TEKNOLOGI BIG DATA

Arranged by Group 14 (G):

1. Mohammad varrel bramasta - 2106733811

2.Mochammad Dyenta D - 2106731245

3. Fatima Khairunnisa - 2106651515

4.Muhammad Irsyad Fakhruddin - 2006468850

DEPARTEMEN TEKNIK ELEKTRO FAKULTAS TEKNIK UNIVERSITAS INDONESIA DEPOK

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dataset Clean_Dataset.csv berisi informasi detail mengenai penerbangan, meliputi maskapai, nomor penerbangan, kota asal, waktu keberangkatan, jumlah transit, waktu kedatangan, kota tujuan, kelas penerbangan, durasi penerbangan, hari tersisa hingga keberangkatan, dan harga tiket. Volume data yang besar serta keragaman dan kompleksitas data tersebut menjadikannya sebagai kasus big data.

Beberapa tantangan yang muncul dalam menganalisis dataset ini di antaranya adalah volume data yang sangat besar, kecepatan akses dan pemrosesan data, keragaman data, serta verifikasi dan validasi data. Oleh karena itu diperlukan arsitektur sistem big data yang tepat agar analisis data dapat berjalan optimal.

Tujuan analisis data penerbangan ini adalah untuk mengidentifikasi tren perjalanan, memahami pengaruh kelas penerbangan dan durasi penerbangan, memprediksi harga tiket, serta menganalisis ketersediaan tempat pada penerbangan. Hasil analisis diharapkan dapat memberikan wawasan penting bagi maskapai, agen perjalanan, serta pengguna jasa transportasi udara.

Untuk mengatasi tantangan tersebut, tools big data seperti Apache Spark dan HDFS dipilih karena kemampuannya dalam mendistribusikan pemrosesan data pada klaster komputer sehingga analisis data dapat berjalan secara paralel dan terukur. Selain itu, Spark SQL digunakan untuk query data terstruktur secara cepat sementara PySpark menyediakan DataFrame API yang memudahkan pembersihan, transformasi, dan analisis data.

Dengan memanfaatkan teknologi big data yang tepat, diharapkan analisis dataset penerbangan ini dapat mengekstraksi informasi berharga yang berguna bagi berbagai pihak terkait industri penerbangan.

1.2 Rumusan Masalah

- Bagaimana melakukan ingestion dataset Clean_Dataset.csv yang berukuran besar ke dalam sistem big data dengan efisien?
- Bagaimana melakukan pembersihan data (data cleaning) dan transformasi data agar sesuai untuk dianalisis lebih lanjut?
- Bagaimana merancang dan mengimplementasikan model analisis dan machine learning menggunakan teknologi big data untuk mengekstraksi informasi penting terkait tren, pola, dan faktor-faktor yang mempengaruhi metrik tertentu pada data penerbangan?

- Bagaimana meningkatkan performa dan skalabilitas sistem big data yang dirancang agar mampu menangani volume data yang besar dengan tetap bersifat responsive?

Rumusan masalah pertama terkait tantangan ingestion data dalam volume besar. Rumusan masalah kedua terkait proses pembersihan dan transformasi data agar siap dianalisis. Rumusan masalah ketiga fokus pada bagaimana melakukan analisis dan pemodelan data menggunakan teknologi big data. Rumusan masalah keempat berkaitan dengan skalabilitas dan optimasi sistem big data yang dirancang.

1.3 Tujuan dan Solusi

Tujuan utama dari analisis kasus big data ini adalah untuk merancang sebuah sistem yang mampu menangani ingestion data dalam volume besar, melakukan pemrosesan dan pembersihan data, mengimplementasikan model analisis dan machine learning, serta menghasilkan output analisis dan visualisasi yang memberikan informasi penting bagi berbagai pihak terkait industri penerbangan.

Untuk mencapai tujuan tersebut, Apache Spark dipilih sebagai teknologi big data utama karena kemampuannya untuk mendistribusikan pemrosesan data pada klaster server secara paralel, sehingga analisis data dapat berjalan secara cepat dan terukur. PySpark menyediakan DataFrame API yang sangat membantu dalam pembersihan, transformasi, dan analisis data. PySpark MLlib menyediakan library machine learning yang komprehensif. Fitur Spark SQL memungkinkan query data secara cepat. HDFS digunakan sebagai sistem penyimpanan data terdistribusi.

