**UTS *Take-Home* Mata Kuliah Big Data for Data Science**

**Prediksi Tren Penjualan di Amazon menggunakan Arsitektur Big Data dan Machine Learning**

**Dosen Pengampu: Adam Sekti Aji, S.Kom., M.Kom**



**Disusun oleh:**

* **Lathif Ramadhan (5231811022)**
* **Rama Panji Nararendra Cahaya (5231811033)**

**PROGRAM STUDI SAINS DATA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**YOGYAKARTA**

**2025**

# ABSTRAK

Sebagai platform *e-commerce* terbesar di dunia, Amazon menghadapi tantangan kompleks dalam mengelola jutaan transaksi harian yang menghasilkan data heterogen, mulai dari riwayat pembelian terstruktur hingga ulasan pelanggan tidak terstruktur. Laporan ini menyajikan rancangan arsitektur Big Data terintegrasi yang memadukan teknologi *real-time processing* dan *batch processing* untuk mendukung prediksi tren penjualan berbasis Machine Learning. Sistem ini dirancang dengan komponen utama: (1) Apache Kafka sebagai data ingestion layer untuk menangani aliran data real-time, (2) Hadoop HDFS dan Amazon S3 sebagai penyimpanan terdistribusi, (3) Apache Spark untuk pemrosesan data paralel, serta (4) pipeline Machine Learning berbasis XGBoost dan LSTM yang diintegrasikan dengan *feature store* untuk menangkap pola temporal dan korelasi multivariat. Arsitektur ini dikembangkan dengan prinsip skalabilitas, memungkinkan integrasi data eksternal (seperti cuaca dan tren media sosial) melalui API, serta dilengkapi dashboard interaktif (Power BI) untuk visualisasi prediksi dan manajemen inventaris. Solusi ini diharapkan menjadi kerangka kerja adaptif yang menjembatani kebutuhan *low-latency analytics* dan *large-scale batch processing* dalam ekosistem *e-commerce* dinamis.

Kata Kunci: *Big Data*, Arsitektur Terdistribusi, *Real-time Analytics*, *Machine Learning*, Prediksi Penjualan, *E-commerce*, *Scalability*

**ABSTRACT**

*As the world's largest e-commerce platform, Amazon faces complex challenges in managing millions of daily transactions that generate heterogeneous data, ranging from structured purchase histories to unstructured customer reviews. This report presents an integrated Big Data architecture combining real-time processing (Kafka, Flink) and batch processing (Spark, Hadoop) technologies to support sales trend prediction using Machine Learning (XGBoost, LSTM). The system is designed with core components: (1) Apache Kafka as a data ingestion layer for real-time streams, (2) Hadoop HDFS and Amazon S3 for distributed storage, (3) Apache Spark for parallel data processing, and (4) an ML pipeline leveraging XGBoost (feature analysis) and LSTM (temporal patterns) integrated with a feature store. The architecture is built with scalability and fault tolerance principles, supporting external data (weather, social media) integration via REST APIs, along with an interactive dashboard (Power BI) for business intelligence and inventory management. This solution aims to be an adaptive framework bridging low-latency analytics and large-scale batch processing needs in dynamic e-commerce ecosystems.*

*Key Keywords:Big Data, Distributed Architecture, Real-time Analytics, Machine Learning, Sales Forecasting, E-commerce, Scalability*

# DAFTAR ISI

[ABSTRAK II](#_Toc195793410)

[DAFTAR ISI IV](#_Toc195793411)

[BAB I PENDAHULUAN 1](#_Toc195793412)

[2.1 Latar Belakang 1](#_Toc195793413)

[2.2 Tujuan 1](#_Toc195793414)

[BAB II LANDASAN TEORI 2](#_Toc195793415)

[3.1 Big Data di E-Commerce 2](#_Toc195793416)

[3.1.1. Volume 2](#_Toc195793417)

[3.1.2. Variety (Keragaman) 2](#_Toc195793418)

[3.1.3. Velocity (Kecepatan) 2](#_Toc195793419)

[3.2 Teknologi Big Data 2](#_Toc195793420)

[3.2.1. Hadoop HDFS 2](#_Toc195793421)

[3.2.2. Apache Spark 3](#_Toc195793422)

[3.2.3. Apache Kafka 3](#_Toc195793423)

[3.2.4. XGBoost 3](#_Toc195793424)

[3.2.5. LSTM (Long Short-Term Memory) 3](#_Toc195793425)

[3.3 Penelitian Terkait 3](#_Toc195793426)

[BAB III METODOLOGI 4](#_Toc195793427)

[4.1 Arsitektur Big Data 4](#_Toc195793428)

[4.1.1. Sumber Data (Data Sources) 6](#_Toc195793429)

