DOKUMEN SPESIFIKASI PROYEK *BIG DATA*

Implementasi Ekosistem Hadoop untuk Pemetaan Daerah Rawan Kemiskinan Berdasarkan Profil Kependudukan di Sumatera

Kelompok 18

Try Yani Rizki Nur Rohmah	122450020
Nabiilah Putri Karnaia	122450029
Priska Silvia Ferantiana	122450053
Naufal Fakhri	122450089

PROGRAM STUDI SAINS DATA

FAKULTAS SAINS

INSTITUT TEKNOLOGI SUMATERA

Mei 2025

Daftar Isi

1. Pendahuluan

1.1 Latar Belakang

Kemiskinan merupakan masalah yang pelik dan multifaset karena dampak dari berbagai faktor sosial, ekonomi, dan demografi. Di Sumatera, disparitas yang terjadi antar Kabupaten/Kota masih menjadi permasalahan nyata, diantaranya adalah Daerah yang terbelakang dalam pendidikan, pendapatan serta akses terhadap layanan dasar seperti air bersih dan sanitasi. Untuk menganalisis peta daerah yang termasuk rawan kemiskinan, setidaknya pertama-tama, butuh analisis yang mendalam serta data yang terstruktur dan detail. Akan tetapi, data kependudukan untuk tingkatan daerah kabupaten sudah sangat besar serta komplek, sehingga tidak memungkinkan untuk menggunakan metode pemrosesan sinkron.

Maka dari itu, penggunaan teknologi big data semacam Hadoop akan lebih diperhitungkan. Selain itu, karena dapat memproses data secara bersamaan dan terdistribusi dengan menggunakan HDFS, MapReduce, Hive, HBase, menjadikan Hadoop sangat efisien untuk pengolahan analisis data kemiskinan yang sangat besar. Proyek ini diterapkan di Sumatera dan untuk menjadi langkah nyata memanfaatkan teknologi tersebut.

1.2 Tujuan Dokumen

Dokumen ini ditujukan untuk memberikan penjelasan mendetail tentang spesifikasi sistem pemetaan daerah rawan kemiskinan berbasis ekosistem Hadoop secara teknis dan fungsional. Dalam dokumen ini diuraikan tahap perancangan arsitektur sistem dan pemrosesan data, sampai pada pelaksanaan pipelining analitik pada Hadoop dengan berbagai komponen seperti HDFS, MapReduce, Hive, dan HBase.

Dokumen ini juga berisi analisis sebagai pemenuhan dari tugas untuk memberi motivasi dalam mengembangkan keterampilan analitis, proyek ini dikerjakan oleh kelompok 18 selaku bagian dari proyek kursus Big Data dalam program studi Sains Data Faculty of Science Institut Teknologi Sumatera. Dalam hal ini, dataset yang menjadi fokus utama diperoleh dari distrik dan kota di pulau Sumatera yang memiliki indikator sosial ekonomi berupa: jumlah penduduk dalam kategori kemiskinan, durasi pendidikan, pengeluaran per capita, dan alokasi terhadap layanan dasar.

Khusus dalam dokumen ini, kerangka untuk sistem yang dirancang agar mudah dirapal bersifat modular, alur data slack testing dari ingestion ke view tracing scrape adalah dokumentasi sistem untuk transparan, bisa dipakai ulang, dan diperluas tanpa terdeteksi batasan fungsi. Diharapkan semua dokumen ini berperan sebagai dokumen teknis pada proyek serta bimbingan untuk mendalami implementasi sistem big data dalam pemetaan sosial data yang diproses secara otomatis.

1.3 Lingkup Sistem

Lingkup sistem dalam proyek ini mencakup seluruh proses pemrosesan data kependudukan yang bertujuan untuk memetakan daerah rawan kemiskinan di wilayah

Sumatera menggunakan pendekatan big data. Sistem dirancang untuk bekerja dalam lingkungan lokal berbasis Docker dan WSL2, dengan menggunakan komponen utama dari ekosistem Hadoop.

Secara garis besar, ruang lingkup sistem meliputi:

- Ingest Data: Mengimport dataset kependudukan yang berasal dari sumber lokal dataset CSV ke dalam sistem file terdistribusi Hadoop (HDFS).
- Data Cleansing & Transformation: Memproses dan menghilangkan data yang tidak perlu, menghapus nilai kosong, menduplikat, serta merekayasa data dengan Apache Spark menjadi lebih efisien.
- Data Storage: Menghasilkan data yang dihimpun secara transformasi serta menyimpannya dalam bentuk Parquet di HDFS dan mendefinisikan skema tabel dalam sistem Hive untuk kebutuhan analitik.
- Analitik dan Klasifikasi: Mengeksekusi job MapReduce atau query Hive untuk penghitungan skenario sosial ekonomi dan klasifikasi kemiskinan menjadi 3 kategori, rendah, sedang, tinggi.
- Query SQL dan Penyimpanan NoSQL: Hive untuk analisis data dan HBase penyimpanan data klasifikasi akhir untuk akses cepat.
- Visualisasi Hasil: Tabel analisis untuk klasifikasi tingkat kemiskinan dan perbandingan ekonomi sosial antar kabupaten atau kota
- Monitoring and Logging: Memanfaatkan Apache Ambari untuk pemantauan.

