BIG DATA APPLICATION IN TRAFFIC LIGHT MANAGEMENT SYSTEM: A PROPOSAL

Argya Nur Adhirajasa	**(22/492170/PA/21079)**
Maulana Arya Alambana	**(22/492322/PA/21104)**
Mahisa Naraya Sakti	**(22/492176/PA/21082)**
Louis Widi Anandaputra	**(22/492218/PA/21090)**
Vian Sebastian Bromokusumo	**(22/496698/PA/21355)**

DEPARTMENT OF COMPUTER SCIENCE AND ELECTRONICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCE
UNIVERSITAS GADJAH MADA
YOGYAKARTA

2024

CHAPTER I
INTRODUCTION TO BIG DATA

1.1 Definition

The term “Big Data” is used when the size of a dataset is so large that it is too complex and voluminous for traditional data processing softwares to handle. The concept of big data revolves around the collection, processing, and analysis of large and complex datasets to uncover hidden patterns, correlations, and other valuable insights. Big data comes from a variety of sources, including social media and networks.

1.2 Characteristics

The definition of big data often includes the following characteristics, also known as “5v’s”, where it is defined as:

1. **Velocity**

   Velocity refers to how quickly a data is generated at a given time and how fast it moves. This data flow comes from sources such as mobile phones, social media, networks, and servers. Velocity covers the data’s speed and describes how the information continuously flows. As an example, a consumer with a tech that has a sensor connected to a network will keep gathering and sending data to the source. With the number of devices performing this action simultaneously, we can see why velocity is a prominent characteristic. It also factors in how quickly the raw big data information is turned into something an organization will benefit from. For something like the healthcare industry, it is important that medical data gathered by patient monitoring be quickly analyzed for a patient’s health.

2. **Volume**

   Volume in big data describes both the size and quantity of the data. However, the definition of how big the data is can change over time, depending on the current technological capabilities. Regardless, it does not change the fact that big data’s volume is colossal, due to the vast number of sources sending the information. As an example, on average, 500

million tweets are shared everyday on twitter, and thus, qualifies as big data.

3. **Value**

   Value refers to the benefits that big data can provide, and it relates directly to what organizations can do with the collected data. All data that is available is meaningless until we can derive the significance from it. The ability to pull value from big data is a requirement, as the value of big data increases significantly depending on the insights that can be gained from the data. Tools such as Apache Hadoop can help in cleaning, and rapidly process a massive amount of data.

4. **Variety**

   Variety refers to the diversity of data types. An organization might obtain data from different data sources that might vary in value. Collected data can be unstructured, semi-structured or structured. Unstructured data is usually not a good fit for a mainstream relational database, semi-structured data is data that has not been organized into specialized repository, while structured data is data that has been organized into a formatted repository. The challenge in variety is that it concerns the standardization and distribution of all data being collected.

5. **Veracity**

   Veracity refers to the quality, accuracy, integrity, and credibility of data. In some scenarios, gathered data could have missing pieces, might be inaccurate, or might not be able to provide real and valuable insight. Sometimes, a large amount of data can cause more confusion than insights if it is incomplete. For example, in the medical field, if data about what drugs a patient is taking is incomplete, the patient's life could be endangered. Veracity, overall, refers to the level of trust there is in the collected data.

CHAPTER II

COMPONENTS OF BIG DATA

Big Data consists of many different components including Data Gathering, Data Ingestion, Data Storage, Data Analysis and Visualization, and Data Implementation.

2.1 Data Gathering

Data gathering is a fundamental part of the Big Data process that involves the collection of raw data from several different sources. The quality and relevance of this data will significantly influence the insights that can be derived from it later in the process. Data sources range from transactional data like business transactions to social media data that will offer insight into public sentiment and trends. Sensor data from IoT devices and web data from user interactions on websites also play an important part as a data source. Effective data gathering combines these various data sources to create a comprehensive dataset.

a. Components

Components of data gathering of both the data sources and the method used to collect it. Transactional data is acquired form business activities and financial records which provides structured information for many analytical processes. Social media platforms produce enormous amounts of user-generated data that could offer insights into trends and market sentiment. Sensor data from IoT devices can be useful in knowing the environment to improve a machine's performance. Web data, machine data, and logs from servers and applications can add to the complexity of the data. Together, these components ensure to cover a wide area of data types and sources, each contributing unique insights.

b. Tools and Technologies

Big data tools and technologies are specifically designed for data gathering. Automated data collection tools, such as web crawlers and APIs, enable the extraction of data from websites and other online platforms. Streaming data platforms like Apache Kafka can facilitate real-time data collection from

streaming sources like social media feeds and IoT sensors. Log management tools, such as Elasticsearch and Logstash, help in gathering machine data and server logs. For sensor data, technologies like MQTT (Message Queuing Telemetry Transport) and CoAP (Constrained Application Protocol) are used to collect data from IoT devices.

Additionally, ETL (Extract, Transform, Load) tools such as Talend and Apache Nifi automate the process of extracting these data from various sources, transforming it into a suitable format, and loading it into a centralized repository for further analysis. These tools and technologies for the data-gathering process ensure an efficient collection of high-quality data from diverse sources.

c. Methodologies

An effective method is crucial for a better gathering of Big Data. A primary methodology often used is the ETL (Extract, Transform, Load), which involves extracting data from various sources, transforming it into a suitable format, and loading it into the data repository. This process can make sure that the collected data is standardized and ready for analysis.

Data integration methodologies like those used in data warehousing combine data from multiple sources into a single comprehensive dataset. Stream processing methodologies enable real-time data gathering by using tools like Apache Kafka and Apache Flink, which allows for the collection and processing of data as it is generated.

Additionally, batch processing methodologies are used for the periodic collection of large volumes of data, which is suitable for tasks that do not require immediate processing. Finally, data governance methodologies are also integral in establishing rules and standards for data quality, integrity, and security during the collection process. By implementing these methodologies, it can be ensured that the data gathered is efficient, reliable, and suited to the analytical goal.

2.2 Data Ingestion

Data ingestion is the process of collecting data from various sources into a storage or processing system where it can be accessed and analyzed. This is an important phase in the data lifecycle that ensures the availability of raw data for further processing and analysis. Data ingestion can handle different types of data, including structured, semi-structured, and unstructured data, and it can occur in real-time or in batch mode, depending on the requirements of the system and the nature of the data.

a. Components

Components in Data ingestion ensure that the data is efficiently collected, processed, and made available for further analysis. These components typically include data sources, ingestion mechanisms, processing frameworks, storage solutions, and data quality management.

Data sources can be various databases, file systems, APIs, or streaming data from IoT devices. Ingestion mechanisms manage how data is captured and transferred into a processing system, often utilizing batch or real-time processing. Processing frameworks, such as Apache Spark or Flink, handle the transformation and analysis of ingested data. Storage solutions like data lakes or warehouses ensure the organized and scalable storage of processed data. Data quality management tools ensure the integrity and accuracy of the data throughout the ingestion process.

b. Tools and Technologies

The tools and technologies utilized in data ingestion are designed to streamline and enhance the process. Apache Kafka is a widely used tool for real-time data streaming, providing high-throughput, low-latency data ingestion capabilities. For batch processing, Apache Nifi offers a solution for automating data flows between systems. Amazon Kinesis is another real-time data ingestion service that can process large streams of data. ETL (Extract, Transform, Load) tools like Talend and Informatica facilitate the integration and transformation of

data from diverse sources into a centralized repository. Cloud-based services like Google Cloud Dataflow and AWS Glue offer scalable data processing and ingestion capabilities, enabling seamless integration with other cloud services for storage and analytics.

d. Methodologies

Data ingestion methodologies are usually based on the nature of the data and the specific requirements of the organization. Batch processing involves collecting and processing data at scheduled intervals, which is suitable for scenarios where data is not immediately needed. Real-time or streaming data ingestion, on the other hand, involves continuous data collection and processing as soon as data is generated, which is crucial for time-sensitive applications such as fraud detection or live analytics. Hybrid approaches combine both batch and real-time methodologies to take advantage of both, ensuring comprehensive data processing and up-to-date insights. Additionally, methodologies like Change Data Capture (CDC) track changes in data sources to ensure that the ingested data reflects the latest updates, maintaining data consistency and accuracy across systems.