Dengan memanfaatkan kemampuan Apache Spark dalam komputasi terdistribusi dan pemrosesan paralel diharapkan sistem yang dirancang dapat menangani volume data yang besar dengan tetap bersifat responsif dan terukur. Productivity tool seperti PyCharm juga digunakan untuk memudahkan implementasi kode analisis data.

Melalui optimalisasi arsitektur dan infrastruktur big data ini, proses analisis kasus data penerbangan dapat difasilitasi secara komprehensif mulai dari ingestion hingga visualisasi output analisis. Informasi penting terkait pola, tren, dan faktor yang memengaruhi metrik tertentu pada data penerbangan diharapkan dapat diekstraksi secara maksimal.

BAB 2

PERANCANGAN ARSITEKTUR

2.1 Konsep Big Data

Istilah big data mengacu pada dataset yang terlalu besar dan kompleks untuk ditangani oleh database dan perangkat lunak tradisional. Teknologi big data ditujukan untuk menangkap, menyimpan, mengelola, dan menganalisis kumpulan data dalam skala yang sangat besar dan beragam bentuknya.

Clean_datasheet sudah memenuhi karakteristik 3V, yaitu meliputi:

A. Volume: Dataset berisi informasi detail penerbangan dalam jumlah yang besar
B. Velocity: Data penerbangan bersifat real-time dan sangat dinamis, sehingga diperlukan kecepatan akses dan pemrosesan data yang tinggi.
C. Variety: Dataset berisi beragam jenis data terstruktur maupun semi-terstruktur, seperti tanggal, waktu, nomor, teks, angka, dan lainnya.

Untuk mengatasi tantangan penyimpanan, pemrosesan, dan analisis big data, diperlukan sistem terdistribusi dengan komputasi dan penyimpanan paralel, toleransi kesalahan tinggi, pemulihan cepat, serta kemampuan untuk menangani data dalam skala yang sangat besar dan bentuk yang beragam. Inilah prinsip utama yang diimplementasikan dalam teknologi big data.

Melalui pemanfaatan teknologi big data, diharapkan nilai bisnis (business value) dapat diekstraksi dari dataset bervolume besar, berkecepatan tinggi, dan bervariasi ini, sehingga memberikan manfaat terhadap berbagai sektor bisnis maupun akademik. Kasus pada analisis data penerbangan ini merupakan salah satu contoh aplikasi dari teknologi dan konsep big data.

2.2 Arsitektur Big Data

Arsitektur big data ini dibuat untuk melakukan analisis data yang besar dengan kompleksitas tinggi bagi sistem data tradisional. Dari case study yang kami pilih, untuk melakukan analisis terhadap dataset penerbangan, dirancang sebuah arsitektur big data dengan komponen-komponen sebagai berikut:

- Data Ingestion
  Apache Spark digunakan untuk melakukan ingestion dataset CSV bervolume besar ke dalam sistem. Data kemudian disimpan di HDFS.
- Data Cleaning & Transformation
  PySpark DataFrame API digunakan untuk pembersihan data serta transformasi data seperti penanganan nilai kosong dan konversi kolom kategorikal.

- EDA & Analysis
  PySpark digunakan untuk exploratory data analysis guna mendapatkan statistik deskriptif, distribusi data, dan visualisasi. PySpark MLlib menyediakan algoritma machine learning untuk pemodelan analisis prediktif.
- Processing Optimization
  Kemampuan komputasi terdistribusi Spark dimanfaatkan agar data processing dapat skalabel secara horizontal. Teknik caching dataframe dan partisi data dioptimalkan.
- Output & Visualization
  Hasil analisis ditampilkan melalui dashboard serta visualisasi interaktif. Output disimpan kembali ke dalam HDFS.

Melalui arsitektur yang dirancang, analisis data penerbangan dapat dilakukan secara terdistribusi dengan memanfaatkan tools big data Spark dan Hadoop ecosystem. Kemampuan scaling out secara horizontal diharapkan dapat menangani pertumbuhan volume data yang signifikan di masa mendatang.