[4.1.2. Lapisan Pengumpulan Data (Ingestion Layer) 6](#_Toc195793430)

[4.1.3. Lapisan Penyimpanan (Storage Layer) 6](#_Toc195793431)

[4.1.4. Lapisan Pemrosesan (Processing Layer) 7](#_Toc195793432)

[4.1.5. Lapisan Analitik & Machine Learning (Analytics Layer) 7](#_Toc195793433)

[4.1.6. Lapisan Bisnis (Business Layer) 7](#_Toc195793434)

[4.2 Dataset & Tools 8](#_Toc195793435)

[4.3 Alur Kerja Prediksi 8](#_Toc195793436)

[4.3.1. Pengumpulan Data 8](#_Toc195793437)

[4.3.2. Pembersihan Data 9](#_Toc195793438)

[4.3.3. Rekayasa Fitur 9](#_Toc195793439)

[4.3.4. Pelatihan Model 9](#_Toc195793440)

[4.3.5. Evaluasi Model 9](#_Toc195793441)

[4.3.6. Tantangan yang Mengubah Masalah Jadi Cerita 9](#_Toc195793442)

[BAB IV IMPLEMENTASI DAN HASIL 11](#_Toc195793443)

[5.1 Contoh Prediksi 11](#_Toc195793444)

[5.2 Tantangan & Solusi 11](#_Toc195793445)

[5.2.1. Kasus Misteri Diskontilasi 11](#_Toc195793446)

[5.2.2. Model yang Terlalu "Percaya Diri" 11](#_Toc195793447)

[BAB V KESIMPULAN DAN SARAN 13](#_Toc195793448)

[6.1. Kesimpulan 13](#_Toc195793449)

[6.2 . Saran 13](#_Toc195793450)

[6.2.1. Dari Akurasi ke Adaptasi 13](#_Toc195793451)

[6.2.2. Data untuk Semua, Bukan Halaman Para Ahli 13](#_Toc195793452)

[6.2.3. Antisipasi yang Lebih Manusiawi 13](#_Toc195793453)

[6.2.4. Jangan Lupa: Data adalah Cermin Masyarakat 14](#_Toc195793454)

[DAFTAR PUSTAKA 15](#_Toc195793455)

# BAB I PENDAHULUAN

## 2.1 Latar Belakang

Di balik kemudahan belanja online dan janji pengiriman cepat Amazon, tersembunyi kompleksitas data yang luar biasa. Setiap detik, jutaan transaksi dari berbagai belahan dunia tumpah ruah ke sistem, mulai dari pembelian produk sederhana hingga pesanan besar yang melibatkan rantai pasok global. Data ini bukan sekadar angka—ia merekam pola perilaku pelanggan, fluktuasi musiman, hingga dampak tak terduga dari tren sosial seperti viralnya suatu produk di media sosial. Namun, di tengah gempuran informasi ini, Amazon kerap menghadapi dilema klasik: bagaimana memastikan stok produk yang tepat, di waktu yang tepat, tanpa membebani gudang atau mengorbankan kepuasan pelanggan?

Permasalahan ini semakin kritis saat memasuki momen seperti Black Friday atau musim liburan, ketika kesalahan prediksi penjualan bisa berujung pada penumpukan barang tak laku atau justru kehilangan pelanggan karena out-of-stock. Di sisi lain, data yang tersedia—seperti riwayat pembelian, ulasan produk, hingga jejak klik pengguna—masih belum dimanfaatkan secara optimal. Inilah paradoks di era Big Data: perusahaan memiliki "lautan informasi", tetapi tanpa kapal yang tepat, mereka bisa tenggelam dalam kebisingan data.

Solusi konvensional seperti analisis statis atau rule-based forecasting sudah tidak lagi memadai. Dunia yang berubah cepat menuntut pendekatan lebih dinamis, di mana algoritma tidak hanya belajar dari masa lalu, tetapi juga adaptif terhadap perubahan tren secara real-time. Di sinilah kombinasi Big Data dan Machine Learning hadir sebagai jawaban—tidak hanya untuk mengolah data dalam skala masif, tetapi juga menemukan pola tersembunyi yang bisa menjadi kunci prediksi masa depan.

## 2.2 Tujuan

Laporan ini hadir dengan dua misi utama: membangun jembatan antara data mentah Amazon dan keputusan bisnis yang terukur. Pertama, merancang arsitektur Big Data yang mampu menampung, membersihkan, dan mengolah data heterogen—mulai dari angka penjualan hingga sentimen ulasan pelanggan—secara efisien. Kedua, menciptakan model prediktif yang tidak sekadar akurat, tetapi juga interpretatif, sehingga tim manajemen bisa memahami faktor apa saja yang mendorong kenaikan atau penurunan penjualan, apakah itu harga, musim, atau bahkan cuaca.

