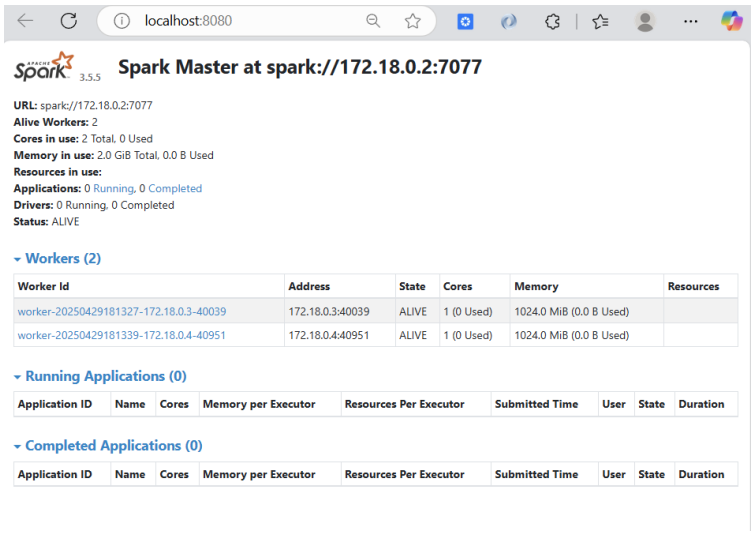
NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA – 10 Data Cleaning dan Transformasi Menggunakan Apache Spark

Siapkan lingkungan Spark Cluster



Buat dan copy data ke dalam container spark

```
PS C:\Users\sukma bagus\downloads> docker cp "C:\Users\sukma ba
gus\Downloads\ecommerce_transactions_1000.csv" boring_joliot:/o
pt/spark_data/
Successfully copied 58.4kB to boring_joliot:/opt/spark_data/
PS C:\Users\sukma bagus\downloads> docker exec -u root -it spar
k-master bash
root@b558c8f2d2a0:/opt/bitnami/spark# ls /opt/spark_data/
ecommerce_transactions_1000.csv
root@b558c8f2d2a0:/opt/bitnami/spark#
```

Praktikum 1 :

1. Load Dataset

1. Load Dataset

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("DataCleaningBigData") \
    .master("spark://spark-master:7077") \
    .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
```

NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA – 10 Data Cleaning dan Transformasi Menggunakan Apache Spark

2. Inspeksi Data

- Lihat struktur schema:

2. Inspeksi Data

```
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)
```

- 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

- Handling Missing Values

  Drop transaksi yang tidak memiliki "transaction_time"

  Isi nilai kosong pada "amount" dengan '0'.

```
**3. Cleaning Data**
```

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

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

- Cleaning Format Email

  Buat kolom baru "email_domain" yang berisi domain email.

  Hapus transaksi yang email-nya tidak valid (tidak memgamdung '@').

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

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

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

NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA – 10 Data Cleaning dan Transformasi Menggunakan Apache Spark

4. Transformasi Data

4. 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

5. Simpan Data Bersih

```
import os

# Membuat folder 'output' jika belum ada
os.makedirs("output", exist_ok=True)

# Menyimpan file CSV
pdf.to_csv("output/cleaned_transaction_1000.csv", index=False)
```

Checklist Praktikum

- Dataset berhasil dibaca ✓
- Struktur schema dipahami ✓
- Missing value berhasil dihandle ✓
- Email valid berhasil diproses ✓
- Kolom baru email_domain dan transaction_date berhasil dibuat ✓
- Data bersih disimpan ulang ✓

Jawaban Pertanyaan

1) Berapa banyak data yang dibuang karena transaction_time kosong?
   - Ada sebanyak 50 data dibuang dari 1000 data, karena transaction_time kosong

```
df = df.dropna(subset=["transaction_time"])
print("Jumlah Row:", df.count())
```

```
Jumlah Row: 950
```

2) Apakah semua data amount sudah bertipe numerik setelah cleaning?
   - Seluruh data amount sudah bertipe numerik setelah dilakukan cleaning
3) Kenapa lebih baik memperbaiki email invalid sebelum menganalisis data transaksi?
   - Ada beberapa alasan yaitu
     1. Validasi Identitas
     2. Menghindari adanya duplikasi dan noise dalam analisis
     3. Menjaga integritas dataset

   Jadi, memperbaiki atau menghapus email invalid membantu memastikan bahwa analisis dilakukan hanya pada data yang benar-benar representatif dan relevan.

NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA – 10 Data Cleaning dan Transformasi Menggunakan Apache Spark

Praktikum 2: Deteksi Outlier Sederhana di Spark

1. Load Data:

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("OutlierDetection") \
    .master("spark://spark-master:7077") \
    .getOrCreate()

df = spark.read.csv("/opt/spark_data/ecommerce_transactions_1000.csv", header=True, inferSchema=True)
df = df.fillna({"amount": 0})
df = df.withColumn("amount", df["amount"].cast("double"))
```