2.3 Tools yang Digunakan

a. Apache Spark

Apache Spark adalah open-source cluster computing framework untuk pemrosesan data dalam skala besar. Spark menyediakan engine untuk pemrosesan data secara terdistribusi di memory, yang mendukung pemrosesan data real-time. Spark juga mencakup library untuk pemrosesan graph, ML, dan lainnya.

b. Apache Hadoop

Apache Hadoop adalah kumpulan open-source software untuk penyimpanan data dan analisis dataset besar menggunakan klaster commodity hardware. HDFS (Hadoop Distributed File System) menyediakan sistem file terdistribusi fault-tolerant dan termonitor secara otomatis.

c. PySpark

PySpark adalah API Python untuk Spark yang memungkinkan pemrograman Spark menggunakan Python API dan akses ke fitur Spark, termasuk Spark SQL, DataFrames, MLlib, dan lainnya. PySpark memudahkan ETL, pemodelan ML, dan analisis data.

d. PyCharm

PyCharm adalah IDE (Integrated Development Environment) moderen untuk bahasa pemrograman Python yang memudahkan pengembangan aplikasi Python. PyCharm mendukung pengembangan Spark dan Hadoop ecosystem menggunakan Python dengan fitur auto-complete, debugger, profiler, dan lainnya.

Dengan memanfaatkan kemampuan utama dari masing-masing tools di atas, analisis data penerbangan dapat dilakukan mulai dari ingestion, pemrosesan, hingga analisis menggunakan prinsip komputasi terdistribusi pada klaster server.

BAB 3

IMPLEMENTASI

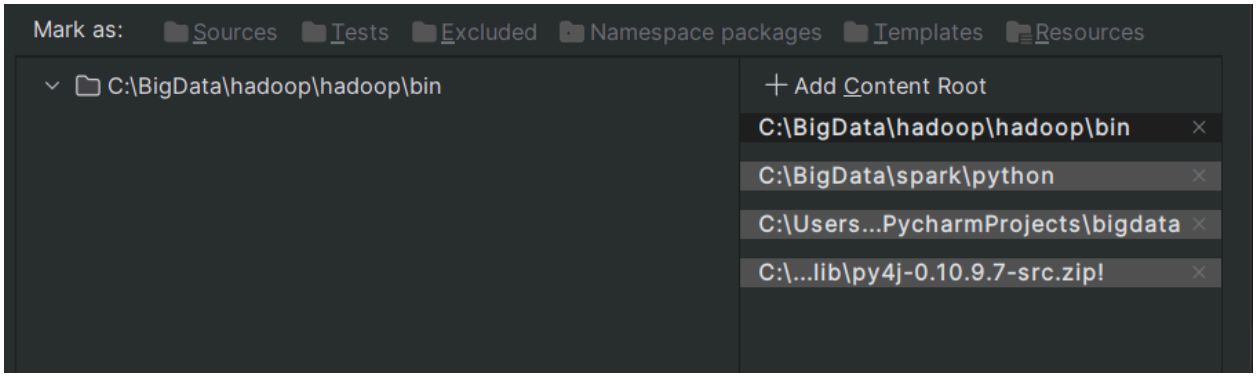
3.1 Konfigurasi Spark pada laptop

```
C:\Users\moham.BRAMASTA.001>spark-shell
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(ne
23/12/15 21:55:16 WARN NativeCodeLoader: Unable to load native-hadoop library for y
asses where applicable
Spark context Web UI available at http://192.168.6.1:4040
Spark context available as 'sc' (master = local[*], app id = local-1702652118345).
Spark session available as 'spark'.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.0
      /_/

Using Scala version 2.12.18 (Java HotSpot(TM) 64-Bit Server VM, Java 11.0.16.1)
Type in expressions to have them evaluated.
Type :help for more information.
```

Pastikan Spark sudah terinstal di Device

3.2 Konfigurasi Pycharm



Pastikan Pycharm sudah terhubung dengan Pyspark dan Hadoop agar dapat menggunakan spark di Pycharm