1.4 Signifikansi Dokumen

Dari sisi proyek ini bermanfaat karena memberi pemahaman mengenai penerapan teknologi big data pada problematika sosial. mahasiswa juga berpraktek dalam pembuatan analitik pipeline.Secara praktis, sistem yang dibangun dapat digunakan untuk perencanaan dan penganggaran program mayor dalam asosiasi kesejahteraan sosial yang berbasis data. Klasifikasi daerah rawan kemiskinan yang dibuat dapat membantu pihk-pihak terkait mendefinisikan program intervensi yang lebih efektif dan efisien, bahkan bagi program-program jangka panjang. Lebih dari yang telah dijabarkan, siswa bisa dikenalkan dengan permasalahan lain yang lebih kompleks yang berhubungan dengan pendidikan atau kesehatan bahkan dalam pemberian bantuan sosial, untuk menjadikan sistem ini lebih umum yang kaya akan lebih banyak penggunaan.

2. Deskripsi Umum

2.1 Perspektif Sistem

Sistem yang dikembangkan dalam proyek ini merupakan sebuah sistem informasi berbasis big data yang dirancang untuk melakukan pemetaan daerah rawan kemiskinan di wilayah Sumatera dengan memanfaatkan teknologi ekosistem Hadoop. Sistem ini

dikembangkan dengan mengadopsi prinsip *distributed computing* yang memungkinkan pemrosesan data skala besar secara paralel dan efisien. Dalam konteks proyek ini, sistem memanfaatkan berbagai komponen utama Hadoop seperti HDFS, Hive, Spark, dan HBase untuk menyimpan, memproses, dan menganalisis data kependudukan dari berbagai kota dan kabupaten di Sumatera.

Sistem dirancang dengan perspektif modular dan scalable, artinya setiap tahapan pemrosesan data, mulai dari ingestion (pengambilan data), pembersihan, transformasi, hingga visualisasi akhir dapat dikembangkan atau diperluas secara terpisah sesuai kebutuhan. Perspektif ini memungkinkan sistem tidak hanya relevan untuk analisis kemiskinan, tetapi juga bisa digunakan untuk studi sosial-ekonomi lainnya di masa depan. Sistem berjalan dalam lingkungan lokal menggunakan Docker dan WSL2, namun tetap mensimulasikan arsitektur cluster Hadoop secara lengkap seperti di lingkungan produksi. Hal ini menjamin bahwa sistem dapat diadaptasi ke skala yang lebih besar jika dibutuhkan.

2.2 Fungsi Sistem Utama

Sistem memiliki beberapa fungsi utama yang dirancang untuk mendukung alur kerja data pipeline end-to-end:

- Import Data
  Mengimpor dataset kependudukan dari file eksternal (CSV) ke dalam sistem file terdistribusi (HDFS). Proses ini dilakukan secara otomatis dan periodik melalui *cron job* atau sistem orkestrasi seperti Apache Airflow.
- Data Cleansing & Transformation
  Data yang telah diimpor akan diproses menggunakan Apache Spark untuk membersihkan nilai kosong, mendeteksi duplikasi, dan menyelaraskan format data. Output dari proses ini disimpan dalam format Parquet untuk efisiensi penyimpanan.
- Data Storage
  Data hasil transformasi disimpan dalam lapisan Silver dan Gold dalam HDFS, yang kemudian dikonversi menjadi tabel-tabel analitik di Hive. Ini memudahkan pengguna untuk melakukan query terhadap data secara langsung.
- Analisis dan Klasifikasi
  Sistem menjalankan query analitik di Hive atau job Spark untuk melakukan klasifikasi daerah rawan kemiskinan berdasarkan indikator seperti pengeluaran per kapita, akses terhadap layanan dasar, dan tingkat pendidikan.
- Query dan Akses Cepat
  Data hasil klasifikasi disimpan dalam HBase, memungkinkan akses cepat dan integrasi dengan sistem visualisasi atau aplikasi downstream lainnya.
- Visualisasi
  Hasil analitik ditampilkan dalam bentuk visualisasi tabular dan grafik menggunakan Apache Superset, sehingga dapat dipahami oleh pengguna dari berbagai latar belakang.

- Monitoring dan Logging
  Seluruh sistem dipantau menggunakan Apache Ambari untuk memastikan performa cluster tetap optimal dan semua layanan berjalan dengan baik.

