**LAPORAN AKHIR PROYEK PENGOLAHAN DATA BESAR**

**Clustering Analysis With Spark MLib API for E-Commerce Dataset**



**Disusun oleh:**

**12S17025 Evola R A Tampubolon**

**12S17050 Kotrel Manurung**

**12S17059 Ekis Naomi Lasma**

**12S17067 Fradina Kristina Sinambela**

**PROGRAM STUDI SARJANA SISTEM INFORMASI**

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO**

**INSTITUT TEKNOLOGI DEL**

**2021**

# **DAFTAR ISI**

[**DAFTAR ISI**](#_heading=h.30j0zll) **2**

[**DAFTAR GAMBAR**](#_heading=h.3znysh7) **3**

[**DAFTAR TABEL**](#_heading=h.2et92p0) **4**

[**BAB 1 PENDAHULUAN**](#_heading=h.kqf02ckvjun9) **5**

[Latar Belakang](#_heading=h.3dy6vkm) 5

[**BAB 2 TINJAUAN PUSTAKA**](#_heading=h.1t3h5sf) **6**

[2.1 Arsitektur Big Data](#_heading=h.4d34og8) 6

[2.1.1 Data Source](#_heading=h.j9u4cu5al29e) 7

[2.1.2 ETL Process](#_heading=h.5qxp8355o9mz) 7

[2.1.3 Database Storage](#_heading=h.pr0iqscwp6ol) 7

[2.1.4 Machine Learning Analysis](#_heading=h.s2omoukpjzk1) 7

[2.1.5 Visualization](#_heading=h.l2oailoplxew) 7

[2.2 Machine Learning Pipeline](#_heading=h.2s8eyo1) 7

[**BAB 3 IMPLEMENTASI**](#_heading=h.17dp8vu) **8**

[3.1 Lingkungan Implementasi](#_heading=h.ghj7aq8vjsni) 8

[3.1.1 Perangkat Keras (Hardware)](#_heading=h.yq139sto71m5) 8

[3.1.2 Perangkat Lunak (Software)](#_heading=h.7rl08r96wam2) 8

[3.2 Implementasi Import Dataset](#_heading=h.atn4nsyfbg7z) 9

[**BAB 4 HASIL DAN PEMBAHASAN**](#_heading=h.3rdcrjn) **10**

[**REFERENSI**](#_heading=h.26in1rg) **11**

# 

# **DAFTAR GAMBAR**

# **DAFTAR TABEL**

# **BAB 1 PENDAHULUAN**

## **Latar Belakang**

Perkembangan teknologi yang semakin meluas menghasilkan berbagai jenis data dalam jumlah yang sangat besar. Pemanfaatan teknologi pun sudah banyak digandrungi oleh berbagai bidang, salah satunya *e-commerce*. Data yang dihasilkan oleh sebuah layanan *e-commerce* sangatlah besar dan beragam, data tersebut pada dasarnya sangat dibutuhkan untuk melakukan analisis yang bertujuan untuk meningkatkan kualitas barang, pelayanan ataupun untuk menghasilkan sebuah laporan yang diperlukan sebagai dokumentasi.

Salah satu solusi yang dapat mengatasi analisis data dalam jumlah besar dan beragam adalah dengan menggunakan teknologi yang sedang berkembang juga yaitu algoritma *clustering*. *Clustering* adalah sebuah algoritma yang membantu untuk menemukan kesamaan atau *similarity* yang tersembunyi antar data, seperti kesamaan antara harga produk atau layanan yang diberikan kepada pelanggan dalam jangka waktu tertentu. Dalam hal ini, algoritma *clustering* sangat membantu dalam melakukan pengelompokan terhadap pelanggan dengan satu pola yang sama. Terdapat beberapa pendekatan yang dapat digunakan untuk menangani case clustering seperti partition based, density based, hierarchical based dan model based approaches. Beberapa algoritma yang diusulkan pada pendekatan tersebut adalah k-means clustering, c-means clustering, DBSCAN, CLARA, BIRCH dan Gaussian Mixture Mode. Dalam pengerjaan proyek ini digunakan algoritma K-means yang merupakan clustering dengan pendekatan partition based dan populer dalam golongannya.