2.3 Data Storage

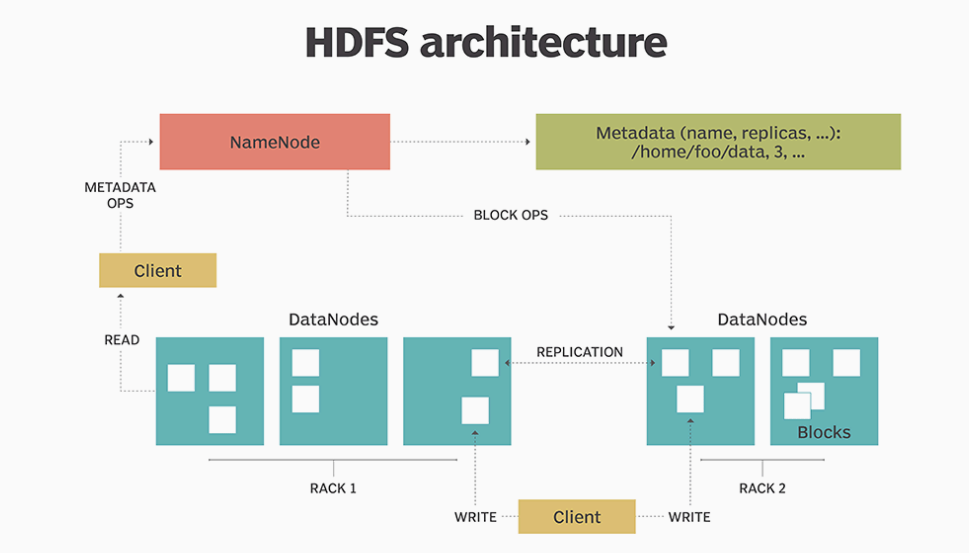
Data storage is the component that provides the physical or logical space to store the data that is ingested into the big data storage and processing architecture. Data storage can be on-premise or cloud-based, and can have different characteristics such as capacity, scalability, availability, durability, and cost.

a. Types of Big Data Storage

1. Hadoop Distributed File System (HDFS)

HDFS is utilized for storage permission. It is mainly designed for working commodity hardware devices, working on a distributed file system design. HDFS is designed in such a way that it believes more in storing the data in a large chunk

of blocks rather than storing small data blocks. HDFS in Hadoop provides fault tolerance and high availability to the storage layer and the other devices present in that Hadoop cluster.



NameNode works as a Master in a Hadoop cluster that guides the Datanode(Slaves). Namenode is mainly used for storing the Metadata i.e. the data about the data. Metadata can be the transaction logs that keep track of the user's activity in a Hadoop cluster.

DataNodes works as a Slave. DataNodes are mainly utilized for storing the data in a Hadoop cluster, the number of DataNodes can be from 1 to 500 or even more than that. The more data Node, the Hadoop cluster will be able to store more data. So it is advised that the DataNode should have High storage capacity to store a large number of file blocks.

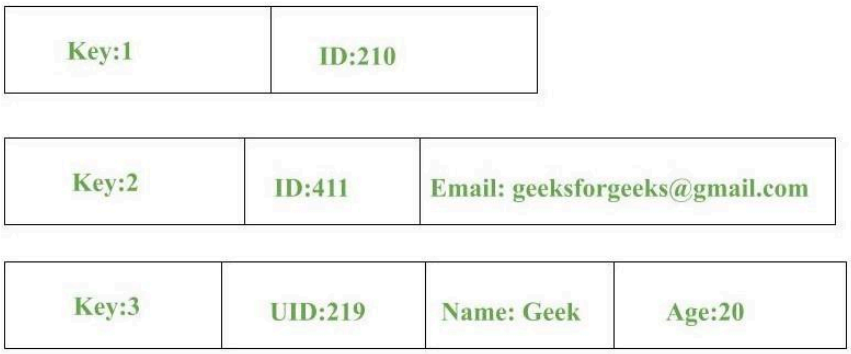
2. NoSQL Database

NoSQL databases are non-tabular and handle data storage differently than relational tables. Non-relational in nature, the core function of NoSQL is to provide a mechanism for storing and retrieving information. data modeling occurs using means not included under the tabular relations associated with relational databases.

Architecture pattern is a logical way of categorizing data that will be stored on the database. The data is stored in NoSQL in any of the following four data architecture patterns:

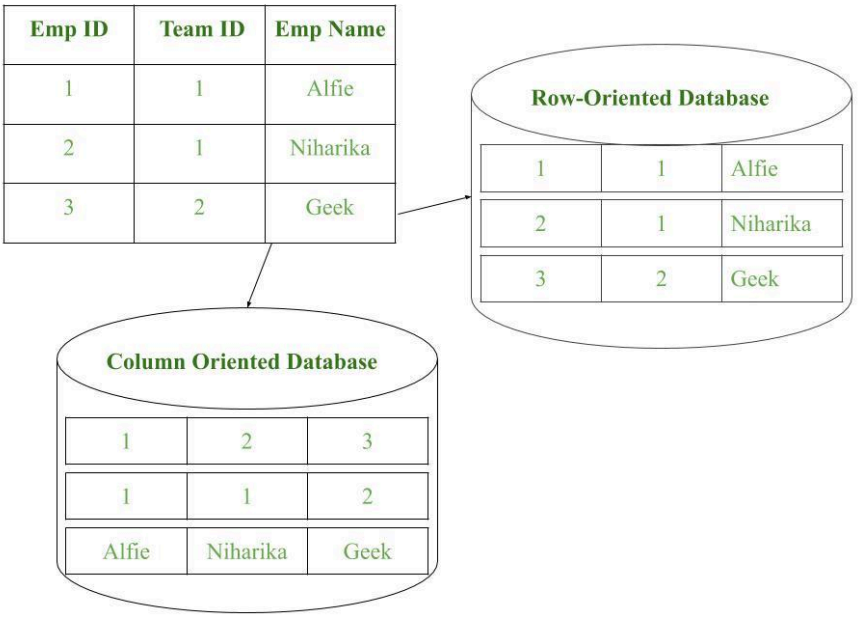
- **Key-Value Store Database**

  This model is one of the most basic models of NoSQL databases. As the name suggests, the data is stored in form of Key-Value Pairs. The key is usually a sequence of strings, integers or characters but can also be a more advanced data type. The value is typically linked or co-related to the key. The key-value pair storage databases generally store data as a hash table where each key is unique. The value can be of any type (JSON, BLOB(Binary Large Object), strings, etc). This type of pattern is usually used in shopping websites or e-commerce applications.