2.3 Karakteristik Pengguna

Sistem ini ditujukan untuk berbagai jenis pengguna dengan tingkat pemahaman teknologi yang berbeda-beda:

- Analis Data dan Peneliti Sosial:
  Memerlukan data terstruktur dan hasil klasifikasi untuk digunakan dalam laporan atau penelitian akademik. Mereka menggunakan fitur query di Hive untuk analisis mendalam.

- Perencana Kebijakan Pemerintah:
  Membutuhkan hasil visualisasi yang mudah dipahami untuk menentukan alokasi anggaran atau kebijakan intervensi sosial di wilayah rawan kemiskinan.

- Administrator Teknologi:
  Bertanggung jawab atas pengelolaan dan pemantauan cluster, termasuk instalasi, konfigurasi, backup, dan manajemen resource sistem.

- Mahasiswa dan Pengajar:
  Menggunakan sistem ini sebagai media pembelajaran dalam mata kuliah Big Data, untuk memahami alur kerja pemrosesan data dalam skala besar menggunakan teknologi Hadoop.

3. Spesifikasi Proyek

3.1 Ringkasan

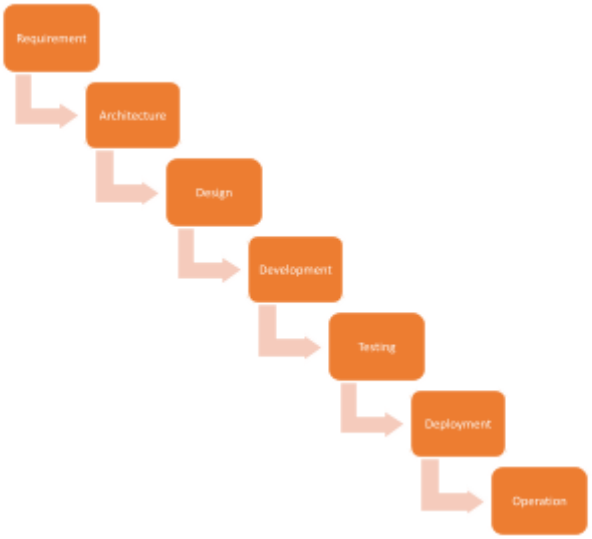
Proyek ini merupakan bagian dari implementasi pembelajaran pada mata kuliah Big Data di Program Studi Sains Data, Institut Teknologi Sumatera. Proyek ini mengusung ide besar pemanfaatan ekosistem Hadoop untuk memetakan daerah rawan kemiskinan berdasarkan data kependudukan. Dengan memanfaatkan kekuatan Hadoop yang dapat mengolah data dalam jumlah besar secara paralel dan terdistribusi, proyek ini menyajikan solusi data-driven dalam menghadapi permasalahan sosial yang kompleks seperti kemiskinan.

Penerapan dilakukan pada wilayah Sumatera, dengan fokus pada indikator sosial ekonomi seperti tingkat pengeluaran per kapita, tingkat pendidikan, dan akses layanan dasar. Data yang digunakan berasal dari dataset publik dan disusun ke dalam arsitektur *data lake* berlapis (bronze, silver, gold). Proyek ini juga bertujuan menjadi portofolio praktikal bagi mahasiswa untuk menguasai teknologi big data modern.

3.2 Metode Proyek

Metode Waterfall adalah salah satu model proses rekayasa perangkat lunak tradisional yang bersifat linear dan berurutan. Proses pengembangan sistem dilakukan tahap demi tahap, dimulai dari:

- Requirement – pengumpulan dan analisis kebutuhan sistem,
- Architecture – perancangan arsitektur sistem,
- Design – desain sistem dan komponen,
- Development – implementasi atau pengkodean,
- Testing – pengujian terhadap sistem yang telah dibangun,
- Deployment – peluncuran atau penerapan sistem ke lingkungan produksi,
- Operation – pemeliharaan dan penggunaan sistem oleh pengguna akhir.



Gambar 1. Metode Waterfall

3.3 Studi Kasus

Studi kasus pada proyek ini dilakukan dengan mengambil fokus pada pemetaan daerah rawan kemiskinan di wilayah Sumatera berdasarkan data kependudukan dan indikator sosial ekonomi. Proyek ini mengkaji bagaimana data-data seperti jumlah penduduk miskin, tingkat pengeluaran per kapita, rata-rata lama sekolah, serta akses terhadap pelayanan dasar (seperti air bersih dan sanitasi) dapat digunakan untuk mengidentifikasi tingkat kerentanan kemiskinan pada masing-masing kabupaten atau kota. Melalui implementasi sistem berbasis ekosistem Hadoop, data yang semula tersebar dan berukuran besar dapat diolah secara efisien menggunakan HDFS, Spark, dan Hive untuk menghasilkan klasifikasi daerah miskin ke dalam tiga kategori, yaitu rendah, sedang, dan tinggi.