3.3 Implementasi Code

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
from pyspark.ml.feature import VectorAssembler, StringIndexer
from pyspark.sql.types import StructType, StructField, StringType, IntegerType,
DoubleType

# Inisialisasi sesi Spark
spark =
SparkSession.builder.appName("BigDataAnalysis").config("spark.sql.debug.maxToSt
ringFields", 1000).getOrCreate()

# Skema dari Clean_Dataset
schema = StructType([
 StructField("_c0", IntegerType()),
 StructField("airline", StringType()),
 StructField("flight", StringType()),
 StructField("source_city", StringType()),
 StructField("departure_time", StringType()),
 StructField("stops", StringType()),
 StructField("arrival_time", StringType()),
 StructField("destination_city", StringType()),
 StructField("class", StringType()),
 StructField("duration", DoubleType()),
 StructField("days_left", IntegerType()),
 StructField("price", IntegerType())
])

# Membaca file CSV ke dalam DataFrame
df = spark.read.format("csv") \
 .option("header", True) \
 .option("delimiter", ",") \
 .option("quote", '"') \
```

```
 .schema(schema) \
 .load("Clean_Dataset.csv")

# Path untuk menyimpan output
output_file_path =
"C:/Users/moham.BRAMASTA.001/PycharmProjects/bigdata/output_Clean_Dataset.txt"

# Mengarahkan output ke file output
import sys
original_stdout = sys.stdout
with open(output_file_path, 'w') as f:
   sys.stdout = f

# Analisis dan mengelola file pada Clean_Dataset.csv

   # 1. Menampilkan 10 baris pertama dari DataFrame
   print("\n5 baris pertama dari Clean_Dataset.csv:")
   df.show(5)

   # 2. Menampilkan skema DataFrame
   print("Skema DataFrame:")
   df.printSchema()

   # 3. Menghitung jumlah baris dalam DataFrame
   row_count = df.count()
   print(f"Jumlah Baris:\n{row_count}")
   print("\n")

   # 4. Menampilkan statistik deskriptif untuk kolom numerik
   numeric_summary = df.describe().toPandas()
   print("Statistik Deskriptif untuk Kolom Numerik:")
   print(numeric_summary)

   # 5. menghitung berapa banyak class economy dan business
   print("Table untuk menghitung berapa banyak kelas ekonomi dan bisnis")
   category_counts = df.groupBy("class").count().show()

   # 6. Melakukan filtering data duration lebih dari 3
   print("Table yang menampilkan Duration yang lebih besar dari 3")
   filtered_data = df.filter(col("duration") > 3).show()

   # 7. Pemodelan Mesin (Logistic Regression)
   print("Table yang menampilkan 10 baris pertama dari DataFrame yang sudah
di-assembly")
   feature_columns = ["duration", "days_left", "price"]
   assembler = VectorAssembler(inputCols=feature_columns, outputCol="features")
   df_assembled = assembler.transform(df)
   df_assembled.show(10, False)
```

```
    # 8. Membuat String Indexer
    print("Table yang menampilkan 10 baris pertama dari DataFrame yang sudah
di-index")
    indexer = StringIndexer(inputCol="class", outputCol="label")
    indexed = indexer.fit(df_assembled).transform(df_assembled)
    indexed.show(10, False)

# Mengembalikan output standar ke keadaan semula
sys.stdout = original_stdout
print(f"Output telah disimpan di: {output_file_path}")

# Menyimpan hasil analisis ke dalam file output
output_file_path = "path/to/output_Clean_Dataset.txt"
```

3.4 Output yang disimpan dalam direktori lokal

- Lima Baris pertama dari datasheet:

```
5 baris pertama dari Clean_Dataset.csv:
+---+--------+-------+-----------+-------------+-----+-------------+---------------+-------+--------+---------+-----+
|_c0| airline| flight|source_city|departure_time|stops| arrival_time|destination_city|  class|duration|days_left|price|
+---+--------+-------+-----------+-------------+-----+-------------+---------------+-------+--------+---------+-----+
|  0|SpiceJet|SG-8709|      Delhi|      Evening| zero|        Night|         Mumbai|Economy|    2.17|        1| 5953|
|  1|SpiceJet|SG-8157|      Delhi|Early_Morning| zero|      Morning|         Mumbai|Economy|    2.33|        1| 5953|
|  2| AirAsia| I5-764|      Delhi|Early_Morning| zero|Early_Morning|         Mumbai|Economy|    2.17|        1| 5956|
|  3| Vistara| UK-995|      Delhi|      Morning| zero|    Afternoon|         Mumbai|Economy|    2.25|        1| 5955|
|  4| Vistara| UK-963|      Delhi|      Morning| zero|      Morning|         Mumbai|Economy|    2.33|        1| 5955|
+---+--------+-------+-----------+-------------+-----+-------------+---------------+-------+--------+---------+-----+
```

- Skema Data Frame:

```
Skema DataFrame:
root
 |-- _c0: integer (nullable = true)
 |-- airline: string (nullable = true)
 |-- flight: string (nullable = true)
 |-- source_city: string (nullable = true)
 |-- departure_time: string (nullable = true)
 |-- stops: string (nullable = true)
 |-- arrival_time: string (nullable = true)
 |-- destination_city: string (nullable = true)
 |-- class: string (nullable = true)
 |-- duration: double (nullable = true)
 |-- days_left: integer (nullable = true)
 |-- price: integer (nullable = true)

Jumlah Baris:
300153
```

- Statistik Deskriptif untuk kolom numerik:

```
Statistik Deskriptif untuk Kolom Numerik:
  summary                _c0  ...           days_left               price
0   count             300153  ...              300153              300153
1    mean           150076.0  ...  26.004750910369044  20889.660523133203
2  stddev  86646.85201148395  ...  13.561003687093626   22697.76736607441
3     min                 0  ...                   1                1105
4     max             300152  ...                  49              123071

[5 rows x 13 columns]
Table untuk menghitung berapa banyak kelas ekonomi dan bisnis
+--------+------+
|   class| count|
+--------+------+
| Economy|206666|
|Business| 93487|
+--------+------+
```

- Duration yang lebih besar dari 3 jam

```
Table yang menampilkan Duration yang lebih besar dari 3
+---+---------+-------+-----------+--------------+-----+-------------+----------------+-------+--------+---------+-----+
|_c0|  airline| flight|source_city|departure_time|stops| arrival_time|destination_city|  class|duration|days_left|price|
+---+---------+-------+-----------+--------------+-----+-------------+----------------+-------+--------+---------+-----+
| 18|  AirAsia| I5-747|      Delhi|       Evening|  one|Early_Morning|          Mumbai|Economy|   12.25|        1| 5949|
| 19|  AirAsia| I5-747|      Delhi|       Evening|  one|      Morning|          Mumbai|Economy|   16.33|        1| 5949|
| 20| GO_FIRST| G8-266|      Delhi| Early_Morning|  one|      Evening|          Mumbai|Economy|   11.75|        1| 5954|
| 21| GO_FIRST| G8-101|      Delhi| Early_Morning|  one|        Night|          Mumbai|Economy|    14.5|        1| 5954|
| 22| GO_FIRST| G8-103|      Delhi|       Evening|  one|      Morning|          Mumbai|Economy|   15.67|        1| 5954|
| 23|Air_India| AI-441|      Delhi|       Evening|  one|        Night|          Mumbai|Economy|    3.75|        1| 5955|
| 27|  AirAsia| I5-744|      Delhi|       Morning|  one|    Afternoon|          Mumbai|Economy|    5.83|        1| 8869|
| 30| GO_FIRST| G8-165|      Delhi| Early_Morning|  one|    Afternoon|          Mumbai|Economy|     8.0|        1|10575|
| 31|   Indigo|6E-2373|      Delhi|     Afternoon|  one|      Evening|          Mumbai|Economy|     6.0|        1|10838|
| 32|  Vistara| UK-813|      Delhi|       Evening|  one|      Morning|          Mumbai|Economy|   14.67|        1|12150|
| 33|  Vistara| UK-817|      Delhi|       Evening|  one|      Morning|          Mumbai|Economy|   16.17|        1|12150|
| 34|  Vistara| UK-819|      Delhi|     Afternoon|  one|      Morning|          Mumbai|Economy|    18.0|        1|12150|
| 35|  Vistara| UK-801|      Delhi|       Morning|  one|      Morning|          Mumbai|Economy|   23.17|        1|12150|
| 36|  Vistara| UK-815|      Delhi|       Morning|  one|      Morning|          Mumbai|Economy|   24.17|        1|12150|
| 37|Air_India| AI-453|      Delhi| Early_Morning|  one|    Afternoon|          Mumbai|Economy|    8.83|        1|11900|
| 38| SpiceJet|SG-2976|      Delhi|       Evening|  one|        Night|          Mumbai|Economy|     4.5|        1|12123|
| 39| SpiceJet|SG-2976|      Delhi|       Evening|  one|      Morning|          Mumbai|Economy|   15.25|        1|12123|
| 40|Air_India| AI-504|      Delhi|         Night|  one|      Morning|          Mumbai|Economy|    11.0|        1|12150|
| 41|Air_India| AI-502|      Delhi|     Afternoon|  one|      Morning|          Mumbai|Economy|   19.08|        1|12150|
| 42|Air_India| AI-506|      Delhi|       Morning|  one|      Morning|          Mumbai|Economy|   22.83|        1|12150|
+---+---------+-------+-----------+--------------+-----+-------------+----------------+-------+--------+---------+-----+
```

- 10 Baris pertama dari dataframe yang sudah di assmbly