2. Hitung Statistik Dasar

   Kita butuh :

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

```
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}S)
```

```
Q1 = 0.0, Q3 = 24763.06, IQR = 24763.06
Lower Bound = -37144.590000000004, Upper Bound = 61907.65000000001
```

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|
+--------------+-------+--------+--------------------+-------------------+
|         T0002|   U253|70921.08| porteramy@yahoo.com|2025-03-30 21:07:41|
|         T0005|   U064|81176.73|   louis64@gmail.com|2025-04-14 08:50:35|
|         T0011|   U093| 82119.7|roberttucker@john...|2025-04-20 02:52:35|
|         T0012|   U279| 63515.6|brucesmith@gmail.com|2025-04-20 09:58:53|
|         T0035|   U180|74468.55|michaelcarey@gmai...|2025-04-01 16:09:24|
|         T0036|   U066|88464.76|stephanie50@yahoo...|2025-04-11 05:50:57|
|         T0049|   U050|93898.14|carlsonjames@gard...|2025-04-05 03:12:16|
|         T0052|   U088|70959.19|jessica48@hotmail...|2025-04-25 00:09:15|
|         T0060|   U265|80521.08|      kaitlynsalazar|2025-04-10 17:07:00|
|         T0063|   U098|87681.99|         rachelhayes|2025-04-13 16:25:19|
|         T0066|   U108|80296.12|    jill11@gmail.com|2025-04-03 09:51:20|
|         T0067|   U183|98103.36|danielramirez@hot...|2025-04-19 08:54:15|
|         T0075|   U131|89574.63|jonesgeorge@yahoo...|2025-04-14 00:16:53|
|         T0076|   U199|95746.19|             eric18|2025-03-29 22:51:17|
|         T0081|   U209|63408.75|    tara00@gmail.com|2025-04-22 15:38:34|
|         T0090|   U043|73488.49|   scott49@gmail.com|2025-04-08 18:42:41|
|         T0095|   U031|72250.11|    ryan82@brown.com|2025-04-06 08:18:57|
|         T0097|   U065|82322.29|    kaustin@soto.com|2025-04-18 15:16:49|
|         T0099|   U108|95527.61|walterelliott@yah...|2025-04-07 10:00:41|
|         T0100|   U044|64732.73|              ayoung|2025-04-10 10:08:57|
+--------------+-------+--------+--------------------+-------------------+
only showing top 20 rows
```

4. Hitung Berapa Banyak Outliers

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