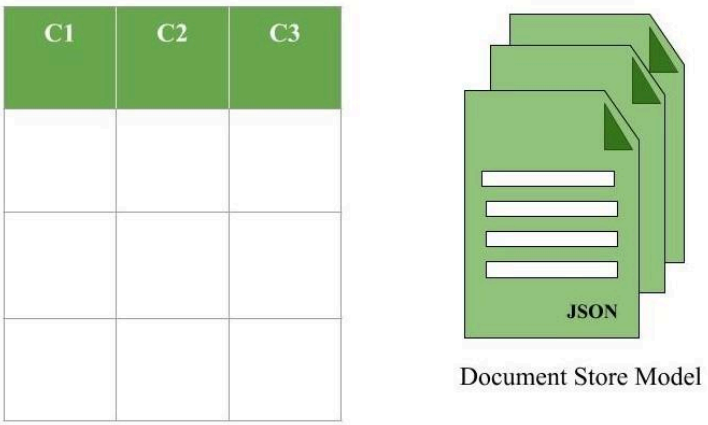
- **Column Store Database**

  Rather than storing data in relational tuples, the data is stored in individual cells which are further grouped into columns. Column-oriented databases work only on columns. They store large amounts of data into columns together. Format and titles of the columns can diverge from one row to other. Every column is treated separately. But still, each individual column may contain multiple other columns like traditional databases.



- **Document Database**

  The document database fetches and accumulates data in form of key-value pairs but here, the values are called as Documents. Document can be stated as a complex data structure. Document here can be a form of text, arrays, strings, JSON, XML or any such format. The use of nested documents is also very common. It is very effective as most of the data created is usually in form of JSONs and is unstructured.

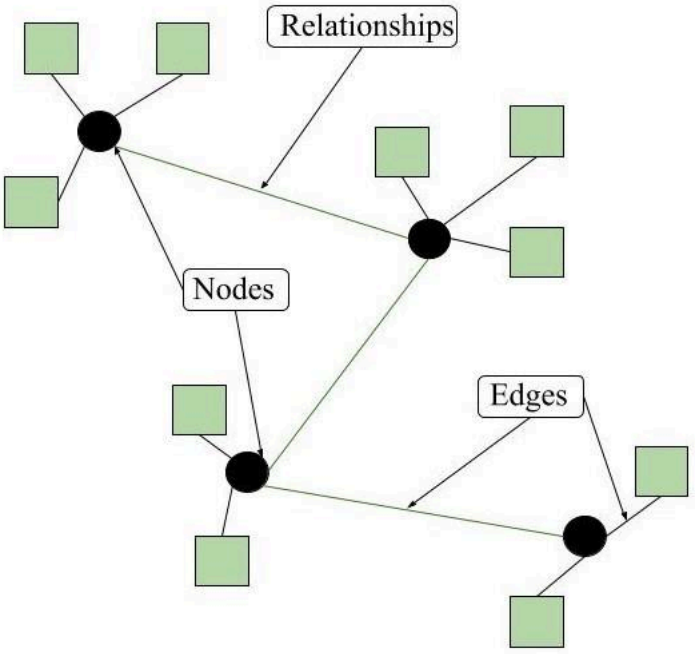


Relational Data Model

Document Store Model

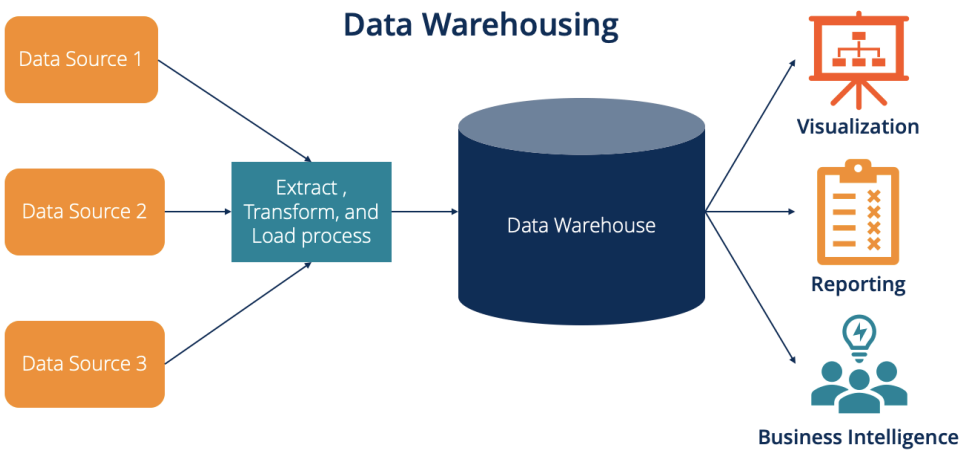
- **Graph Databases:**

This architecture pattern deals with the storage and management of data in graphs. Graphs are basically structures that depict connections between two or more objects in some data. The objects or entities are called as nodes and are joined together by relationships called Edges. Each edge has a unique identifier. Each node serves as a point of contact for the graph. This pattern is very commonly used in social networks where there are a large number of entities and each entity has one or many characteristics which are connected by edges. The relational database pattern has tables that are loosely connected, whereas graphs are often very strong and rigid in nature.



3. Data Warehouse

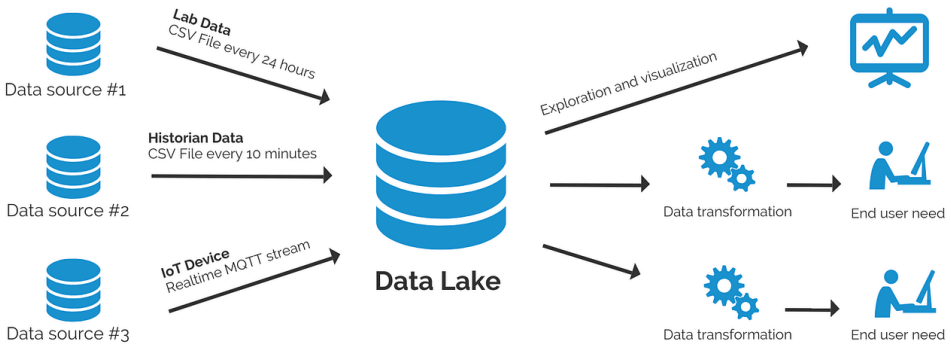
A data warehouse architecture is made up of tiers. The top tier is the front-end client that presents results through reporting, analysis, and data mining tools. The middle tier consists of the analytics engine that is used to access and analyze the data. The bottom tier of the architecture is the database server, where data is loaded and stored.



There are 2 different ways to store data in the data warehouse. First, data that is accessed frequently is stored in very fast storage (like SSD drives). Second, data that is infrequently accessed is stored in a cheap object store, like Amazon S3. The data warehouse will automatically make sure that frequently accessed data is moved into the "fast" storage so query speed is optimized.

4. Data Lake

A data lake is a central location that holds a large amount of data in its native, raw format. Compared to a hierarchical data warehouse, which stores data in files or folders, a data lake uses a flat architecture and object storage to store the data. Object storage stores data with metadata tags and a unique identifier, which makes it easier to locate and retrieve data across regions, and improves performance. By leveraging inexpensive object storage and open formats, data lakes enable many applications to take advantage of the data.



