

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

```
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|     0|    0|               0|
+--------------+-------+------+-----+----------------+
```

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

```
from pyspark.sql.functions import col, when, count

df.select(
    count(
        when(~col("email").contains("@"), True)
    ).alias("invalid_email_count")
).show()
```

```
+-------------------+
|invalid_email_count|
+-------------------+
|                  0|
+-------------------+
```

4. Transformasi Data

```
#Transformasi data

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

```
#Simpan data bersih
df.write.csv("cleaned_transactions_1000.csv", header =True, mode="overwrite")
```

Pertanyaan & Jawaban

1. Berapa banyak data yang dibuang karena transaction_time kosong?

Sebelum cleaning data

```
#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|
+--------------+-------+------+-----+----------------+
```

Proses cleaning data

```
df.filter(df["transaction_time"].isNull())
```

```
[34]:

DataFrame[transaction_id: string, user_id: string, amount: double, email: string, tra
nsaction_time: timestamp, email_domain: string, transaction_date: date]
```

Sesudah cleaning data

```
df.select([count(when(col(c).isNull(), c)).alias(c) for c in df.columns]).show()
```

```
+--------------+-------+------+-----+----------------+------------+----------------+
|transaction_id|user_id|amount|email|transaction_time|email_domain|transaction_date|
+--------------+-------+------+-----+----------------+------------+----------------+
|             0|      0|     0|    0|               0|           0|               0|
+--------------+-------+------+-----+----------------+------------+----------------+
```

50 data transaction_time berhasil dibuang

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

```
from pyspark.sql.types import DoubleType
df = df.withColumn("amount", df["amount"].cast(DoubleType()))
```

```
[37]:
```

```
df.filter(df["amount"].isNull()).show()
```

```
+--------------+-------+------+-----+----------------+------------+----------------+
|transaction_id|user_id|amount|email|transaction_time|email_domain|transaction_date|
+--------------+-------+------+-----+----------------+------------+----------------+
+--------------+-------+------+-----+----------------+------------+----------------+
```

Iya, semua data telah bernilai numerik

3. Kenapa lebih baik memperbaiki email invalid sebelum menganalisis data transaksi?

   Dikarenakan, email berguna sebagai identifikasi pengguna. Sehinga apabila terdapat masalah invalid email dapat menghasilkan analisis yang salah

Praktikum 2

1. Load Data

```
#Load Data
from pyspark.sql import SparkSession

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

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

2. Hitung Statistik Dasar

```
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

```
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 Berapa Banyak Outliers

```
[44]:
```

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

```
Jumlah Outliers:  331
```

Tugas Praktikum

1. Tampilkan top 5 transaksi dengan amount terbesar ?

```
# Tugas Praktikum

# Nomer 1
df.orderBy(df["amount"].desc()).show(5)
```

```
+--------------+-------+------+--------------------+-------------------+
|transaction_id|user_id|amount|               email|   transaction_time|
+--------------+-------+------+--------------------+-------------------+
|         T0019|   U280|   NaN|   hgarcia@yahoo.com|2025-04-12 00:43:15|
|         T0028|   U110|   NaN|elizabethmclean@p...|2025-04-26 14:43:19|
|         T0020|   U057|   NaN|    paul68@yahoo.com|2025-04-15 11:48:24|
|         T0014|   U130|   NaN|    dawn56@roman.net|2025-04-15 19:21:50|
|         T0022|   U157|   NaN|    ysilva@gmail.com|2025-04-05 14:14:18|
+--------------+-------+------+--------------------+-------------------+
only showing top 5 rows
```

2. Hitung jumlah total transaksi ?

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

```
Jumlah transaksi: 1000
```

3. Hitung jumlah outlier ?

```
q1, q3 = df.approxQuantile("amount", [0.25, 0.75], 0.01)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr

outliers = df.filter((df["amount"] < lower_bound) | (df["amount"] > upper_bound))
jumlah_outlier = outliers.count()

print("Jumlah outlier:", jumlah_outlier)
```

```
Jumlah outlier: 331
```

4. Hitung persentase outlier terhadap seluruh transaksi ?

```
persentase = (jumlah_outlier / total_transaksi) * 100
print(f"Persentase outlier: {persentase:.2f}%")
```

```
Persentase outlier: 33.10%
```