```
Table yang menampilkan 10 baris pertama dari DataFrame yang sudah di-assembly
+---+--------+-------+-----------+--------------+-----+-------------+----------------+-------+--------+---------+-----+----------------+
|_c0|airline |flight |source_city|departure_time|stops|arrival_time |destination_city|class  |duration|days_left|price|features        |
+---+--------+-------+-----------+--------------+-----+-------------+----------------+-------+--------+---------+-----+----------------+
|0  |SpiceJet|SG-8709|Delhi      |Evening       |zero |Night        |Mumbai          |Economy|2.17    |1        |5953 |[2.17,1.0,5953.0]|
|1  |SpiceJet|SG-8157|Delhi      |Early_Morning |zero |Morning      |Mumbai          |Economy|2.33    |1        |5953 |[2.33,1.0,5953.0]|
|2  |AirAsia |I5-764 |Delhi      |Early_Morning |zero |Early_Morning|Mumbai          |Economy|2.17    |1        |5956 |[2.17,1.0,5956.0]|
|3  |Vistara |UK-995 |Delhi      |Morning       |zero |Afternoon    |Mumbai          |Economy|2.25    |1        |5955 |[2.25,1.0,5955.0]|
|4  |Vistara |UK-963 |Delhi      |Morning       |zero |Morning      |Mumbai          |Economy|2.33    |1        |5955 |[2.33,1.0,5955.0]|
|5  |Vistara |UK-945 |Delhi      |Morning       |zero |Afternoon    |Mumbai          |Economy|2.33    |1        |5955 |[2.33,1.0,5955.0]|
|6  |Vistara |UK-927 |Delhi      |Morning       |zero |Morning      |Mumbai          |Economy|2.08    |1        |6060 |[2.08,1.0,6060.0]|
|7  |Vistara |UK-951 |Delhi      |Afternoon     |zero |Evening      |Mumbai          |Economy|2.17    |1        |6060 |[2.17,1.0,6060.0]|
|8  |GO_FIRST|G8-334 |Delhi      |Early_Morning |zero |Morning      |Mumbai          |Economy|2.17    |1        |5954 |[2.17,1.0,5954.0]|
|9  |GO_FIRST|G8-336 |Delhi      |Afternoon     |zero |Evening      |Mumbai          |Economy|2.25    |1        |5954 |[2.25,1.0,5954.0]|
+---+--------+-------+-----------+--------------+-----+-------------+----------------+-------+--------+---------+-----+----------------+
```

- 10 Baris pertama dari DataFrame yang sudah di index

Table yang menampilkan 10 baris pertama dari DataFrame yang sudah di-index

_c0	airline	flight	source_city	departure_time	stops	arrival_time	destination_city	class	duration	days_left	price	features	label
0	SpiceJet	SG-8709	Delhi	Evening	zero	Night	Mumbai	Economy	2.17	1	5953	[2.17,1.0,5953.0]	0.0
1	SpiceJet	SG-8157	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.33	1	5953	[2.33,1.0,5953.0]	0.0
2	AirAsia	I5-764	Delhi	Early_Morning	zero	Early_Morning	Mumbai	Economy	2.17	1	5956	[2.17,1.0,5956.0]	0.0
3	Vistara	UK-995	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.25	1	5955	[2.25,1.0,5955.0]	0.0
4	Vistara	UK-963	Delhi	Morning	zero	Morning	Mumbai	Economy	2.33	1	5955	[2.33,1.0,5955.0]	0.0