```
Jumlah Outliers:  158
```

NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

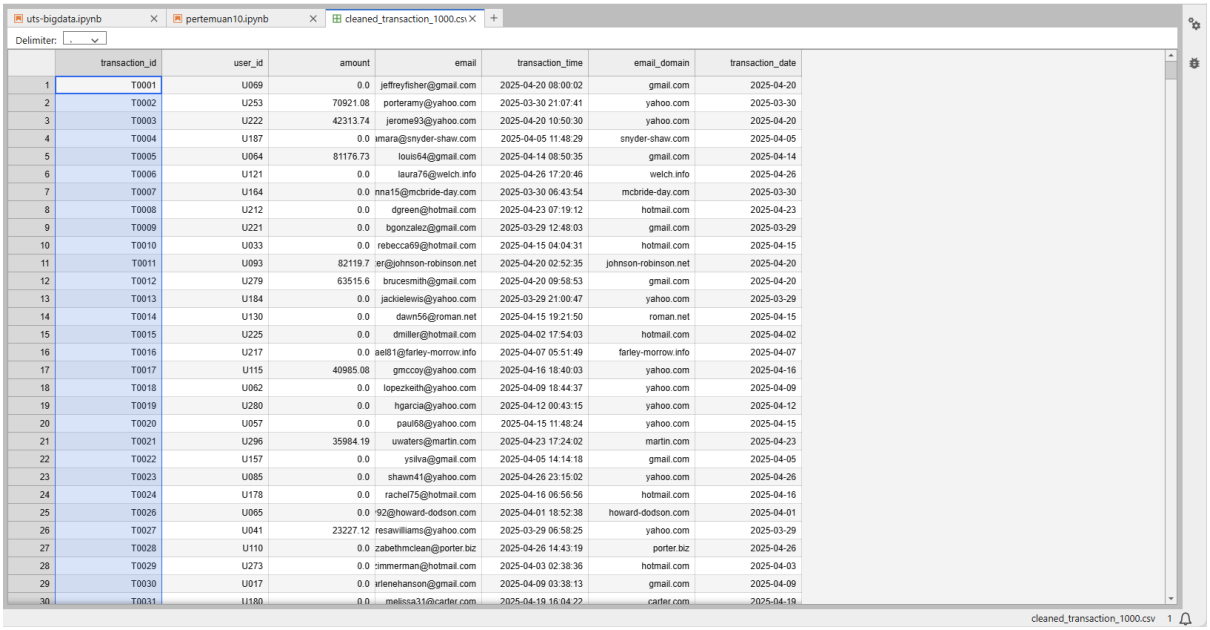
BIG DATA – 10 Data Cleaning dan Transformasi Menggunakan Apache Spark

Penjelasan Ringkas:

- IQR (Interquartile Range) = Jarak antara Q3 dan Q1 → rentang normal data.
- Outlier = Data di luar batas normal.
- Spark pakai .approxQuantile() karena menghitung quantile di dataset besar lebih cepat.

Tugas Praktikum

Pengerjaan tugas ini menggunakan data yang sudah bersih dari hasil cleaning pada praktikum diatas.



	transaction_id	user_id	amount	email	transaction_time	email_domain	transaction_date
1	T0001	U069	0.0	jeffreyfisher@gmail.com	2025-04-20 08:00:02	gmail.com	2025-04-20
2	T0002	U253	70921.08	porteramy@yahoo.com	2025-03-30 21:07:41	yahoo.com	2025-03-30
3	T0003	U222	42313.74	jerome93@yahoo.com	2025-04-20 10:50:30	yahoo.com	2025-04-20
4	T0004	U187	0.0	mara@snyder-shaw.com	2025-04-05 11:48:29	snyder-shaw.com	2025-04-05
5	T0005	U064	81176.73	louis64@gmail.com	2025-04-14 08:50:35	gmail.com	2025-04-14
6	T0006	U121	0.0	laura76@welch.info	2025-04-26 17:20:46	welch.info	2025-04-26
7	T0007	U164	0.0	nna15@mcbride-day.com	2025-03-30 06:43:54	mcbride-day.com	2025-03-30
8	T0008	U212	0.0	dgreen@hotmail.com	2025-04-23 07:19:12	hotmail.com	2025-04-23
9	T0009	U221	0.0	bgonzalez@gmail.com	2025-03-29 12:48:03	gmail.com	2025-03-29
10	T0010	U033	0.0	rebecca69@hotmail.com	2025-04-15 04:04:31	hotmail.com	2025-04-15
11	T0011	U093	82119.7	er@johnson-robinson.net	2025-04-20 02:52:35	johnson-robinson.net	2025-04-20
12	T0012	U279	63515.6	brucesmith@gmail.com	2025-04-20 09:58:53	gmail.com	2025-04-20
13	T0013	U184	0.0	jackielewis@yahoo.com	2025-03-29 21:00:47	yahoo.com	2025-03-29
14	T0014	U130	0.0	dawn56@roman.net	2025-04-15 19:21:50	roman.net	2025-04-15
15	T0015	U225	0.0	dmiller@hotmail.com	2025-04-02 17:54:03	hotmail.com	2025-04-02
16	T0016	U217	0.0	ael81@farley-morrow.info	2025-04-07 05:51:49	farley-morrow.info	2025-04-07
17	T0017	U115	40985.08	gmccoy@yahoo.com	2025-04-16 18:40:03	yahoo.com	2025-04-16
18	T0018	U062	0.0	lopezkeith@yahoo.com	2025-04-09 18:44:37	yahoo.com	2025-04-09
19	T0019	U280	0.0	hgarcia@yahoo.com	2025-04-12 00:43:15	yahoo.com	2025-04-12
20	T0020	U057	0.0	paul68@yahoo.com	2025-04-15 11:48:24	yahoo.com	2025-04-15
21	T0021	U296	35984.19	uwaters@martin.com	2025-04-23 17:24:02	martin.com	2025-04-23
22	T0022	U157	0.0	ysilva@gmail.com	2025-04-05 14:14:18	gmail.com	2025-04-05
23	T0023	U085	0.0	shawn41@yahoo.com	2025-04-26 23:15:02	yahoo.com	2025-04-26
24	T0024	U178	0.0	rachel75@hotmail.com	2025-04-16 06:56:56	hotmail.com	2025-04-16
25	T0026	U065	0.0	92@howard-dodson.com	2025-04-01 18:52:38	howard-dodson.com	2025-04-01
26	T0027	U041	23227.12	reawilliams@yahoo.com	2025-03-29 06:58:25	yahoo.com	2025-03-29
27	T0028	U110	0.0	zabethmclean@porter.biz	2025-04-26 14:43:19	porter.biz	2025-04-26
28	T0029	U273	0.0	immerman@hotmail.com	2025-04-03 02:38:36	hotmail.com	2025-04-03
29	T0030	U017	0.0	rlenehanson@gmail.com	2025-04-09 03:38:13	gmail.com	2025-04-09
30	T0031	U180	0.0	melissa31@carter.com	2025-04-19 16:04:22	carter.com	2025-04-19

1. Tampilkan top 5 transaksi dengan amount terbesar ?