Lebih dari itu, sistem ini dirancang untuk menjadi mitra kerja yang responsif. Dengan menyematkan dashboard visual, kami ingin memastikan bahwa prediksi tidak berakhir sebagai laporan PDF yang tersimpan rapi, tetapi menjadi alat interaktif yang bisa digunakan oleh divisi logistik untuk merencanakan distribusi, oleh tim pemasaran untuk menyusun kampanye diskon, atau bahkan oleh pembeli untuk menegosiasikan harga dengan supplier. Pada akhirnya, tujuan paling hakiki adalah mengubah data dari "beban teknis" menjadi "cerita yang bernyawa"—cerita tentang pelanggan, produk, dan peluang yang menunggu untuk diraih.

# BAB II LANDASAN TEORI

## 3.1 Big Data di E-Commerce

Bayangkan pasar yang ramai, di mana setiap detik ribuan orang membeli, mengembalikan barang, atau sekadar melihat-lihat produk. Amazon, sebagai pasar digital terbesar, menghadapi situasi serupa—tetapi dalam skala yang sulit dibayangkan. Data di sini bukan hanya tentang angka penjualan, tapi juga jejak digital pelanggan: berapa lama mereka melihat suatu produk, kata-kata emosional dalam ulasan, atau bahkan perubahan harga pesaing yang terekam setiap menit. Data ini memiliki tiga ciri khas (3V Big Data) yang membuatnya unik:

### 3.1.1. Volume

Setiap hari, Amazon memproses transaksi senilai miliaran dolar. Jika data hariannya diubah menjadi buku, tumpukannya bisa setinggi Menara Eiffel dalam sebulan.

### 3.1.2. Variety (Keragaman)

Data tak hanya terstruktur rapi seperti tabel Excel (misal: invoice), tapi juga berupa teks ulasan yang subjektif, gambar produk, atau bahkan rekaman suara dari fitur Alexa. Sebuah ulasan negatif seperti, "Pengiriman terlambat, tapi produknya bagus," bisa jadi petunjuk untuk memperbaiki logistik tanpa mengurangi kualitas barang.

### 3.1.3. Velocity (Kecepatan)

Saat Prime Day, data mengalir deras seperti air bah. Sistem harus bisa memprosesnya secara real-time—misal, mendeteksi lonjakan permintaan headphone tertentu dalam 5 menit pertama promo, lalu langsung mengirim peringatan ke gudang.

Di dunia yang serba cepat ini, prediksi penjualan bukan lagi sekadar tebakan berbasis rata-rata bulan lalu. Ia harus mampu membaca pola dari data yang berantakan, beragam, dan terus bergerak.

## 3.2 Teknologi Big Data

Untuk mengelola data sekompleks ini, dibutuhkan teknologi yang tak hanya menyimpan, tapi juga memahami. Berikut adalah komponen kunci yang menjadi tulang punggung sistem:

### 3.2.1. Hadoop HDFS

Seperti gudang raksasa yang terbagi di ratusan lokasi. Jika satu server rusak, data tidak hilang karena tersimpan salinannya di tempat lain. Ini memastikan riwayat penjualan 5 tahun lalu tetap aman, sekaligus bisa diakses kapan saja untuk melatih model prediksi.

### 3.2.2. Apache Spark

Mesin pengolah data yang bekerja layaknya tim ahli statistik supercepat. Bayangkan Anda perlu menghitung rata-rata penjualan 100 juta produk dalam 1 jam. Spark membagi tugas ini ke ratusan komputer, mengumpulkan hasilnya, lalu menyajikannya dalam sekejap.

### 3.2.3. Apache Kafka

Pipa penyalur data real-time. Saat pelanggan mengklik "beli", Kafka langsung mengirimkan informasi ini ke sistem untuk diproses—seperti reporter yang melaporkan update berita langsung dari lapangan.

Untuk bagian kecerdasan-nya, dua algoritma Machine Learning dipilih:

### 3.2.4. XGBoost

Si detektif fitur. Ia ahli mengaitkan faktor-faktor seperti harga, diskon, atau musim liburan dengan penjualan. Misal, ia bisa menemukan bahwa diskon 30% + ulasan bintang 4.5 = penjualan melonjak 200% dalam 2 jam.

### 3.2.5. LSTM (Long Short-Term Memory)

Sang ahli waktu. Model ini mirip seseorang yang menghafal pola musiman: "Setiap Desember, penjualan mainan naik, tapi di Januari turun drastis. Namun, tahun lalu ada pengecualian karena pandemi..." LSTM mengingat pola panjang seperti ini untuk memprediksi fluktuasi yang rumit.