5. Data Lake vs Data Warehouse

	Data Lake	Data Warehouse
Information Types	Many types of unstructured and structured data from business applications and databases, websites, IoT and mobile devices, and social media	Structured relational (tabular) data from operational systems and databases
Structure	Data stored in its original raw form: data is structured only when it is analyzed	Schema defined in advance determines the way information is structured
Information Availability	Data is available for analysis very quickly	Data may take longer to become available for analysis because it is processed before being imported to the data warehouse
Uses	A variety of users can explore the data to delve into new questions. Users include data scientists, analysis, and developers	Primarily used by business groups and developers to ask specific predetermined types of questions
Analysis	A wide variety of tools, including machine learning, statistical analysis, and graph analytics	Analytics tools include business intelligence products, machine learning, and dashboards

2.4 Data Analysis and Visualization

Data analysis and visualization is the process of exploring, understanding, and interpreting data conditions and distributions to extract meaningful information. This process is crucial for any data-driven applications, and is applicable to a large range of knowledge domains.

a. Components

The components in data analysis would be summarized into four main parts, exploring, understanding, interpreting, and communicating.

1. Exploring
    a. Data Collection

       Data can be collected from various sources, including pre-existing datasets from platforms like Kaggle or the UCI Machine Learning Repository, or by extracting data from other sources such as data lakes, cloud data storage, or external APIs (data ingestion).

    b. Data Cleaning

       Raw data often contains missing values, noise, and redundant or irrelevant features. Data cleaning involves handling missing values, removing duplicates, correcting errors, and standardizing data formats to ensure the accuracy and reliability of the dataset.

    c. Data Integration

       Combining data from different sources to create a unified dataset. This involves merging datasets, resolving discrepancies, and ensuring consistency.
2. Understanding
    a. Exploratory Data Analysis
        - Summary Statistics: Calculating measures like mean, median, mode, standard deviation, and range to understand the central tendency and dispersion.

- Correlation Analysis: Examining relationships between variables using correlation coefficients and scatter plots.
- Data Visualization: Creating visual representations such as histograms, scatter plots, and box plots to identify patterns, trends, and anomalies.

b. Feature Engineering and Selection
   - Feature Engineering: Creating new features from existing data to improve the performance of analysis and predictive models.
   - Feature Selection: Identifying and selecting the most relevant features for analysis to enhance model accuracy and reduce complexity.

c. Normalization and Scaling
   - Normalization: Adjusting data values to a common scale, typically between 0 and 1.
   - Scaling: Transforming data to fit a specified range, often to improve the performance of machine learning algorithms.

3. Interpreting
   a. Data Analysis

      Applying statistical methods and machine learning techniques to extract insights and make predictions. This involves using various algorithms and models, such as regression analysis, clustering, classification, and more.

   b. Model Evaluation

      Assessing the performance of analysis and predictive models using evaluation metrics like accuracy, precision, recall, F1-score, and confusion matrices.

c. Data Visualization

Creating interactive and static visualizations to represent findings effectively. Tools like Tableau, Power BI, Plotly, and Matplotlib can be used to build charts, graphs, and dashboards.

4. Communicating
    - Storytelling with Data: Crafting a narrative around the data to communicate insights clearly and compellingly. This involves highlighting key findings, using appropriate visualizations, and providing context.
    - Creating Reports: Compiling the analysis and visualizations into comprehensive reports. These can be in the form of documents, presentations, or interactive dashboards.
    - Presenting Findings: Communicating results to stakeholders through presentations, emphasizing key insights, and suggesting actionable recommendations.
    - Documentation: Documenting the entire process, including methods used, assumptions made, and limitations of the analysis to ensure transparency and reproducibility.

b. Tools and Technologies

To facilitate the methodologies in which data analysis and visualization are performed, there are several tools and technologies that are most commonly used.

1. Python Libraries
    - Pandas, utilized for data analysis
    - NumPy, utilized for linear algebra operations
    - Matplotlib, visualization tools
    - Seaborn, advanced visualization
    - Scikit-learn, machine learning pre-processing until model selection and evaluation tool

- Plotly
- Bokeh

2. R
    - ggplot2
    - dplyr
    - Shiny
3. Tableau
4. Power BI
5. Excel
6. Big Data Tools
    - Apache Hadoop
    - Apache Spark
    - Apache Hive
    - Kafka
    - Cassandra

c. Methodologies

In the realm of data analysis, several analytics methodologies are used.

1. Descriptive Analytics
    - Purpose: To summarize historical data to understand what has happened in the past.
    - Techniques: Calculating metrics such as averages, totals, percentages, and identifying trends through visualizations like bar charts, line charts, and histograms.
    - Tools: Excel, Tableau, Power BI, Python (Pandas, Matplotlib, Seaborn).

2. Predictive Analytics
    - Purpose: To forecast future outcomes based on historical data.

- Techniques: Using statistical models and machine learning algorithms like regression analysis, time series analysis, and classification models to predict future trends and behaviors.
- Tools: Python (Scikit-learn, TensorFlow, Keras), R (caret, randomForest), Apache Spark, SAS.

3. Prescriptive Analytics
    - Purpose: To recommend actions based on predictive analytics to achieve desired outcomes.
    - Techniques: Optimization algorithms, simulation models, and decision analysis to suggest the best course of action.
    - Tools: Python (PuLP, Gurobi), R (lpSolve), AIMMS, IBM ILOG CPLEX.

4. Diagnostic Analytics
    - Purpose: To investigate the reasons behind past outcomes.
    - Techniques: Drill-down analysis, data discovery, data mining techniques, and correlation analysis to identify relationships and causes.
    - Tools: Tableau, Power BI, Python (Pandas, NumPy), SQL.

5. Inferential Statistics
    - Purpose: To make inferences about populations using sample data.
    - Techniques: Hypothesis testing, confidence intervals, regression analysis, and ANOVA to draw conclusions and make predictions about a population.
    - Tools: Python (SciPy, Statsmodels), R (stats package), SAS, SPSS.

6. Machine Learning
    - Purpose: To build models that can learn from data and make predictions or decisions without being explicitly programmed.
    - Techniques: Supervised learning (regression, classification), unsupervised learning (clustering, dimensionality reduction), and reinforcement learning.
    - Tools: Python (Scikit-learn, TensorFlow, Keras), R (caret, mlr), Apache Spark, H2O.ai

7. Data Mining
    - Purpose: To discover patterns and knowledge from large datasets.
    - Techniques: Association rule learning, clustering, anomaly detection, and sequence mining.
    - Tools: Python (Orange, Scikit-learn), R (arules, rpart), Weka, RapidMiner

8. A/B Testing
    - Purpose: To compare two versions of a variable to determine which one performs better.
    - Techniques: Randomized controlled trials, hypothesis testing, and statistical significance testing to evaluate the impact of changes.
    - Tools: Optimizely, Google Optimize, Python (SciPy, Statsmodels), R (stats package).

2.5 Machine Learning in Big Data Application

As data can grow to a really large scale in big data applications, standard analysis may not suffice business needs. Methods that implement a model learning over various data acquired through a business process in the big data environment may be utilized to enhance various aspects of the business. This process of digesting data into a model that act as an estimator, regulator, or even descriptor is called machine learning. Regarded as the most successful rebranding of linear algebra, machine learning has been implemented in various industries through various big data technologies. Examples of big data technologies that implement machine learning are the various recommendation systems in many social media platforms. In the case of a social media platform such as Instagram, not only does the platform need to accommodate millions of users accessing their platform, providing posts and reels in a low-latency manner, but data from users specifically cookies are collected and stored for then a recommender model would be able to give more recommendations of posts and reels to the users.