3.4 Arsitektur Sistem dan Tools yang Digunakan

Arsitektur sistem dalam proyek ini dirancang berbasis Data Lake berlapis (Medallion Architecture) dengan memanfaatkan ekosistem Hadoop yang dirancang untuk pengelolaan data kependudukan skala besar dengan efisien melalui batch processing, dimulai dari ingestion data mentah hingga visualisasi hasil analitik. Sistem ini dijalankan secara lokal menggunakan Docker Compose. Sistem dibagi ke dalam beberapa lapisan utama berdasarkan model Medallion Architecture:

1. Bronze Layer (Raw Data)
    - Menyimpan data mentah dari berbagai sumber (CSV) tanpa transformasi.
    - Tools:
        - Bash: otomatisasi tugas seperti download data, upload ke HDFS, penjadwalan job.
        - HDFS: penyimpanan terdistribusi untuk file mentah dalam format CSV.
2. Silver Layer (Cleaned and Transformed Data)
    - Menyimpan data yang telah dibersihkan, distandarisasi, dan divalidasi.
    - Tools:
        - Apache Spark: melakukan ETL, lalu disimpan dalam format Parquet.
        - Apache Hive: membuat struktur tabel dari data mentah yang telah dibersihkan.
3. Gold Layer (Analytics-Ready Data)
    - Menyimpan data agregasi siap pakai untuk kebutuhan analitik dan visualisasi.
    - Tools:
        - Apache Spark: melakukan agregasi kompleks.
        - Apache Hive: membuat tabel agregasi untuk analisis.
4. Visualisasi dan analisis

    Visualisasi dan penyajian insight dilakukan menggunakan Tableau Desktop. Hasil klasifikasi tingkat kemiskinan dari proses Hadoop disimpan dalam bentuk tabel (CSV/Excel/Parquet), kemudian diekspor dari Hive atau file lokal, dan diimpor ke Tableau untuk divisualisasikan.

5. Manajemen dan Orkestrasi

    Sistem dijalankan secara manual dan terjadwal oleh pengguna. Proses monitoring dilakukan melalui command line dan log file tanpa menggunakan Apache Ambari.

4. Metode Umum

4.1 Analisis Kebutuhan (*Requirement Analysis)*

Tujuan:

1. Memahami struktur dan kompleksitas data sosial kependudukan.
2. Mengidentifikasi stakeholder utama: pemerintah daerah, peneliti sosial, dan analis data.

3. Mendeteksi pola ketimpangan ekonomi antar kabupaten/kota di Pulau Sumatera.
4. Menentukan output berupa pemetaan dan ringkasan statistik kemiskinan.

- **Tabel Kebutuhan Fungsional**

Kebutuhan Fungsional	Prioritas
Sistem dapat membaca dan memuat file CSV dataset kemiskinan ke HDFS.	Tinggi
Sistem dapat membersihkan data dari nilai kosong dan duplikat menggunakan PySpark.	Tinggi
Sistem dapat menyimpan data hasil pembersihan dalam format parquet di HDFS	Tinggi
Sistem dapat menjalankan query Hive untuk analisis indikator sosial ekonomi	Tinggi
Sistem dapat melakukan klasifikasi wilayah berdasarkan persentase kemiskinan	Tinggi
Sistem dapat menyimpan hasil klasifikasi ke dalam database HBase untuk akses cepat	Sedang
Sistem dapat menghasilkan tabel rangking dan grafik sederhana non spasial	Sedang
Sistem mendukung penambahan data baru secara berkala ke dalam pipeline	Tinggi

- **Tabel Kebutuhan Non Fungsional**

Kebutuhan Non-Fungsional	Prioritas
Sistem dapat berjalan pada lingkungan lokal berbasis Docker dan WSL 2 secara stabil.	Tinggi
Komponen Hadoop Stack (HDFS, Hive, HBase) dapat diatur dan dimonitor secara terpusat.	Sedang
Proses ETL dan query dijalankan secara batch dengan waktu pemrosesan yang efisien.	Tinggi
Penyimpanan data menggunakan format Parquet untuk efisiensi ruang dan kecepatan baca.	Tinggi
Data pipeline dapat direplikasi oleh	Sedang

pengguna lain melalui dokumentasi GitHub.	
Sistem mendukung modularitas (tiap tahap dapat dikembangkan atau diganti secara mandiri).	Rendah
Semua script dan konfigurasi terdokumentasi dengan baik dan tersimpan dalam repository.	Tinggi

4.2 Perancangan Sistem: Arsitektur dan Desain (*System Design*)

Arsitektur Data

Tabel Arsitektur Medallion – Batch Processing dengan Data Lake

(NOTE: ISI TABELNYA UDH BERUBAH, kl mau revisi silahkan)

Lapisan	Deskripsi	Komponen Utama (Tools)	Format Data	Tujuan
Bronze	Menyimpan data mentah dari dataset Kaggle tanpa transformasi.	HDFS, Bash	CSV	Menyimpan data asli untuk arsip dan jejak audit.
Silver	Menyimpan data yang telah data yang telah dibersihkan, distandarisasi, dan divalidasi.	- Apache Spark (ETL), Apache Hive	Parquet	Data terstruktur siap dianalisis. Terintegrasi, bersih, dan efisien dalam penyimpanan.
Gold	Menyimpan data agregat per wilayah/atribut untuk klasifikasi dan visualisasi.	- Apache Hive (Analytic Queries)- Apache Spark (Aggregation)	Parquet / ORC	Menyediakan insight siap pakai untuk pengambilan keputusan.