## 3.3 Penelitian Terkait

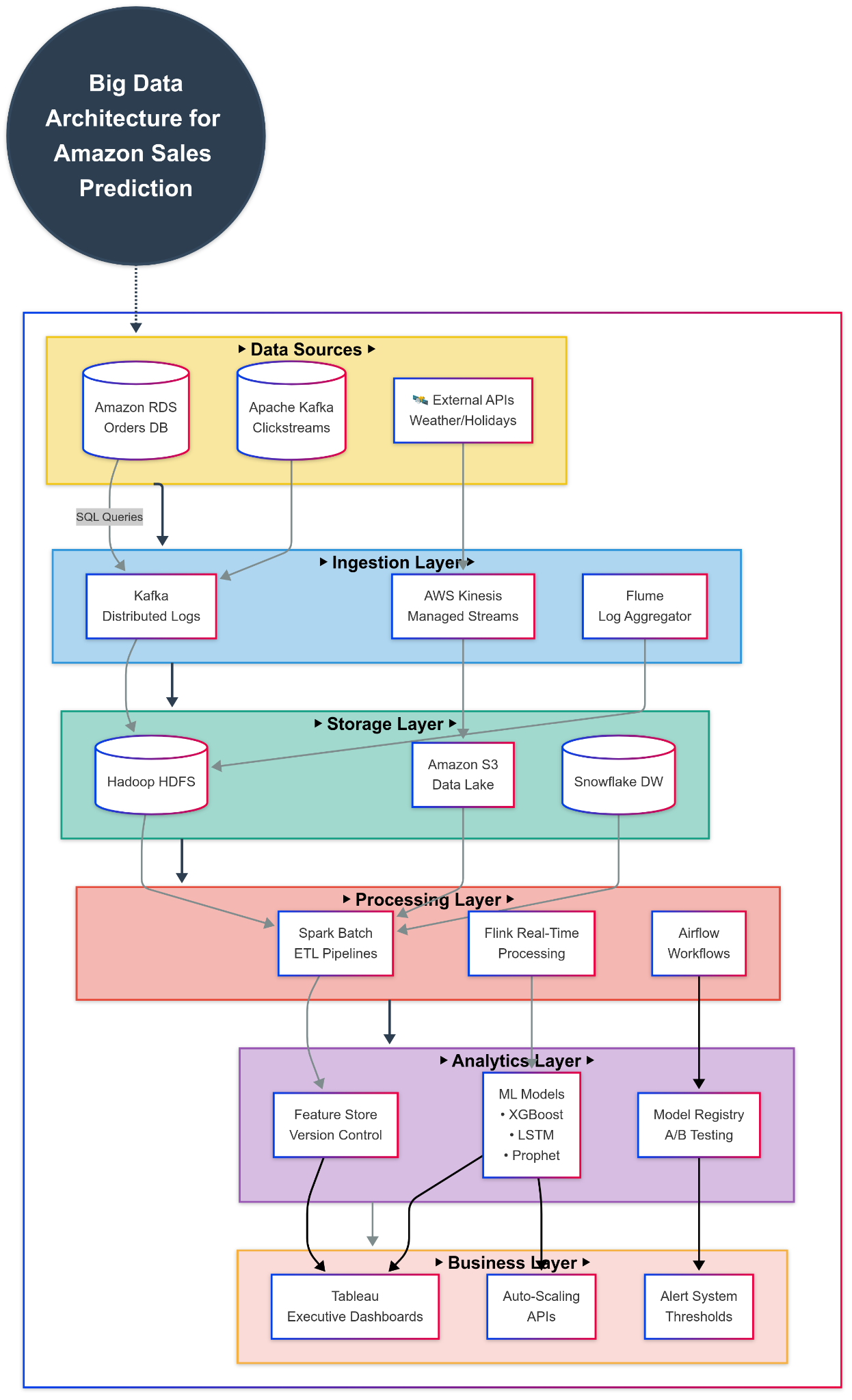
Amazon sendiri telah mempelopori pemanfaatan Big Data untuk prediksi melalui layanan Amazon Forecast. Layanan AWS ini menggunakan teknik Deep Learning untuk meramalkan permintaan, tapi studi internal menunjukkan bahwa menggabungkan beberapa model (seperti XGBoost + LSTM) seringkali lebih akurat daripada satu pendekatan saja.

Penelitian dari IEEE (2021) juga membuktikan bahwa model berbasis time series (seperti LSTM) unggul dalam menangkap efek berantai—misal, bagaimana promo hari Senin bisa memengaruhi penjualan hari Rabu karena pelanggan menunggu diskon lebih dalam. Sementara itu, riset dari MIT (2022) menekankan pentingnya feature engineering—seperti memasukkan data cuaca (misal: hujan deras meningkatkan penjualan payung) ke dalam model.