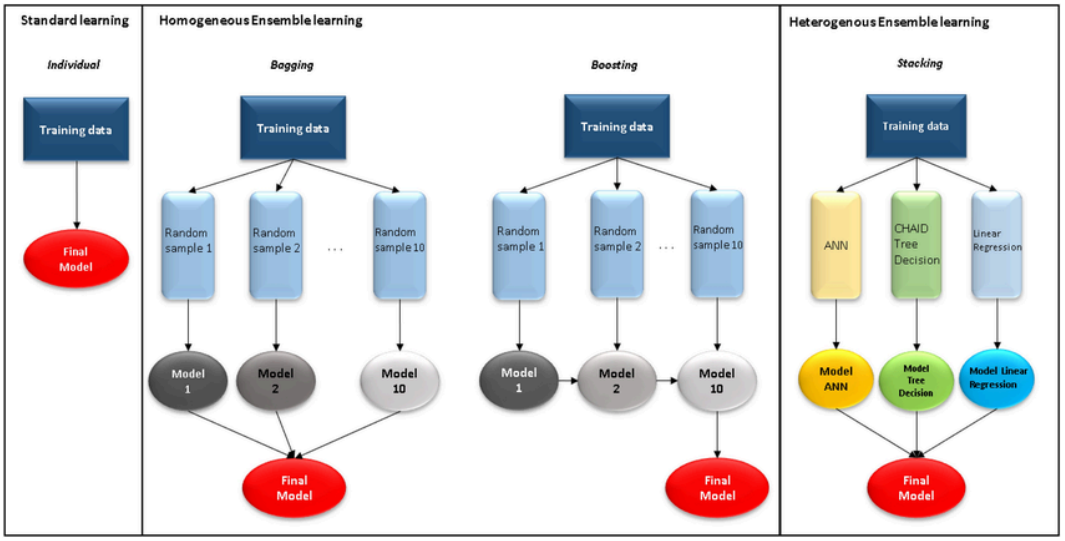
1. Introduction to Machine Learning

Various types of machine learning methods can be implemented. As a subset of artificial intelligence (AI), machine learning has three major types in terms of its learning process. Supervised learning became the most notable for those who are starting to learn about machine learning. Sample data is given and an estimator would be built upon it, hoping it will be able to predict future data with unknown outcomes. Other types of machine learning would also consider unsupervised learning and reinforcement learning.

Learning Method	Description	Example Use Case
Supervised Learning	A certain part of the data became the target to be learned and predicted by the model in the future with new data	Binary and multiclass classification, numerical regression, image segmentation, text generation, and summarization
Unsupervised Learning	No parts of the data	clustering for customer

	became the label. Mostly deals with distance, similarities, and density between each data point.	segmentations, data dimensionality reduction, and anomaly detection.
Reinforcement Learning	Can be supervised or unsupervised but the model is being updated according to a certain reward/penalty (loss function)	gaming, a recommendation system, and advertising, chat GPT website, robotic automation processes

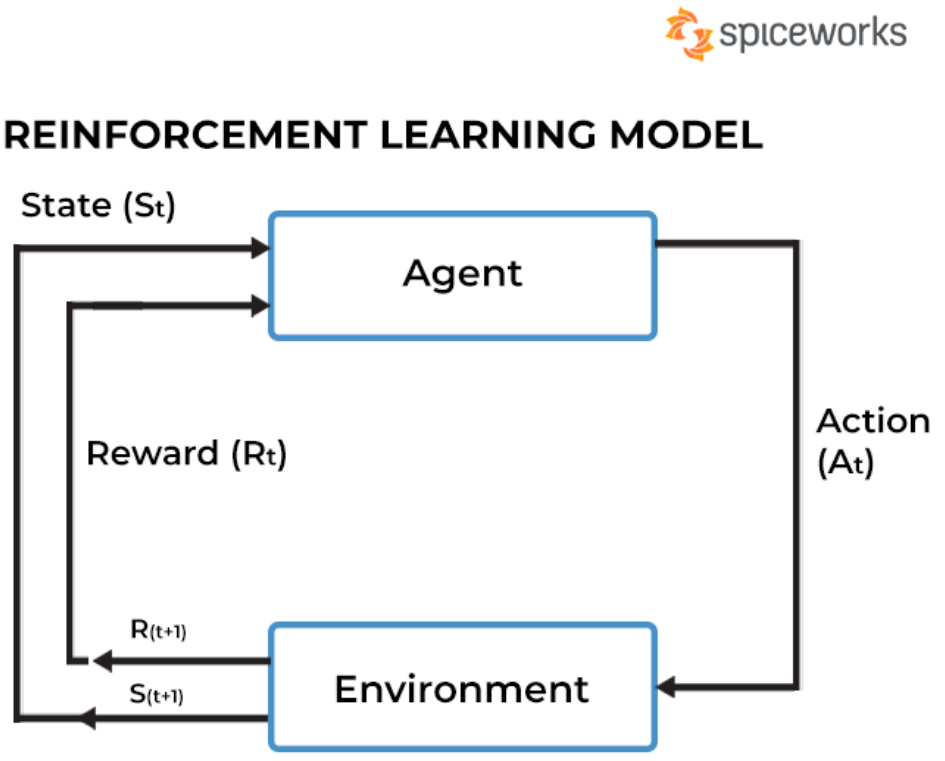
Supervised learning would consider the task of predicting something based on previous patterns. Traditional models being used for supervised learning would include Support Vector Machines, Decision Trees and Random Forests, Naïve Bayes, and K-Nearet Neighbors. Each mentioned model exhibits different characteristics and approaches such as ensemble learning, which tries to provide the best estimator by combining multiple models having the shortcomings and strengths of each model complement each other.



The branch of unsupervised learning in machine learning would not have a guided information on what to predict in the future compared to machine learning. The tasks included in unsupervised learning would consider finding similarities between data points while calculating the distance. It would try to find hidden and interesting patterns of a given data, or even finding one that should not belong to the data (anomalies/outliers). Most clustering algorithms is used in businesses that

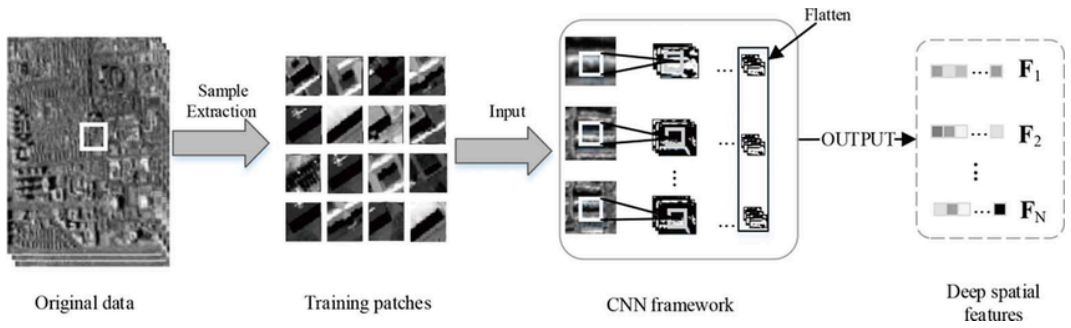
implement big data technologies to cluster groups of customers. The clustering is not merely a singular exhaustive process but most likely it is a part of multiple step business process.