Desain Infrastruktur

Arsitektur Cluster Lokal (Hadoop Cluster)

Komponen	Jumlah Node	Peran dan Deskripsi
Hadoop Namenode	1	Master node untuk HDFS, mengelola metadata file.
Hadoop Datanode	2	Penyimpanan data yang terdistribusi.
ResourceManager	1	Komponen YARN yang menjadwalkan job Spark/Hive.
NodeManager	2	Mengelola eksekusi task pada masing-masing worker node.
Apache HiveServer2	1	Layanan query engine untuk Hive SQL yang dapat diakses melalui Beeline/BI tools.
Apache Spark Master	1	Koordinator eksekusi job Spark batch.

Apache Spark Worker	2	Mengeksekusi job Spark (ETL, transformasi, agregasi).
Apache Hive Metastore	1	Metadata store untuk skema dan tabel Hive.
Ambari Server	1	Monitoring dan manajemen layanan Hadoop ecosystem.
Superset	1	Untuk visualisasi data gold layer.

Tabel Daftar Teknologi Apache Projects yang digunakan

No	Teknologi	Kategori	Fungsi Utama
1	Hadoop HDFS	Storage	Menyimpan data mentah (bronze), hasil transformasi (silver), dan agregasi (gold) secara terdistribusi.
2	Hadoop YARN	Resource Management	Mengatur sumber daya (CPU/RAM) dan menjadwalkan eksekusi job Spark secara terdistribusi.
3	Apache Hive	Query Engine / Metadata	Menyediakan SQL-like interface (HiveQL) dan metadata management untuk query analitik.
4	Apache Spark	ETL / Analytics Engine	Melakukan pembersihan data, transformasi, dan agregasi batch processing.

Orkestrasi Directed Acyclic Graphs (DAG) di Airflow

dag_export_pipeline:

├── task_1: fetch_csv_from_source
├── task_2: load_to_hdfs_bronze
├── task_3: spark_transform_to_silver
├── task_4: spark_aggregate_to_gold
├── task_5: hive_refresh_tables
└── task_6: notify_team_or_export_to_dashboard

Orkestrasi Alur Kegiatan Sistem

Urutan	Aktivitas	Layer	Tools
1	Ambil file dari Kaggle dan unggah ke HDFS	Bronze	Bash, HDFS CLI
2	Simpan data mentah tanpa ubahan	Bronze	HDFS
3	Validasi schema dan lakukan pembersihan data	Silver	Apache Spark, Hive
4	Ubah format ke Parquet dan simpan di lokasi refined	Silver	Spark + HDFS
5	Agregasi dan perhitungan metrik ekspor berdasarkan waktu dan kategori	Gold	Spark SQL / Hive
6	Simpan hasil agregasi akhir dalam bentuk tabel analitik	Gold	HDFS, Hive
7	Visualisasi dan eksplorasi data	Gold	Tableau Desktop
8	Monitoring dan manajemen cluster	Seluruh layer	Command Line

4.3 Implementasi (Implementation)

Mewujudkan arsitektur dan perancangan sistem menjadi sistem nyata (real deployment) pada lingkungan lokal (Docker Desktop VM) berbasis Linux, menggunakan tools utama dari Hadoop ecosystem dan Apache Foundation.

Lingkungan Implementasi

Komponen	Spesifikasi
OS	Ubuntu Server 22.04 (via Docker VM) – OS komputer: Windows 11
Cluster	Pseudo-distributed Hadoop Cluster dengan 2 Node
Orkestrasi	Shell Script + Crontab (opsional: Apache Nifi atau Airflow)
Core Tools	Hadoop, HDFS, Hive, Spark, Ambari, Superset, Airflow
Penyimpanan	HDFS sebagai distributed storage
Format Data	CSV (bronze), Parquet/ORC (silver & gold)

Instalasi dan Setup Cluster

a. Persiapan Cluster Lokal

1. Install Docker Desktop + WSL2 di Windows 11
2. Jalankan container berbasis image Hadoop
3. Gunakan docker-compose untuk spin-up:

    ✔ 2 NameNode

    ✔ 2 DataNode

    ✔ 1 Spark Master

    ✔ 1 Spark Worker

    ✔ 1 Hive Metastore + HiveServer2

    ✔ Ambari

    ✔ Superset

    ✔ Airflow

b. Struktur Docker-Compose

buat file docker-compose.yml yang memuat baris struktur instruksi docker.

