

Big Data

Data Cleaning dan Transformasi Menggunakan
Apache Spark (Jobsheet 10)

Dosen :

M. Hasyim Ratsanjani, S.Kom., M.Kom.



Dibuat Oleh :

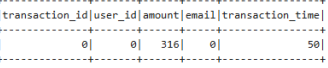
Nurhaliza Anindya Putri 2241720016

Kelas : TI – 3D

POLITEKNIK NEGERI MALANG


2025

1. Implementasi Python Spark (di Dockers)

Praktikum 1		
No	Gambar	Deskripsi
1.	<pre> : from pyspark.sql import SparkSession # Inisialisasi SparkSession spark = SparkSession.builder.appName("DataCleaningBigData").getOrCreate() # Membaca dataset df = spark.read.csv("ecommerce_transactions_1000.csv", header=True, inferSchema=True) # Menampilkan 5 baris pertama df.show(5) </pre> 	Load dataset
2.	<ul style="list-style-type: none"> • Lihat struktur schema: <pre> # Struktur Scema df.printSchema() </pre> <pre> root -- transaction_id: string (nullable = true) -- user_id: string (nullable = true) -- amount: double (nullable = true) -- email: string (nullable = true) -- transaction_time: timestamp (nullable = true) </pre> <ul style="list-style-type: none"> • Hitung missing values setiap kolom: <pre> # Hitung Missing Values per Kolom from pyspark.sql.functions import col, isnan, when, count df.select([count(when(col(c).isNull(), c)), alias(c) for c in df.columns]).show() </pre>  <ul style="list-style-type: none"> • Hitung jumlah total data: <pre> # Hitung Jumlah Total Data print("Jumlah baris:", df.count()) </pre> <pre> Jumlah baris: 1000 </pre>	Inspeksi data
3.	<p>A. Handling Missing Values</p> <ol style="list-style-type: none"> Drop transaksi yang tidak memiliki transaction_time. Isi nilai kosong pada amount dengan 0. 	Cleaning Data

	<pre># Drop transaksi yang tidak memiliki transaction_time. df = df.dropna(subset=["transaction_time"]) # Isi nilai kosong pada amount dengan 0. df = df.fillna({'amount': 0})</pre> <p>B. Cleaning Format Email</p> <ol style="list-style-type: none"> Buat kolom baru email_domain yang berisi domain email. Hapus transaksi yang email-nya tidak valid (tidak mengandung '@') <pre>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)</pre>	
4.	<ul style="list-style-type: none"> Ubah kolom amount menjadi tipe DoubleType. Tambahkan kolom baru transaction_date dari transaction_time. <pre>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"))</pre>	Transformasi Data
5.	<pre># Simpan Data Bersih df.write.csv("cleaned_transaction_1000.csv", header=True, mode="overwrite")</pre>	Simpan Data Bersih
Pertanyaan		
No	Soal	Jawab
1.	Berapa banyak data yang dibuang karena transaction_time kosong?	<pre># 1. Berapa banyak data yang dibuang karena transaction_time kosong? df.filter(df["transaction_time"].isNull()).count() 0</pre>
2.	Apakah semua data amount sudah bertipe numerik setelah cleaning?	Ya, karena sudah dilakukan cast ke DoubleType di langkah transformasi.
3.	Kenapa penting memperbaiki email invalid sebelum analisis data transaksi?	<p>Email invalid bisa menyebabkan:</p> <ul style="list-style-type: none"> Error saat pengelompokan berdasarkan domain. Hasil analisis pelanggan menjadi bias. Tidak bisa digunakan

		untuk kampanye marketing berbasis email.
Praktikum 2: Deteksi Outlier Sederhana di Spark		
No	Gambar	Deskripsi
1.	<pre>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"))</pre>	Load Data
2.	<p>Kita butuh:</p> <ul style="list-style-type: none"> ● Q1 (25th percentile) ● Q3 (75th percentile) ● IQR (Interquartile Range) <pre>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</pre>	Hitung Statistik Dasar
3.	<p>Cari data amount yang lebih kecil dari lower bound atau lebih besar dari upper bound.</p> <pre>outliers = df.filter((df.amount < lower_bound) (df.amount > upper_bound)) outliers.show()</pre> <pre>+-----+-----+-----+-----+-----+ 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 daun56@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 michael13@hotmail... 2025-04-17 21:55:07 T0064 U295 NaN andreal3@gallegos... 2025-04-20 11:05:49 +-----+-----+-----+-----+-----+ only showing top 20 rows</pre>	Deteksi Outliers
4.	<pre>print("Jumlah Outliers:", outliers.count()) Jumlah Outliers: 331</pre> <p>Penjelasan Ringkas</p> <ul style="list-style-type: none"> ● IQR (Interquartile Range) = Jarak antara Q3 dan Q1 → rentang normal data. ● Outlier = Data di luar batas 	Hitung Berapa Banyak Outliers

	<p>normal.</p> <ul style="list-style-type: none"> Spark pakai .approxQuantile() karena menghitung quantile di dataset besar lebih cepat. 	
Tugas Praktikum		
No	Gambar	Deskripsi
1.	<pre># Top 5 transaksi dengan amount terbesar: df.orderBy(col("amount").desc()).show(5)</pre>  <pre>only showing top 5 rows</pre>	Tampilkan top 5 transaksi dengan amount terbesar ?
2.	<pre># Hitung jumlah total transaksi: df.count()</pre> <p>1000</p>	Hitung jumlah total transaksi ?
3.	<pre># Hitung jumlah outlier: outliers.count()</pre> <p>331</p>	Hitung jumlah outlier ?
4.	<pre># Hitung persentase outlier: (outliers.count() / df.count()) * 100</pre> <p>33.1</p>	Hitung persentase outlier terhadap seluruh transaksi ?