```
# 1. Top 5 transaksi dengan amount terbesar
from pyspark.sql.functions import col, desc

top_5 = df.orderBy(col("amount").desc()).limit(5)
top_5.show()
```

```
+--------------+-------+--------+--------------------+-------------------+------------+----------------+
|transaction_id|user_id|  amount|               email|   transaction_time|email_domain|transaction_date|
+--------------+-------+--------+--------------------+-------------------+------------+----------------+
|         T0437|   U233|99830.84|franklincraig@gma...|2025-03-31 01:07:47|   gmail.com|      2025-03-31|
|         T0175|   U224|99410.65|natalie63@hotmail...|2025-04-10 14:15:20| hotmail.com|      2025-04-10|
|         T0320|   U046|99399.22|bonniemack@yahoo.com|2025-04-05 21:15:08|   yahoo.com|      2025-04-05|
|         T0451|   U293|98343.68|  sean46@walters.com|2025-04-17 14:27:35| walters.com|      2025-04-17|
|         T0067|   U183|98103.36|danielramirez@hot...|2025-04-19 08:54:15| hotmail.com|      2025-04-19|
+--------------+-------+--------+--------------------+-------------------+------------+----------------+
```

NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA – 10 Data Cleaning dan Transformasi Menggunakan Apache Spark

2. Hitung jumlah total transaksi ?

```
# 2. Jumlah Total Transaksi
from pyspark.sql.functions import sum

total_amount = df.agg(sum("amount").alias("total_amount")).collect()[0]["total_amount"]
total_transaksi = df.count()

print(f"Jumlah total transaksi: {total_transaksi}")
print(f"Jumlah total amount   : {total_amount}")
```

```
Jumlah total transaksi: 867
Jumlah total amount   : 16922579.86999999
```

3. Hitung jumlah outlier

```
# 3. Hitung jumlah outlier
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}")

# Jumlah outlier
outliers = df.filter((col("amount") < lower_bound) | (col("amount") > upper_bound))
print("Jumlah Outliers: ",outliers.count())
```

```
Q1 = 0.0, Q3 = 30029.83, IQR = 30029.83
Lower Bound = -45044.745, Upper Bound = 75074.57500000001
Jumlah Outliers:  86
```

4. Hitung persentase outlier terhadap seluruh transaksi ?

```
# 4. Hitung persentase outlier terhadap seluruh transaksi
jml_outlier = outliers.count()
persentase_outlier = (jml_outlier / total_transaksi) * 100
print(f"Outlier : {persentase_outlier:.2f}%")
```

```
Outlier : 9.92%
```

NAMA : SUKMA BAGUS WAHASDWIKA
NIM : 2241720223
KELAS : TI - 3D

BIG DATA – 10 Data Cleaning dan Transformasi Menggunakan Apache Spark

Catatan Penting Praktikum 1:

- Handling data besar: Proses cleaning jangan terlalu banyak chaining .filter() berurutan — lebih baik batch.
- Ingat: setiap operasi di Spark lazy — data baru di-load setelah .show(), .count(), atau .write()

Catatan Penting Praktikum 2:

- Untuk dataset sangat besar, .approxQuantile() lebih cepat daripada .quantile() biasa.
- Outlier tidak selalu error — tergantung konteks (misal pembelian Rp 1.000.000 memang mungkin saja di e-commerce).
- Bisa lanjut dengan perintah: drop atau flag pada data outlier sesuai kebutuhan analisis.