Struktur Folder HDFS

Layer	Path HDFS	Format
Bronze	`/data/bronze/`	CSV, JSON
Silver	`/data/silver/`	Parquet
Gold	`/data/gold/`	Parquet
Temp (untuk temporary)	`/data/tmp/`	-

Implementasi Pipeline Batch

1. **Bronze Layer (Ingestion)**

✔ Jalankan bash script via crontab setiap jam/hari (contoh):

```
curl -o ekspor.csv https://data.example/api/ekspor
hdfs dfs -put -f ekspor.csv /data/bronze/ekspor_$(date +%F).csv
```

curl untuk scrapping atau unduh, hdfs dfs -put untuk upload berkas data.

2. **Silver Layer (Cleansing & Normalization)**

✔ Gunakan Spark Job:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName("CleanEkspor").getOrCreate()
df = spark.read.csv("/data/bronze/*.csv", header=True)
df_clean = df.dropDuplicates().na.fill("N/A")  # Cleaning
df_clean.write.parquet("/data/silver/ekspor_clean.parquet", mode="overwrite")
```

Simpan code pyspark dalam file .ipynb atau .py

3. **Gold Layer (Aggregation & Analytics)**

✔ Agregasi OLAP:

```
df = spark.read.parquet("/data/silver/ekspor_clean.parquet")
agg = df.groupBy("negara", "bulan").agg({"nilai": "sum"})
agg.write.parquet("/data/gold/ekspor_summary.parquet", mode="overwrite")
```

Simpan code pyspark dalam file .ipynb atau .py

Implementasi Query dan Visualisasi

✔ Hive Table (DDL)

```
CREATE EXTERNAL TABLE ekspor_summary (
  negara STRING,
  bulan STRING,
  total_nilai DOUBLE
)
STORED AS PARQUET
LOCATION '/data/gold/ekspor_summary.parquet';
```

Gunakan HiveQL untuk menjalankan kueri.

Pengembangan ETL

✔ Upload file mentah → HDFS (bronze)

✔ ETL Spark → silver layer (cleansing, parsing)

✔ Agregasi Spark → gold layer (komoditas, negara, volume)

- ✔ Pembuatan Hive Table dari gold layer

Dashboard

Import ke **Apache Superset** atau **Jupyter + PyHive** untuk eksplorasi.

Monitoring dan Logging

- ✔ **Log Spark**: Tersedia di UI http://localhost:8080
- ✔ **Ambari UI**: Monitoring cluster, resource, dan service
- ✔ **Log File**: Tersimpan otomatis di folder /logs/ di container jika dikonfigurasi

Output Implementasi

Output	Lokasi
Data mentah ekspor	`/data/bronze/`
Data bersih (normalized)	`/data/silver/`
Ringkasan nilai ekspor (OLAP)	`/data/gold/`
Tabel Hive (ekspor_summary)	Hive Metastore
Visualisasi ekspor	Superset / Jupyter Dashboard

4.4 Penerapan (*Deployment*)

Mengimplementasikan sistem ke lingkungan lokal berbasis Docker VM yang mensimulasikan cluster Hadoop untuk kebutuhan batch processing berbasis data lake.

Strategi Deployment

Tahapan	Deskripsi
1. Setup Environment	Instalasi Docker Desktop + Ubuntu VM, setup docker-compose cluster
2. Build & Configure	Jalankan dan konfigurasikan layanan Hadoop, Spark, Hive, HDFS, Superset
3. Data Ingestion	Gunakan `curl` dan `bash` script dalam cron job untuk mengambil data mentah ke Bronze layer
4. Spark Transformation	Deploy job Spark batch untuk cleansing (Silver) dan agregasi (Gold)
5. Hive Integration	Buat tabel Hive dari hasil Spark Gold Layer
6. Visualisasi	Deploy Apache Superset dan hubungkan ke Hive metastore

Teknologi Deployment

Tools	Peran
Docker	Virtualisasi container untuk seluruh layanan Hadoop cluster
Docker Compose	Manajemen multi-container untuk Hadoop, Hive, Spark
Bash + Crontab	Menjadwalkan ingestion data ke HDFS
Spark Submit	Menjalankan job ETL batch
Hive Metastore (MySQL/PostgreSQL)	Menyimpan metadata dan query table Gold layer

Apache Superset	Visualisasi akhir

Struktur File di Server