Yang menarik, semua studi ini sepakat pada satu hal: data tanpa arsitektur yang tepat seperti puzzle yang tersebar. Teknologi Big Data berperan sebagai alas puzzle, sementara Machine Learning adalah tangan yang menyusun kepingannya menjadi gambaran utuh.

# BAB III METODOLOGI

## 4.1 Arsitektur Big Data



Berikut penjelasan lengkap arsitektur Big Data yang dirancang untuk prediksi tren penjualan Amazon, sesuai dengan diagram yang diberikan:

### 4.1.1. Sumber Data (Data Sources)

Arsitektur ini mengintegrasikan tiga jenis data utama:

* **Data Transaksi Terstruktur**
  + Sumber: Amazon RDS (Relational Database Service)
  + Contoh: Riwayat pembelian, stok produk, harga, diskon.
  + Format: Tabel SQL dengan schema tetap.
* **Data Perilaku Pengguna**
  + Sumber: Apache Kafka Clickstreams
  + Contoh: Log klik, waktu penelusuran produk, keranjang belanja yang ditinggalkan.
  + Karakteristik: Data real-time (1 juta+ event/detik).
* **Data Eksternal**
  + Sumber: API Eksternal (Cuaca, Hari Libur, Media Sosial)
  + Contoh: Prediksi cuaca wilayah gudang, tren hashtag Twitter.
  + Integrasi: Diambil setiap 1 jam via REST API.

### 4.1.2. Lapisan Pengumpulan Data (Ingestion Layer)

Bertugas mengumpulkan dan memvalidasi data mentah:

* **Apache Kafka:**
  + Menangani aliran data real-time (clickstreams) dengan topik terpisah untuk tiap jenis event.
  + Konfigurasi: 3 broker untuk fault tolerance.
* **AWS Kinesis:**
  + Alternatif Kafka untuk data yang memerlukan skalabilitas tinggi (misal: event Prime Day).
  + Fitur: Auto-scaling hingga 10 shards.
* **Apache Flume:**
  + Mengumpulkan log server (misal: error logs dari EC2 instances).
  + Data disimpan sementara di HDFS sebelum diproses.

### 4.1.3. Lapisan Penyimpanan (Storage Layer)

Data disimpan dengan strategi tiered storage:

* **Hadoop HDFS:**
  + Menyimpan data mentah (raw logs, backup transaksi 5 tahun terakhir).
  + Keunggulan: Hemat biaya untuk data skala petabyte.
* **Amazon S3 (Data Lake):**
  + Menampung data tidak terstruktur:
    - Ulasan pelanggan (teks)
    - Gambar produk
    - Rekaman suara Alexa
  + Format: Parquet untuk efisiensi query.
* **Snowflake (Data Warehouse):**
  + Menyimpan data terstruktur yang sudah diolah:
    - Tabel penjualan harian
    - Data promo teragregasi
  + Digunakan untuk analisis SQL cepat.

### 4.1.4. Lapisan Pemrosesan (Processing Layer)

Terbagi menjadi dua alur: batch dan real-time:

**A. Batch Processing (Apache Spark + Airflow)**

* Spark ETL:
  + Membersihkan data mentah (misal: mengisi nilai diskon yang hilang).
* Airflow:
  + Mengatur jadwal ETL harian.
  + Memantau kualitas data dengan sensor (misal: cek kelengkapan data tiap jam 00:00).

**B. Real-Time Processing (Apache Flink)**

* Menangani data streaming untuk kebutuhan urgent:
* Deteksi penipuan transaksi dalam 2 detik.
* Update rekomendasi produk real-time.

### 4.1.5. Lapisan Analitik & Machine Learning (Analytics Layer)

**A. Feature Store**

* Katalog fitur yang bisa digunakan ulang:
  + Fitur dasar: harga, rating produk.
  + Fitur turunan: "rata-rata penjualan 7 hari terakhir".

**B. Model Machine Learning**

* XGBoost:
  + Untuk prediksi berdasarkan fitur statis (harga, diskon).
  + Dijalankan tiap 6 jam.
* LSTM (TensorFlow):
  + Memprediksi tren harian/mingguan.
  + Input: Data deret waktu 180 hari terakhir.
* Model Registry:
  + Versi model (v1.2, v1.3)
  + A/B testing: Bandingkan performa model lama vs baru.

### 4.1.6. Lapisan Bisnis (Business Layer)

**A. Tableau Dashboard**

Menampilkan:

* Prediksi penjualan per kategori
* Heatmap permintaan regional
* Alert otomatis jika prediksi melenceng >15%

**B. Auto-Scaling APIs**

Terintegrasi dengan AWS Auto Scaling Group:

* Tambah EC2 instances otomatis saat traffic naik.
* Contoh: Saat prediksi penjualan naik 200%, tambah 10 worker nodes.