Apache Spark adalah sebuah platform *cluster computing* yang dimanfaatkan untuk melakukan pengelolaan terhadap big data dengan waktu cepat. Spark diyakini dapat menjalankan program hingga 100 kali lebih cepat di memori atau 10 kali lebih cepat pada disk dibandingkan dengan penggunaan Hadoop. Dalam artian lain, spark dapat membantu menyederhanakan tugas yang

MLlib adalah sebuah *library machine learning* yang menyediakan berbagai algoritma dan dirancang untuk dapat menskalakan cluster untuk klasifikasi, regression, clustering, collaborative filtering, dan lainnya.

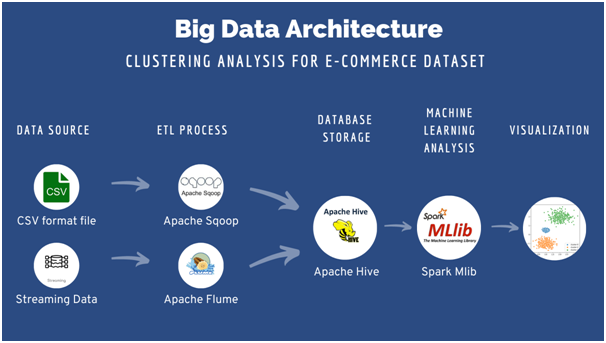
Pada pengerjaan proyek kali ini digunakan sebuah kumpulan data transaksional yang berisi pencatatan transaksi yang terjadi pada rentang waktu 01/12/2010 hingga 09/12/2011. Data tersebut didapatkan dari pedagang yang terdaftar tidak membuka online store dan bertempat di UK. Perusahaan tersebut menjual hadiah unik atau cenderamata untuk berbagai kesempatan. Kebanyakan pelanggan perusahaan tersebut adalah grosir. Data yang digunakan didukung dan dipublikasikan oleh The UCI Machine Learning Repository.

belum selesai ya… Ekis.

# **BAB 2 TINJAUAN PUSTAKA**

## **2.1 Arsitektur Big Data**

Arsitektur *big data* merupakan suatu struktur logis dan fisik yang dapat dirancang untuk menangani data yang akan disimpan untuk dapat diakses dan dikelola dalam suatu struktur data besar pada lingkungan teknologi informasi. Perancangan arsitektur *big data* merupakan proses ataupun aktivitas yang kompleks karena hasil dari perancangan tersebut dapat dijadikan sebagai sebuah referensi dalam menentukan solusi dari permasalahan infrastruktur data besar. Bervariasinya sumber data yang ada pada masa sekarang ini memiliki standar yang masing-masing bervariasi. Hal ini dapat menimbulkan adanya permasalahan pengolahan data yang melibatkan banyak sumber data. Arsitektur *big data* yang dirancang harus dapat bekerja tidak hanya untuk berbagai sumber data, namun juga untuk kemungkinan adanya perubahan skema data struktur yang digunakan untuk men-transport dan menyimpan data. Berikut merupakan rancangan arsitektur *big data* yang dirancang oleh tim pengembang.



Gambar 1 Perancangan Arsitektur *Big Data*

## **2.1.1 *Data Source***

*Data source* pada arsitektur *big data* merupakan sumber data yang akan digunakan pada pemrosesan data besar. *Data Source* dapat dibedakan menjadi dua jenis tipe data utama dalam susunan big data, yaitu data terstruktur dan data tidak terstruktur. Data terstruktur *(structured data)* secara umum mengacu pada data yang memiliki panjang dan format yang telah terdefinisi sebelumnya. Sebaliknya, data yang tidak terstruktur *(unstructured data)* merupakan data yang tidak mengikuti suatu susunan format tertentu. Selain itu, sumber dari data dapat dibagi kedalam dua buah kategori, yakni data yang dihasilkan oleh mesin (komputer) dan data yang dihasilkan oleh manusia. [1] Dalam pengerjaan proyek ini, sumber data yang digunakan berasal dari *E-Commerce* Dataset yang diperoleh dari UCI *Machine Learning dataset* yang dapat diakses dari Kaggle.