Reinforcement learning is an agent based approach to machine learning. The model becomes an agent that learns through the environment where the environment is hard to define. The model may need to re-learn the environment to produce the best output. One such example is on a popular video streaming platform such as Youtube where recommendations system is put to use. The recommendation model may not have a good understanding of the preferences a user might have. It might have initial understanding but as time goes on more user data can be collected, which the model should be able to re-learn the preferences of the user. It can be said that the model goes to another state as its current state has different understanding of the environment or data from the previous state.



2. Utilizations of Deep Learning

As data can also be unstructured in a big data technology environment, the representation of vector space a standard machine learning model might digest may be hard to be extracted from the unstructured data acquired through business processes. Data such as images, audio, signals, and texts are all unstructured data that needs to have further processing to be able to be represented as a vector space. Deep learning that was developed from artificial neural networks has become a popular method for extracting such features from unstructured data. The process of extracting features from images can be done through Convolutional Neural Networks (CNN) and texts have the luxury of using a transformers model as a sequence processing unit.

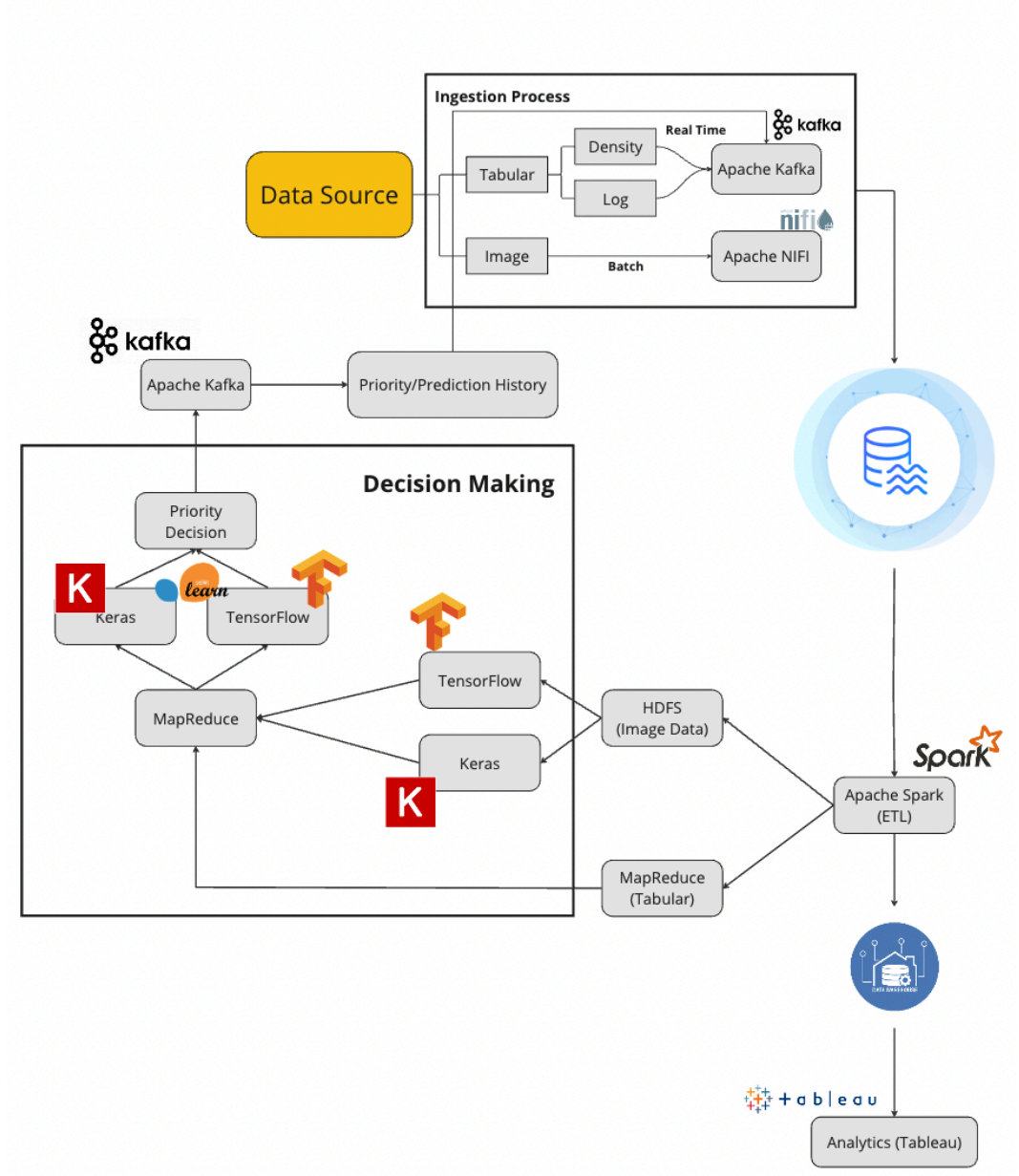


Challenges in implementing deep learning on a big data architecture of technologies is the big scale of computational resource needed to perform deep learning. In addition to the resource-heavy nature of deep learning, the amount of data to be used in deep learning processes, especially in a reinforcement learning approach, would need a high-resource platform to run on. Approaches such as TF-Lite were develop for a lightweight usage, enabling the other resource of the big data architecture to be utilized for other tasks.

CHAPTER III

APPLIED BIG DATA ON TRAFFIC MANAGEMENT SYSTEM

Sustainable Development Goal (SDG) 11 focuses on making cities and human settlements inclusive, safe, resilient, and sustainable. One of the many critical aspects of urban sustainability is the optimization of traffic flow to reduce congestion, emissions, and travel time. This work would propose how data are used to let a decision making model output the best sequence for traffic management, specifically traffic light management.



3.1 Big data in the Proposed System

Some cities in Indonesia are experiencing a significant level of traffic jams, which results in even more pollution, longer traveling time, and thus lowered life standards. While traffic light systems have been set at certain time intervals, they are not flexible enough for traffic flow throughout the day.

To address the issue of inflexible traffic light systems and improve traffic conditions, we propose a solution utilizing big data architecture. This solution aims to regulate traffic lights to ensure that it takes minimal time to adjust according to the traffic conditions. Through the ingestion of data gathered from different spots on the roads, the system can modify the signals in a way that will allow for a smoother flow of traffic, hence enhancing the efficiency on the roads.

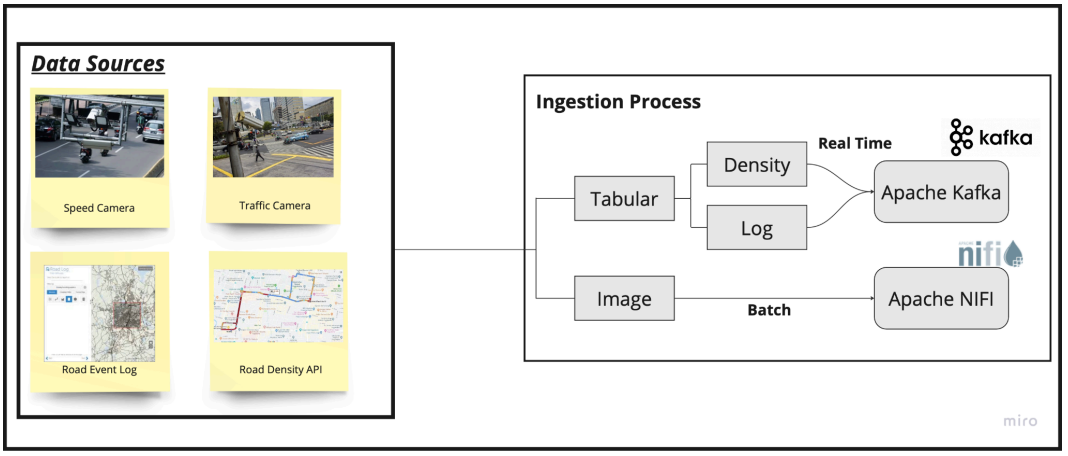
3.2 Data Gathering

For the traffic management system, the data collected is in the form of tabular data, and image data. The primary data sources include speed cameras, traffic cameras, road event logs, and road density API, where each of these data sources plays a crucial role in providing real-time data that are essential for accurate traffic monitoring and analysis for better utilization of the traffic lights. Speed cameras and traffic cameras continuously capture data and images of vehicle speeds and traffic conditions, while the road event log records information such as accidents, roadworks, and road closure. Furthermore, the road density API offers a real-time update on the road density, providing a dynamic overview of traffic flow across different roads and intersections.