```
/opt/bigdata/
├── docker-compose.yml
├── data/
│   ├── bronze/
│   ├── silver/
│   └── gold/
├── scripts/
│   ├── ingest.sh
│   └── etl_spark.py
└── logs/
```

4.5 Pengujian (*Testing*)

Memastikan seluruh proses pipeline batch berjalan dengan baik dari ingestion hingga visualisasi.

Jenis Pengujian

Jenis Pengujian	Tujuan
Unit Test	Memastikan script individual seperti `etl_spark.py` berjalan benar
Integration Test	Validasi antara ingestion → HDFS → Spark → Hive
Data Quality Test	Cek missing values, duplikasi, dan format
Performance Test	Waktu eksekusi ingestion dan job Spark batch
End-to-End Test	Simulasi aliran data penuh dari ingestion ke dashboard Superset

Kasus Uji (Test Cases)

No	Kasus Uji	Deskripsi	Status
1	Data berhasil diambil dari API	File CSV terekam di `/data/bronze/`	✅
2	Spark berhasil membersihkan data	Dataset `silver` tidak memiliki missing/null	✅
3	Spark Gold menghasilkan ringkasan agregat	Data tersimpan di `/data/gold/`	✅
4	Hive dapat membaca tabel gold layer	Query SELECT berhasil dari Hive	✅
5	Superset tampilkan grafik ekspor per negara	Dashboard dapat divisualisasikan	✅
6	File log tertulis saat job dijalankan	File `/logs/job.log` terbentuk	✅

Tools Testing

Tools	Fungsi
Jupyter Notebook	Validasi manual output Spark dan Hive
Bash + Log File	Logging ingestion dan Spark jobs
Hive CLI	Uji query tabel Gold
Superset	Validasi data akhir dari user interface

4.6 Analitik (*Analytics*)

Melakukan analisis lanjutan terhadap data ekspor bahan pangan yang telah diproses hingga Gold Layer, menggunakan tools **Apache Spark MLlib** untuk menghasilkan model prediktif dan wawasan berbasis machine learning, **Hive** Menyimpan hasil analisis model dan input agregat, **Jupyter VSCode** Eksperimen awal analitik dan eksplorasi data, dan **Superset** untuk visualisasi hasil prediksi atau clustering insight.

Tujuan Analitik ML

- ✔ Memprediksi **volume ekspor** untuk bulan/tahun berikutnya berdasarkan tren historis.
- ✔ Mengelompokkan **negara tujuan ekspor** berdasarkan pola pembelian (clustering).
- ✔ Menilai **faktor utama** yang mempengaruhi tinggi-rendahnya ekspor (feature importance).

Data Input untuk MLlib

1. Sumber: Tabel **Gold Layer** di Hive (hasil agregasi dan pembersihan).
2. Format: Parquet/ORC.
3. Kolom:
    - ✔ negara_tujuan
    - ✔ tahun
    - ✔ bulan
    - ✔ komoditas
    - ✔ volume_ton
    - ✔ harga_usd
    - ✔ total_usd

Metodologi Analitik (Pilih 1)

No	Analisis	Algoritma Spark MLlib	Tujuan
1	Prediksi volume ekspor	Linear Regression / Random Forest	Mengetahui estimasi ekspor di bulan/tahun berikutnya
2	Segmentasi negara ekspor	KMeans Clustering	Mengelompokkan negara berdasarkan kesamaan perilaku impor
3	Faktor penting ekspor	Feature Importance (Tree-based)	Mengetahui fitur mana yang paling berpengaruh terhadap ekspor

Tahapan Analisis

1. **Load Data**: Baca data Gold Layer dari Hive/Parquet menggunakan Spark SQL.
2. **Preprocessing**:
    - ✔ Normalisasi fitur (MinMaxScaler)
    - ✔ Encoding fitur kategorikal (komoditas, negara_tujuan)
    - ✔ Handle missing/null (jika ada)
3. **Splitting Data**:

✔ 80% Training, 20% Testing (dengan randomSplit)

4. **Modeling**:
   ✔ Latih model regresi dan clustering
5. **Evaluasi**:
   ✔ Metrik: RMSE, MAE (regresi), Silhouette Score (clustering)
6. **Saving Model**:
   ✔ Simpan model di /models/ menggunakan model.save("path")
7. **Inference**:
   ✔ Jalankan prediksi dan simpan hasil di tabel analitik Hive atau folder Gold/Insight.

4.7 Dataset *(Contoh)*

Data dari berbagai sumber di kumpulkan menjadi satu dalam data lake melalui bacth processing.

Contoh Deskripsi Data

Kolom	Tipe Data	Deskripsi
`negara_tujuan`	String	Negara tujuan ekspor
`tahun`	Integer	Tahun ekspor
`bulan`	Integer	Bulan ekspor
`komoditas`	String	Jenis komoditas pangan yang diekspor (misal: beras, gula, kelapa)