5	Vistara	UK-945	Delhi	Morning	zero	Afternoon	Mumbai	Economy	2.33	1	5955	[2.33,1.0,5955.0]	0.0
6	Vistara	UK-927	Delhi	Morning	zero	Morning	Mumbai	Economy	2.08	1	6060	[2.08,1.0,6060.0]	0.0
7	Vistara	UK-951	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.17	1	6060	[2.17,1.0,6060.0]	0.0
8	GO_FIRST	G8-334	Delhi	Early_Morning	zero	Morning	Mumbai	Economy	2.17	1	5954	[2.17,1.0,5954.0]	0.0
9	GO_FIRST	G8-336	Delhi	Afternoon	zero	Evening	Mumbai	Economy	2.25	1	5954	[2.25,1.0,5954.0]	0.0

BAB 4

KESIMPULAN

Analisis data penerbangan dengan volume besar dan keragaman tinggi pada case study ini menunjukkan bahwa teknologi dan arsitektur big data dibutuhkan agar analisis data dapat dilakukan secara optimal. Dataset Clean_Dataset.csv berisi informasi detail penerbangan yang sangat berharga jika dianalisis lebih lanjut.

Apache Spark dipilih sebagai mesin pemrosesan data utama karena kemampuan komputasi terdistribusinya. PySpark menyediakan DataFrame API yang sangat membantu untuk pemrosesan data. HDFS digunakan sebagai sistem penyimpanan data yang terdistribusi. Tools ini mampu menangani data dalam volume besar.

Melalui penerapan arsitektur big data yang dirancang, analisis data penerbangan dapat dilakukan mulai dari proses ingestion, pembersihan, transformasi, EDA, hingga pemodelan machine learning untuk mengekstraksi informasi penting seputar pola, tren, dan prediksi yang bermanfaat bagi industri penerbangan.

Rekomendasi ke depan antara lain adalah menambahkan analytics dashboard sebagai output analisis agar hasilnya dapat divisualisasikan dan diakses oleh pengguna secara interaktif melalui portal web. Infrastruktur juga perlu dimonitor dan dievaluasi performanya secara berkala agar tetap optimal dan responsif seiring pertumbuhan data yang sangat pesat di masa mendatang.

REFERENSI

[1] Shubham Bathwal, “Flight Price Prediction,” Kaggle.com, 2022. https://www.kaggle.com/datasets/shubhambathwal/flight-price-prediction/ (accessed Dec. 04, 2023).

[2] A. Shah, “What is Apache Spark? - Ashish Shah - Medium,” Medium, May 2018. https://medium.com/@ashish1512/what-is-apache-spark-e41700980615 (accessed Dec. 04, 2023).

[3] “Introduction to PySpark Distributed Computing with Apache Spark,” GeeksforGeeks, Aug. 16, 2017. https://www.geeksforgeeks.org/introduction-pyspark-distributed-computing-apache-spark/ (accessed Dec. 04, 2023).

[4] B. Saini, “What is Apache Hadoop in Big Data - CodeX - Medium,” Medium, Feb. 04, 2021. https://medium.com/codex/what-is-apache-hadoop-in-big-data-1c542e32d3df (accessed Dec. 04, 2023).

[5] Big Data Framework, “The Four V’s of Big Data | Enterprise Big Data Framework©,” Enterprise Big Data Framework©, Mar. 12, 2019. https://www.bigdataframework.org/the-four-vs-of-big-data/ (accessed Dec. 04, 2023).