## **2.1.2 ETL *Process***

Dataset yang dimiliki kemudian diproses berdasarkan pemrosesan ETL dengan menggunakan Apache Sqoop untuk pemrosesan *batch* data, dan Apache Flume untuk pemrosesan data *streaming*. Pemilihan Apache Sqoop sebagai *tool* dalam pemrosesan ETL untuk data *batch* adalah karena data yang akan diolah merupakan dataset terstruktur yang berbentuk batch (tidak streaming) sehingga nantinya akan memerlukan sebuah basis data sementara yang digunakan untuk menampung data *batch.*

## **2.1.3 *Database Storage***

Kami berinteraksi dengan database storage menggunakan JDBC. Agar dapat memungkinkan, kami harus memiliki driver JDBC yang diperlukan untuk database yang kami inginkan untuk *interface* dengan disertakan dalam jalur *classpath*. Kami menggunakan metode pemuatan yang disebutkan sebelum format harus diubah dari yang default (Parquet) ke jdbc menggunakan metode format pada pembaca. Dan juga dapat menggunakan metode jdbc dan meneruskannya ke instance kelas Properties yang akan menampung properti koneksi.

## **2.1.4 *Machine Learning Analysis***

Spark MLlib digunakan untuk melakukan pembelajaran mesin di Apache Spark. MLlib terdiri dari algoritma dan utilitas populer. MLlib di Spark merupakan library machine learning yang dapat diskalakan untuk membahas algoritma berkualitas tinggi dan kecepatan tinggi. Algoritma pembelajaran mesin seperti regresi, klasifikasi, pengelompokan, penambangan pola, dan pemfilteran kolaboratif.

Spark.ml adalah Machine Learning API utama untuk Spark. Pustaka Spark.ml menawarkan API level lebih tinggi yang dibangun di atas DataFrames untuk membangun pipeline ML.

**ML Algorithms**

Algoritma ML membentuk inti dari MLlib. Termasuk algoritma pembelajaran umum seperti klasifikasi, regresi, pengelompokan, dan pemfilteran kolaboratif. MLlib menstandarkan API untuk mempermudah penggabungan beberapa algoritma ke dalam satu pipeline, atau alur kerja. Konsep utamanya adalah Pipelines API, di mana konsep pipeline terinspirasi oleh proyek scikit-learn.

**Transformer:**

Transformer adalah algoritma yang dapat mengubah satu DataFrame menjadi DataFrame lain. Secara teknis, Transformer mengimplementasikan metode transform (), yang mengonversi satu DataFrame ke yang lain, umumnya dengan menambahkan satu atau beberapa kolom. Sebagai contoh: Transformator fitur mungkin menggunakan DataFrame, membaca kolom (misalnya, teks), memetakannya ke dalam kolom baru (misalnya, vektor fitur), dan mengeluarkan DataFrame baru dengan menambahkan kolom yang dipetakan.

Model pembelajaran mungkin menggunakan DataFrame, membaca kolom yang berisi vektor fitur, memprediksi label untuk setiap vektor fitur, dan mengeluarkan DataFrame baru dengan label prediksi yang ditambahkan sebagai kolom.

**Estimator:**

Estimator adalah algoritma yang dapat dipasang pada DataFrame untuk menghasilkan Transformer. Secara teknis, Estimator mengimplementasikan metode fit (), yang menerima DataFrame dan menghasilkan Model, yang merupakan Transformer. Misalnya, algoritma pembelajaran seperti LogisticRegression adalah Estimator, dan pemanggilan fit () melatih LogisticRegressionModel, yang merupakan Model dan karenanya menjadi Transformer. Transformer.transform () dan Estimator.fit () keduanya tanpa kewarganegaraan. Di masa depan, algoritma stateful dapat didukung melalui konsep alternatif.

Setiap instance Transformer atau Estimator memiliki ID unik, yang berguna dalam menentukan parameter (dibahas di bawah).

## **2.1.5 *Visualization***

*Visualization* mencakup ekstraksi fitur, transformasi, pengurangan dimensi, dan pemilihan.

Ekstraksi Fitur untuk mengekstrak fitur dari data mentah. Transformasi Fitur mencakup penskalaan, renovasi, atau modifikasi fitur dan Pemilihan Fitur melibatkan pemilihan subset fitur yang diperlukan dari sekumpulan besar fitur.

## **2.2 Machine Learning Pipeline**

Pipeline menyatukan beberapa Transformer dan Estimator untuk menentukan alur kerja ML. Ini juga menyediakan fitur untuk membuat, mengevaluasi, dan menyesuaikan Pipelines ML.