`volume_ton`	Float	Volume ekspor dalam satuan ton
`harga_usd_per_ton`	Float	Harga jual per ton dalam USD
`total_usd`	Float	Total nilai ekspor (volume × harga) dalam USD

Dataset 1: ekspor_bahan_pangan.csv

negara_tujuan	tahun	bulan	komoditas	volume_ton	harga_usd_per_ton	total_usd
Jepang	2023	1	Beras	1200	400	480000
Malaysia	2023	1	Gula	900	350	315000
Australia	2023	1	Kelapa	1000	300	300000
Jepang	2023	2	Beras	1250	410	512500
Malaysia	2023	2	Gula	920	345	317400
Australia	2023	2	Kelapa	980	310	303800
Jepang	2023	3	Beras	1180	415	489700
Malaysia	2023	3	Gula	950	340	323000
Australia	2023	3	Kelapa	970	320	310400

Dataset 2: cuaca_ekspor.csv

Data cuaca di pelabuhan saat ekspor berlangsung (penting untuk analisis delay atau risiko transportasi)

tanggal	pelabuhan	suhu_celsius	kelembaban	kecepatan_angin_kmh	kondisi_cuaca
2023-01-10	Tanjung Priok	30	75	10	Cerah
2023-01-10	Belawan	32	80	15	Hujan ringan
2023-01-11	Tanjung Perak	31	70	8	Berawan

Dataset 3: biaya_logistik.csv

Biaya-biaya logistik yang terkait dengan kegiatan ekspor tiap bulan dan pelabuhan.

bulan	tahun	pelabuhan	biaya_pengemasan_usd	biaya_transport_usd	biaya_total_usd
1	2023	Tanjung Priok	5000	20000	25000
1	2023	Belawan	4800	18000	22800
1	2023	Tanjung Perak	5200	21000	26200

Dataset 4: kurs_mata_uang.csv

Kurs mata uang sebagai acuan nilai ekspor terhadap nilai tukar saat itu.

tanggal	mata_uang	nilai_tukar_usd_idr
2023-01-10	IDR	15500
2023-02-10	IDR	15300
2023-03-10	IDR	15000

Dataset 5: permintaan_global.csv

Permintaan global per komoditas berdasarkan negara tujuan.

negara_tujuan	komoditas	bulan	tahun	indeks_permintaaan	tren_yoy
Jepang	Beras	1	2023	88.5	3%
Malaysia	Gula	1	2023	91.2	1.5%
Australia	Kelapa	1	2023	85.0	2.8%

Format Penyimpanan

Berikut format data setelah di ingest ke Hadoop setiap lapisannya

1. **Bronze Layer**: (Raw) CSV di HDFS /bronze/ekspor/2023-01.csv
2. **Silver Layer**: (Cleaned) Parquet /silver/ekspor/2023.parquet
3. **Gold Layer**: (Aggregated) ORC/Parquet dengan hasil agregasi per komoditas & negara

4.8 Aliran Data (*Pipeline*) (buat dalam bagan visual bukan seperti ini)

[1] Data Source (CSV, JSON, XML, API, FTP, DB)
- ekspor_bahan_pangan.csv
- cuaca_ekspor.csv
- biaya_logistik.csv
- kurs_mata_uang.csv
- permintaan_global.csv

[2] Bronze Layer (HDFS - Raw Zone)

- Simpan file mentah ke HDFS (/datalake/bronze/)
- Format tetap CSV/JSON (belum diproses)
- Metadata tracking dicatat (timestamp, sumber, ukuran) via Hive Metastore

[3] Ingestion & Staging (Apache Spark + Hive) – Bronze ke Silver
- Gunakan Spark untuk baca data dari HDFS (bronze)
- Lakukan parsing, validasi skema, handling missing values
- Standardisasi waktu, format tanggal, dan satuan
- Tulis ke Silver Layer dalam format kolumnar (Parquet/ORC)

[4] Silver Layer (HDFS - Clean Zone)
- Path: /datalake/silver/
- Data sudah bersih dan distandardisasi
- Join antar dataset (ekspor + cuaca + biaya + kurs)
- Simpan hasil ke format Parquet untuk efisiensi query

[5] Enrichment & Transformation (Apache Spark) – Silver ke Gold
- Hitung agregasi bulanan, nilai ekspor (dalam USD/IDR)
- Integrasi permintaan global
- Siapkan data untuk analitik & ML

[6] Gold Layer (HDFS - Curated Zone)
- Path: /datalake/gold/
- Dataset siap pakai untuk analitik dan machine learning
- Format: Parquet atau Hive Tables
- Gunakan Hive untuk query BI, dan Spark MLlib untuk model prediksi

[7] Analytics Layer
- Apache Hive (OLAP + BI)
- Apache Superset atau Grafana (dashboard)
- Apache Spark MLlib (klasifikasi/prediksi permintaan/volume ekspor)

[8] Monitoring & Orchestration
- Apache Airflow → Workflow terjadwal
- Apache Ambari → Monitoring cluster & resource

5. Lampiran

5.1 Repositori Portofolio

Silahkan isi deksripsi url atau barcode yang mengarahkan ke repositori github proyek.

5.2 Lampiran Code

Silahkan isi file-file code yang digunakan di proyek.