3.3 Data Ingestion

In terms of data ingestion processes, it is divided into two different parts depending on the type of data collected. As mentioned previously, the data collected for the traffic management system is divided into tabular and image data. The tabular data from the road density API and the road event log which consisted of structured data formats such as CSV or JSON, are ingested in real-time using Apache Kafka. Kafka's distributed platform ensures high

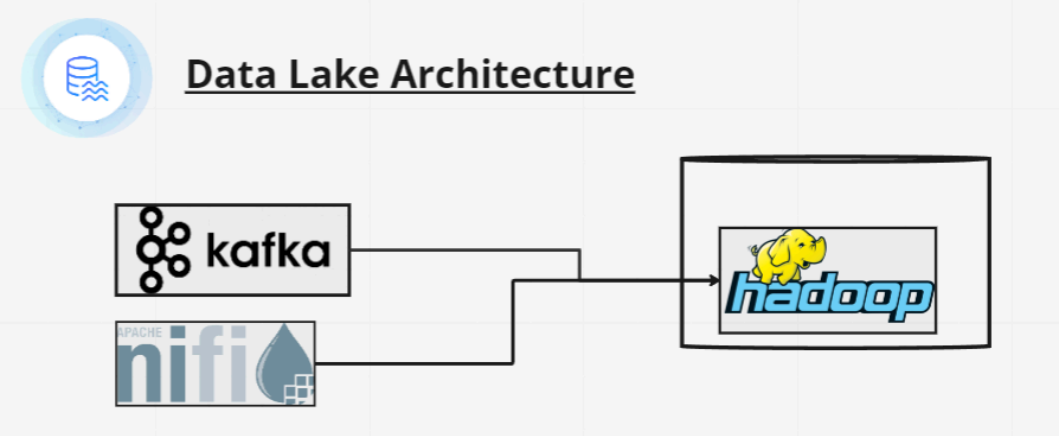
throughput and low latency, making it ideal for real-time data ingestion and processing. As data flows in from road density API and event logs, Kafka efficiently collects, organizes, and streams this data into a data lake system for further analysis and storage.



On the other hand, the image data gathered from cameras is ingested in batches using Apache NiFi. NiFi's robust data flow management capabilities allow it to handle the complexities of ingesting and processing large volumes of image data. By employing a batch processing approach, NiFi can manage the periodic collection of image files, ensuring they are properly formatted, compressed, and transmitted to storage and processing systems. This method balances the need for timely data updates with the computational demands of processing high-resolution images. With both of the ingestion processes, it provides a scalable and efficient solution for the traffic management system, leveraging a real-time data ingestion for immediate insights, and batch processing for detailed image analysis.

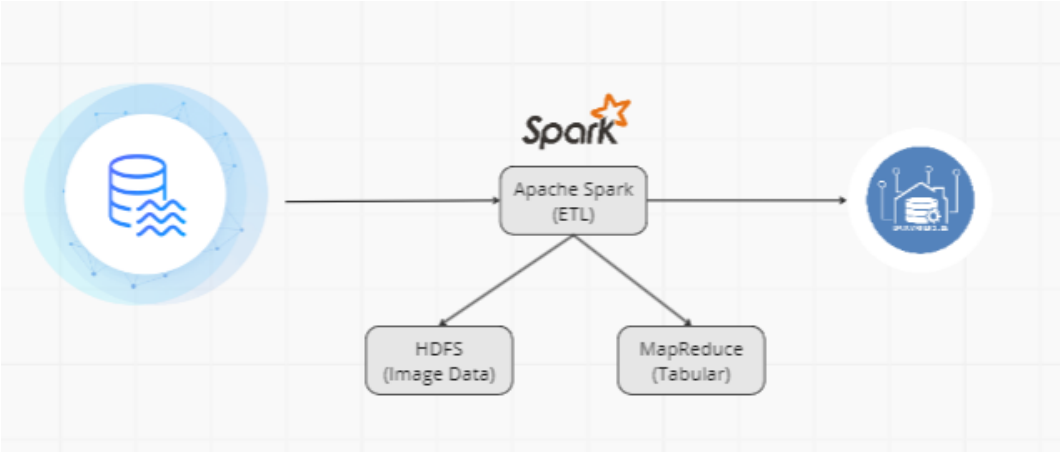
3.4 Data Storages

For storing the collected data, we use two main components which are a Data Lake and a Data Warehouse. The Data acquired from the sources that have been ingested is then passed through to the Data lake.



A data lake is a repository that allows for the storing of all structured and unstructured data at any scale. This allows us to store data as-is, without having to structure it first. In the data lake, the tabular data acquired with Kafka and the Image data acquired with NIFI are then passed to Apache Hadoop, which is an open-source framework that enables the distributed storage and processing of large datasets using a cluster of computers. Hadoop acts as the primary storage repository which provides the necessary infrastructure to store and analyze vast amounts of data. Hadoop's ecosystem, including HDFS (Hadoop Distributed File System), ensures that the data is managed efficiently and can be processed in parallel across the cluster.

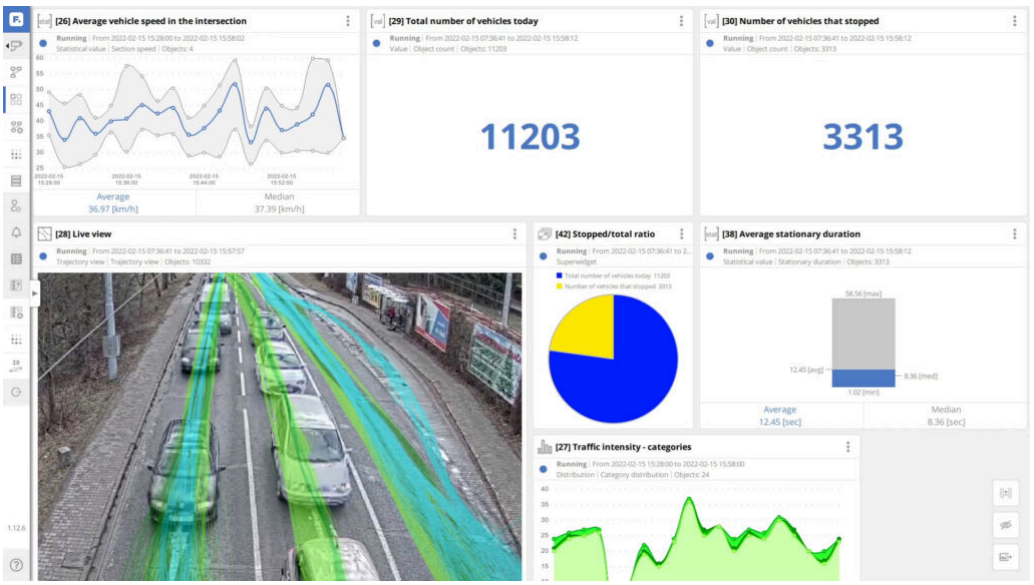
From the data lake, the data is then passed on to Apache Spark for further processing, which is an open-source, distributed computing system designed for big data processing and analytics. It provides an interface for programming entire clusters with implicit data parallelism and fault tolerance. Spark is known for its in-memory processing capabilities, which makes it significantly faster than traditional disk-based processing frameworks like Hadoop MapReduce.