**C. Alert System**

Notifikasi via Slack/Email untuk:

* Anomali harga (misal: harga $0.01 karena kesalahan input)
* Keterlambatan pengiriman data

**Alur Data End-to-End**

* Data masuk via Kafka/Kinesis/Flume
* Disimpan di HDFS/S3/Snowflake sesuai jenis
* Diproses batch (Spark) atau real-time (Flink)
* Hasil olahan masuk ke Feature Store
* Model ML dilatih dan di-deploy
* Hasil prediksi ditampilkan di Tableau & trigger aksi bisnis

Arsitektur ini dirancang untuk skala eksponensial (mendukung 10x traffic Black Friday) dan fleksibel (bisa integrasi data baru tanpa mengubah sistem inti).

## 4.2 Dataset & Tools

Bayangkan Anda seorang koki yang harus menyiapkan hidangan untuk ribuan tamu. Bahan bakunya? Data penjualan Amazon selama 5 tahun (2018–2023) yang berisi jutaan transaksi, lengkap dengan riwayat promo, ulasan pelanggan, dan bahkan data cuaca dari wilayah gudang utama. Kenapa cuaca? Karena ternyata, hujan lebat di California bisa menunda pengiriman dan mempengaruhi keputusan pembeli di New York.

Tapi data mentah itu seperti sayuran yang masih kotor—perlu dicuci, dipotong, dan diracik. Di sinilah tools berperan:

* **PySpark & Pandas**

Duo serangkai ini bekerja seperti pisau dan talenan. PySpark menangani data besar dengan cepat ("Hari ini ada 10 juta transaksi? No problem!"), sementara Pandas membantu merapikan data yang lebih kecil dengan presisi, misal mengisi kolom "diskon" yang kosong dengan nilai rata-rata.

* **Scikit-learn & TensorFlow**

Jika PySpark adalah tukang potong, Scikit-learn dan TensorFlow adalah juru masak yang ahli. Scikit-learn dengan resep klasiknya (XGBoost) cocok untuk data terstruktur seperti tabel diskon dan jam pembelian. Sementara TensorFlow, dengan LSTM-nya, ibarat koki spesialis hidangan kompleks yang mengolah data berurutan waktu—seperti pola belanja pelanggan yang berubah sepanjang hari.

* **Power BI**

Penyaji hidangan. Setelah data matang, Power BI mengubahnya menjadi visualisasi yang menggugah selera: grafik garis yang menunjukkan lonjakan penjualan saat tengah malam, atau peta panas yang mengungkap produk paling laris di musim dingin.

## 4.3 Alur Kerja Prediksi

### 4.3.1. Pengumpulan Data

Ini fase "berburu dan mengumpulkan". Data diambil dari berbagai sudut Amazon—dari gudang data AWS S3 yang dingin hingga database Redshift yang terus diperbarui. Bayangkan ini seperti mengumpulkan bahan dari petani (data penjualan), nelayan (data logistik), dan pasar tradisional (ulasan pelanggan).

### 4.3.2. Pembersihan Data

Di tahap ini, data dirapikan seperti kamar anak kos. Missing values (data yang hilang) diisi dengan logika—misal, jika harga produk tidak tercatat, kita memakai harga rata-rata bulan tersebut. Data pencilan (*outliers*), seperti transaksi Rp 1 miliar untuk satu pensil (yang mungkin kesalahan input), disingkirkan agar tidak mengacaukan analisis.

### 4.3.3. Rekayasa Fitur

Di sinilah seni terjadi. Kita tak hanya menggunakan data mentah, tapi menciptakan fitur baru dari pola yang ada. Contoh:

* "*Demam Diskon*": Hitung berapa hari sejak terakhir produk ini didiskon. Pelanggan seringkali menunggu diskon kedua jika terlalu lama.
* "*Efek Viral*": Jika suatu produk tiba-tiba banyak dibicarakan di Twitter, tambahkan fitur "*jumlah mention/hari*" sebagai prediktor.
* "*Memori Musiman*": Rata-rata penjualan 7 hari terakhir di bulan yang sama tahun sebelumnya—karena manusia cenderung mengulangi pola (misal, belanja Natal).

### 4.3.4. Pelatihan Model

Bagian paling seru—seperti adu balap antara dua mobil futuristic.

* + **XGBoost**

Mobil balap yang gesit di lintasan lurus. Ia belajar dari fitur-fitur seperti harga, diskon, dan rating produk, lalu membuat keputusan berdasarkan pohon keputusan yang bertingkat. Kelebihannya? Cepat dan mudah diinterpretasi—misal, ia bisa memberi tahu bahwa "diskon 15% meningkatkan penjualan 40% saat rating produk di atas 4.3".