Dalam pembelajaran mesin hal ini bertujuan menjalankan urutan algoritma untuk memproses dan belajar dari data. MLlib merepresentasikan alur kerja seperti Pipeline, yang terdiri dari urutan Tahapan Pipeline (Transformers dan Estimator) untuk dijalankan dalam urutan tertentu.

# **BAB 3 IMPLEMENTASI**

Pada bab ini akan dijelaskan mengenai lingkungan implementasi, hingga penerapan dan peng-implementasian *clustering* untuk dataset *E-Commerce* menggunakan *Spark Machine Learning Library API.*

## **3.1 Lingkungan Implementasi**

Lingkungan implementasi pada pengerjaan projek ini mencakup perangkat keras (*Hardware*) dan juga perangkat lunak (*Software)* yang digunakan dalam proses implementasi. Selanjutnya akan dijelaskan detail perangkat keras dan perangkat lunak yang digunakan pada proses pengerjaan projek.

## **3.1.1** **Perangkat Keras *(Hardware)***

Daftar perangkat keras yang digunakan untuk proses implementasi pada projek ini dapat dilihat pada tabel berikut ini.

Tabel 1 Perangkat Keras

|  |  |
| --- | --- |
| **Hardware** | **Spesifikasi** |
| Laptop (PC) | Acer |
| Processor | Core i3 |
| RAM | 4 GB |

## **3.1.2** **Perangkat Lunak *(Software)***

Spesifikasi perangkat lunak yang digunakan dalam proses implementasi pada projek ini dapat dilihat pada tabel berikut ini.

Tabel 2 Perangkat Lunak

|  |  |
| --- | --- |
| *Software* | Spesifikasi |
| Sistem Operasi | Windows 10 |
| Editor | Jupyter Notebook, Google Colaboratory |
| *Documentation* | Microsoft Office 2013 |
| Bahasa Pemrograman | Python |

## **3.2 Implementasi Import Dataset**

Proses pertama dalam pengerjaan projek adalah dengan memulai sesi dari spark untuk proses implementasi projek. Terlebih dahulu melakukan import pyspark dan *library-library* yang akan digunakan untuk pengerjaan projek yakni seperti *library* k-means, library K-Means Model, library ClusteringEvaluator dan matplotlib.

|  |
| --- |
| #import findspark  #findspark.init()  import pyspark  from pyspark.ml.clustering import KMeans  from pyspark.ml.clustering import KMeansModel  from pyspark.ml.evaluation import ClusteringEvaluator  import matplotlib.pyplot as plt  %matplotlib inline |

Setelah seluruh *library* yang dibutuhkan dalam pengerjaan proyek telah di-*import*, kemudian tentukan berapa jumlah *core* yang akan dipakai pada proses *clustering* dengan spark mlib dan juga tentukan nama aplikasi. Dapat dilihat pada potongan *code* berikut ini.

|  |
| --- |
| import pyspark  from pyspark.sql import SparkSession  spark = SparkSession.builder \  .master("local[2]") \  .appName("Proyek PBD Cluster E-Commerce") \  .getOrCreate() |

## **3.3 Implementasi *Import Dataset***

Proses selanjutnya adalah memuat dataset dari file local dengan ekstensi .csv ke dalam spark *dataframe* dengan menggunakan fungsi spark.read. Berikut merupakan potongan kode yang dapat dijalankan untuk peng-implementasian *import* dataset.

|  |
| --- |
| **from google.colab import drive**  **drive.mount('/content/drive/')**  **data = spark.read.format("csv").option("header", "true").load("/content/drive/MyDrive/PROYEK PBD/dataset/data.csv")** |

## **3.4 Implementasi *Exploratory Data***

Pada tahapan ini akan dilakukan investigasi awal sehingga akan diperoleh kondisi dataset yang akan digunakan. Proses ini bertujuan untuk mempersiapkan pembentukan model *machine learning*. Berikut merupakan potongan *code* yang dapat digunakan untuk tahapan implementasi *exploratory data.*

|  |
| --- |
| **data.show(10)**  **data.printSchema()**  **#check deskripsi dataset**  **data.describe().show()** |

# **BAB 4 HASIL DAN PEMBAHASAN**

# **REFERENSI**

|  |  |
| --- | --- |
| [1] | W. Aprilius, "Big Data dan Perawatan Kesehatan Studi Awal Menuju Perawatan kesehatan Masa Depan," *ULTIMA InfoSys,* vol. VI, no. 1, pp. 64-70, 2015. |