Besides further processing after spark, the data is also passed on to a data warehouse, which is designed to store, manage, and analyze large volumes of structured data. It integrates data from various sources, transforms it into a consistent format, and supports complex queries and analysis for business intelligence (BI) and reporting purposes. Data warehouses are optimized for read-heavy operations, ensuring fast query performance and reliable data retrieval.

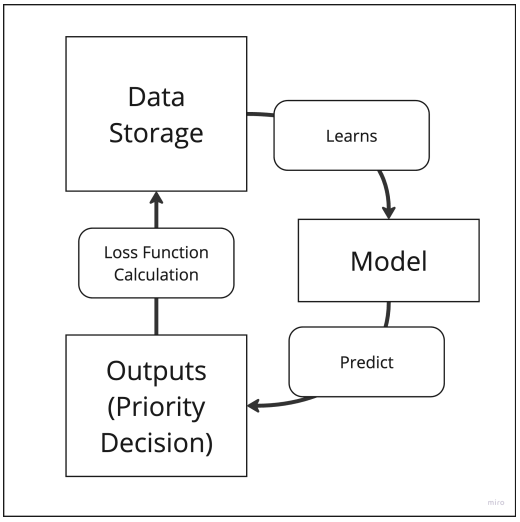
3.5 Data Analysis and Visualization

After the data is loaded to the data warehouse, various data analytics are performed. The use of data warehouses already supports data analytics because of its architecture. For easier stakeholders' consumption, our approach utilizes Tableau for dashboarding. Such implementations would be able to allow the stakeholders to see the conditions based on a certain road. Summarized informations are then able to help stakeholders, specifically the governing organizations for traffic management to make better decisions based on a data-driven approach.



3.6 Machine Learning Implementation

To allow decision management in traffic lights based on real-time traffic data, a machine learning approach is taken. The output of the process would be a priority decision, determining which traffic lights will indicate green and red. This Priority Decision is then also ingested back to the data source as Priority Decision History, which will aid in future predictions. In some way building a Reinforcement Learning algorithm.



The machine learning aspect in our approach is used twice. The first is the use of Convolutional Neural Networks to process image data, in identifying

important vehicles, such as ambulances, fire trucks, etc., and count the vehicles. Second, the outputs from the first step are represented as a vector space feature to allow concatenation to the original tabular data that contains density information, time, etc. The tabular data are used for prediction by utilizing standard machine learning classification processes. This method would implement technologies such as sci-kit learn and Pandas. The output of this stage will be data of importance scale (in regards to the existence of important vehicles) and the number of vehicles in a given junction in tabular form.

Once the output has been determined by the machine learning model, a loss function will be used to determine if the priority decision is enough. This loss function will calculate factors such as density, average speed, and more to be determined. The loss function and the priority decision data will be sent back to the storage system where the model could re-learn it. This process should not require heavy computational resources as training over the image classification model with deep learning as standard machine learning is utilized to give output the priority decision. Therefore, once the model achieves a high level of understanding of how to manage traffic, it will be able to optimize traffic altogether.

CHAPTER IV
CONCLUSION

Big data technologies integration into systems like traffic management provides notable advantages in enhancing the urban environment's alignment. Using data extracted from several sources such as speed cameras, traffic cameras, and road density API, the proposed system enables qualitatively adaptive optimization of traffic signals with the goal of improving traffic flow. The concept of data lakes and data warehouses integrated with Apache Hadoop and Apache Spark frameworks describes the ability to process and store large amounts of information. Furthermore, with the integration of Convolutional Neural Network and reinforcement learning, the system enables it to make real-time decisions to improve traffic situations and therefore improve traffic flow and aid in sustainable development of urban communities. This approach also emphasizes that big data can play an important role in solving the more difficult urban problems, not only this particular problem, but also indicating the potential for further developments of smart city measures.

REFERENCES

[1] GeeksforGeeks. (2021, September 21). NoSQL Data Architecture Patterns. GeeksforGeeks. https://www.geeksforgeeks.org/nosql-data-architecture-patterns/

[2] Ashtari, H. (2022, October 18). NoSQL Basics: Features, Types, and Examples - Spiceworks Inc. Spiceworks Inc. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-nosql/

[3] What is a Data Warehouse? - Data Warehouse Explained - AWS. (n.d.). Amazon Web Services, Inc. https://aws.amazon.com/what-is/data-warehouse/

[4] Riasetiawan, Mardhani. (2021). Modul Pembelajaran Analisis Big Data.

[5] Tableau. (2018). What Is Data Visualization? Definition, Examples, and Learning Resources. Tableau Software. https://www.tableau.com/learn/articles/data-visualization

[6] Terra, J. (2023, February 20). Characteristics of Big Data: Understanding the Five V's. Simplilearn.com. https://www.simplilearn.com/5-vs-of-big-data-article

[7] Robinson, S., & Gillis, A. S. (2023, November 17). 5V's of big data. Data Management. https://www.techtarget.com/searchdatamanagement/definition/5-Vs-of-big-data

[8] Team, K. (2024, May 10). Real-Time Data Ingestion and Batch processing with Apache NiFi for Data Lake. Ksolves. https://www.ksolves.com/blog/big-data/nifi/real-time-data-ingestion-and-batch-processing-with-apache-nifi-for-data-lake

[9] GeeksforGeeks. (2024, March 18). How to Use Apache Kafka for Real-Time Data Streaming? GeeksforGeeks. https://www.geeksforgeeks.org/how-to-use-apache-kafka-for-real-time-data-streaming/

[10] Khan, Faisal. (2021). Apache Kafka with Real-Time Data Streaming.

[11] Soloveichik, H. (2023, July 19). Guide to Data Ingestion: Types, Process & Best Practices. IBM Blog. https://www.ibm.com/blog/guide-to-data-ingestion/

[12] Data gathering: a complete guide | Adverity. (n.d.). https://www.adverity.com/data-gathering-a-complete-guide

[13] Ways to conduct data gathering | SurveyMonkey. (n.d.). SurveyMonkey. https://www.surveymonkey.com/market-research/resources/ways-to-conduct-data-gathering/

[14] Cunningham, P., Cord, M., Delany, S.J. (2008). Supervised Learning. In: Cord, M., Cunningham, P. (eds) Machine Learning Techniques for Multimedia. Cognitive Technologies. Springer, Berlin, Heidelberg.

[15] Ghahramani, Z. (2004). Unsupervised Learning. In: Bousquet, O., von Luxburg, U., Rätsch, G. (eds) Advanced Lectures on Machine Learning. ML 2003. Lecture Notes in Computer Science(), vol 3176. Springer, Berlin, Heidelberg.

[16] LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. Nature 521, 436–444.

[17] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). The MIT Press.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 30). Curran Associates, Inc.