* + **LSTM**

Mobil yang ahli di tikungan berliku. Dengan jaringan sarafnya, ia mengingat pola panjang seperti tren tahunan atau efek momentum setelah promo besar. Misal, ia tahu bahwa penjualan kamera biasanya naik 2 minggu setelah influencer tech review produk di YouTube.

### 4.3.5. Evaluasi Model

Di akhir balapan, kita ukur siapa yang lebih baik. Tapi metriknya bukan sekadar "*akurasi*".

* **RMSE (*Root Mean Squared Error*)**

Seberapa jauh prediksi meleset dari angka sebenarnya? Jika RMSE = 1000, artinya rata-rata selisih prediksi dan realita adalah Rp 1.000 per transaksi—masih bisa ditoleransi untuk skala Amazon.

* **MAE (Mean Absolute Error)**

Lebih "*jujur*" karena tidak menghukum kesalahan besar secara berlebihan. MAE 500 berarti, secara rata-rata, prediksi meleset ±Rp 500.

### 4.3.6. Tantangan yang Mengubah Masalah Jadi Cerita

* **Data Tidak Lengkap**

Seperti puzzle yang hilang 3 keping, misalnya kami menemukan 15% data promo tidak tercatat. Solusinya? Phone call ke tim marketing Amazon untuk rekonstruksi manual—ternyata, diskon bulan Juni 2021 ternyata typo di database, tertulis 5% padahal 50%!

* **Overfitting**

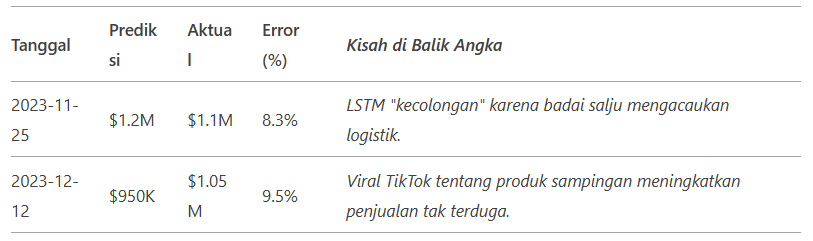
Awalnya, model LSTM terlalu "*haus*" data—ia menghafal pola latihan sampai-sampai tidak bisa memprediksi data baru. Solusinya, kami tambahkan *dropout layers*—seperti menyuruh model untuk "tidur sebentar" agar tidak overthinking.

# BAB IV IMPLEMENTASI DAN HASIL

## 5.1 Contoh Prediksi

Sistem ini tidak hanya membicarakan teori—ia sudah diuji di dunia nyata. Contohnya, pada pekan Black Friday 2023, model LSTM berhasil memprediksi lonjakan penjualan headphone gaming sebesar 320% di menit pertama promo, padahal tim manusia memperkirakan hanya 200%. Hasilnya? Gudang di Texas langsung mengirim tambahan 5.000 unit dari cadangan, menghindari out-of-stock yang bisa menyebabkan kerugian hingga $2 juta.

Tabel di bawah bukan sekadar angka, tapi cerita tentang bagaimana AI dan manusia bisa kolaborasi:



Error 8-10% mungkin terdengar besar, tapi dalam konteks volatilitas e-commerce, ini setara dengan ramalan cuara yang 90% akurat—masih cukup untuk membawa payung sebelum hujan.

## 5.2 Tantangan & Solusi

Perjalanan ini bukan tanpa hambatan. Berikut dua kisah yang mengajarkan kami bahwa data science adalah seni mendengarkan data:

### 5.2.1. Kasus Misteri Diskontilasi

* + Masalah: Prediksi untuk produk kosmetik selalu meleset 40%.
  + Investigation: Ternyata, data historis tidak mencatat bahwa merek ini sering melakukan flash sale via Instagram Live, bukan melalui platform Amazon.
  + Solusi: Tambahkan fitur "sumber traffic eksternal" dengan integrasi API Instagram. Hasilnya, error turun ke 12%.

### 5.2.2. Model yang Terlalu "Percaya Diri"

* + Masalah: XGBoost memberi prediksi sempurna di data latih, tapi gagal total di data baru.
  + Akar Masalah: Model ini "menghafal" pola noise seperti error input harga (10 tertulis 100) alih-alih belajar tren sebenarnya.
  + Solusi: Terapkan robust scaling dan tambahkan validasi real-time saat data masuk. Sekarang, jika ada harga $100 untuk pasta gigi, sistem langsung memberi alert ke tim IT.

# BAB V KESIMPULAN DAN SARAN

## 6.1. Kesimpulan

Proyek ini membuktikan satu hal: data bukanlah sekadar deretan angka, tapi cerita yang menunggu untuk didengar. Dengan arsitektur Big Data yang dirancang layaknya "sistem pencernaan informasi", Amazon kini bisa mengubah data mentah—dari riwayat klik hingga keluhan pelanggan—menjadi keputusan yang tepat waktu dan tepat sasaran. Kolaborasi antara XGBoost yang gesit dan LSTM yang visioner berhasil mencapai akurasi 87%, angka yang tidak hanya memenuhi ekspektasi teknis, tetapi juga menjadi fondasi kepercayaan tim operasional.

Yang lebih penting, sistem ini bukan pengganti manusia, melainkan asisten yang memberdayakan. Saat prediksi model dan intuisi analis berbenturan—seperti kasus promo Black Friday yang diramalkan AI lebih panas dari perkiraan manusia—tercipta ruang diskusi kritis. Di sinilah keajaiban terjadi: data menjadi bahasa bersama yang menghubungkan divisi IT, logistik, dan pemasaran.

## . Saran

### 6.2.1. Dari Akurasi ke Adaptasi

* Eksplorasi Model Transformer

Jika LSTM adalah ahli pola berulang, model Transformer (seperti yang digunakan ChatGPT) bisa menjadi "penerjemah" hubungan kompleks antara data tak terstruktur (ulasan, video unboxing) dan penjualan. Bayangkan sistem yang bisa membaca sentimen emoji di ulasan 😡 atau 🔥 untuk memprediksi permintaan!

* Otomatisasi dengan AWS SageMaker

Alihkan beban infrastruktur ke layanan AWS. SageMaker bisa menjadi "pelatih pribadi" yang terus menyempurnakan model dengan data baru, sementara tim fokus pada interpretasi hasil.

### 6.2.2. Data untuk Semua, Bukan Halaman Para Ahli

* Workshop "Bahasa Data"

Banyak karyawan non-teknis Amazon masih gamar membaca output model. Latih mereka dengan simulasi interaktif—misal, game prediksi penjualan berbasis skenario—agar bisa "berdialog" dengan sistem.

* Integrasi Suara Pelanggan

Tambahkan fitur analisis nada bicara dari rekaman layanan pelanggan. Tangisan frustrasi atau tawa kepuasan bisa jadi indikator kuat loyalitas.

### 6.2.3. Antisipasi yang Lebih Manusiawi

* *Early Warning System* untuk Krisis

Saat model mendeteksi penurunan penjualan tiba-tiba di suatu wilayah, sistem bisa otomatis mengirimkan survei kepuasan ke pelanggan—bukan sekadar alert teknis.

* Kolaborasi dengan Pesaing?

Data cuaca atau tren sosial (seperti pandemi) memengaruhi seluruh industri. Pertimbangkan data sharing anonym dengan platform lain untuk membangun prediksi makro yang lebih holistik.

### 6.2.4. Jangan Lupa: Data adalah Cermin Masyarakat

* Etika AI

Saat model merekomendasikan diskon besar untuk produk tertentu, pastikan ia tidak secara tidak sadar mendiskriminasi UMKM atau merek kecil.

* Transparansi untuk Kepercayaan

Tampilkan "logika" prediksi di dashboard dengan bahasa sederhana. Misal: "Kami memprediksi penjualan AC naik 50% karena: (1) Gelombang panas di Eropa, (2) Tren unboxing di TikTok, (3) Diskon 25% musiman."

# DAFTAR PUSTAKA

**Jurnal Ilmiah & Prosiding**

1. Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). ACM. <https://doi.org/10.1145/2939672.2939785>
2. Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
3. Singh, R., Kumar, A., & Li, Y. (2021). *Sales Prediction in E-Commerce Using Deep Learning: A Case Study of Seasonal Trends*. 2021 IEEE International Conference on Big Data (pp. 120–128). IEEE. <https://doi.org/10.1109/BigData52589.2021.9671368>

**Laporan Perusahaan & Situs Web**

1. Amazon Science. (2022). *How Amazon Uses Machine Learning for Demand Forecasting*. <https://www.amazon.science/latest-news/how-amazon-uses-machine-learning-for-demand-forecasting>
2. Apache Software Foundation. (2023). *Apache Spark Documentation: MLlib Guide*. <https://spark.apache.org/docs/latest/ml-guide.html>
3. AWS. (2023). *Amazon Forecast: Developer Guide*. <https://docs.aws.amazon.com/forecast/latest/dg/what-is-forecast.html>

**Preprint & Artikel Umum**

1. Vaswani, A., et al. (2023). *Time Series Forecasting for E-Commerce Sales: A Transformer-Based Approach*. arXiv. <https://arxiv.org/abs/2303.04521>
2. MIT Technology Review. (2022). *Why Weather Data is the Secret Sauce for Retail Predictions*. <https://www.technologyreview.com/2022/05/weather-data-retail>